

# LightWave™ 10



LAYOUT



# LightWave<sup>®</sup> 10

## Reference Manual

License.key number

Win32	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	—	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	—	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	—	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Win64	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	—	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	—	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	—	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
MAC	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	—	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	—	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	—	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

This publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose without prior written consent from NewTek, Inc.

© Copyright 2010, NewTek, Inc. All Rights Reserved

LightWave 3D™

NewTek  
5131 Beckwith Blvd  
San Antonio, TX 78249  
Tel.: 1-800-862-7837 - Fax: 210-370-8001  
[www.newtek.com](http://www.newtek.com)



## LightWave 3D® Software License and Limited Warranty

PLEASE READ CAREFULLY BEFORE INSTALLING AND/OR USING THIS SOFTWARE. BY INSTALLING AND/OR 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, RETURN THIS PACKAGE TO THE PLACE WHERE YOU OBTAINED IT WITHIN 15 DAYS OF PURCHASE FOR A FULL REFUND.

### 1. Grant of License

The enclosed computer program(s) (the "Software") is licensed, not sold, to you by NewTek, Inc. (NEWTEK) for use only under the terms of this License, and NEWTEK reserves any rights not expressly granted to you. You own the disk(s) on which the Software is recorded or fixed, but the Software and all copyright rights therein, foreign and domestic, is owned by NEWTEK or its suppliers and is protected by United States copyright laws and international treaty provisions.

The copyright restrictions of this license extend to any further updates, software patches, or bug fixes made available to you by NEWTEK, whether distributed by floppy disc, CD ROM, DVD ROM or in an electronic format via BBS, ftp, email, etc.

This License allows you to use one copy of the Software on a single computer at a time. To "use" the Software means that the Software is either loaded in the temporary memory (i.e., RAM) of a computer, or installed on the permanent memory of a computer (i.e., hard disk, CD ROM, DVD ROM, etc.).

You may use at one time as many copies of the Software as you have licenses for. You may install the Software on a common storage device shared by multiple computers, provided that if you have more computers having access to the common storage device than the number of licensed copies of the Software, you must have some software mechanism which locks out any concurrent user in excess of the number of licensed copies of the Software (an additional license is not needed for the one copy of Software stored on the common storage device accessed by multiple computers).

You may make one copy of the Software in machine readable form solely for backup purposes. The Software is protected by copyright law. As an express condition of this License, you must reproduce on the backup copy the NEWTEK copyright notice in the following format "(c) 1990 - 2010 NEWTEK"

You may permanently transfer all your rights under this License to another party by providing such party all copies of the Software licensed under this License together with a copy of this License and all written materials accompanying the Software, provided that the other party reads and agrees to accept the terms and conditions of this License.

### 2. Restrictions

The Software contains trade secrets in its human perceivable form and, to protect them, YOU MAY NOT REVERSE ENGINEER, DECOMPILE, DISASSEMBLE, OTHERWISE REDUCE THE SOFTWARE TO ANY HUMAN PERCEIVABLE FORM. YOU MAY NOT MODIFY, ADAPT, TRANSLATE, RENT, LEASE, LOAN, RESELL FOR PROFIT, OR CREATE ANY MODIFICATIONS OR OTHER DERIVATIVE WORKS BASED UPON THE SOFTWARE OR ANY PART THEREOF.

### 3. Termination

This License is effective until terminated. This License will terminate immediately without notice from NEWTEK or judicial resolution if you fail to comply with any provision of this License. Upon such termination you must destroy the Software, all accompanying written materials and all copies thereof. You may also terminate this License at any time by destroying the Software, all accompanying written materials and all copies thereof.

### 4. Export Law Assurances

You agree that neither the Software nor any direct product thereof is being or will be shipped, transferred or re-exported, directly or indirectly, into any country prohibited by the United States Export Administration Act and the regulations thereunder or will be used for any purpose prohibited by the Act.

### 5. Limited Warranty and Disclaimer, Limitation of Remedies and Damages.

YOU ACKNOWLEDGE THAT THE SOFTWARE MAY NOT SATISFY ALL YOUR REQUIREMENTS OR BE FREE FROM DEFECTS. NEWTEK WARRANTS THE MEDIA ON WHICH THE SOFTWARE IS RECORDED TO BE FREE FROM DEFECTS IN MATERIALS AND WORKMANSHIP UNDER NORMAL USE FOR 90 DAYS FROM PURCHASE, BUT THE SOFTWARE AND ACCOMPANYING WRITTEN MATERIALS ARE LICENSED "AS IS." ALL IMPLIED WARRANTIES AND CONDITIONS (INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE) ARE DISCLAIMED. YOUR EXCLUSIVE REMEDY FOR BREACH OF WARRANTY WILL BE THE REPLACEMENT OF THE MEDIA OR REFUND OF THE PURCHASE PRICE. IN NO EVENT WILL NEWTEK OR ITS DEVELOPERS, DIRECTORS, OFFICERS, EMPLOYEES OR AFFILIATES BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, AND THE LIKE), WHETHER FORESEEABLE OR UNFORESEEABLE, ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE OR ACCOMPANYING WRITTEN MATERIALS, REGARDLESS OF THE BASIS OF THE CLAIM AND EVEN IF NEWTEK OR AN AUTHORIZED NEWTEK REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The above limitations will not apply in case of personal injury only where and to the extent that applicable law requires such liability. Because some jurisdictions do not allow the exclusion or limitation of implied warranties or liability for consequential or incidental damages, the above limitations may not apply to you.

### 6. General

This License will be construed under the laws of the State of Texas, except for that body of law dealing with conflicts of law. If any provision of this License shall be held by a court of competent jurisdiction to be contrary to law, that provision will be enforced to the maximum extent permissible and the remaining provisions of this License will remain in full force and effect.

### 7. Trademarks

NewTek, LightWave, LightWave 3D, LightWave CORE, LightWave 10, Viewport Preview Renderer, VPR, Virtual Studio Tools, TriCaster, TriCaster TCXD300, TriCaster TCXD850, 3PLAY, TriCaster PRO, TriCaster STUDIO, TriCaster BROADCAST, TriCaster DUO, SpeedEDIT, Virtual Set Editor, VSE, iVGA, LiveText, LiveSet, LiveMatte, TimeWarp, VT, [VTS], Video Toaster, Toaster, 3D Arsenal, Aura, are trademarks of NEWTEK. All other brand names, product names, or trademarks belong to their respective holders.

### 8. US Government Restricted Provision

If this Software was acquired by or on behalf of a unit or agency of the United States Government this provision applies. This Software:

- (a) Was developed at private expense, and no part of it was developed with government funds,
- (b) Is a trade secret of NEWTEK for all purposes of the Freedom of Information Act,
- (c) Is "commercial computer software" subject to limited utilization as provided in the contract between the vendor and the government entity, and
- (d) In all respects is proprietary data belonging solely to NEWTEK.

For units of the Department of Defense (DoD), this Software is sold only with "Restricted Rights" as that term is defined in the DoD Supplement to the Federal Acquisition Regulations, 52.227-7013 (c) (1) (ii).

Use, duplication or disclosure is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at 52.227-7013. Manufacturer: NEWTEK, 5131 Beckwith Boulevard, San Antonio, TX 78249.

If this Software was acquired under a GSA Schedule, the US Government has agreed to refrain from changing or removing any insignia or lettering from the software or the accompanying written materials that are provided or from producing copies of manuals or disks (except one copy for backup purposes) and:

(e) Title to and ownership of this Software and documentation and any reproductions thereof shall remain with NEWTEK,

(f) Use of this Software and documentation shall be limited to the facility for which it is required, and,

(g) If use of the Software is discontinued to the installation specified in the purchase/delivery order and the US Government desires to use it at another location, it may do so by giving prior written notice to NEWTEK, specifying the type of computer and new location site. US Governmental personnel using this Software, other than under a DoD contract or GSA Schedule, are hereby on notice that use of this Software is subject to restrictions which are the same as or similar to those specified.

## Copyright and Trademarks

LightWave® 10

© Copyright 1990-2010 NewTek, Inc. All rights reserved.

Additional components Copyright: Steve Worley (1995-2010), Worley Labs (1995-2010), RainMaker (2003-2010) and Daisuke Ino (1995-2010)

FxMonkey™, HyperVoxels™, IntelligEntities™, Motion Mixer™, MultiMeshes™, Particle FX™, P.A.V.L.O.V.™, Skelegons™, SkyTracer 2™, Vmaps™ are trademarks of NewTek, Inc.

LightWave 3D is a registered trademark of NewTek, Inc. LightWave is a trademark of NewTek, Inc. All rights reserved. Windows and Windows Vista are registered trademarks of Microsoft in the United States and/or other countries. Intel and Intel Core are trademarks of Intel Corporation in the U.S. and/or other countries. AMD Athlon is a trademark of Advanced Micro Devices in the U.S. and/or other countries. Mac and Snow Leopard are registered trademarks of Apple Inc. NVIDIA and GeForce are trademarks of NVIDIA Corporation in the U.S. and other countries. All other trademarks are the property of their respective owners.





## Acknowledgements

---

### Engineering:

Chuck Baker, Deuce Bennett, Jamie Finch, David Forstenlechner, Matt Gorner, Mark Granger, Jonas Gustavsson, Bob Hood, David Ikeda, Antti Järvelä, Jarno van der Linden, Marvin Miller, Rob Powers, Prem Subrahmanyam, Jay Roth, Jon Tindall, David Vrba.

### Engineering Contributors:

Richard Brak, Andrew Cross, Josh Darling, Steve Forcucci, Art Howe, Peter Jespersen, Christopher Lutz, David Matthews, James Prentice, Mike Reed, Jarom Schow, William Vaughan, Scott Wheeler, James Willmott.

### Content Assembly:

Leigh Bamforth, Jarrod Davis, Lino Grandi, Luis Santos, Graham Toms, Gildardo Triana

### Installer:

Bob Hood, Dave Vrba

### Documentation:

Marvin Miller, Matt Gorner, Chuck Baker, William Vaughan

### Product Marketing:

Donetta Colboch, David Maldonado

### Special Thanks To:

Don Ballance, Steve Doubleday, Pat Grogan, Tim Jenison, Michael Kornet, Ralph Messana, Rex Olson, Jim Plant, Graham Toms, Andrew Yapple, Franck Lafage, Marc Gaillard, Hiroyoshi Abe, Ron Thornton, Jennifer Hachigian Jerrard, Dave Jerrard, Richard Morton, Reiner Shug, Lee Stringer and to all the Beta Team Members and HardCORE Participants

NewTek, Inc  
5131 Beckwith Blvd - San Antonio,  
TX 78249, USA  
1-800-862-7837 - +1-210-370-8000  
[www.newtek.com](http://www.newtek.com)

NewTek Europe  
Europarc - 17, Av. Léonard de Vinci  
33600 Pessac - FRANCE  
Tel.: +33 557 262 262 - Fax: +33 557 262 261  
[www.newtek-europe.com](http://www.newtek-europe.com)







# Table of Contents



## Contents

---

Copyright and Trademarks.....	II
Acknowledgements .....	III
Table of Contents.....	V





# Volume II . . . . . 1

## Layout . . . . . 1

## Chapter 13: Introduction to Layout 3

Introduction . . . . .	4
The Layout Interface . . . . .	10
The Grid . . . . .	13
Schematic View . . . . .	14
Close/Save Window . . . . .	15
Feedback Agent: . . . . .	15

## Chapter 14: Camera Basics . . . 17

Camera Basics. . . . .	18
Introduction . . . . .	18
Multiple Cameras . . . . .	18
CameraSelector. . . . .	18
Camera Types. . . . .	19
Camera Settings . . . . .	23
Pixel Aspect Ratio . . . . .	24
Frame Aspect Ratio . . . . .	25
Antialiasing . . . . .	27
Motion Blur Effects. . . . .	31
Stereoscopic Rendering . . . . .	33
Depth of Field . . . . .	34

## Chapter 15: LightProperties . . 37

Light Properties. . . . .	38
Lighting Introduction . . . . .	38
Basic Light Attributes. . . . .	38
Lights display size . . . . .	39
Light Exclusion. . . . .	40
Saving Lights . . . . .	41
Light Color and Intensity . . . . .	46
Shadows. . . . .	47
Lens Flares. . . . .	49
Volumetric Lights . . . . .	53
Luxigons. . . . .	55



# Chapter 16: Scene Management

57

File Menu .....	58
Save Commands .....	59
Importing Items into Layout .....	60
Exporting Items from Layout .....	61
Image List .....	67
Content Manager .....	67
Package Scene .....	68
Quit .....	68
Organizing Your Items: List Manager .....	69
Items Tab .....	70
Load Scene .....	70
Add .....	71
Replace .....	74
Delete .....	74
Graph Editor .....	75
Graph Editor: Adjusting the Curve Edit Window .....	79
Channel Bin Pop-up Menu .....	84
Curve Edit Window Pop-up Menu .....	85
Key Pop-up Menu .....	85
Graph Editor: Editing Color Channels .....	85
Graph Editor: Curves Tab .....	86
Expressions .....	90
Graph Editor: Modifier Tab .....	103
Scene Editor .....	107
Scene Editor: Dope Sheet .....	122
Classic Scene Editor .....	128
Utilities Tab .....	131
Commands .....	131
LScript .....	131
Plugins .....	133



## Chapter 17: Object Properties 141

Object Properties .....	142
Object Properties Panel Introduction .....	142
Object Properties: Geometry Tab .....	142
Object Properties: Custom Objects.....	146
Object Properties: Deform Tab.....	152
Object Properties: Displacement Plugins .....	164
Object Properties: Render Tab .....	185
Object Properties: Edges Tab .....	189
Object Properties: Lights Tab .....	191
Object Properties: Dynamics Tab.....	191

## Chapter 18: Previsualization . 193

Creating a Preview Animation .....	194
Layout Transport Controls .....	195
Motion Blur and Depth of Field Preview .....	195
VPR (Virtual Preview Renderer).....	196
VIPER.....	197
Enable Viper .....	197
Surface Preview Mode.....	197
Other Preview Modes .....	198
Preview Options .....	198
Virtual Studio .....	199
Visor.....	202

## Chapter 19: Rigging ..... 205

Setup Tab.....	206
General.....	206
Add .....	207
Modify .....	209
Detail.....	216
Edit.....	218
Motions Options Panel .....	224
Controllers and Limits Tab .....	236
Bone Properties Panel.....	237
Use Bones From Object.....	237
Clear All bones .....	237
Falloff type .....	237
Use Morphed Positions.....	237
Faster Bones .....	237
Current bone .....	237





Bone Active.....	238
Rest Position.....	238
Rest Rotation.....	238
Rest Length.....	238
Bone Weight Map.....	238
Use Weight Map Only.....	238
Weight Normalization.....	238
Strength.....	238
Multiply Strength by Rest Length.....	238
Limited Range.....	238
Joint Compensation.....	239
Joint Comp for Parent.....	239
Muscle Flexing.....	239
Parental Muscle Flexing.....	239
Twist.....	239
IK Boost Tool.....	239
IKB Menu Options.....	241
IK Booster and the Dope Track.....	242

## Chapter 20: Animation..... 253

Animation.....	254
Keyframing.....	254
Navigating a Scene.....	257
Playing a Scene.....	257
Creating Keyframes.....	258
Creating and Modifying Keys Automatically.....	258
Editing Motion Paths Directly in a Viewport.....	259
Deleting Keyframes.....	259
Delete Motion Key Plugin.....	260
Saving and Loading Motion Files.....	261
DopeTrack.....	262
BioVision Motion Capture Support.....	265
Keyframer.....	266
Modify Tab.....	268
Translate.....	268
Rotate.....	271
Transform.....	272
General.....	273
Tools.....	274
Motions.....	277
Motion Mixer.....	277
Relativity Expression Module.....	286



# Chapter 21: Dynamics. . . . . 301

Introduction . . . . .	302
Adding Dynamics to your Scene and Objects . . . . .	302
Object Properties: Dynamics Tab . . . . .	304
Cloth Dynamics . . . . .	305
Soft Body Dynamics . . . . .	312
Hard Body Dynamics . . . . .	317
ParticleFX (Emitter) . . . . .	322
Dynamics: Collision Controller . . . . .	328
Dynamics: Gravity Controller . . . . .	331
Dynamics: Wind Controller . . . . .	332
Additional Dynamics Utilities . . . . .	334

# Chapter 22: Environments . . 345

Backdrop Options . . . . .	346
HyperVoxels . . . . .	352

# Chapter 23: Customizing Layout377

Customizing Layout . . . . .	378
Edit Drop Down Menu . . . . .	378
Undo . . . . .	378
Edit Menu Layout . . . . .	378
Window Config Loader . . . . .	379
Layout General Options . . . . .	380
Layout Display Options . . . . .	383
OpenGL Options . . . . .	388
Schematic View Tools . . . . .	395
View Tab . . . . .	396
Viewports . . . . .	396
View Layout . . . . .	397
Layout Grid Tools . . . . .	397
Layout Selection Tools . . . . .	398
Customize Color . . . . .	399

Glossary . . . . .	i
Glossary of 3D Terms . . . . .	ii







---

## Volume II

---

## Layout

---



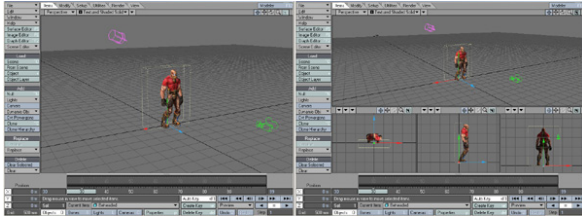


## Chapter 13: Introduction to Layout



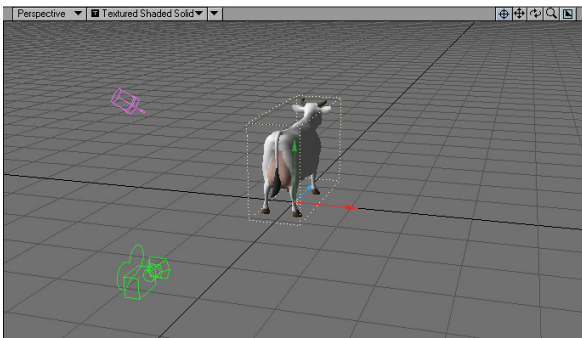
## Introduction

LightWave's Layout has been designed to provide the most efficient interface possible for your 3D animation work. By default, there is a single large viewport, but you can display multiple viewports if you desire. A viewport provides you with visual feedback about the virtual world you are creating. How well this corresponds to what the final output will look like is completely configurable by you. This can range from bounding box stand-ins to wireframe representations, all the way to textured and solid-shaded displays. How you view your creations will vary depending on their complexity, your machine's capabilities, and other factors.



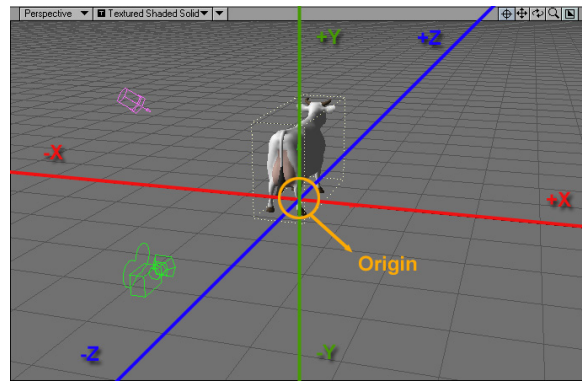
## LightWave's Virtual World

LightWave's world is defined using three axes: X, Y, and Z. Here we've loaded the **Cow** object and haven't rotated it. At its default position, from front to back, the cow's body is aligned along the Z axis and is facing the positive Z direction. The X axis runs left to right, with right as the positive side. The Y axis runs up and down, with up as the positive side.



It is common to see objects that have a front and back (e.g., vehicles, spaceships and animals) facing in the positive Z direction. As you will understand later, this orientation works best with LightWave's motion features.

The center of the world, called the *Origin*, is defined by the XYZ coordinates 0, 0, 0 and represents the intersection of all three axes. Any position in LightWave's virtual world can be defined by positive and negative XYZ values. The cow is standing right over the Origin. (The **Size** and **Stretch** tools also use XYZ (scaling) values along those axes; note that these are applied relative to the object's **Pivot Point**, also known as the *Local Origin*, discussed in the next section, rather than the *World Origin*.)



## World and Local Axes

Objects in a scene also have axes and, thus, an Origin, called the local Origin. When an object is first loaded, its local Origin is lined up with the world Origin. Moreover, its local axes are lined up with the world axes. However, once you move or rotate the object, this is no longer the case. LightWave provides functions that let you move and rotate items using global or local axes. Now, most of the time you'll use World, but sometimes using local will be invaluable.

To illustrate the difference, let's say you are standing in the middle of a one-room house facing the front door. If you held your right arm straight out, it would point to the right side of the house, and your left arm would point to the left side of the house.

Now, let's say you turned 90 degrees clockwise. (You'd be facing the right side of the house.) If I told you to point your right arm towards the house's right side (global axes), you'd move it straight out in front of you. However, if I told you to point your right arm to your right (local axes), you would point to the back of the house.

## Your Point of View

By default, Layout uses a single viewport. Later on in this section, you'll learn how to use up to four simultaneous viewports. You can choose between several different points of view (POV) for each viewport using the View Type pop-up menu at each viewport's top-left corner. Manipulating items in virtual 3D space on a 2D display (i.e., your monitor) can be difficult at times, so you will switch between nearly all of these as you edit your scene.



It is sometimes easier to work in just two dimensions at a time. The options with the axis notations (e.g., **Top (XZ)**) are the "orthogonal" views, which let you move items in only two dimensions (horizontally or vertically), along the XY, XZ, or ZY axes. The **(none)** setting blanks out the viewport. Note that there are two options for each axis set. This allows you to look in either direction along the perpendicular axis (e.g., **Top (XZ)** and **Bottom (XZ)**). For these, Y is the perpendicular axis.)



The Perspective view is a *forced-perspective* view. That is, it gives you a three-dimensional look at your scene.

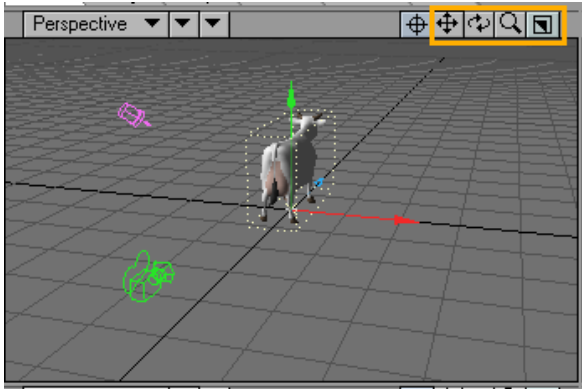


NOTE: The orthogonal and the Perspective views are *dependent on each other*. Changing the position of one will affect the other.

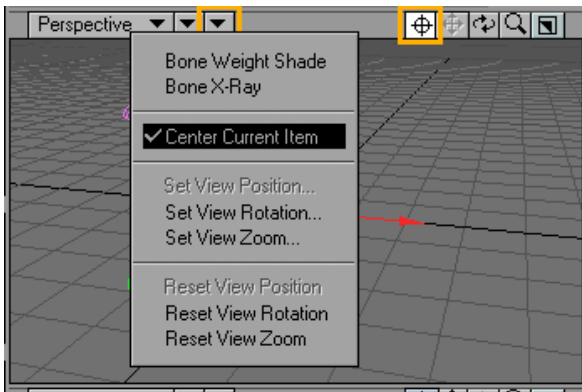
There are also *pseudo-physical* POVs. When setting up a light, you'll often want to *look through it* to see exactly what it points at. In such a case, you'll use the **Light** view to look through the current light. You'll always want to see your scene from the **Camera** view at some point since that is the perspective used in your rendered images.

## Changing Your Point of View

With the **View Control** drag buttons located on the upper-right edge of a viewport, you can interactively alter the orthogonal and perspective POVs. The buttons are dimmed to indicate that they are not available when you use the Light or Camera views, where they have no effect since those are based on their respective item's POV in the scene itself.



**Center Continuously** centers the viewport on the selected item. You may also activate the **Center Current Item** option on the pop-up menu next to the view selector.



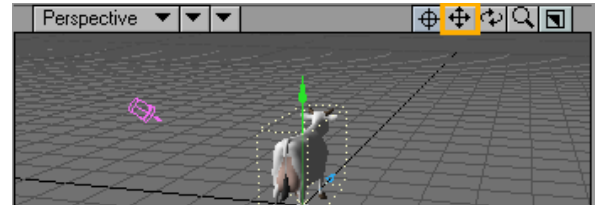
For an object, the centering is based on its pivot point, which is discussed later. This is not always the center of an object. If you deactivate this mode, the existing POV position will remain until changed. As such, you can use this feature to establish a starting point if the need arises.

## Move

**Orthogonal view:** Moves your POV horizontally when you drag left or right and vertically when you drag up or down.

Keyboard shortcut: **ALT**

**Perspective view:** Moves your POV horizontally when you drag left or right and farther/closer when you drag up or down with the LMB. Moves your POV vertically when you drag up or down with the RMB.



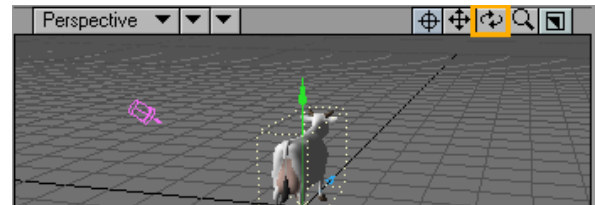
NOTE: Since you are changing your POV, the scene items will appear to move in the opposite direction of your mouse movements.

## Rotate

**Orthogonal views:** not applicable.

**Perspective view:** Rotates your POV's heading when you drag left or right and its pitch when you drag up or down with the LMB. Rotates your POV's bank when you drag left or right with the RMB.

Keyboard shortcut: **Alt**



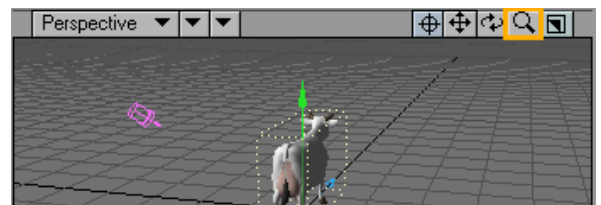
## Zoom

**All views:** Zooms in and out when you drag left and right. (You can also use the < and > keys.)

Keyboard shortcut: **Ctrl + Alt**



HINT: Use the keyboard shortcut "a" to zoom the view out to where all items in the scene are visible.



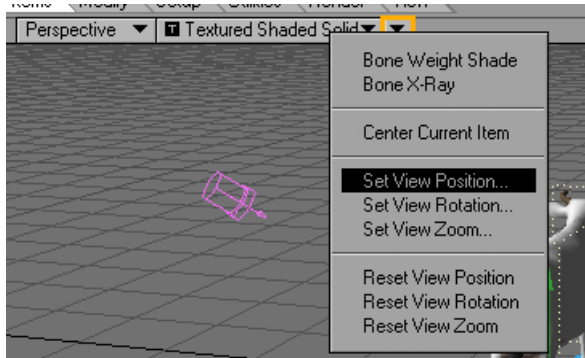




## Taking Aim

Each orthogonal and perspective view is based on a single *aimpoint*. In other words, you are always looking at the same point in 3D space, no matter which view type you use in that viewport (except **Light** and **Camera** view, of course.) That point is also the center of the view rotation. That's why, if you move around in the **Back** view and then change the **View Type** to **Perspective**, you'll find the view has moved in that view type as well.

The position, rotation (affects only **Perspective** mode), and zoom of each viewport can now be specified numerically using the **Set View...** menu items in the viewport titlebar.



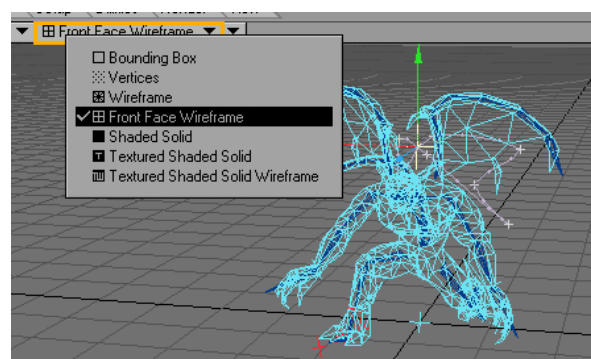
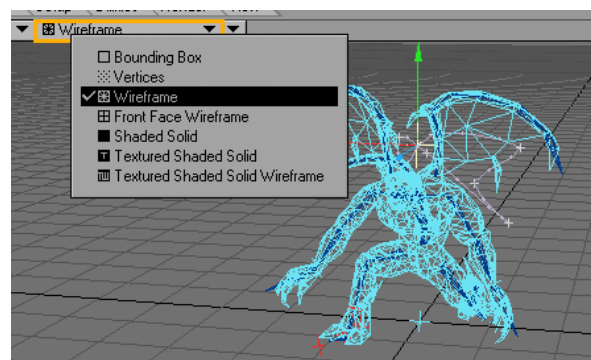
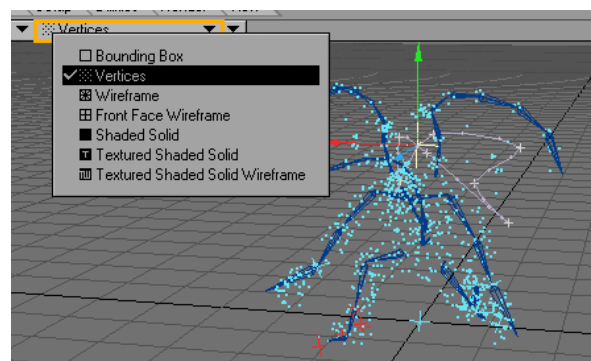
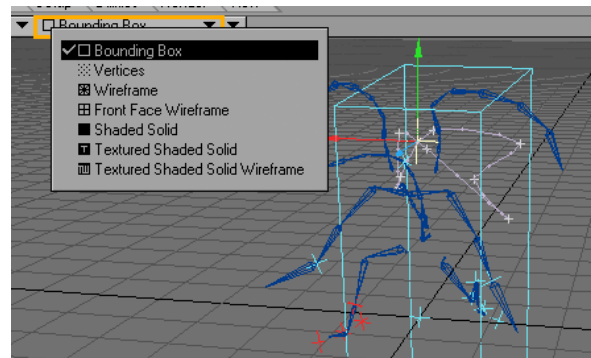
NOTE: If you are using multiple viewports, each has its own independent aimpoint and set of position, rotation, and zoom values.

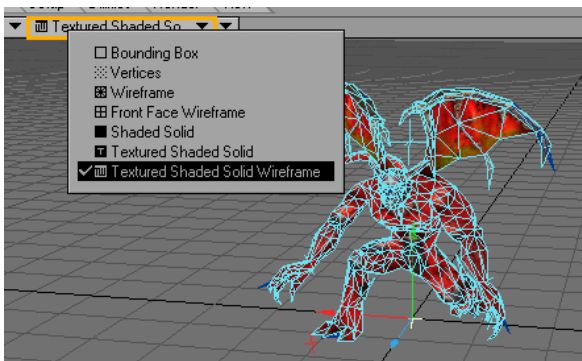
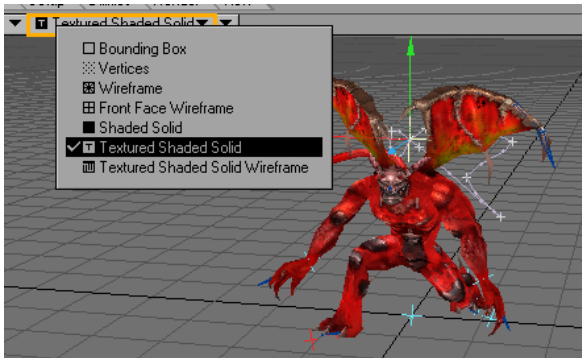
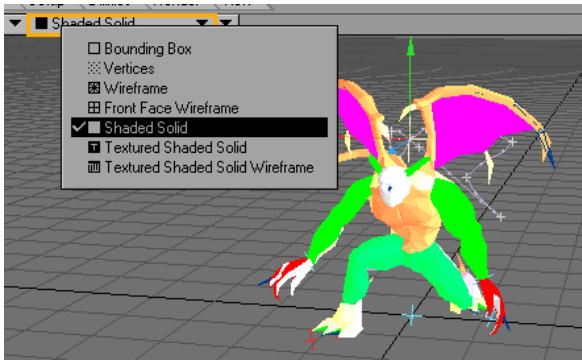
## Resetting Views

Also on the pop-up menu, above, are options to reset a view's position (Move), rotation, and zoom to default values.

## Viewport Display Mode

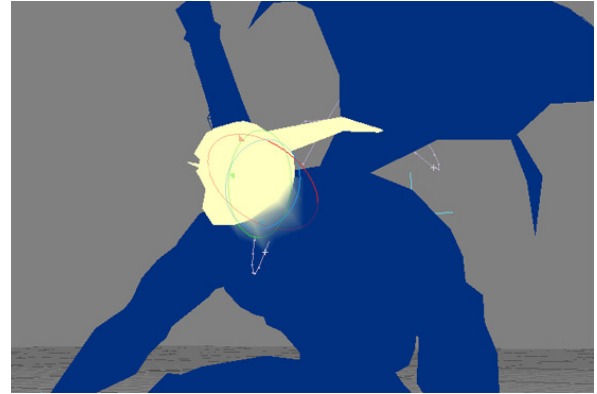
You can also set the display mode used by the viewport using the **Viewport Options** pop-up menu next to the View Type selector. This is much faster than using the **Scene Editor Panel**.





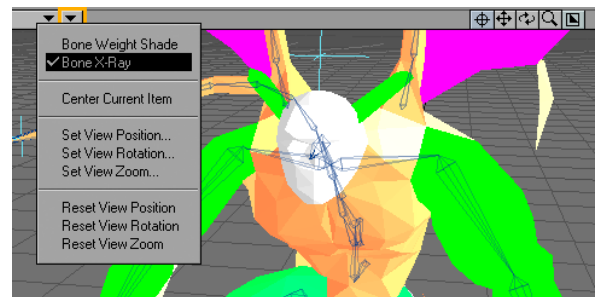
## Bone Weight Shade

Activating **Bone Weight Shade** in the Viewport Options pop-up menu will show the selected bone's influence range in any shaded viewport. The influence coloring is based on each bone's color, which can be changed in the **Scene Editor**. A bright yellow is used for the currently selected bone. Note that the bone must be active to see this effect. This mode will override the normal texture display.



## Bone X-Ray

There is an option to make bones visible inside opaque objects. To use, activate the Bone X-Ray option on the titlebar. Note that bones may still be invisible if the object surface is similarly colored/shaded. Change the bone colors using the **Scene Editor**, if this becomes a problem.





## Recipe for a Scene

LightWave animations (or still images) always start as a *Scene* — basically, a collection of objects, lights, cameras, and images, which can move and change over some specified length of time. Creating a basic LightWave scene involves the following steps:

- Adding items (e.g., objects and lights) to a scene
- Setting the starting position for all items in the scene
- Setting the length of the scene
- Placing items in key positions at certain points in time
- Previewing the motions of the items
- Setting and testing render settings
- Rendering the final animation

## Selecting an Item in Layout

Usually, you work on one item at a time, the *current item*, and you need to tell LightWave which item it is. But before you learn how to do that, you need to know that Layout items are grouped into four different types: objects, bones, lights, and cameras. When you work on any item, the edit mode buttons along the bottom (i.e., **Objects**, **Bones**, **Lights** or **Cameras**) are set to the current item's type.

### To select an item:

There are several ways to select an item in Layout:

Click on the item in a viewport;

Click on the item's name in the **Scene Editor Panel (Scene Editor)**; or

Manually select the edit mode and then select the item from the **Current Item** pop-up menu.

Note that you cannot select a locked item (a little lock icon appears next to name).

Use the Item Picker master plugin

You can select items in a viewport by clicking on any polygon edge rather than just on a pivot point.



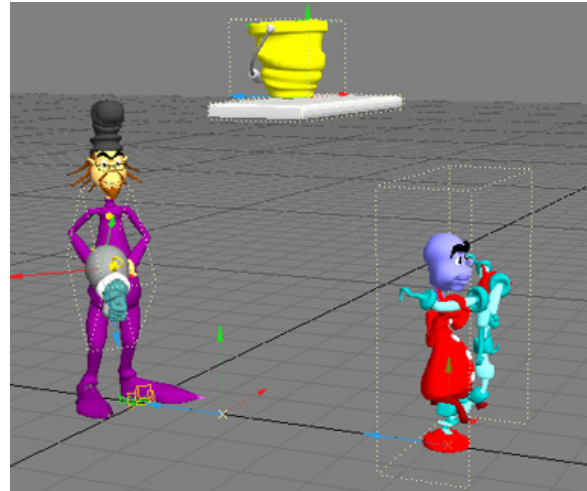
**NOTE:** You can use your UP and DOWN cursor keys to cycle through the **Current Item** list. All items, except objects, are highlighted in yellow when selected. When an object is selected, a dotted-yellow bounding box will appear around it.



**NOTE:** You can select a bone by clicking near its midpoint, rather than its pivot point, making it possible to pick different bones that branch from the same point in a hierarchy.

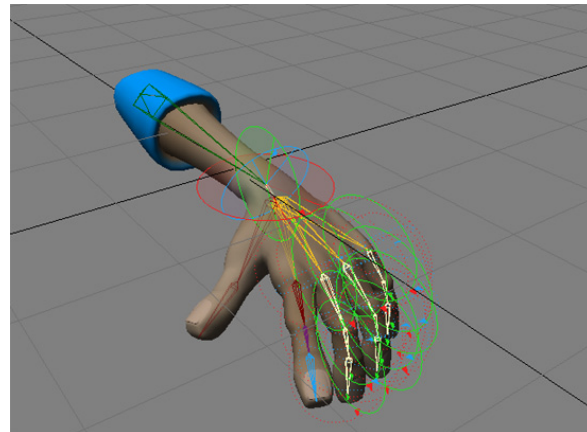
## Selecting Multiple Items

You can select multiple items of the same type, like all objects or all lights, and perform edits on them simultaneously.



Holding the **Shift** (or **Ctrl**) key will allow you to select multiple items of the same type.

This can save a lot of time under the right circumstances. Such operations include Move, Rotate, and Size, as well as certain item properties, like Unseen by Rays, Unseen by Camera, Self Shadow, Cast Shadow, Receive Shadow, Bone Active, Affect Diffuse, Affect Specular, Affect Caustics, Affect OpenGL, and so on.



Multiple bones in the hand object selected and rotated

If the toolbar command is a state-type command and the selected items have mixed settings, the button will be shaded diagonally. Clicking the button will toggle the state of the current item and make all other selected items the same. Clicking again will toggle the state of all items.

## Selecting by Name

Pressing the apostrophe (') key launches a special selector dialog. Simply type-in a few characters that uniquely identify the desired item and click **OK**. You can select any type of item.

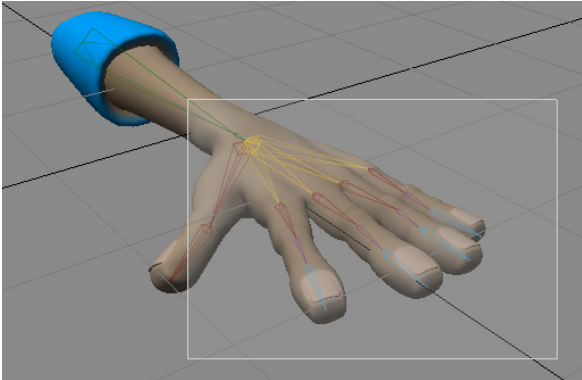


## Unselecting Items

In Layout, one item is always selected. It becomes unselected when you select a different object.

### Item Selection: Bounding Box

A selection bounding box can be created by dragging your middle mouse button in any viewport. Items whose points appear within the box will be selected, and if the Shift key is held down, the items will be added to the existing selected items, if they are the same type. If items of more than one type are within the box, preference is given to those items matching the current edit mode. In addition to the viewports, this also works in other areas of the interface, such as the **Graph Editor**.

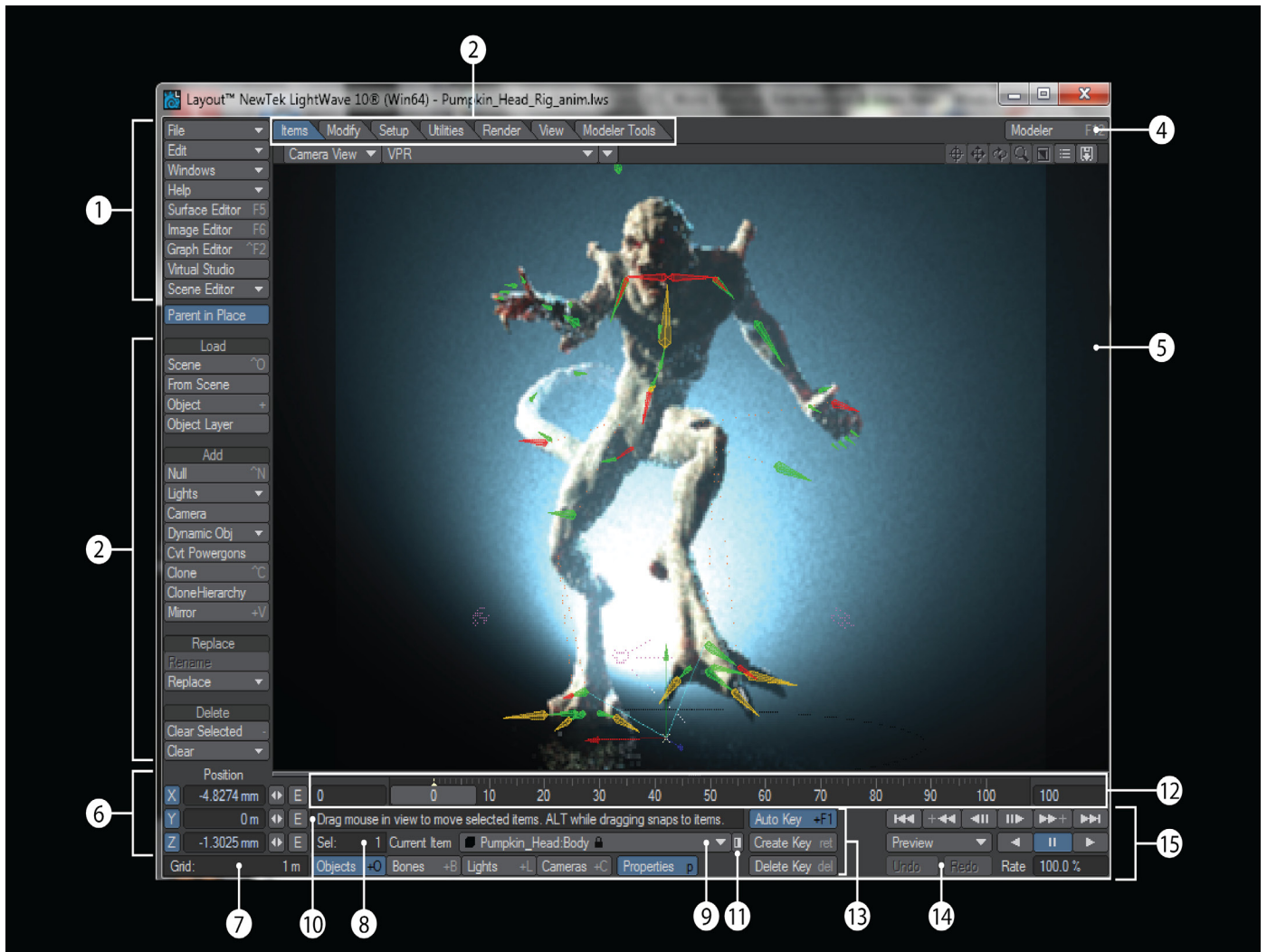


HINT: If you have a scroll wheel, it may also function as a middle mouse button if you press down on it. This obviously depends on the make and model of your mouse.



## The Layout Interface

LightWave's Layout has been designed to provide the most efficient interface possible for your 3D animation work. By default, there is a single large viewport, but you can display multiple viewports if you desire. A viewport provides you with visual feedback about the virtual world you are creating. How well this corresponds to what the final output will look like is completely configurable by you. This can range from bounding box stand-ins to wireframe representations, all the way to textured and solid-shaded displays. How you view your creations will vary depending on their complexity, your machine's capabilities, and other factors.



1. Top Tool Bar, 2. Main Tool Bar, 3. Menu Tabs, 4. Modeler Access, 5. Workspace, 6. Numeric Adjustment Controls, 7. Grid Display, 8. Selection Info, 9. Current Item and Item Properties, 10. Tool Tips, 11. List Manager, 12. Time Line and Dope Track, 13. Keyframe options, 14. Undo/Redo, 15. Preview Controls.





## 1 and 2. ToolBar

The toolbar sits at the side of the screen. The buttons presented will vary depending on which menu tab you select along the top. The **Top Tool Bar** buttons will appear no matter which Tab is selected. You can completely hide (or unhide) the toolbar by pressing **Alt+F2**.

## 3. Layout Menus

The **Menu Tabs** located at the top of the interface will determine which tools appear in the Tool Bar. Generally, menu group names that are verbs contain commands based on the type of action they perform. Menu group names that are nouns contain commands based on the type of object they affect. The menu tab names are all intended to be read as verbs.



Items — Add objects, lights, and other items to your scene here.

Modify — Move, Rotate, Scale, and more.

Setup — Bone tools and other rigging tools.

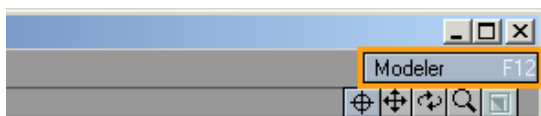
Utilities — Lscript and Plugins Access.

Render — Render Options, VIPER access and more.

View — Display options and selection tools can be found here.

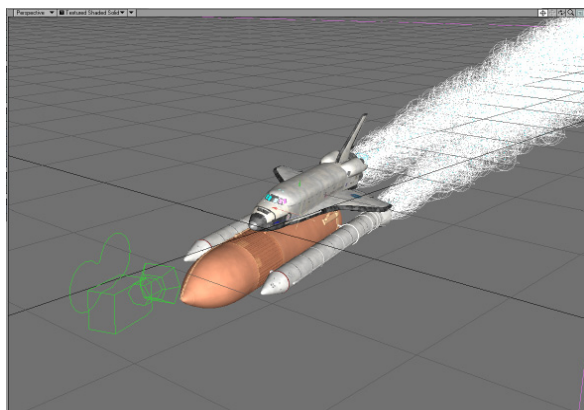
## 4. Modeler Access

This command will switch to your Modeler window if it is open or opens Modeler if you do not already have it open.



## 5. Workspace

The Workspace consists of Viewports. By default, there is a single large viewport, but you can display multiple viewports if you desire. A viewport provides you with visual feedback about the virtual world you are creating.



## Viewport Titlebar

You can change the **View Type** and **Rendering Style** of a viewport without going to the **Display Options Panel** by using the pop-up menus on the left side of each viewport titlebar. You can also activate Weight Shade and Bone X-ray modes.



The buttons on the right side affect centering, panning, rotating, zooming, and minimising/maximising your view.



## 6. Numeric Adjustment Controls

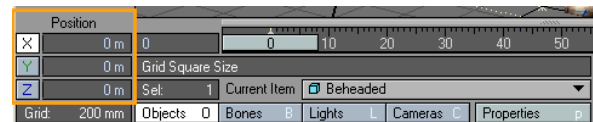
You can change **Position** and **Rotation** settings numerically by entering the desired values in the **XYZ** or **HPB** fields, located in the lower left corner. The function of these fields changes depending on the editing being done.



## Protecting from Changes

Whether you are moving, rotating, or stretching an item, or moving its pivot point, Layout lets you independently deactivate any of the three components used for those settings. You deactivate the components by clicking on them on the information display in the lower-left corner of the screen.

This protects the deactivated component(s) from changing as you move your mouse.



**NOTE:** You can also confine changes by using any of the orthogonal views (i.e., Front, Top, and Side). In these views, you can generally make changes only along the display's horizontal and vertical axes using your mouse.

## 7. Grid Display

The grid serves as a visual reference when you move items around, but it will never render in a final image (no matter how much you wish it to). The grid lines are darker every tenth square for visual reference. The Origin is located at the center of the grid.

You can find the current size of the grid squares in the information field at the very lower-left corner of the screen. The size of each grid square is adjustable as is its overall size on the **Display Options Tab** of the **Preferences Panel** (**Edit > Display Options**). If you adjust the size of the grid squares you also adjust the relative size of the camera and the lights within the scene.



## 8. Selection Info

The Selection Info displays how many items are selected.



## 9. Current Item and Item Properties

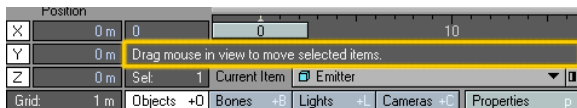
Usually, you work on one item at a time, the *current item*, and you need to tell LightWave which item it is. But before you learn how to do that, you need to know that Layout items are grouped into four different types: objects, bones, lights, and cameras. When you work on any item, the **Edit Mode** buttons along the bottom (i.e., **Objects**, **Bones**, **Lights** or **Cameras**) are set to the current item's type.

Once you select the **Current Item**, click the **Properties** button to view the items properties based on the **Edit Mode**.



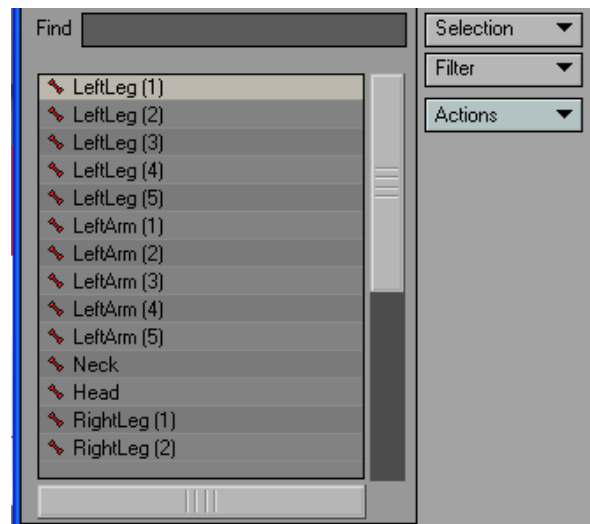
## 10. Tool Tips

Displays helpful information for current tool.



## 11. List Manager

Located directly to the right of Current Item pulldown menu, The List Manager allows for quick and easy management of the items in your scene.



## 12. Timeline and Dope Track

The Timeline, located beneath the viewport area uses the Frame Slider which can be dragged to change the current frame or shuttle through your scene. The input field to the left of the timeline is the first frame in your scene and the input field to the right is the last frame. You can change these settings by simply changing the values in either field.

The **DopeTrack** allows you to modify keys. You can adjust not only their position in time and the scene, but also alter their properties. You can set up markers here to tell you when things should happen in the scene and bake ranges of keys to make your animation concrete. You can open or close the DopeTrack by clicking on the thick bar above the time line.



## 13. Keyframe options

To automatically create or modify keys you must activate the **Auto Key** option on the main interface. This is the global on/off switch for automatically creating keyframes. It works in conjunction with the **Auto Key Create** setting (**General Options Tab** of the **Preferences Panel**).

Use the **Create Key** and **Delete Key** commands to add and remove keyframes.







## 14. Undo/Redo

The **Undo** function will reverse motion changes (**Move**, **Rotate**, **Size**, or **Stretch** tool) that you have made. The **Redo** function will re-execute a sequence of actions that have been undone, beginning with the action most recently undone. You can set the number of undo levels that you wish to have available in the **General Options** tab of the **Preferences Panel**, discussed later, in the **Edit Menu** section. If there are no actions available to undo, the **Undo** button will be ghosted. The **Redo** button is ghosted if there are no actions available to redo.



**HINT:** If you are not using the Auto Key feature, you can quickly reset all aspects of a frame (i.e., position, rotation, etc.) if you haven't created the key yet. Just press the Right cursor key and then the Left cursor key. This advances the current frame and then goes back to the original frame. The frame will return to the last keyframed state, or if the frame is not a keyframe, to its in-between state.

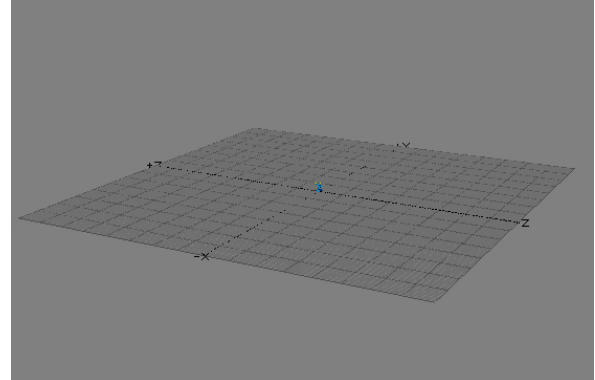
## 15. Preview Controls

Besides dragging the frame slider to navigate through your scene, you can also use the transport controls. In addition to playing a scene using the transport controls, you can also create special preview animations. In most cases, these will give you a more accurate preview of your final animation.



## The Grid

A grid of squares, made up of an planar axis at 0,0,0, is visible in any of the orthogonal views, as well as the Perspective view. The grid serves as a visual reference when you move items around, but it will never render in a final image. The grid lines are darker every tenth square for visual reference. The Origin is located at the center of the grid.



You can find the current size of the grid squares in the information field at the very lower-left corner of the screen. The size of each grid square is adjustable as is its overall size on the **Display Options Tab** of the **Preferences Panel** (**Edit > Display Options**).



### The Grid and Relative Camera/Light Sizes

The size of lights and cameras (that is, how they appear in the viewport) are relative to the size of the grid squares, with the exception of Area Lights and Linear Lights, which are independent of the grid square size. If you have very large grid squares, you will also have very large lights/camera compared to objects, and vice versa.



**NOTE:** The size of the lights/cameras do not affect their functionality.

### The Grid Square Size Effect on Positioning

The **Grid Square Size** (**Display > Options: Display Options**) also determines the incremental change as you drag your mouse. Thus, a smaller size lets you edit your object's position with greater accuracy than using a larger one. If you find that you can't edit an object with the accuracy required, try lowering the **Grid Square Size**. However, this will also affect the Orthogonal and Perspective view modes.



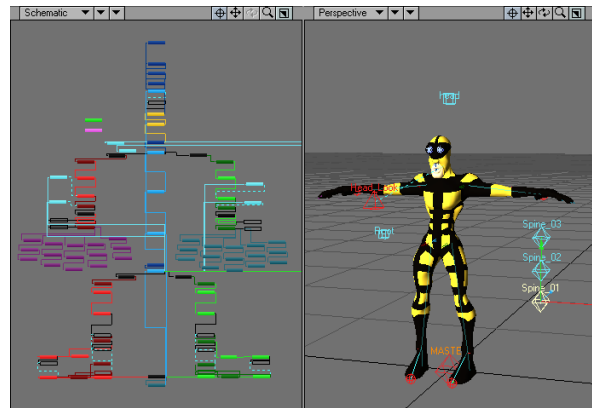
## Grid Square Size Auto-Adjustment

When you start creating a new scene, your **Grid Square Size** will automatically adjust itself upwards only, if necessary. This can be problematic when you use objects that differ significantly in relative size, like planets and spaceships. Objects may seem to disappear, when in reality they are just too small or too big to see in the viewport.

Once you manually set the **Grid Square Size** or save and reload a scene, the automatic sizing adjustment is deactivated. As such, you may want to load the smaller objects first and then manually change the **Grid Square Size** to the same value. Then, load the larger objects.

## Schematic View

The Schematic viewport type is a two-dimensional view showing all items in the scene as rectangles that can be selected and moved into any arrangement. You access this mode by selecting **Schematic** from the **Viewport Options** pop-up menu on a viewport's titlebar.



Each rectangle appears in the associated item's wireframe color, and hidden items are shown as outlined instead of filled rectangles. Solid lines link children to their parents and end effectors are connected to their goals with dashed lines.



NOTE: Also see the **Display Options** and **Schematic View** tool generic plugin.

Navigation is consistent with other 2D viewports (e.g., **Alt/Alt+Shift** =move, **Ctrl+Alt**=zoom, **Center Current Item**, etc.). Press the (a) key to fill all items into view or (g) key to center.

### Parenting in Schematic View

Parent items can be assigned by holding your Ctrl key down and clicking on the desired parent for the current item. Ctrl-clicking in a blank area *unparents* the item.

### Other Schematic View Options

When you right-click an item, it displays a pop-up menu. This menu can clear, clone, rename, open the **Properties Panel** for the item, and set the item's wireframe color.



## Close/Save Window

**Show Changes Only** shows only items that have had changes applied in the work session.

**Scene File Version** will save the file in the version selected. The version 6.0 covers versions 6.0 through 9.0.

**Save All Changes** will save all changes made in a session.

**Items** shows the items in a scene. Clicking on the arrow will expand or collapse the menu tree.

**Mod** shows which items have been changed.

**Save State** has different save options, which are available by right-clicking on the Save State section.

Scene files: Do not save, Save, Save As

Object files: Do not save, Save, Save As, Incremental

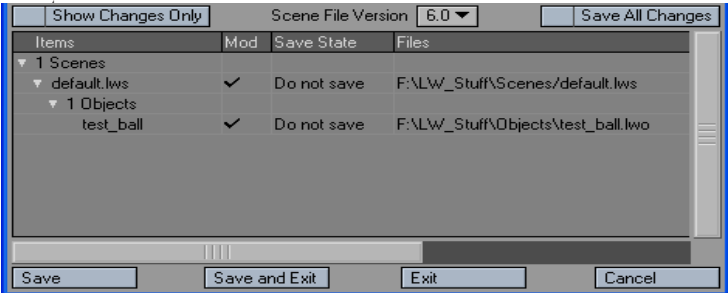
**Files** is the file path where the items will be saved. Double clicking there will open a save options window.

Save will save based on the options chosen above.

Exit will exit without saving.

Save and Exit will both save based on the options above and also exit Layout.

Cancel will close this window without performing any of the above



## Feedback Agent:

The Feedback Agent helps users to provide the 3D Team with the most possible information about LightWave issues they encounter. It can create a log of the problem or crash situation, and can email the log to NewTek's bug reporting system.

Notes on use of the Feedback Agent

- \* Please Edit the "Exception in LightWave" portion of the default subject line to include a description of the problem, such as "Crash in Modeler when hit all keys on keyboard at once."

- \* Please be sure to list steps to duplicate the problem in the "Brief Description" field.

- \* Please attach the content with which the problem was encountered.

- \* You will not usually have to fill in the "To:" field. Feedback Agent knows to send to the LW Bugs address if this is left blank.

- \* You may need to fill in your smtp-server information if Feedback Agent does seem to be able to send out email.





---

## Chapter 14: Camera Basics

---



## Camera Basics

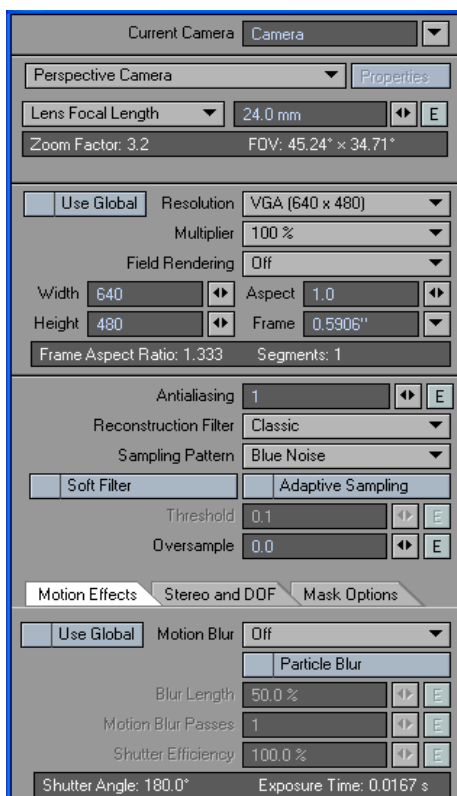
### Introduction



The camera is the conduit for your imagination to a final image. When creating your render you need to decide what it will be used for, what its destination will be. Will it be an image for film or broadcast, print or computer screen?

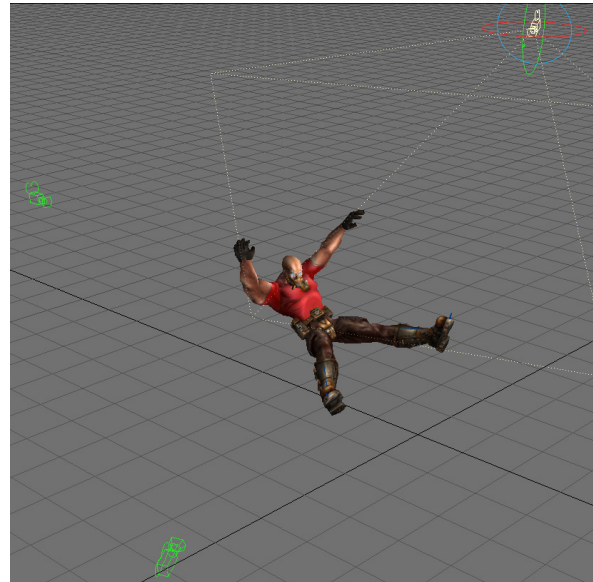
Whatever your image's destination, there are two defining characteristics for it. One is how big it is, the other is the shape of the pixels that make it up.

You can make all these decisions in the **Camera Properties** window, which will appear when you pick a camera and hit the **P** button on your keyboard, or the **Properties** button on your LightWave screen.



## Multiple Cameras

You can have as many cameras as you like in a scene (well, there's a standard limit at 100, but you can increase this in the LW9.CFG config file to an absolute maximum of 32,000). However, you can only render the current camera view. You add extra cameras into the scene by clicking on the **Items > Add > Camera** button, or you can clone existing ones (with Ctrl C). If you have more than one camera in a scene, you can choose the camera you wish to be active, when in Camera mode, either in the **Camera Properties Panel** or the **Current Item** button under the main Layout window.



## CameraSelector

This Master plugin allows you to switch between different cameras in Layout. The current camera is still the one being rendered but this can be used to preview camera switches in-scene or in a Preview.

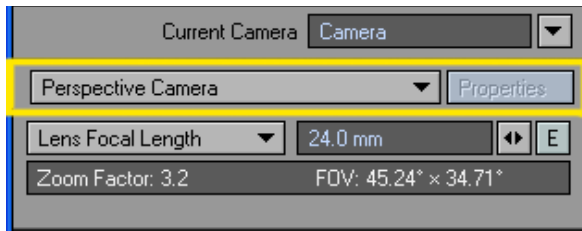
To access the CameraSelector, click on the **Master Plugins** button in the **Utilities Tab**. This will bring up a window with a drop down menu in which you can select **CameraSelector**. Double clicking on its entry in the Master Plugins list window will open the CameraSelector plugin's window. To create a list of camera changes all you need do is to scrub through your scene selecting the camera you want for each shot. Simply click on the **Add** button in the CameraSelector window for each camera change you wish to take place. If you then start playing your scene, you will see that the camera switches at the time you stated. If you wish to remove a camera change, select the change you want to get rid of and hit **delete** in the CameraSelector window.

If you would like to see your scene play through just one camera view, you can turn off Camera Selector by clicking the **Enable Dynamic Preview** switch, during scene playback.

For CameraSelector to work its magic, it needs information not normally available to master plugins. This is why it automatically creates a null object called "SpecialTriggerNull" whose only purpose in life is to serve CameraSelector. If you wish to use a different object, you can select it on the **Trigger** drop down menu but, normally, there is no reason to do so.



## Camera Types



## Classic Camera

Classic Camera refers to the only camera type available previous to LightWave 9.

## Perspective Camera

Similar to the Classic Camera, however there are some differences to note. The Perspective camera renders from the top of the frame down and complex scenes tend to render faster.

## Zoom Factor

The **zoom factor** drop down menu allows you to set a zoom factor equivalent to a real world camera lens. It has four different types of zoom factor for you to choose from, but LightWave defaults to a zoom factor of 3.2, equal to a 24 mm lens. LightWave users that are familiar with real world camera equipment may find that using the Lens Focal Length type on the drop down menu is the easiest to use. Those who are solely used to LightWave's way of doing things may be more comfortable using the Zoom Factor type. You can also use the **Horizontal and Vertical FOV (Field of View)** settings. These set the degree of angle of view.

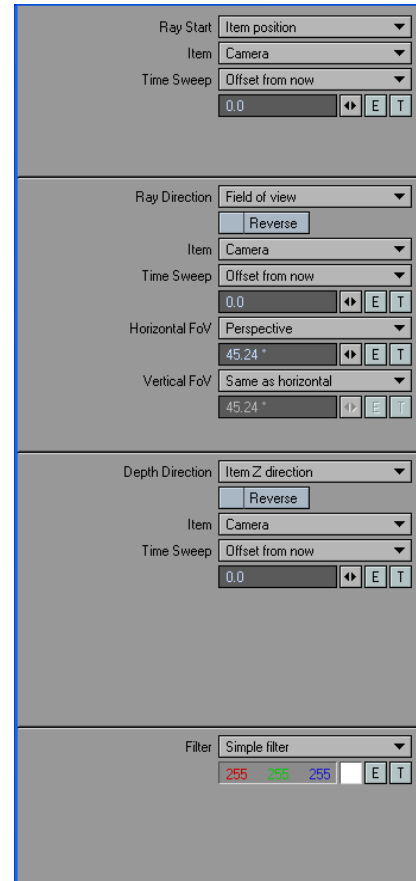


Left:138 mm lens, Right: 8 mm lens

**Smaller Zoom Factor or Lens Focal Length** values will produce a wider angle lens effect while larger values give a narrow field of view, similar to a telephoto lens effect. You can create an envelope to achieve effects such as reverse zooms where you pull the camera away while concentrating the field of view. The envelope will be based on **Zoom Factor** regardless of which mode on the drop down you choose to use.

## Advanced Camera

The advanced camera is a multi-purpose camera. With the advanced camera, you can recreate real cameras and lenses. You can shift the lens distortion over time. Custom lenses can be created by using a mesh object.



**Ray Start**—This item defines the starting position for every Ray in the scene(world coordinate based)

-Item Position: this is used for pinhole type cameras(like the standard LW camera), that is a single point in space as a ray start position.

-Item: the item in the scene to be used as a camera

-Time Sweep: this can be used to control when in time to read the item position.

-UV Position on a mesh: camera plane defined as a UVmap on a mesh, I would probably recommend trying to stay within a semi square plane as a starting point as you can get highly unpredictable results otherwise, the mesh itself can be animated in any manner(quite fun to play with)

-Mesh: well the mesh used for the camera plane.

UV Map: Select the UV map you want to use

Item XY: X and Y start position is taken from the Items coordinate system

Item: what item to use as ray start





Time Sweep: same as the item position Sweep function.

Custom: definable with XYZ using numerical input, envelopes or textures

Ray Direction

This defines where the rays should go once they leave the starting position, given in world coordinates.

Reverse: The direction of the ray is reversed.

Towards Item Position: Each ray goes through the origin of the item.

Towards UV Position on a mesh: Each ray goes through the matching UV coordinates on a mesh.

Mesh UV Polygon Normal: same as above but goes through the polygon normal instead of UV coordinates.

Mesh UV smooth normal: same as above but with normal smoothing.

Field of View: Behaves mostly like a normal camera, aligning the rays to move along the Z axis in the FOV defined by the settings. This has a couple of sub modes you can play with as well:

-Perspective: Standard perspective view with definable X and Y FOV

-Orthographic: Planar projection mode, area rendered is defined by size of the ray start item.

-Cylinder: Y axis is orthographic while the X axis roams free, this is useful for making things like panoramas.

-Spherical: Produces a similar effect as a fisheye lens. The vertical settings will be disabled. The horizontal value controls how many degrees the fisheye lens covers. Note that this mode affects both horizontal and vertical FoV.

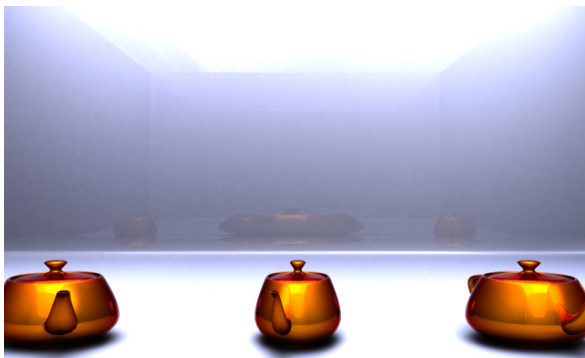
The field of view has a default value of 45.24.

Through item XY: rays set to go through item XY, dependant on ray start.

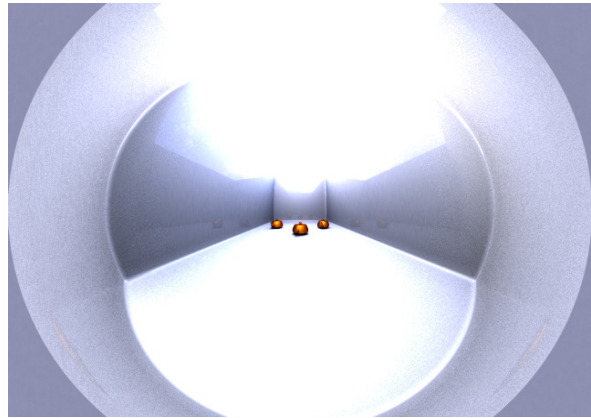
Custom: Definable with XYZ using numerical input, envelopes or textures.

Orientation Reference: Determines the orientation of the rays in relation to the selected item.

Example of Time Sweep



Example of Spherical Camera



Example of Camera using UVMesh



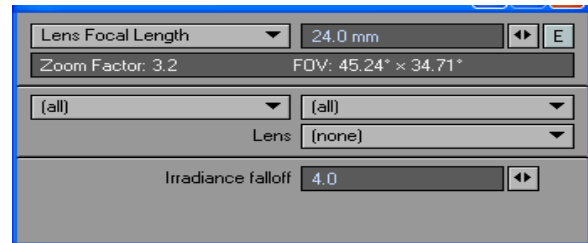




## Orthographic Camera

For the Orthographic Camera, the direction is the same for all rays, and the position is determined by the location of the corresponding pixel on the imaginary screen.

## Real Lens Camera



The Real Lens Camera Setting in the Camera Properties Panel will simulate a physical camera lens.

## Zoom Factor Menu

The zoom factor drop down menu allows you to set a zoom factor equivalent to a real world camera lens. It has four different types of zoom factor for you to choose from, but LightWave defaults to a zoom factor of 3.2, equal to a 24 mm lens. LightWave users that are familiar with real world camera equipment may find that using the Lens Focal Length type on the drop down menu is the easiest to use. Those who are solely used to LightWave's way of doing things may be more comfortable using the Zoom Factor type. You can also use the Horizontal and Vertical FOV (Field of View) settings. These set the degree of angle of view.

## Camera and Lens Menus



The first filter (top-left) will select the manufacturer of the camera. The second (top-right) filter will select the camera body. The third (bottom-right) filter will select the lens type.



Note: If the first two filters are left at the default "All" selection, the Lens filter will show all lens types.

**Irradiance falloff** simulates the darkening towards the boundary of the image, much like a real camera. What is happening on a technical level is the brightness of a pixel is reduced as a function of the angle between the ray and the film plane.

The brightness is proportional to the cosine of the angle between the film plane normal and the ray direction, taken to some power given by the falloff value. So a falloff of 0 effectively disables it as the brightness will always be 1. Higher falloff values make the brightness drop off sharper and faster.



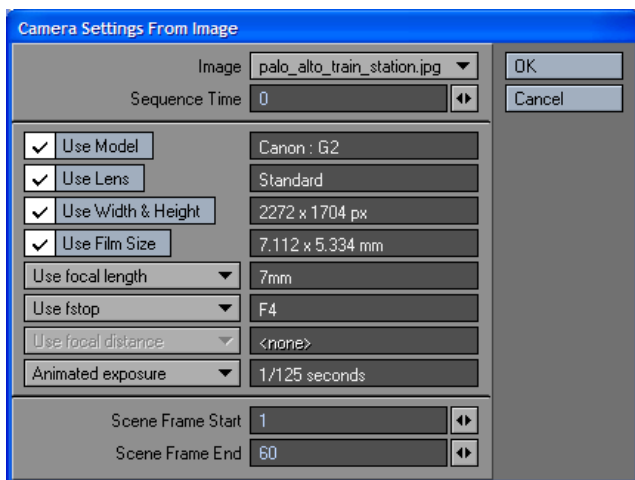
## Load Image

The Real Lens camera can take its settings from the EXIF tags in an image or image sequence.

The model selection dropdown in the Real Lens camera panel has an "(from image)" option. Selecting this opens a panel in which an image can be selected, and the settings to use from it picked.

There is also support for getting settings from image sequences. Image sequences should first be set up in the image editor. You can set the time in the sequence from which to get the settings.

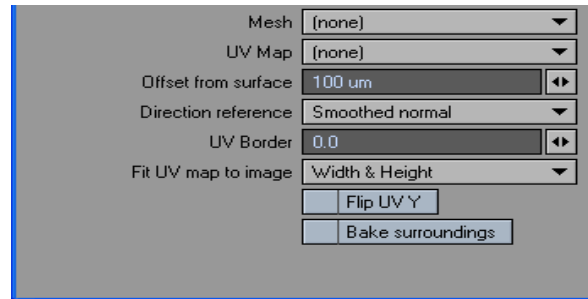
Some settings can be animated, which creates an envelope. Keyframes are set for all frames between the Scene Frame Start and End frame numbers (inclusive). If an image sequence is used, for each frame it will use the settings from the image matching the frame according to the sequence's settings.



### Limitations and cautions:

- \* If "Use Model" is greyed out, even though a make and model is shown, it's because the Real Lens camera doesn't recognise it.
- \* There is no EXIF tag that says what lens is used. LightWave does put the lens info in the MakerNote for a rendered image, and it will use that if available. Otherwise it will default to Standard.
- \* Focal distance is only rarely stored in the appropriate EXIF tag. Some of the most popular digital camera makers keep the focal distance in a proprietary MakerNote (presumably to get you to use their crappy photo software), if at all. Again, LightWave rendered images do store the focal distance correctly (if rendered with DoF).
- \* The focal length is the actual focal length, not converted to a 35mm film equivalent.
- \* Exposure is mapped to blur length using the current scene FPS setting, and causes photorealistic motion blur to be turned on.

## Surface Baking Camera



This camera considers each pixel as a UV coordinate, computing the position on a mesh polygon matching that UV coordinate, and shooting a ray at that position from close range.

Unlike the Surface Baker shader it is multi-threaded, with the rendering done by the raytracing renderer, it works with ScreamerNet, and works with VIPER. Furthermore, as it produces a normal render, all the render buffers are available for saving.

The disadvantage is that only one mesh and UV map can be rendered at a time. However, you can set-up multiple cameras, each with its own mesh and map set, and render an animation that is setup to switch to a different camera each frame.

- Mesh: This is the object that you want to bake.
- UV Map: This is the map on that object that you want to use as the baker map.
- Offset from Surface: This is the distance from the mesh surface the ray should be shot from. Too large a value may mean that the ray encounters an intervening surface instead of the intended surface. Too small a value increases the possibility of floating point errors causing the ray to slip through the mesh surface instead of hitting it, or the ray direction to become erratic.
- Direction Reference: Ray directions can be set to be equal to the polygon normal, the smoothed polygon normal, or in the direction towards the origin of the mesh. The ray will still hit the same spot on the mesh. The difference is in the direction in which the ray approaches that point.
- UV Border: This is the 'overdraw' of the UV polygon boundaries. Using this setting, you can remove the visible seams in your UV projection.
- Fit UV map to image: This control just determines how to fit the map to the rendered image size. Fit both width and height, or just width or height.
- Flip UV Y: this inverts the map by flipping it around the "Y" axis.
- Bake Surroundings: This allows you to flip the rays around, thereby baking the surroundings instead of the mesh. Use for baking a reflection map or environment map.



## Shiftcamera

The Shiftcamera is a tilt/shift type of camera capable of 2-point perspective views, popular for exterior renderings of building designs. It works the same as the Perspective camera, but removes the vertical perspective from the view. If part of the OpenGL preview is cut off, change the size of the grid size with Fixed Near Clip turned off.



NOTE: The camera must be kept horizontal, do not bank the camera, as unexpected results may occur in a render.

## Camera Settings

### Resolution

If you have **Use Global** checked, the Camera will use the resolution setting from the Render Globals Panel

The **Resolution Preset** drop down menu will present you with a series of pre-defined resolutions to choose from for your render. It will automatically set the **Width**, **Height** and **Pixel Aspect** ratio fields.



NOTE: You can add your own presets to this list, but it requires you to delve into the LW9.cfg file. If this thought doesn't scare you, then we'll proceed.

If you look in your LW9.cfg file you see that near the top there are several lines that look like this:

```
ResolutionPreset 1920 1080 1 0 0 1920 1080 HDTV (1920 x 1080)
```

The first two numbers are the size of the frame, the next one is the pixel aspect ratio and then the next four are the default Limited Region frame. The last bit of text is the title of the preset that will appear in the drop down menu.

Feel free to make your own lines. For instance here's one you may wish to add to get the ball rolling:

```
ResolutionPreset 2480 3508 1 0 0 2480 3508 A4 page (300dpi)
```

As you can tell from the title, this Resolution Preset gives you a full A4 page at 300dpi.

The width and height fields can be set to anything between 16 and 16,000 pixels. Be aware that larger resolutions can make serious demands on the memory of your machine.

The **Resolution Multiplier** gives you a much more consistent way of quickly checking a scene rather than changing the width and height fields when you want a small test render. It takes into account the scaling of things such as particle, line, and edge thickness, as well as the glow radius.

If you have selected a resolution preset and you alter the width or height fields, it will override any preset and the menu will then show the word Custom. If you have already set a resolution multiplier, it will then operate on the **Width** and **Height** settings you have chosen.

The resolution multiplier does not scale an image after it has been rendered. Therefore, it can be used in a situation where the boss



asks for an image about “two thirds as large again”. Deciding which resolution to use on a project is largely down to its intended use. An image for broadcast can almost always use the appropriate PAL or NTSC resolution presets. An image for print will always vary depending on the size of the final image, whereas film is usually one or two size settings. Both provide high-resolution images that can take a long time to render and a lot of memory. If, however, you are rendering an animation for display on a computer, you will often want to use a lower resolution for reasons such as the running speed of the final animation and its file size.

## Print Assistant

In the **Additional** menu in the **Utilities Tab**, you should find the **Print Assistant** plugin. This will enter width and height details for your render based on inch or Pica measurements at a specified dpi rate.



NOTE: You can also enter print sizes directly into the width and height fields using LightWave's ability to do maths in these fields. For instance, the seemingly complicated sum:

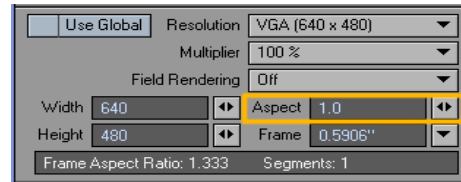
$$8.26 * 300$$

gives the width of an A4 page in inches at 300dpi. If you would rather work in metric, the width of an A4 page is 21 cm, so:

$$21 * 300 / 2.54$$

will give you roughly the same result (the /2.54 converts the sum into inches). The centimeter value is more precise since an A4 page's size is worked out based on metric rather than imperial measurements.

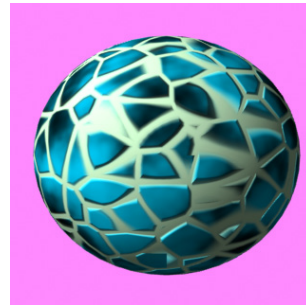
## Pixel Aspect Ratio



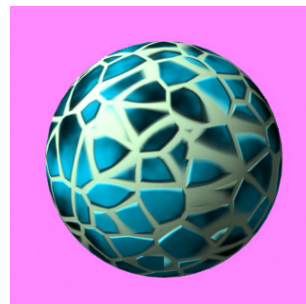
Once you've been using a computer for a while you forget that pixels actually come in different shapes. Ones for NTSC TV are tall and thin; ones for PAL TV tend to be a bit fatter, while ones for print are the same as those for computer screens – square as square can be.

The **Pixel Aspect Ratio** setting in LightWave is calculated by dividing the width of a pixel by its height. A pixel intended for print or a computer screen is square, as we said, so its aspect ratio is 1.0. Because NTSC pixels are taller than they are wide, the aspect ratio tends to be between 0.86 and 0.9. PAL ones, on the other hand, tend to vary between 1.01 and 1.06. Values for widescreen displays are considerably wider in both NTSC and PAL.

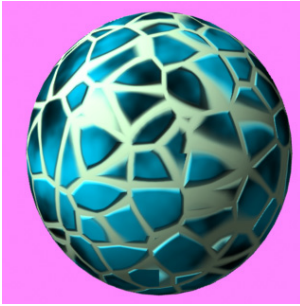
Why worry about the pixel aspect ratio? After all a pixel is a pixel, right? Well yes, but if you look at a perfectly round ball that has a radius of 50cm and you are using an NTSC resolution preset, the ball will look squashed on a computer monitor, whereas it will look perfectly round on your NTSC monitor. When selecting one of the resolution presets you will notice that the pixel aspect ratio changes along with the resolutions for width and height. As for things looking squashed or stretched on your computer monitor, I'm afraid it's either something you'll have to get used to, or you will need an output to a proper broadcast monitor to reassure yourself.



NTSC (0.9)



Computer monitor (1.0)



PAL (1.0667)

Same ball, different monitors.

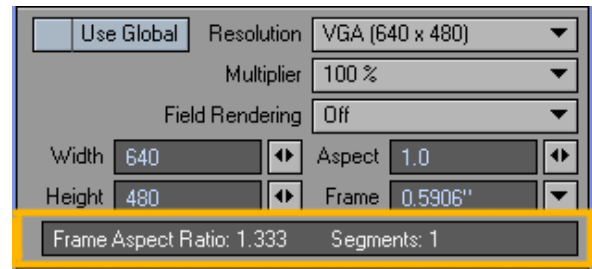
### Aperture Height

You can change the aperture height of your virtual camera in LightWave to match the optical characteristics of a real world camera, especially for film work. Changing this setting will only affect the Depth of Field effect and the lens focal length.



NOTE: Aperture Height is always listed in inches, even if you are using a metric unit system.

## Frame Aspect Ratio



Before we move on, don't confuse the pixel aspect ratio with the frame aspect ratio figure, often referred to simply as the aspect ratio. The way to work this out is to take the pixel width of a picture, divide it by the pixel height and multiply the result by the pixel aspect ratio. As an example, a standard VGA screen is 640 x 480. This equates to a frame aspect ratio of 1.333, which is the result of the following sum  $(640/480) \times 1$  and converting it to a ratio. You will often see this figure quoted on the back of DVD cases to indicate the width of the display compared to its height (which indicates how much of your TV screen will be covered by black bars).

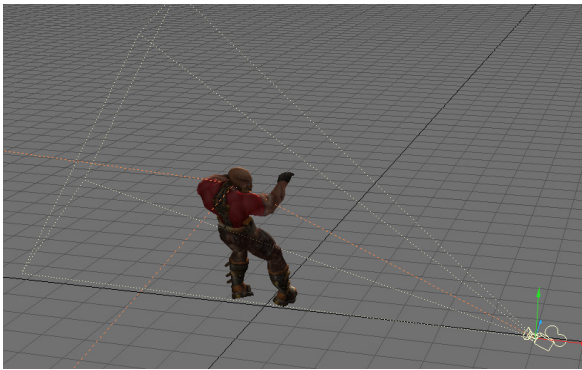
### Camera Settings in a Viewport

When you use the camera view (numeric keypad 6), areas of the viewport that are outside the render area will be colored with the overlay color chosen in **Display Options (D)**. You can have horizontal bars showing the exclusion or they can be vertical bars, depending on the frame aspect of the render you are making.

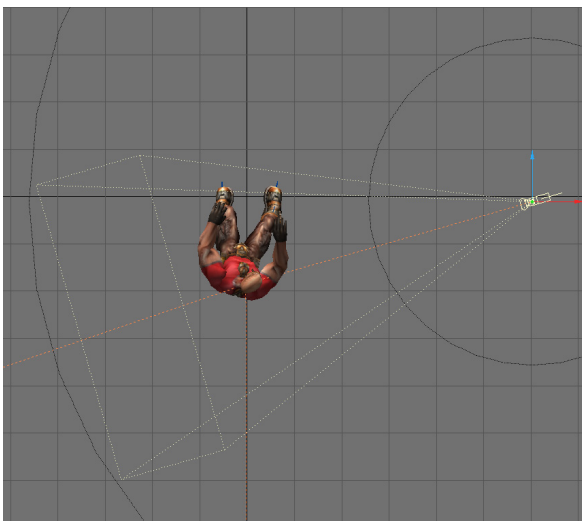


Furthermore you will be able to show safe areas for overscan and underscan displays by turning on the **Show Safe Areas** switch in the **Display Options (D)**. You will also have a grid displayed to allow you to better divide up the frame by selecting **Show Field Chart**.

If you are using a view other than the camera view, you will see the camera represented on-screen together with a pyramid showing its field of view. This pyramid is only shown when the camera is the selected item.



If you are using an orthogonal projection and you have **Show Fog Circles** option switched on in the **Display Options (D)**, you will be able to see the area around the camera affected by fog. If you press **Ctrl F6** and choose a standard fog type while in a top view mode, for example, you will be able to see circles around the camera indicating a minimum and maximum fog radius around the camera.



### Rendering a limited region

You can render a limited rectangular region of your scene if you only wish to test your work, rather than taking the time to render a full image. Simply hit the **L** key and you will see that a dotted line will surround the renderable area of the frame. You can change the size and shape of this area by clicking and dragging the left mouse button on one of the dotted lines that surround the area.



Above: What Layout looks like, Middle: Render Limited Region Borders, Right: Render Limited Region No Borders

There are two different types of limited region that you can use, either with or without borders and you cycle through these choices by repeatedly hitting the **L** key or by choosing the drop down menu in the **Render Globals** window. The difference between a limited region with a border and one without it is the fact that a limited region with a border puts your limited region on a black page the size of a full render, whereas a limited region without borders will just render the shape you desire as the full image. The frame aspect ratio in **Camera Properties** will remain at the aspect ratio for a full frame, but all other options, such as **antialiasing** and **Masks** still apply.

The limited region with border allows you to “patch” only a segment of a frame, rather than having to re-render the whole frame for re-compositing. This can be a major time-saver.

### Memory Considerations

Limited Region allocates only enough memory to render the horizontal limited region area. If you stitch parts of an image together, you can effectively render images that are much larger than those you could render one pass. This is especially useful for high-resolution print images or in low memory situations. However, note that some post-processing filters require a full-sized image. In such cases, you may be able to apply those filters to the “stitched” image in an additional step. The way to do this is to take your final rendered image and save it to disk. Then clear your scene – or better yet, quit and restart LightWave – and load this image into an empty scene. Make it the camera backdrop and add whichever post-process filter you wish to use, and then render again. Since you aren’t rendering all the objects, textures, Image Maps, etc., the memory requirements will be a lot lower.

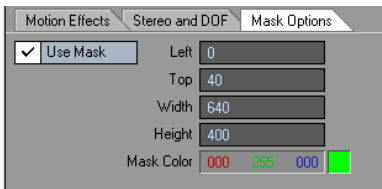




## Masking out a region

Using a mask is a little like rendering a limited region, but allows you to define a color for the area outside the region you define. Set the **Mask Options** in the **Camera Properties** window. Click on the **Use Mask** button to open up the settings for use. The figures you enter dictate the render area; everything outside it will be the color you choose. You can use this feature to get a letterbox-style effect.

Note: You must have Mask Options enabled in the Render Global Panel to use the Mask Options in the Camera Panel.



## Antialiasing

If aliasing (the stepping effect on diagonal lines seen when image resolution isn't high enough) is the crime, then anti-aliasing is the cure. Aliasing occurs because a pixel is rectangular in shape which means that there will always be a jagged effect with anything other than rectangular shapes. This effect can be reduced by working at high resolutions but cannot be removed. All image-processing programs, including LightWave, employ a device called anti-aliasing in order to reduce this stair-stepping effect. It compares the RGB values of two neighboring edge pixels and adds an average-valued pixel between them. This fools the eye into seeing a smooth line.

Found in the antialiasing menu in the Render Globals window and in the Camera Properties Panel, the PLD options offer new alternatives for rendering scenes. The original LightWave antialiasing features are now referred to as "Classic" in Layout. The Reconstruction Filters add even more options for your final image.



NOTE: As of LightWave v9.2, PLD and Classic Antialiasing are only available in the Classic Camera. A numeric anti-aliasing method is now available in the Advanced Cameras.

**Antialiasing** in the Advanced Cameras has been changed. You can now input any number of antialiasing levels or using the slider arrow with a left-click and drag.



Using any of the Advanced Cameras, **Antialiasing** will also occur in one pass for all levels, instead of one pass for each level. If you are using the Perspective Camera, for example, with an AA level of 9 will render in one pass, while using the Classic Camera with a PLD-9 level will render in 9 passes.

The advantage of a single pass in the Advanced Cameras is that the renderer does not have to reconstruct the geometry data for each AA level. Multi-pass rendering is required with the Classic Camera.

The **Reconstruction Filter** works the new camera modes as it has in previous versions of LightWave with one exception. If you set the antialiasing samples to 1 and set the **Reconstruction Filter** to any mode other than Classic, you will get the old PLD-1 antialiasing. This works best with the **Fixed** Sampling Pattern.

Major changes have been made to the core of the industry-leading LightWave renderer. In keeping with our tradition of innovation in this area, NewTek has introduced a new technology to allow significant improvements to Antialiasing called "PLD antialiasing" (Pixel Lattice Deformation.) The technology allows rendering to be undertaken with a much smaller number of passes, but without sacrificing quality.

The PLD antialiasing modes can be used either to render a scene using fewer passes, or to keep the number of passes constant and get a higher quality image.

In previous versions of LightWave, it was only possible to render the image in 1 (no antialiasing), 5, 9, 17 or 33 passes. This meant that



the jump from a preview image with no antialiasing to one with low antialiasing meant up to a 5x jump in rendering time. The PLD modes help address this not only by allowing much higher quality images with fewer passes, but also by opening up a much larger number of rendering pass options to users. This allows a user to find a much better trade-off between rendering time and image quality, thus significantly improving the production work-flow.

## Pixel Lattice Deformation (Classic Camera)

When an image render begins the antialiasing process, a determination must be made as to what type of samples will be taken and how long antialiasing sampling will occur. The new form of render pass for LightWave is the Pixel Lattice Deformation, or PLD.

PLD takes information about how the edges in the scene fall onto the lattice of output pixels. The pixel lattice is then deformed and realigned to best represent the underlying geometry. This generally helps with edges in images and textures. One limit to this method is that it cannot reconstruct details that have not been sampled at all, such as highly undersampled data. The advantage gained with PLD is that it gives you an image with some level of antialiasing without the need for extra samples being taken.

Compared to the "Classic" modes:

- Low and Enhanced Low: 5 passes
- Medium and Enhanced Medium: 9 Passes
- High and Enhanced High: 17 Passes
- Extreme and Enhanced Extreme: 33 Passes

The Classic modes have five levels of antialiasing from None to Extreme. These levels are then subdivided into two: normal and enhanced antialiasing. Enhanced will usually give a better effect for a given level of antialiasing because it uses almost twice as many samples for each pixel and filters them more intelligently, giving a better result at an additional cost in rendering time.



Left: No antialiasing, Right: PLD 1-Pass

Low or medium levels of anti-aliasing usually suffice for video resolution images, but if depth of field or motion blur is required in a render, the high level of anti-aliasing may be useful. The extreme level of anti-aliasing is useful for the highest level of fidelity in motion blur or depth of field.

Anti-aliasing can also be achieved by rendering at a higher resolution than needed with no anti-aliasing, and then shrinking the resulting image using another graphics package or LightWave itself. To use LightWave for this, simply load the rendered image into an empty scene and set the render resolution to less than the image's original resolution. You do not need to turn on anti-aliasing. Now render at the final desired resolution – something that will take place very quickly – and you will have your anti-aliased image.

## Advanced Camera Antialiasing

Using any of the Advanced Cameras, **Antialiasing** will also occur in one pass for all levels, instead of one pass for each level. If you are using the Perspective Camera, for example, with an AA level of 9 will render in one pass, while using the Classic Camera with a PLD-9 level will render in 9 passes.

The advantage of a single pass in the Advanced Cameras is that the renderer does not have to reconstruct the geometry data for each AA level. Multi-pass rendering is required with the Classic Camera.

The **Reconstruction Filter** works the new camera modes as it has in previous versions of LightWave with one exception. If you set the antialiasing samples to 1 and set the **Reconstruction Filter** to any mode other than Classic, you will get the old PLD-1 antialiasing. This works best with the **Fixed** Sampling Pattern.

## Sampling Pattern

There are two choices for how the antialiasing samples are arranged within a pixel. These are selected from the Sampling Pattern menu.

**Blue Noise** generates a semi-random sample pattern and is best used when you are using a large number of samples (16 or more) or **Adaptive Sampling** is enabled. The Fixed pattern generates samples on a fixed grid. This mode works best with fewer samples or when the image contains straight edges. There is a limit of 64 fixed samples. If you select **Fixed** sampling and more than 64 samples, the additional samples are generated with **Blue Noise**. Since Blue Noise is semi-random, it does not produce Moré patterns when there are thin parallel lines close to each other.

## Blue Noise

**Blue Noise** is a randomized sampling pattern. For a given pixel, **Blue Noise** will sample a semi-random point on the grid for a ray. The random point will be a minimal distance away from each other random sampling point. The sampling continues until the difference between two points are within a certain threshold or the maximum number of samples has been reached. A higher antialiasing setting will produce a finer grid for sampling, providing smoother antialiasing.

## Fixed

**Fixed** sampling takes ray information from the center of the grid. As more information is gathered from the grid, the aliasing produced is reduced.

## Classic

The classic sampling pattern uses the original LightWave sampling pattern, which is another type of fixed pattern.





## Reconstruction Filters

In addition to these significant changes to the raw antialiasing engine, the LightWave 3D renderer now includes a full implementation of reconstruction filters. These filters determine how the rendering data is combined into a final image at a sub-pixel level in order to produce a final image. Reconstruction filters have been a subject of research over many years and the names of these filters are often derived from the research that led to their development. The new LightWave filters include the most important “state-of-the-art” techniques and are outlined in the table below.

**Box Filtering:** This is the most common form of filtering, and the method that most traditional applications use to reconstruct an image from raw rendering samples. These samples are simply placed in a pixel “box” and averaged. Although this technique is fast, it can exhibit significant artifacts on motion and when there is fine image detail. From a signal processing point of view, this is a very poor technique for reconstructing an image from the raw data that comprises it.

This mode is very close to the traditional LightWave modes.

**Gaussian Filtering:** Gaussian filter based image construction takes the samples that compose the image and builds the final pixel data by weighting their contributions based on a Gaussian kernel of approximately one pixel in size. This technique typically performs quite well, although images tend to have a “soft” look. In practice this mode can be very valuable for video output where it can help hide some of the artifacts introduced by fielded or reduce bandwidth content.

**Mitchell Filtering:** Mitchell filtering is a technique that is now very popular and was suggested as an alternative to Lanczos filtering (see below.) It does not suffer many of the ringing artifacts of Lanczos filtering and generally is a very good starting point for most situations.

**Lanczos Filtering:** Lanczos filtering is arguably the technique that yields the closest to the perfect results for image reconstruction. This technique is based on a windowed sinc function (a sinc function being the optimal infinite image reconstruction filter.) Unfortunately in practice Lanczos image reconstruction tends to produce overly emphasized edges and “ringing” in high contrast areas of the image.

The filters listed above are all implemented in “Sharp”, “Normal” and “Smooth” modes that give users a huge variety of options to fine tune the look of their rendering to their particular needs and desired look.



NOTE: Reconstruction Filters can be used with Classic and PLD antialiasing and even work with no antialiasing option selected.



NOTE: Each of the Reconstruction Filters renders differently, so you might want test each of the different filters to see which one best fits your project.

All of LightWave’s “Classic” rendering modes and options are still available to ensure that scenes rendered with previous versions of the product can be rendered identically; or for use in rendering scenes where some of the advanced new options might not be applicable.

## Soft filter

In the **Render Globals** window under the antialiasing level drop down menu you will see an option for a soft filter. Selecting this will render your objects with a soft look, akin to a film image. It is not an appropriate substitute for antialiasing except in real emergencies. Note that it only affects objects, background images will not be affected.

## Depth Buffer Antialiasing

The **Depth Buffer Antialiasing** command (Effects command group) has been added to specify how the Z-buffer values from multiple antialiasing passes should be combined when creating a final depth buffer for output. Executing the command will display a small dialog. The **Minimum Value** setting (the standard behaviour) causes each pixel to store the closest of the depths found at that pixel in all passes. The **Average Value** setting causes each pixel to store the average of the pixel’s depths in all passes. The **Depth Buffer Antialiasing** setting is recorded for a scene in the Render Globals panel.

## Antialiasing Using Edge Detection

By default, LightWave uses **Adaptive Sampling** (edge detection) to determine which areas of an image should be antialiased. This focuses the antialiasing process primarily on the edges of objects. With **Adaptive Sampling** active, you can tell LightWave which areas to antialias by entering a sampling Threshold value. If the **Adaptive Sampling** option is inactive, LightWave will antialias the entire frame.

The adaptive sampling **Threshold** functions by comparing the brightness of two neighbouring pixels. A value of 0 will antialias everything in the scene, but values between .0314 and .1255 work well in most situations — the higher the level, the fewer edges are detected and, thus, the lower the rendering time.

A value of 1 is the maximum brightness difference in a 24-bit color space; however, internally LightWave can work with pixels brighter than RGB 255, 255, 255. Since you might want antialiasing only when nearby pixels differ by more than 1, the adaptive sampling Threshold can be set higher than 1. If you want to ensure that extra antialiasing is never performed, use a large **Threshold** value.

When **Adaptive Sampling** and the **Show Rendering in Progress (Render Globals Panel)** options are active, you can see the area where LightWave has detected edges highlighted in white on the rendering screen. By adjusting the sampling **Threshold**, you can increase or decrease the amount of white areas (and thus the antialiasing) to correspond to areas you know contain prominent edges. These white lines do not appear if you are using one of the enhanced antialiasing modes.

The **Adaptive Sampling in LightWave 9.2** has been changed considerably. When you select the Adaptive Sampling option, the image will be rendered in multiple passes. The first pass will render with the number of antialiasing samples you have selected. Each pass after that doubles the number of samples. For example, if the antialiasing is set to 2, the adaptive passes will render 2, 2, 4, 8, 16 and so on. Only the pixels that exceed adaptive **Threshold** will receive additional sampling during each pass. The **adaptive sampling** passes will stop when all of the pixels are within the adaptive **Threshold**. Because the number of samples increases each adaptive pass, later passes may take longer to render than earlier ones. For most images, it is best to set the antialiasing samples to 1 when adaptive sampling is enabled. An adaptive **Threshold** of 0.1 is good for draft quality work and 0.01 would produce production quality antialiasing.



Hint: When you have higher antialiasing that tends to get applied to an image, you don't need to use as much sampling in rendering options like radiosity, nodes with multiple sampling, or light quality. Use lower settings for these and let the higher number of antialiasing fill it in for you. Radiosity set to 2x6 and AA set to 2 will generate the same number of samples as radiosity set to 4x12 with AA set to 1. The render times will be about the same, but the image will be a bit cleaner with the higher antialiasing. If there's grain, the Adaptive Sampling passes will help clean that up pretty quick. The antialiasing can do that work for you, where it's needed most.

How the **Adaptive Sampling** mode works:

**Adaptive Sampling** works by rendering multiple passes and comparing each pixel to its neighbors and to the current pixel in the previous pass. The **Threshold** value is used to determine which pixels need additional samples. When a pixel is within the **Threshold** of its neighbors or the same pixel from the previous pass, it is considered to be done. Since the neighbor pixels can change on each pass, it is quite possible that a pixel that was once considered to be done may need additional sampling in later adaptive passes. Each **Adaptive Sampling** pass doubles the total number of samples. The **Adaptive Sampling** stops rendering passes when all the pixels are done or the maximum number of passes is reached.

The **Antialiasing** value works with the **Adaptive Sampling** as follows:

In the first pass of the **Adaptive Sampling**, the number of samples rendered is equal to the Antialiasing level (level 0 = 1 sample). The second adaptive pass will render this number of samples again. The third adaptive pass will double the total number of samples again and so on. Although raising the **Antialiasing** will cause the first adaptive pass to render slower, the pixels will converge (become done) faster so the total rendering time may not increase all that much.

In most scenes that use Adaptive Sampling, setting the **Antialiasing** level to 0 will produce the desired results. Its possible that the initial level of adaptive sampling could miss tiny bright features (highlights for example). This could create temporal aliasing (flashing dots) in your animation. To prevent this, you can increase the **Antialiasing** level (try 4 or 6) and still keep the benefits of **Adaptive Sampling**.

The number of adaptive passes you see is determined by two factors. First, the maximum number of passes is set with the adaptive tolerance. If the **Tolerance** is 0.1, for example, the maximum number of passes will be 5. Second, the adaptive passes can stop early if they detect that no further work needs to be done. That is all the pixels have converged to the desired quality.

**Oversampling** is similar to the Enhanced modes for the Classic camera, where you oversample the pixel. the value will go from 0 (no oversampling) to 1 (sample current pixel and surroundings).

## Noise Reduction and PLD-n Work Around

Some features, such as Noise Reduction, do not work the same in the new antialiasing as they did in earlier versions of LightWave. There is a simple work around for this problem. You can cause the

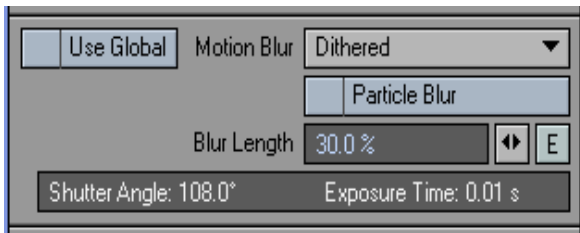
new antialiasing to work more like classic antialiasing by setting the Antialiasing to 1 and setting the desired antialiasing level in the Motion Blur Passes. You must also select Classic Motion Blur, set the Blur Length to 0 (assuming you don't want motion blur) and disable Adaptive Sampling. If you also set the Reconstruction Filter to Box (or any other mode) instead of Classic and select the Fixed Sampling Pattern, you will get the old PLD-n antialiasing.

There is currently no way to access the Enhanced antialiasing style with the new antialiasing. We are aware of this limitation and are considering different options for implementing it.

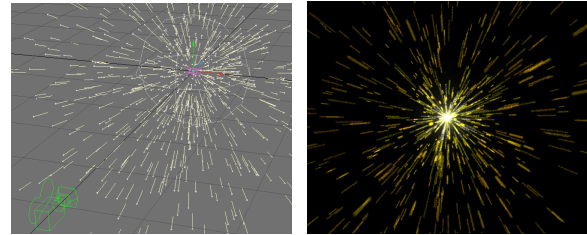


## Motion Blur Effects

When using a camera to film fast-moving objects, these objects are often blurry. This is because they continue moving while the shutter of the camera is open. It is this feature that LightWave aims to replicate with **Motion Blur**.



HINT: Using Soft Filter in combination with Dithered or Photoreal Motion Blur creates an even better effect.



Left: a Particle explosion, Right: Rendered with Particle Blur on

**Motion blur** becomes essential when animating, especially for use with live action. It prevents the crisp quality that normally pervades computer-generated animation and helps an animation appear more fluid.



Use a camera view and render a Motion Blur preview by hitting Shift F9.

LightWave's motion blur system takes everything that can change over time into account. From shadows, to surfaces, from light intensities to object or camera movement. It accounts for curved motion and does not blur in a linear fashion, but rather following the path that the motion is taking.

For motion blur to work, some level of antialiasing needs to be enabled. LightWave uses these antialiasing passes to generate the additional images used by motion blur. You will be able to see the process working if you are rendering in a render view. For each antialiasing pass, LightWave seems to move the objects a little and then composites them all together to get the motion blurred image. Because only five steps (a low level of antialiasing) can give a stepped effect, higher levels of antialiasing are recommended. There are three types of motion blur — normal, dithered and Photoreal. Dithered provides a better quality result with double the number of images to dither in between, and doesn't take as long as using the next level of antialiasing, but provides results just as good if not better. Introduced in LightWave v9.2, Photoreal provides the best quality blur. Instead of rendering a new pass for each motion blur sample, Photoreal Motion Blur allows for multiple samples within each render pass.



NOTE: Photoreal Blur will not work with the Classic Camera. You must use one of the Advanced Cameras, such as the Perspective Camera



## Photoreal settings

Motion Blur Passes are the number of render passes taken for motion blur.

Shutter Efficiency determines the amount of time, per frame, the shutter will be open. A value of 100% means the shutter is open 100% of the frame time, so the light exposure is equal for the entire frame. A value of 50% means the shutter does not open fully until midway of the frame, so the beginning and end of the light exposure are darker.

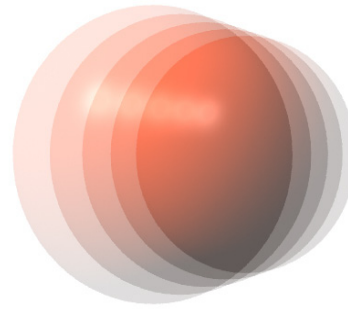
The Blur Length sets the time that the simulated camera shutter will be open during a frame. 100% would mean that the shutter is open for the entire frame. 50% would mean that the shutter is open for half the frame and is a more typical setting. Increasing this value increases the length of the motion blur streaks. It is possible to enter values larger than 1.0 if you wish to capture more than one frame worth of action.

The quality of the streaks is determined by the antialiasing settings. For best results, enable the Adaptive Sampling option and set your desired quality with the Adaptive Threshold value. Set the Antialiasing samples to a fairly low number. Antialiasing 1 will be fastest and should work for most scenes. If there are very long motion blur streaks in the image, you may need to increase the Antialiasing to 2 or 4 in order to get rid of all the gaps.

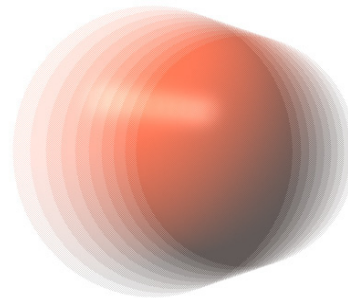
The Motion Blur Passes sets how many sub-frames will be rendered. Each sub frame will be anti-aliased with the settings described above. These sub-frames are blended together to produce the final image. This is the same way the classic motion blur has worked in earlier versions of LightWave with the addition that each sub-frame can now be anti-aliased.

Shutter Efficiency determines the amount of time, per frame, the shutter will be open. The Shutter Efficiency is limited to a range of 0.5 to 1.0. A value of 100% means the shutter is open 100% of the frame time, so the light exposure is equal for the entire frame, closing instantly at the beginning and end of each frame. A value of 50% means the shutter does not open fully until midway of the frame, so the beginning and end of the light exposure are darker. This produces a more realistic but shorter looking motion blur streak.

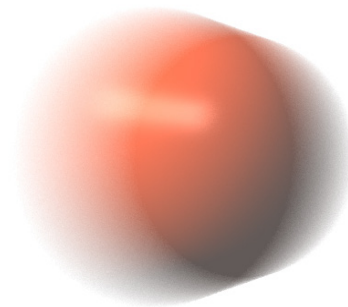
If your scene contains object deformations, you should use multiple Motion Blur Passes when using Photoreal Motion Blur. This will save you a lot of rendering time and produce more precise motion blur streaks.



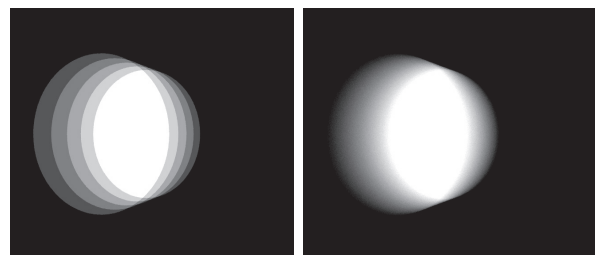
**Classic Motion Blur**



**Dithered Motion Blur**



**Photoreal Motion Blur**



**Alpha of Classic Blur**

**Alpha of Photoreal Blur**

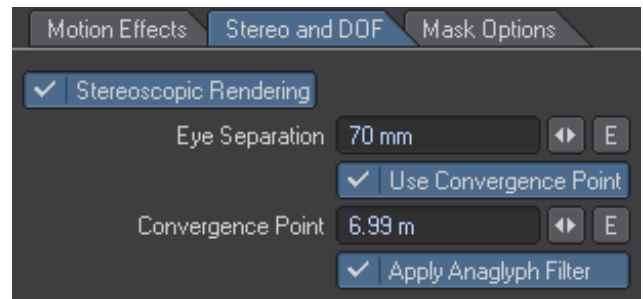


## Stereoscopic Rendering

Stereoscopic rendering gives you the familiar two-image set seen from slightly different perspectives that can be combined in a variety of ways to produce a 3D image. Further explanation of stereoscopy is outside the remit of this manual, but there are plenty of resources on the web for people interested in this field. When rendering using the **Stereoscopic Rendering** function you will get two images for each frame of your animation suffixed with either an L or R for Left and Right eye images respectively.

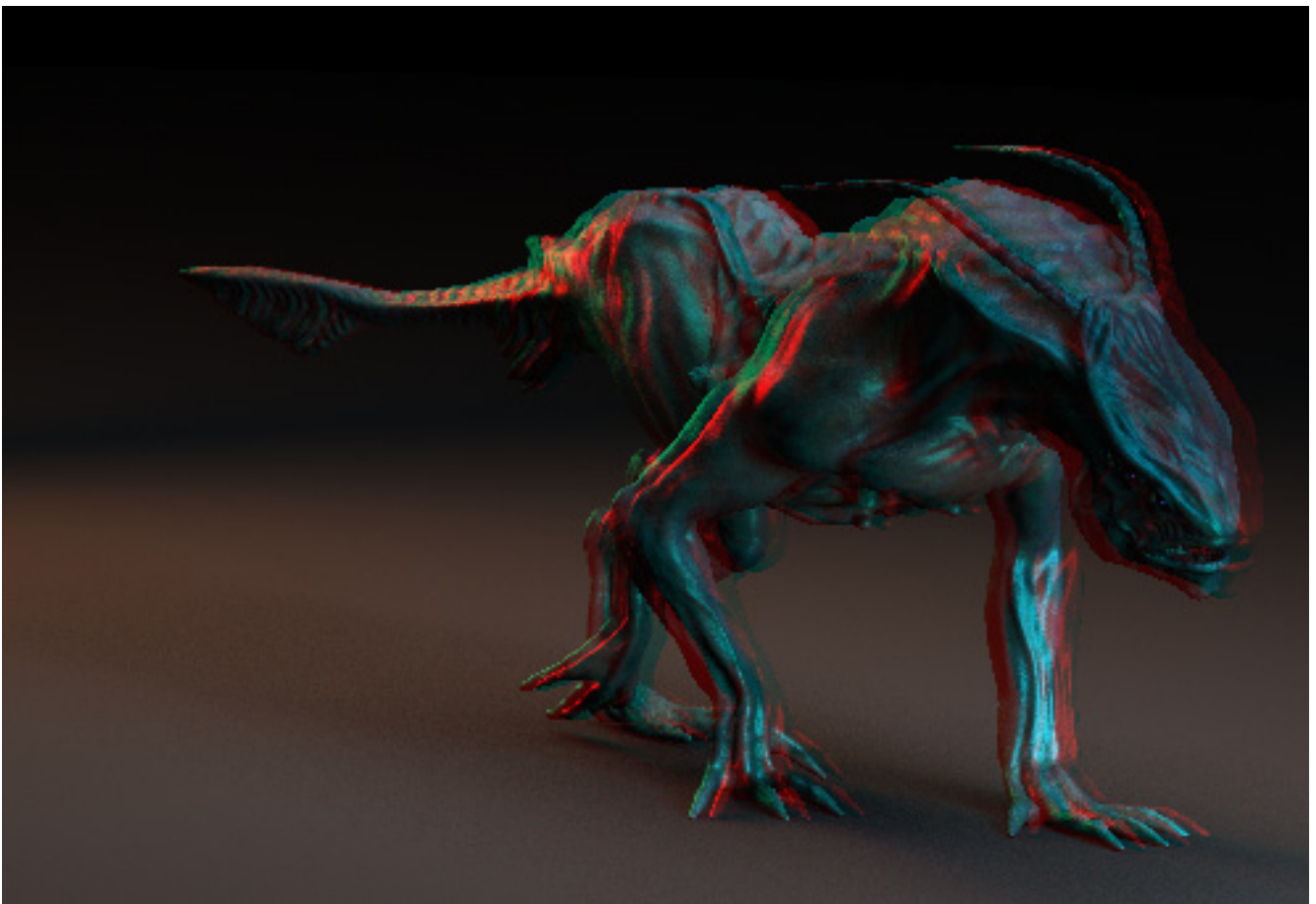
The stereo camera always renders the Left Camera first then the Right Camera. So when using F9 it will render the Left image then you must press continue and it will then render the right camera image. When you have a RGB file format and path chosen it will save the sequence of Right and Left images into that folder.

Also, when you check on **Apply Anaglyph Filter** it will render the Left Camera Image and then composite that with the Right Camera image to create an Anaglyph Red / Blue image saved as the Right Camera Image. If rendering with the Anaglyph filter for best results you should use an image format without an alpha because the resulting final Anaglyph image is composed of both right and left camera so the alpha would cause the image to not display properly. If you need an alpha then you should render just straight Right and Left stereoscopic images (uncheck the Apply Anaglyph Filter) so that two separate images (Right and Left) will be generated with their proper alpha channels for compositing.



The **Eye Separation** field is set by default to 70 mm — the average distance between the centers of the pupils in a human adult. You can change this value to exaggerate the 3D effect.

The **Convergence Point** is the point a specified distance away from the center of the camera that each eye focuses on. If it is checked off, the eyes will point directly forward.

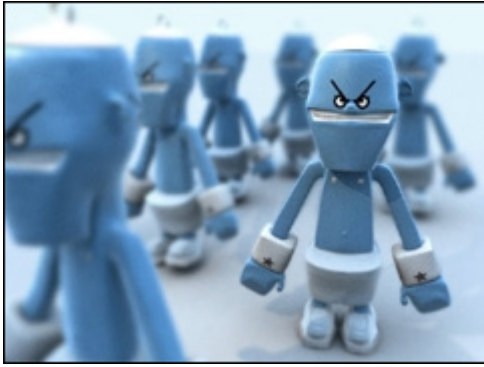


**NOTE:** Use the Anaglyph Stereo image filter if you wish to make a 3D image viewed through red/blue glasses.

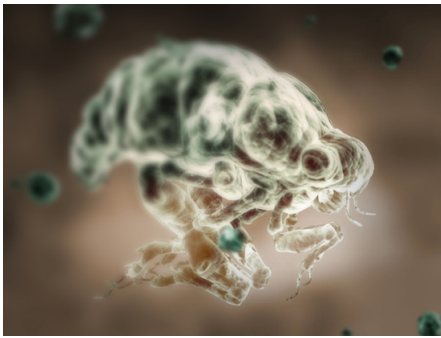




## Depth of Field



LightWave allows you to have a whole series of objects in a scene at differing distances from the camera and to render them all in perfect focus! The human visual system can't even do that, and that's the problem. For your images to look as realistic as possible, they need to use a feature of reality called **Depth of Field**. Fortunately, you can emulate this functionality inside LightWave. You need to render an image with antialiasing set to more than **Low** for the effect to be available, and then you can turn on **Depth of Field**. If this is going to give you unacceptably long rendering times, please also check out the image filter **Digital Confusion**.



**Depth of Field** refers to the area of acceptable sharpness in front of and behind the actual area in focus. When you enable **Depth of Field** you can change two settings. The first setting, **Focal Distance**, refers to the distance from the camera to the object that you would like to be in focus.

The second option is **Lens F-Stop**. Camera focus typically encompasses a range, from near to far, that we call "in focus". Objects nearer than this, or farther than this, appear out of focus. The **Lens F-Stop** value determines the range of focus around the **Focal Distance** (the near and far distances from the camera in which objects still appear in focus).

If you are familiar with real world cameras, you know that the f-stop sets the diameter of the lens aperture. An f-stop of  $f/4$  (which corresponds to a LightWave **Lens F-Stop** of 4) indicates an aperture diameter that is a quarter of the lens' focal length (LightWave's **Lens Focal Length**). Higher f-stop numbers refer to a smaller aperture, because the number is the denominator of a fraction. The aperture (f-stop) control on a real camera affects both the brightness and sharpness of an image. In the LightWave world, the **Lens F-Stop** works in the context of **Depth of Field** where it affects only sharpness.

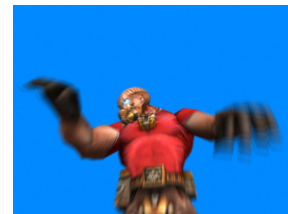
The larger the **Lens F-Stop** value, the larger the depth of field, that is, the greater the distance between the near and far distances where objects appear in focus. Conversely, the smaller the **Lens F-Stop**, the smaller the range of focused area.

In general, remember that the **Depth of Field** becomes progressively greater as the **Lens F-Stop** setting increases, the **Focal Distance** value increases and/or the **Zoom Factor** (and therefore the **Lens Focal Length**) setting becomes smaller.

When starting to experiment with **Depth of Field**, try using a short **Focal Distance** setting (shorter than the distance to the object you wish to be in focus) so that the depth of field is more pronounced. You can also look at the scene in an orthogonal viewport and be able to see a ring around the camera indicating the focal distance of your lens — the distance at which things are in perfect focus. Don't worry that the display will get confused if you are also showing fog circles in the viewport; the lines indicating the fog are visibly different to that of the focal distance.



**HINT:** Rather than waiting for realistic renders when setting up your depth of field, use the Quickshade rendering method in the Rendering Options window to get a quick preview of how blurry or sharp the objects are in your scene.



Left: Quickshade render, Right: Full render



**HINT:** You can get an accurate measure of the distance between the camera and your selected object using the Ruler or Range Finder custom objects.

You can also set a null to be parented to the camera and linked to the focal distance envelope of the camera so that you can control the focal distance by moving the null. This way you can get an interactive distance tool to control the focal range. Here's how you do it:

Add a null to your scene and parent it to your camera in the **Motion Options** window (**M**). In Layout, turn off the X and Y axes so that the null can only be moved along its Z-axis. Name this null "CameraFD";

Go to the **Object Properties** window (**P**) and assign the **Range Finder** custom object to the null.

You can turn on the link to camera if you wish;

Go to the **Camera Properties** window. Turn on at least medium antialiasing to enable the depth of field function and make sure that the **Focal Distance** is set to 0 m;

Hit the **Envelope** button next to this field so that you can link the distance between the null and the camera to the focal distance used.

Click on the **Modifiers Tab** in this window and use the **Channel Follower Modifier**. Double click on this in the list under the **Add Modifier** drop down menu to set up the **Channel Follower Modifier**;

From the list of channels, choose the CameraFD.Position.Z channel, make sure that the time lag is set to 0, scale to 100% and start frame and end frame to 0 and -1 respectively.

You can now key the position of the null and the camera's focal range will be set to the distance between the null and the camera meaning that you can pinpoint a single object in a busy scene to be focused



upon. The smaller the value for the **Lens F-Stop**, the smaller the range of focus.

## Depth of Field/Motion Blur Viewport Mode

---

A new feature has been added to viewport menus to preview Motion Blur. It functions like the Shift F9 option, but is always on when activated. The number of preview passes is controlled in the Preferences Panel. You must use the Camera View to use the Motion Blur preview.







---

## Chapter 15: Light Properties

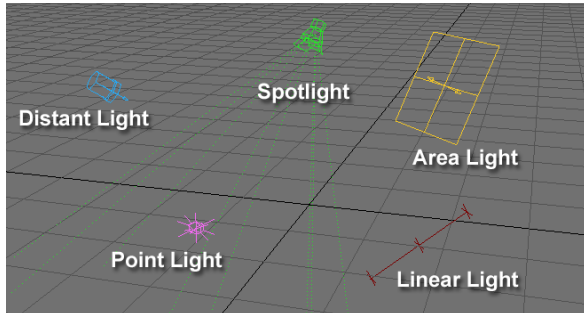
---



## Light Properties

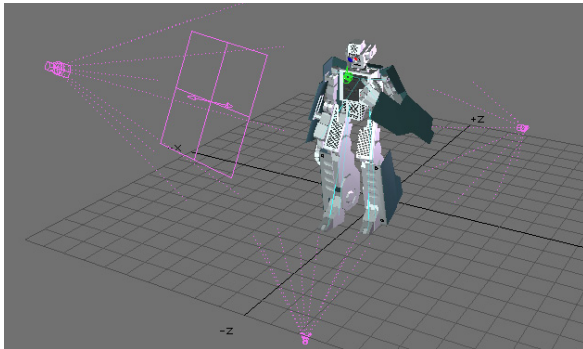
### Lighting Introduction

Along with objects and the camera, lights are a basic element in any LightWave Scene. In Layout, a Scene must always have at least one light. In fact, LightWave will not allow you to remove the last light source. If you really don't need it, you can just turn its **Light Intensity** to zero or deactivate the light on the **Scene Editor**.



LightWave has several types of lights, and you can give them different names, colors, and attributes. Along with lens flares, volumetric and shadow casting options, you can control light with a high degree of precision. All of these features are found within the **Light Properties Panel**. Along with the several lights, LightWave also features radiosity and caustic light effects, which can add tremendous realism to your scenes.

Remember that LightWave lights do not act exactly like lights in the real world. For instance, you cannot see a LightWave light source, only its illuminating effect in the scene, unless you add a lens flare to the light source. This is actually a handy feature because unlike a movie set, you can place lights anywhere, including in front of the camera!



You can also place lights *inside* objects. For example, you may have a solid ball inside a box and you may want the inside walls of the box to be lit. Placing a point light inside the ball will shine light *through* the ball onto the inside of the box. Likewise, you could light the faces of a string of dominoes by placing one light aimed at the face of the first domino in line. Of course, you may not want the light to go past the first domino or through the ball and so there are options that allow for this realistic behaviour as well.

LightWave lights also differ from the real world because you can have *negative* lights that take away color, as well as diffuse and specular shading.

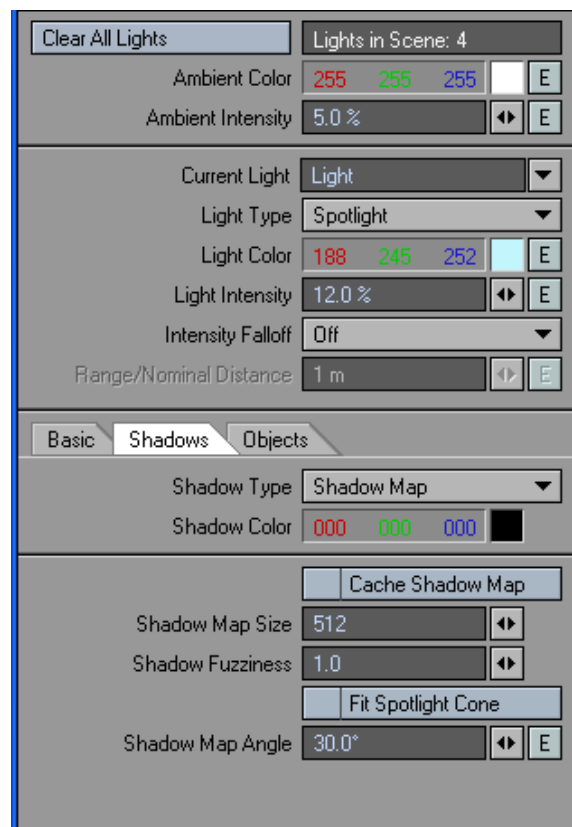
## Basic Light Attributes

Deactivate **Affect Diffuse** to prevent a light from affecting the general color or brightness of a scene. This is particularly useful when you add lights to a scene for creating specular highlights. Often you want a light that creates a nice specular highlight on a surface, but you don't want it to affect the rest of the lighting.

### Ambient Light

Ambient light does not come from any light source, and it produces no shadows. It is a general, ever-present light source that gets into every nook and cranny; it provides a type of *fill light* for objects and shadows. The **Ambient Light** settings are on the Render Globals Panel > **Global Illumination Tab**. Selecting **Ambient Color** lets you choose a color for the ambient light. Changing the **Ambient Intensity** value lets you determine the amount of ambient light in a scene.

LightWave's ambient light does not exist in the real world because it does not come from any specific source or direction. Many animators use lower intensity lights aimed in the opposite direction of the main light source to simulate reflected bounce light as opposed to using LightWave's ambient light feature. Basically, these are ways to *fake* radiosity — the bouncing of light off surfaces. Although LightWave has a radiosity option (discussed starting on the next page), these *faking* techniques offer greater control and faster rendering times.



Deactivate **Affect Specular** to prevent the light from creating specular highlights; the light still affects the color and brightness of the scene, however. This is very nice for adding lights in a scene to approximate the look of radiosity. A scene often needs a very high number of lights to give it the appropriate realism and warmth. However, adding too many lights can cause objects with a high **Specularity** setting to reveal the light's presence with multiple hot spots. By disabling specular for these lights you can overcome the problem. For similar reasons, there is the **Affect Caustics** option. **Affect Caustics** can also be turned off to limit caustics calculation to only desired lights, drastically speeding up calculation time.



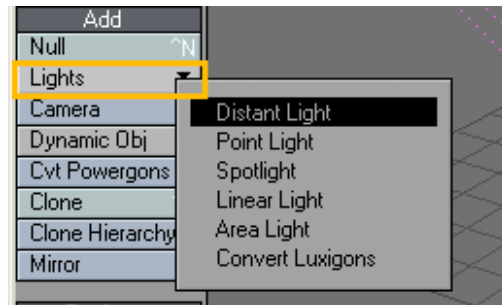
The **Affect OpenGL** option is for display purposes only. It will not change the effect a light has on a rendered image. With this option you can prevent the light from affecting Layout's display, which uses OpenGL. This is important since you can only use up to eight lights to affect your OpenGL display (see the **Display Options Tab** of the **Preferences Panel**).

## Lights display size

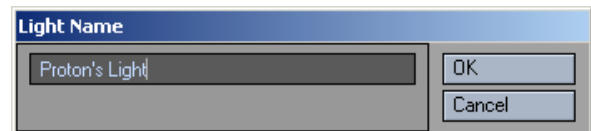
Lights are displayed on the Layout screen in a size relative to the Grid. If you increase or decrease the **Grid Square Size** on the **Display Options Tab** of the **Preferences Panel (Edit > Display Options)**, all lights change size to match the grid. This does not affect the light source's coverage or intensity at all, only its visual representation. Also, the true light source is actually located at a center point within the visual representation of the light.

### To add a light:

Choose **Items > Lights** and select the desired light type from the submenu.



NOTE: A dialog prompts you for a name when you add lights.  
Click OK to accept the default.



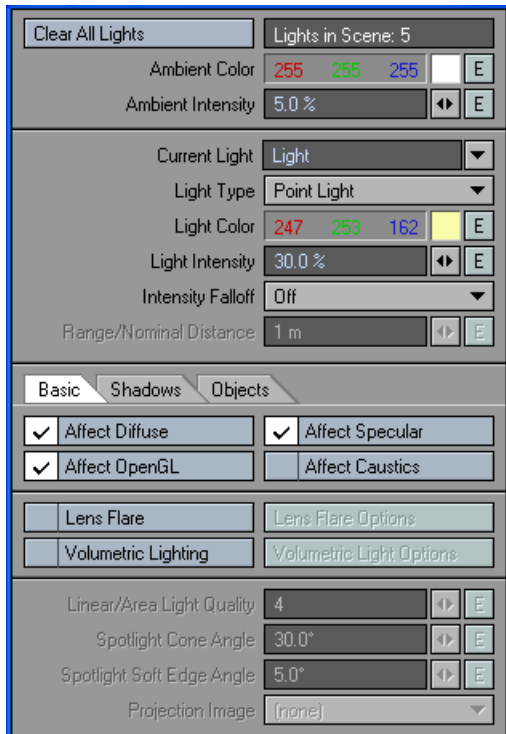
### To remove a light:

Select the light and choose **Items > Clear Selected Items** or **Clear All Lights**.



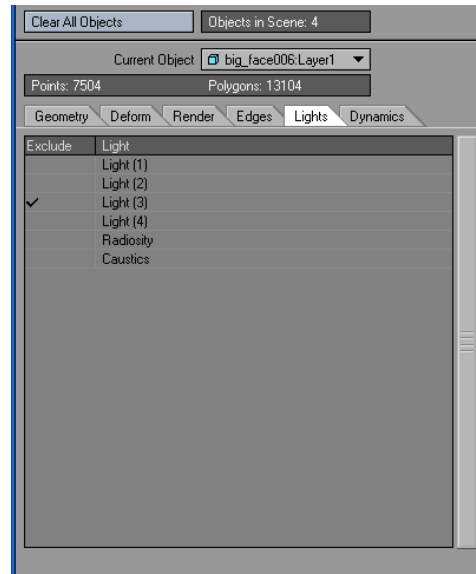
### To adjust a light's properties:

Select the light and open its **Properties Panel (P)**. If the panel is open, you can choose different lights to adjust using the **Current Light** pop-up menu.

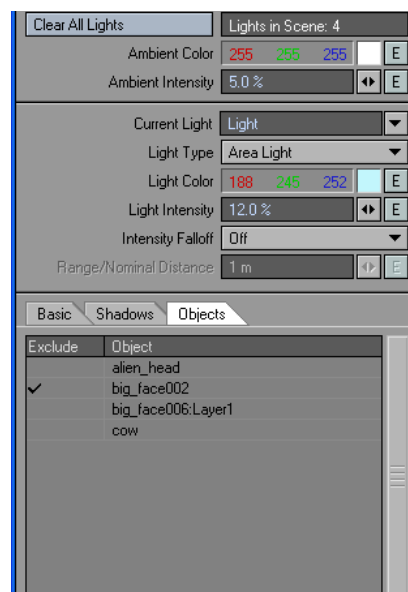


## Light Exclusion

You can exclude any set of lights, as well as radiosity and caustics, from the shading of any object. Simply click in the **Exclude** column to activate the exclusion.



You can also exclude or include all lights in the objects properties panel or all objects in the lights properties panel or invert your selection by right-clicking on the Lights heading



If for some reason you need the excluded lights to continue to cast shadows, deactivate the **Shadow Exclusion** option. When this *global* setting is not activated, all lights will cast shadows even if they are excluded. You will need to add the **Shadow Exclusion** command (Lights command group) to a keyboard shortcut or menu to access it. Note that adding to a menu may be preferable so that you can see its current state.

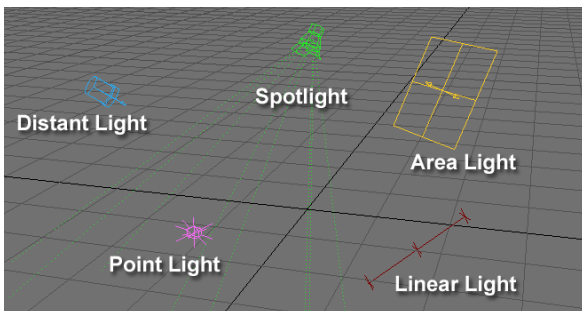


## Saving Lights

Choose **File > Save > Save Current Light** to save the current light to a file. You save normal **Light** settings as well as **Lens Flare** and **Volumetric Light** settings. You can add the light to a scene by choosing **File > Load > Load Items from Scene** and selecting the previously saved light file – which is really just a scene file with only light information.

### Light Types

LightWave has several types of lights, each with their own characteristics, features, and abilities. Once you add a light, you can change its type by changing the **Light Type** on the **Light Properties Panel**.



### The Distant Light

A **Distant light** is somewhat like the light from the sun. Surfaces receive an infinite amount of parallel light rays traveling in the direction that the Distant light points to. Distant lights are handy when you want equal illumination on objects in a scene.



The location of a Distant light doesn't matter — only its rotation is relevant. Because of this, you may place one Distant light in a scene pointing straight down and all objects in your scene are lit as if from above — the light can be a million meters or one meter above the object. You will see the exact same results on the object. In most cases, you will need only one Distant light in a scene, although you can add more. Generally, you will get much more realistic results using the other types of lights, if you need more.



NOTE: Because LightWave, by default, places one Distant light aimed from the upper left to the lower right in a scene, you can simply load any object and immediately hit the Render (**F9**) button (without setting any key frames) to see what the object looks like.

### The Point Light

A **Point light** sends light out from a central location equally in all directions. Light bulbs, camp fires, and fireflies are good examples for Point lights. In a way, a Point light is the opposite of a Distant light. It doesn't matter how you rotate a Point light since it casts light in all directions, but it does matter where it is located.



### Intensity Falloff

Point lights, as well as Spotlights, can be set to fall off over a specified distance. You activate the option on the **Intensity Falloff** pop-up menu. The falloff can be **Linear** or non-linear. The non-linear option **Inverse Distance** reduces intensity as the light moves farther from its source. The **Inverse Distance ^2** uses a higher, more natural level of reduction.

When using Linear falloff, the **Range/Nominal Distance** values set the distance from the light where the light's intensity is zero. When using Inverse or Inverse Distance ^2 falloff, the **Range/Nominal Distance** value determines the point where the Light Intensity reaches the Light Intensity setting, so the closer to the light source the brighter it gets..

**In orthogonal views, the falloff area is visible as a circle around the light. The light will fall off to zero exactly at the edge of the circle if using linear falloff. With inverse falloff, the circle shows exactly where the falloff begins.**



NOTE: If you do not use Intensity Falloff, the light will travel forever, unless a shadow option is active.





## The Spotlight

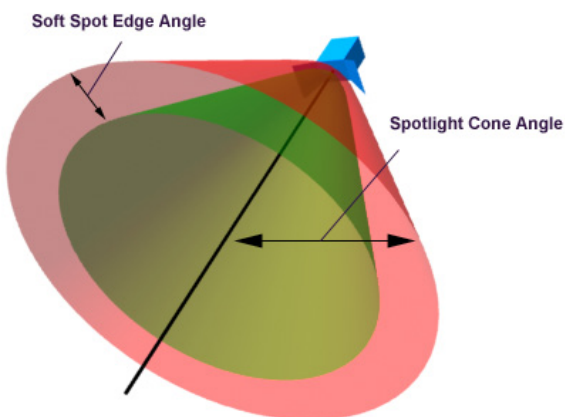
**Spotlights** behave somewhat like their real world counterparts, sending light out in a specified direction and size. Spotlights are probably the most useful type of light available to the LightWave animator. Flashlights, car headlights, and searchlights are all good examples of Spotlights.



As you might have guessed, a Spotlight can project light onto objects using a *cone of light*. The size of the cone is determined by the **Spotlight Cone Angle** and **Spot Soft Edge Angle** values.

The **Spotlight Cone Angle** determines the width of the cone of light. It is equal to the angle from the edge of the Spotlight to an imaginary line projecting straight out from the middle of the light source. So a 30-degree **Spotlight Cone Angle** actually defines a 60-degree arc of light.

The **Spot Soft Edge Angle** determines the width of the *falloff zone* from the illuminated cone to the Spotlight edge. It is equal to the angle from the Spotlight edge to the line projecting straight out from the spotlight. Within this area, the spotlight slowly fades away to no light, thereby creating a soft edge.



A **Spot Soft Edge Angle** of 0 degrees creates a Spotlight with a hard edge, while a setting less than or equal to the **Spotlight Cone Angle** creates a soft-edged light.



NOTE: Spotlights also have the same Intensity Falloff options as Point lights. See the preceding discussion.

## Virtual Projector

Another cool feature of Spotlights lets you use the light to project an image onto an object, much like a movie projector. You select the image to project from the **Projection Image** pop-up menu.



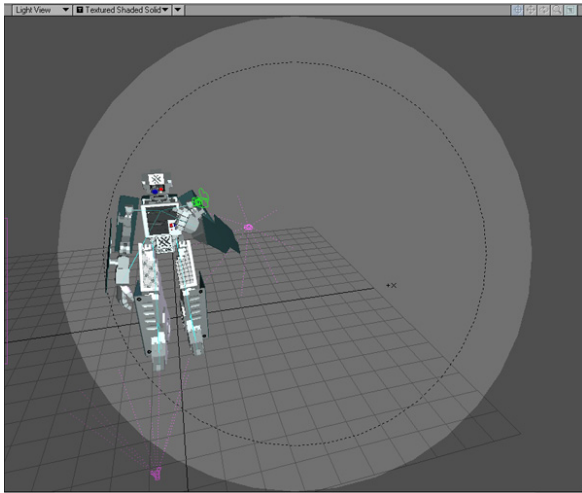
NOTE: The projected image will not conform to the circular shape of the cone angle. To do this, you could place a cookie-cutter object (e.g., a flat box with a round hole) in front of the spotlight or use a paint package to edit the image and fill the unwanted area with black. You may also want to use the Blur Filter to soften the edges.



## Spotlight Viewport Display

You can manipulate how Spotlights display in Layout. First, when you selected the **Spotlight Cone Angle**, it is represented by lines emanating from the light. If you interactively adjust the cone angle the shape of the lines will expand and contract.

If you use the **Light View** mode (on a viewport's titlebar), you see a circle that represents the **Spot Cone Angle**. If you adjust the **Spot Cone Angle** in this view, the circle remains the same size. As such, it looks like you are zooming in/out.



If **Fit Cone** is not active for the spotlight and the **Map Angle** is equal to or smaller than the **Spot Cone Angle**, you will see a square outline. This represents the **Map Angle**.

## Linear and Area Lights

If you could turn a two-point polygon into a light, you'd have something similar to a **Linear** light. Light is sent out equally in all directions, except at the ends. This type of light is great to use in objects like fluorescent strip lighting.



Now, if you could turn a four-point double-sided polygon into a light, you'd have an **Area** light. Light is sent out equally in all directions, except along the edges. This type of light might be used for flat light panels. Area lights behave, by far, the most like real world lights, but come at a cost of increased render times.



Unlike the Distant, Point, and Spotlight lists, you can size these lights, just as you would an object. Another distinction is that ray-traced shadows will have physically accurate edges that are sharpest near the object that is casting them and get softer further away. A smaller area light will result in sharper shadows than a large area light.

The range of the **Linear/Area Light Quality** setting (for linear and area lights) is adjustable from 1 to 5 (4 is default), which corresponds to 1, 4, 9, 16, and 25 samples per area light. Lower settings reduce render times, although soft shadow edges may appear grainier. A setting of 1 is fast, but noisy; you will need to use high anti-alias levels.



**NOTE:** When you use these types of lights, you probably need to use a Medium or High Anti-alias setting (Camera Properties) to smooth out the shadow. Keeping the light size as small as possible will also help.

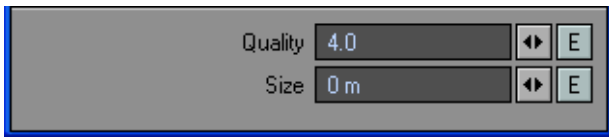
Area Lights now have a toggle in the Global Renders Panel Render tab to become single-sided. The icon for Area Lights will update and indicate if the area light is single or double-sided. The switch toggles all area lights in the scene.



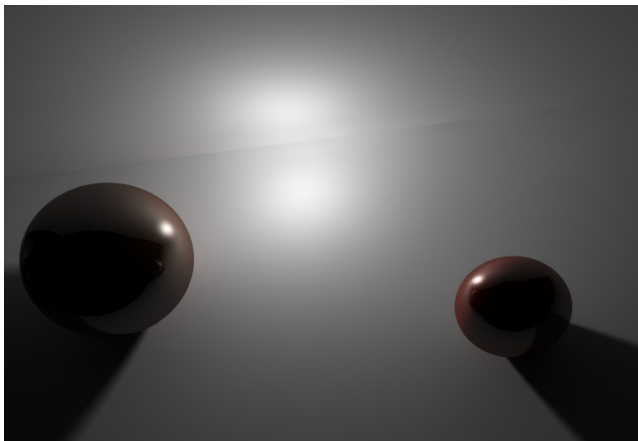
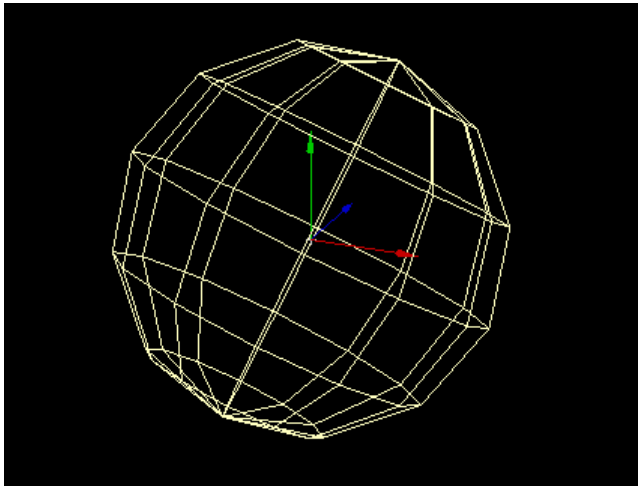


## Spherical Light

The spherical light emits light rays from the edges of a sphere. The default size is 0, which essentially makes it a point light at that size.



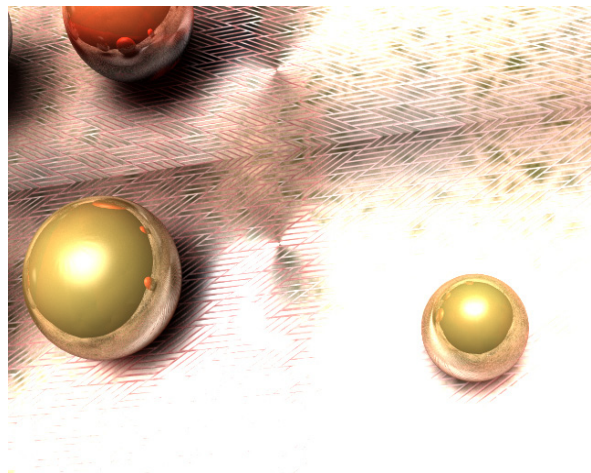
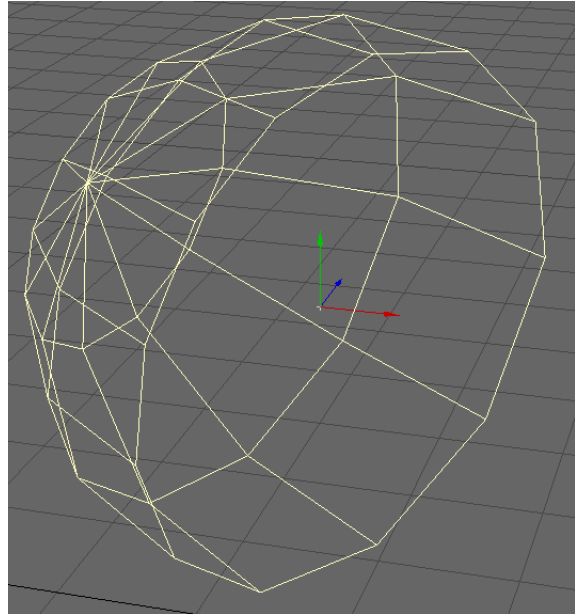
The position and size of the spherical light is important. For example, if you have a spherical light as the only light in a scene, and the light intersects geometry, the section of geometry "inside" the sphere will not be lit.



Spherical light

## Dome Light

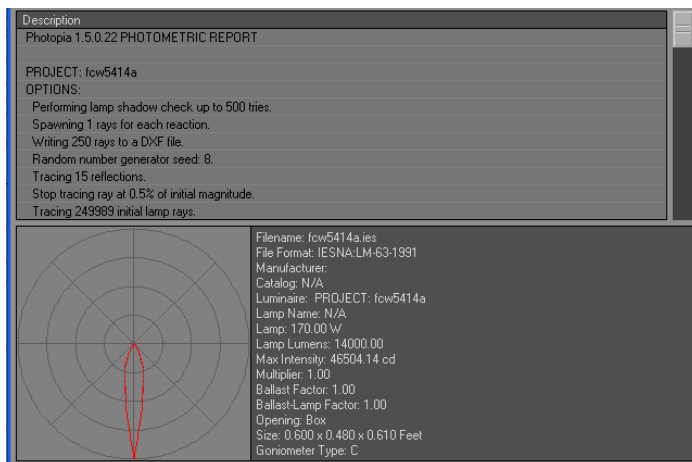
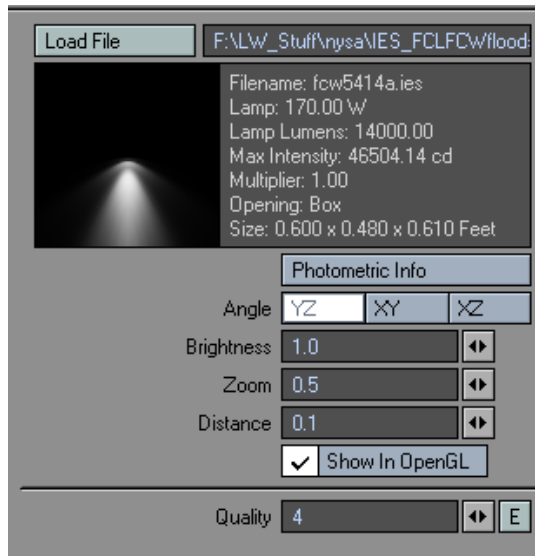
A dome light lights the scene from all sides within the dome. It is exclusively the rotation of the light that determines the light's effect in the lighting solutions. Position and scale do not contribute any effect to the lighting solution. Set the Z-axis upwards to light a scene from the top.





## Photometric Lights

Photometric lights are custom lights used to replicate real lights. The bottom section of the Light Properties panel allows you to load Photometric lights and control different aspects of the lights..



Photometric Info provides information about the loaded light.

Angle, Brightness, Zoom and Distance affect only the Thumbnail Preview window and do not change the display in Open GL or in a render.

Show in OpenGL will turn on/off the Photometric light for OpenGL display in the viewport.



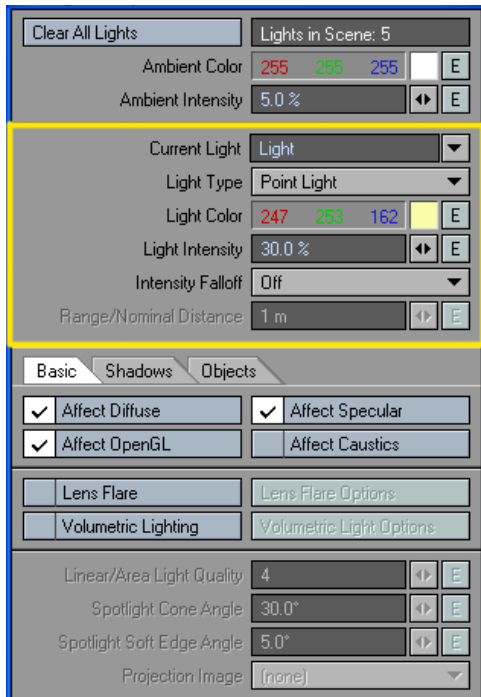
IES lights.

NOTE: A wide variety of IES lights, are available online. A search for "IES Lights" in a search engine should bring up a list of different web sites providing



## Light Color and Intensity

The **Light Color** setting on the **Light Properties Panel** modifies the color for the current light.



**Light Intensity** lets you set the brightness for the current light. You can drag the mini-slider to set a value between 0% and 100%, or enter a numeric value manually. You can even enter values greater than 100%. This is sometimes necessary, particularly when using radiosity (Render **Globals Panel**).



NOTE: The default light is set at 100%; however, lights added subsequently have a 50% Light Intensity setting by default.

Light intensity is additive, so if multiple lights hit a surface, their intensities are added together. As such, too high a **Light Intensity** value tends to wash out a scene, particularly when you have multiple lights. Eventually, the rendered image becomes solid white when the values of light sources exceed a certain brightness.



NOTE: A surface hit with multiple lights which exceed a total of 100% is not necessarily bad, and usually occurs in most scenes. Elements like surface Diffuse values, shadows, light falloff, and so on, will all tend to diminish the initial light intensities.

### Light Intensity Tool

You can use the **Light Intensity Tool** to adjust the intensity of selected light(s) by dragging your mouse. This tool needs to be added to the menu config by using the Edit Menus window (**Alt F10**) and dragging the **Light Intensity Tool** from the Tools Group into your menus where you want it.

### Negative Lights

The **Light Intensity** can also be set to a negative value. This takes away Diffuse and Specular shading. Moreover, if you use a colored light, that is, something other than white, a negative light subtracts the light color from the surfaces it affects.

### The Envelope Please

Numeric light properties can use envelopes to control their values over time. A highlighted **E** button signifies that an envelope is in use. Clicking on the **E** button allows you to make changes to an envelope in the **Graph Editor**. **Shift + LMB** on a highlighted **E** button removes the envelope.

Point lights, as well as Spotlights, linear lights and area lights, can be set to fall off over a specified distance. You activate the option on the **Intensity Falloff** pop-up menu. The falloff can be **Linear** or non-linear. The non-linear option **Inverse Distance** reduces intensity as the light moves farther from its source. The **Inverse Distance <sup>^2</sup>** uses a higher, more natural, level of reduction.



NOTE: See remarks under **Intensity Falloff** above on page 41.

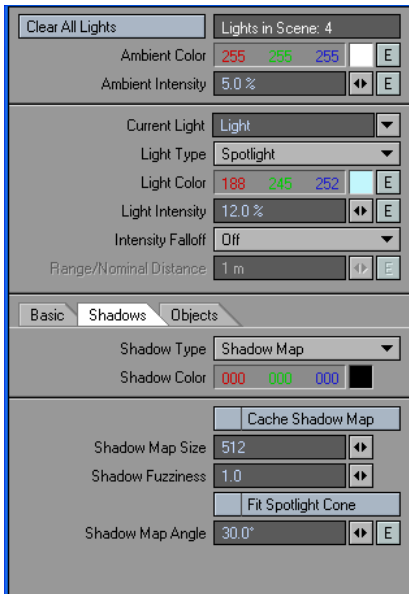


NOTE: If you do not use Intensity Falloff, the light will travel forever, unless a shadow option is active.

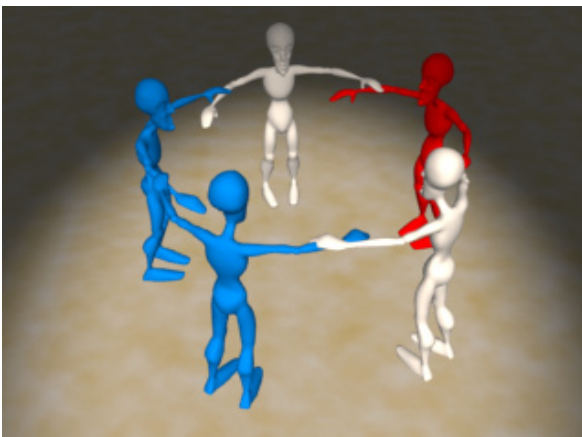


## Shadows

In LightWave, the *light* goes through objects unless the light is set to generate shadows. So unless you use a light with **Intensity Falloff**, the light will continue on forever.



When a light generates shadows, it respects objects that block the beam, which keeps the light from continuing on its merry way. A shadow is created when you have a lighted area next to an unlighted or partially lighted area.



*With Shadows Inactive*

Say you have a scene containing a house with a fully-detailed interior. Without shadows, if you have a light emulating the sun, its light always affects the interior of the house, even if you position the light itself outside of the house.

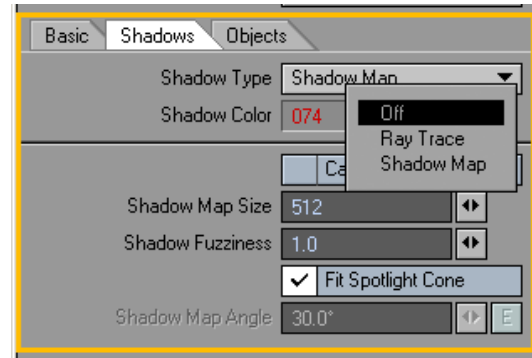
In order for an object to cast ray-traced shadows onto other objects, it must contain polygons that face the object receiving the shadow. In other words, the surface normal of a polygon must face the surface receiving the shadow. This is rarely a problem with solid objects, but if you cast shadows of flat planes, and do not see a shadow, make sure to use the **Double Sided** surfaces attribute or build the object with polygons facing both ways.



NOTE: Volumetric lights pass directly through objects when there are no shadows.

## Shadow Type

You initially set lights to generate shadows via the **Shadow Type** setting. By default, any light added to a scene has its **Shadow Type** set to **Ray Trace**. Ray tracing a light produces accurate shadows. Distant, Point, and Spotlights produce ray-traced shadows with hard edges. Linear and Area lights produce ray-traced shadows with physically accurate shadow edges that are hard near the object casting them and softer further away.



NOTE: When ray tracing shadows, Ray Trace Shadows option (Render Options Panel) must also be active.



NOTE: Single-point polygons (particles) and two-point (lines) polygons do not cast ray-traced shadows. They will, however, cast shadows with Shadow Maps.

## Shadow Mapping

An alternative to ray-traced shadows is the **Shadow Map** option, available only for Spotlights. These are not as accurate as their ray-traced cousins, but will render much more quickly and can have a soft edge. **Shadow Mapping** is a derivative of the same procedure used to calculate areas that are hidden from the camera view by objects. Instead of the camera's view, LightWave determines whether areas are hidden in the light's view. If so, these areas will be in shadow.



NOTE: Make sure that Enable Shadow Maps is active on the Global Illumination Panel (Lights > Global: Global Illum) or Shadow Maps will not appear.

## Transparent Objects

One of the biggest limitations of **Shadow Maps** is that they do not respect transparent surfaces or dissolved objects. The shadow appears as if the object is solid. For transparent shadows, you must use a ray-traced light source.

## Cache Shadow Map

**Cache Shadow Map** specifies that the **Shadow Map** for the light should be calculated only once during each render session, no matter how many motion blur passes or frames are rendered. This saves rendering time, but should be used only when the light and all objects that are illuminated by it are not moving.



Shadow Map



Ray Trace Shadows on semi-transparent aliens

## Shadow Map Size

The **Shadow Map Size** value determines the resolution of the **Shadow Map**. This number represents one side of a square view, therefore the default of 512 generates a **Shadow Map** that is 512 by 512 pixels. The higher the setting the finer the detail of the **Shadow Map**. Too low a setting results in artifacts that manifest as pixilated shadows with jagged edges, or shadows that *jump* while animated.

Remember that the **Shadow Map Size** has a direct correlation to memory requirements. The amount of required memory is equal to four times the square of the **Shadow Map Size** value. Therefore a value of 512 consumes one megabyte of memory ( $512 \times 512 \times 4 = 1,048,576$  bytes = 1 megabyte). The reason for the multiple of four is to account for the four channels used - Red, Green, Blue and Alpha). A size of 1,024 consumes four megabytes of memory.

## Shadow Map Area

When **Use Cone Angle** is active, the default, the area covered by the **Shadow Map** is determined by the **Spotlight Cone Angle**. In other words, the entire area lit by the Spotlight is calculated in the **Shadow Map**.

Deactivating the **Use Cone Angle** option lets you enter an independent **Map Angle**. Use this option when you want to illuminate a large area by the Spotlight but do not have enough RAM to effectively **Shadow Map** the entire area, or when you need only a small area shadowed.



**HINT:** The best-looking Shadow Maps occur when you have the largest Shadow Map Size you can afford and the smallest Map Angle possible for the given Spotlight.

Spotlight.

## Smoothing Out an Edge

You can set the edge sharpness or smoothness of shadows cast by a **Shadow Map** by changing the **Shadow Fuzziness** value. Higher values give a fuzzier edge while lower values yield a sharper edge. A value of 0 results in no smoothing and you will see the square pixels of the **Shadow Map**.



**NOTE:** Also see the discussion "Object Shadow Options."

## Lightening Shadows

An easy technique to lighten a light's shadow is to clone the light and set the **Shadow Type** to **Off** for the clone. Then adjust the **Light Intensity** of both lights so that they add up to the value for the original single light. The more intensity you give to the non-shadow light, the lighter the shadow.

## Shadow Color

A **Shadow Color** setting is available on the **Light Properties Panel**. If a shadow-casting light uses the default black color (0, 0, 0), it will have no direct effect on areas that are blocked by opaque objects. This matches the behavior of previous versions of LightWave (and the real world). Using colors lets you tint shadows without having to adjust ambient lighting. This option works with both ray tracing and **Shadow Mapping**.

The original light color dissolves into the shadow color as the amount of shadow increases from zero to 100%. To change the density of shadows without tinting them, set the **Shadow Color** to be the same as the **Light Color** (Light Properties) and then adjust the **Shadow Color** value (using the HSV colorspace controls. Note: Right-click on the **Color** settings to change to HSV from RGB).





## Lens Flares

LightWave was the first professional 3D package to incorporate lens flares into its arsenal. Simply put, a lens flare is an artifact that appears in the lens elements of a camera when you aim it toward a source of light. It is by all definitions a defect—a limitation of the camera lens. However, by imitating this defect, you can add the realism of using an actual camera to LightWave animations.



**HINT:** While LightWave's lens flares are easy to use and the effect is often very appealing, remember that lens flares are an artifact that most film and video directors try resolutely to avoid. Judicious use of lens flares can enhance your work, but overuse can quickly detract from it.

You use lights to position lens flares since lights cause them. Lens flares are implemented as a light property. You can use any type of light; however, Linear and Area lights will generate only a single lens flare, as do the others. The light retains all of its normal lighting functions, but when it appears within a frame, a lens flare is generated.



**NOTE:** Make sure that Enable Lens Flares is active on the Global Illumination Tab in the Render Globals Panel or LightWave will not create lens flares.

Understand that lens flares are an additive effect, which means they are added to a rendered image after LightWave calculates the scene's appearance. Because of this, you can accidentally create too many flares, or flares so *hot* that they wash out other items in the scene. Of course, this may be a desired effect, such as when a large explosion takes place.

Since lens flares are an additive effect, they will not show up in an alpha channel saved image. Moreover, since flares theoretically exist only in the lens of the camera, they will not show up in any reflections or refractions in an object's surfaces.



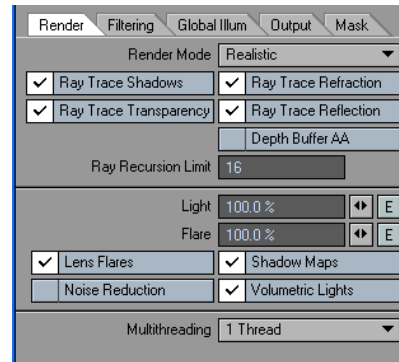
**NOTE:** You can add lens flares to the alpha channel of your render by adding the Flare2Alpha LScript to the Image Processing > Image Filters window.

## Lens Flare Options

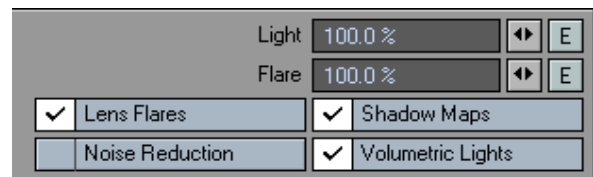
The central glow of a lens flare light source is white, while the glow around it is tinted by the light's **Light Color**. Unlike real lens flares, LightWave gives you tremendous control over how your flares look. Additionally, since lens flares are used by LightWave animators for more than camera lens artifacts, there are some options that help the flares look more like real physical phenomena, like fire, glowing, and explosions.

### To set up a lens flare:

**Step 1:** Open the Render **Globals Panel Render Tab**.



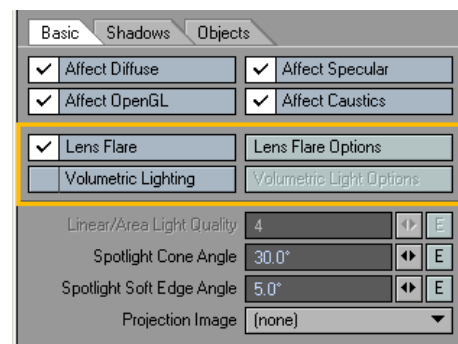
**Step 2:** Make sure there is some percentage of **Global Lens Flare Intensity**. 100% should work in most instances.



**Step 3:** Activate Enable Lens Flares.

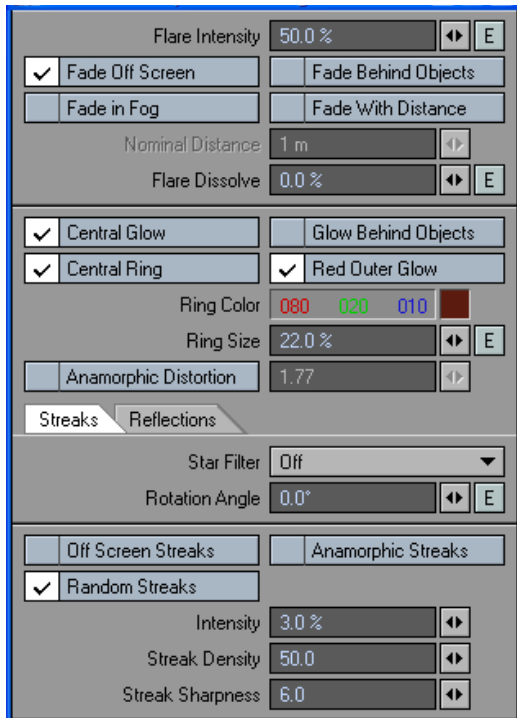
**Step 4:** Select the light.

**Step 5:** Click **Lights Properties Panel > Lens Flare** to turn the option on for the selected light.





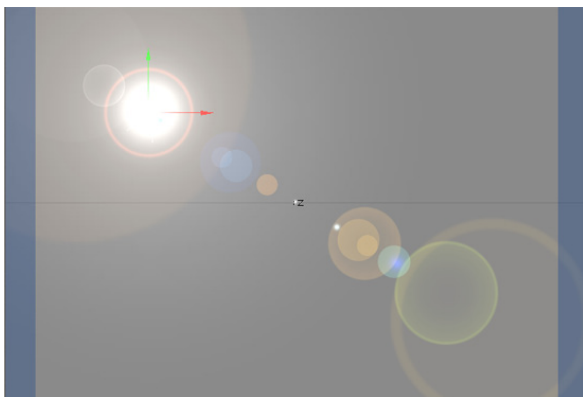
**Step 6:** Click **Lights Properties Panel > Flare Options** and set the options as desired.



NOTE: You can also access the Lens Flare state and Lens Flare Options Panel from the Lights Properties Panel.

### Lens Flare Viewport Preview

You can see an approximation of your **Lens Flare** settings in any viewport using the Camera view. You must activate the **OpenGL Lens Flare** option on the **Display Options Tab** of the **Preferences Panel** (**Edit > Display Options**). Note that this is just an approximation of how the actual lens flare will appear and the actual rendered effect will likely be somewhat different.



NOTE: Not all lens flare options can be seen in the preview, so you need to do test renders (**F9**).

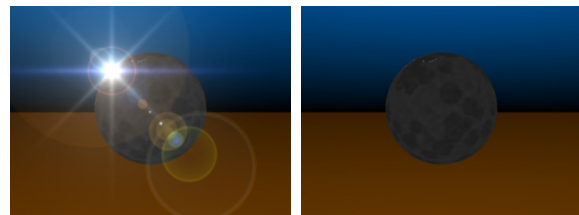
### Fade and Dissolve Options

**Flare Intensity** sets the brightness of the lens flare. The default value is a good starting point.

Select **Fade Off Screen** when you want a lens flare light source to reduce its flare intensity automatically as the light source enters or exits at the edge of the screen. This simulates the properties of actual lens flares within a film camera. When this option is not active, the lens flare will remain constant as it moves off screen.

Activate **Fade In Fog** when you want a lens flare to automatically reduce its flare intensity as it is affected by the minimum and maximum distances set for fog on the **Effects Panel's Volumetrics Tab** (**Windows > Volumetrics and Fog Options**). The further into the fog the lens flare is, the less bright it will be. Once past the maximum fog distance, the lens flare is completely dissolved by the fog. If **Fade in Fog** is not selected, the lens flare will remain bright no matter how far away it is.

Select **Fade Behind Objects** when you want a lens flare to reduce intensity automatically as the light moves behind other objects in the scene. This simulates the properties of actual lens flares within a film camera. Flares even change color when passing behind stained glass windows with the **Fade Behind Objects** option, which uses ray tracing to determine when lights are obscured by objects. If you do not select this option, flares will appear through objects.



Left: Normal Flare Settings, Right: Fade Behind Objects selected

Selecting **Fade With Distance** will automatically fade a lens flare as its distance from the camera increases. If you bring a flare closer to the camera, it grows brighter.

The **Nominal Distance** field, active only when **Fade With Distance** is selected, is the distance from the camera where the flare is at its input intensity. For instance, if your **Flare Intensity** is 100%, and your **Nominal Distance** is 10 meters, moving the flare to a distance of 20 meters causes the intensity to drop to 50% (at twice the distance it drops to one-half the brightness). Conversely, at half the distance, 5 meters, it climbs to twice the intensity, 200%.

Enter a **Flare Dissolve** value to adjust the transparency of the lens flare effect. This option is handy when you wish to see large streaks of light coming from the lens flare but do not want a bright hot spot at the center of the flare. The higher the **Flare Intensity**, the larger the streaks (if selected), and the brighter the flare. **Flare Dissolve** values below 0% or above 100% are not useful, since the flare is either fully visible or fully invisible and cannot be more so.



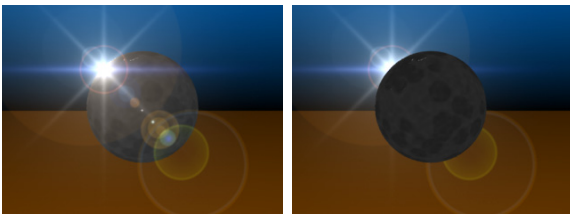


## Glow Options

**Central Glow** is a glow of light at the center of the light source flare. This is the color of the light source. **Red Outer Glow** is a luminous, soft-edged red glow around the light source, available only when **Central Glow** is active. You may want this option off for deep undersea environments, where the color red is not visible.

**Glow Behind Objects** simulates a glow that surrounds a light source. This effect is different from that of a true lens flare, and should not be confused with that effect. A true lens flare is a phenomenon that occurs within the camera lens and thus appears to be in front of all objects in a scene. When the light source that causes the flare is obscured by objects between it and the camera lens, the flare fades or disappears depending on whether the light is partially or completely obscured (an effect that is handled by the **Fade Behind Objects** option).

**Glow Behind Objects** is designed to simulate glows physically located at the light source rather than within the lens. These glows are caused by the illumination of a medium surrounding the light (such as murky water or foggy atmosphere) as opposed to true lens flares, which are caused by diffraction and reflections among the glass elements inside a lens assembly. Unlike true lens flares, glows, at a distance, can be partially visible even if the light source itself is obscured.



Left: Normal Flare Settings, Right: Glow Behind Objects selected

## Central Ring

**Central Ring** is a small ring of light, like a halo, surrounding the light source. The **Ring Color** option controls its color. The ring's size is set by the **Ring Size** value, which defaults to 22%.

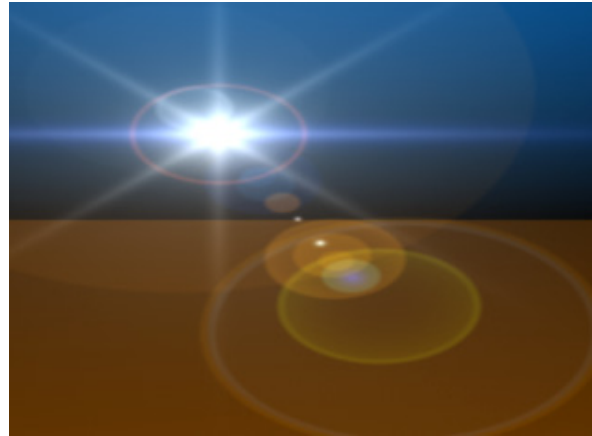


HINT: Use an enveloped Ring Size for shock waves or other similar anomalies.

## Anamorphic Distort

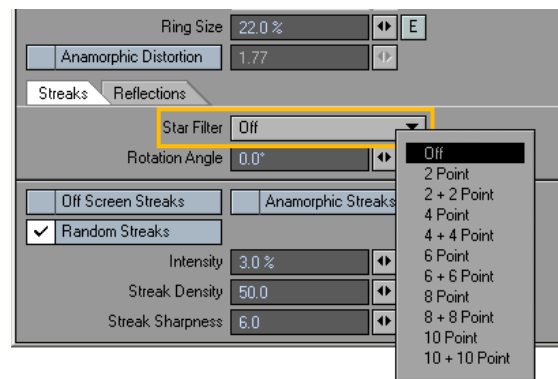
**Anamorphic Distortion** causes the lens flare to stretch horizontally. This simulates the effect of the wider lens flares you see in motion pictures filmed in Panavision. The **Distortion Factor** acts as an aspect ratio control. The height of a flare is determined by its **Intensity** setting and proximity to the camera. The relative width of the flare is determined by the **Distortion Factor**.

The default value, 1.77, sets the flare's width at 1.77 times the flare's height. A default value of 3 causes the flare's width to be three times wider than it is tall, and so forth. The higher the value, the more elongated the flare. To distort the flare vertically, enter a value less than 1. The lower the value, the taller the flare.



## Streaks

**Star Filter** is a pop-up that lets you choose the number of *points* on the star-shaped streaks that emanate from the flare. The + *n* secondary options add in-between minor streaks. You can rotate the streaks with **Rotation Angle**. Positive values rotate the streaks clockwise as seen from the camera. Negative values rotate the streaks counterclockwise.



**Off Screen Streaks** allow lens flares that move off the visible screen area to continue to cast occasional streaks across the LightWave camera and into the scene.



NOTE: You must activate Fade Off Screen to use Off Screen Streaks.

**Anamorphic Streaks** are elliptical horizontal blue streaks emanating from the lens flare light source. This simulates the effect of similar streaks seen in motion pictures filmed in Panavision.



The **Random Streaks** options adds dozens of tiny random streaks of light emanating from the light source. The intensity of these streaks is governed by the **Streak Intensity**, which is a percentage of the brightness of the lens flare. The higher the value, the brighter and larger the streaks. The default value of 3.0% produces streaks that closely match those found in film.



**Streak Density** sets the number of random streaks. The number you enter is used as a rough approximation to determine the actual number of streaks, but your results will be close to the value entered. Higher values make for more streaks.

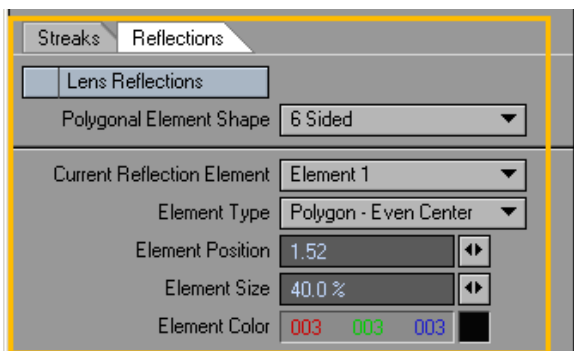
**Streak Sharpness** determines how defined the random streak edges fall off and blend into the background. Low values blur the streaks together, making for softer, wider streaks. Higher values create very distinct streaks. The default value of 6.0 closely matches streak effects found in film.



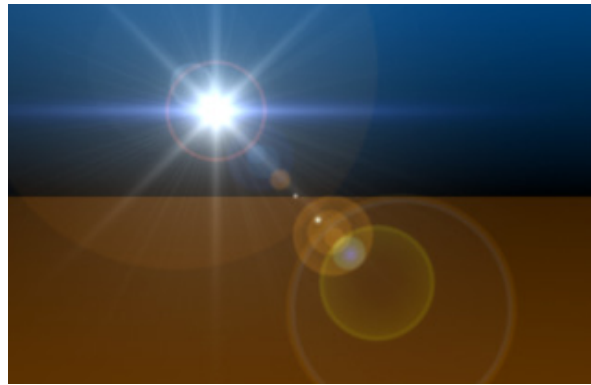
NOTE: A very low Streak Sharpness value (like .0001) will blur the streaks into each other to create a glow-like effect.

## Lens Reflections

The **Lens Reflections** option adds reflections of light in the LightWave camera, as if it were shooting the scene through a standard camera lens assembly. This effect emulates the multiple lens elements that make up a typical lens. Such flares commonly occur whenever you aim a camera at an intense light source.



If either the light source or the camera is moving, then the reflections will move across the screen also. This can be a dramatic effect.



HINT: To get the most visibility out of Lens Reflections, place the lens flare light near the side or corner of the camera's view.

The **Polygonal Element Shape** will change the shape of the reflections. The *sided* settings simulate lenses with *n*-blade irises. Since there's just one iris per lens on a real camera, all polygonal reflections in a particular flare have the same shape. Settings are global for the scene, which means that different lights can't have different patterns.

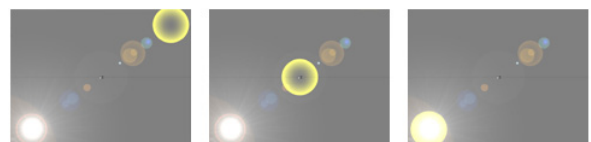


Some

of the available Shapes

You can modify the default **Element Type**, **Element Position**, **Element Size**, and **Element Color** for each of the sixteen reflection elements. To edit an element, first select it using the **Current Reflection Element** pop-up menu. The elements are aligned in a straight line going through the center of the camera's view through the light.

By default, the Elements 1 through 16 are aligned starting near the light, then through the center and on to the opposite side. However, each element's position could be anywhere, if you want it to be. An **Element Position** of 0 is at the center of the screen, 1 at the light's position and -1 on the opposite side. You can use values beyond 1 or -1 to move elements past those positions.



Element 15 (Yellow) Left: Position = -1, Middle: Position = 0, Right: Position = 1

An **Element Size** of 100% vertically fits the element to the camera resolution. It can be clipped or there can be space to the left and right, depending on the relative **Camera Resolution Height** setting.



NOTE: A reflection using a Bright Center Element Type will not be the same size as the Even Center.



The **Element Type** setting determines the shape and density characteristics of the reflection.



Circle- Bright Center, Even Center, Dim Center



Circular- Ring



Polygon- Bright Center, Even Center, Dim Center



Circular- Ring



Rainbow Ring



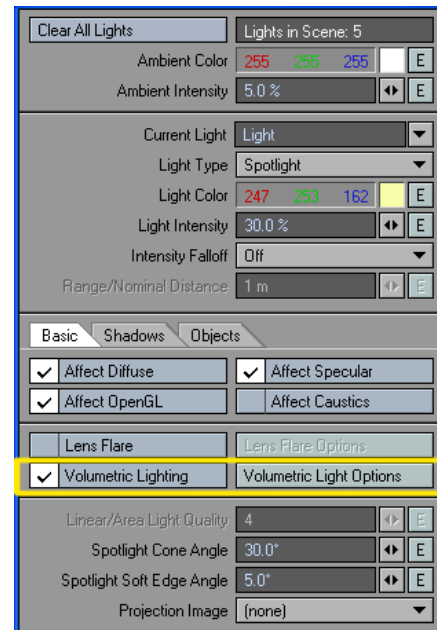
NOTE: If you select a Polygon Element Type, you should also choose a shape other than Circular as the Polygonal Element Shape.



NOTE: The viewport display will not show all lens reflection attributes.

## Volumetric Lights

**Volumetric Lighting** turns a selected light into a *volumetric light*. This gives lights — or more accurately their beams — physical volume. These effects are common in everyday life and can play a key role in creating dramatic and realistic environments.

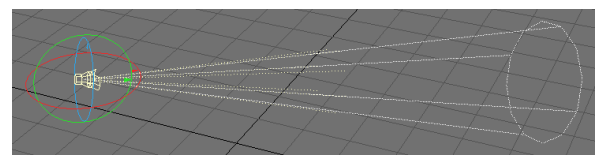


NOTE: Use Enable Volumetric Lights On/Off to enable or disable all volumetric lights in the scene.



NOTE: Also see the Volumetric Antialiasing option on the Volumetrics Tab of the Effects Panel, explained on page 104.

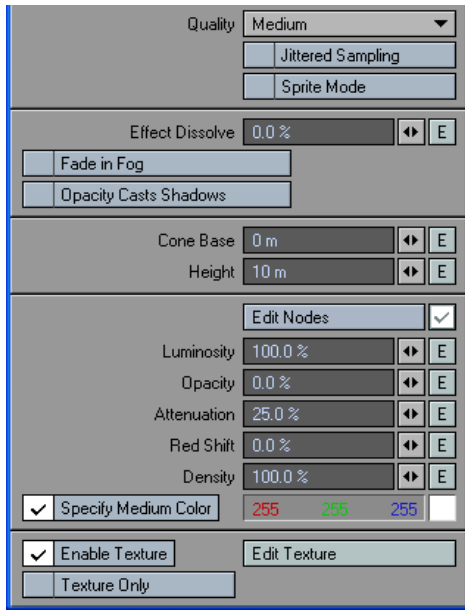
An outline of the volume for volumetric lights is displayed in Layout's viewports.



NOTE: Linear and Area lights cannot be volumetric.



Click the **Volumetric Light Options** button to display a selected light's volumetric options.



**Sprite Mode:** This mode will render the effect much faster. However, there are a few limitations, the biggest being that volumetric shadows will not be cast. Also, in some cases, the light textures will look unnatural — this occurs mostly with spot lights when looking in the direction of the light. Note that the **Quality** setting becomes unavailable when using this mode.

**Quality:** Use the lowest **Quality** setting that achieves acceptable results. Obviously, higher settings will take longer to render.

**Effect Dissolve:** **Effect Dissolve** reduces the amount of the effect.

**Fade in Fog:** Activate **Fade in Fog** for volumetric lights to dissolve into fog, just like lens flares can.

**Opacity Casts Shadows:** This option causes volumetric lights to cast *ray-traced* shadows based on the volumetric light's **Opacity** setting. If **Opacity** is set to 0, no shadow will result. (Since shadows are ray-traced, the illuminating light must have **Shadow Type** set to **Ray Trace** and **Ray Trace Shadows** needs to be active on the **Render Options Panel**.)

**Radius:** **Radius** is the radius of the volumetric beam for Distant and Point lights.

**Cone Base:** If the light is a Spot, the **Cone Base** setting (replaces the **Radius** setting) lets you adjust where the base of the cone begins.

**Height** is the length of the beam for Spotlights and Distant lights.

**Edit Nodes:** Activated, this option allows you to use options from the Node Editor.

**Luminosity** is the strength of the effect. Values can be positive or negative. Negative values have no physical sense, but you can use them to create interesting effects. A 0 value means that no light is emitted from the medium, which can be useful if you want dark smoke or dust effects.

**Opacity** is the effect's surface opacity, that is, how non-transparent the effect is. A high value will cause an object inside and behind to blend into the effect. Negative values are allowed, and can cause dense areas to become very bright. You may need to increase

**Luminosity** to compensate for increased **Opacity** settings.

**Attenuation** determines how fast the effect declines in intensity as the light leaves the emitting point. The default value is 25%. Low **Attenuation** values will make dense areas become very bright and totally saturated. High values will make dense areas darker and make the volume boundaries brighter.



NOTE: Some level of attenuation always exists, even if you set Attenuation to 0%.

The **Red Shift** setting works in conjunction with **Attenuation**, controlling the interior behaviour of light. When light scatters from one point, it must travel inside the medium from the point of emission to the viewer. During this travel, the light attenuates depending on the length of the travel and on the light's wavelength.

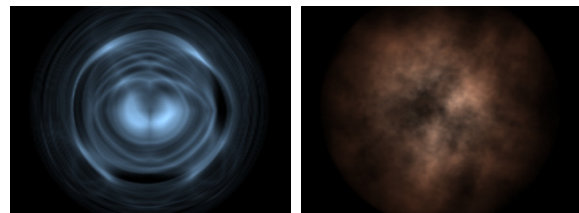
The sky is a good example of this: at sunset, light from the sun gets red because the thickness of the atmosphere crossed at the horizon is more than at the zenith. Red light is attenuated less over long distances, thus the horizon is red while the sky is blue. The **Red Shift** parameter is used for this: 0 values means that **Attenuation** is not wavelength dependent, which means that light will be attenuated with no color changes. Positive values will make the color shift towards the red, while negative values will make a shift towards the blue.

**Density** controls the global density of the medium. This essentially works like a multiplier for **Luminosity** and **Opacity**.

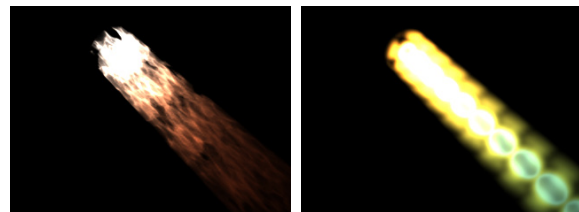
**Specify Medium Color** — Activate this option to set the color of the medium (i.e., the substance) through which the volumetric light is transmitted.

## Light Textures

To add a texture to the volumetric effect using the **Texture Editor**, click **Edit Texture**. The **Enable Texture** option must be active to apply the **Texture** settings. Deactivating it will not affect any existing settings. The **Texture Only** option means that the texture is not blending with the base color, but replacing it totally.



Left: Marble Texture on Point Light, Right: Fractal Noise Texture on Point Light



Left: Gradient and Turbulence on a Distant Light



## The VIPER Window and Presets

To use the Preset Shelf or VIPER window, activate those options on the main Layout interface, or by pressing **F7** for Viper and **F8** for Presets. With these weapons you can quickly set up lighting as you like.



NOTE: You can preview animated textures with VIPER.

VPR will also let you preview lighting as you set up the scene.

## Volumetric Shadows within Lights

To achieve volumetric shadows within lights, make sure you enable **Shadow Maps**, if you are using shadow map shadows. If you are using ray-traced shadows, activate the **Ray Trace Shadows** option on the **Render Options Panel (Rendering > Render Options)**.

Note that it is much better to use **Shadow Maps** because they render much faster and you can control the fuzziness of the shadows. If you find that the shadow “rays” are creating aliasing patterns, you can easily correct this by increasing the **Shadow Fuzziness** value (Light Properties). This is a better solution than increasing the **Volumetric Light Quality** setting, which increases render times.

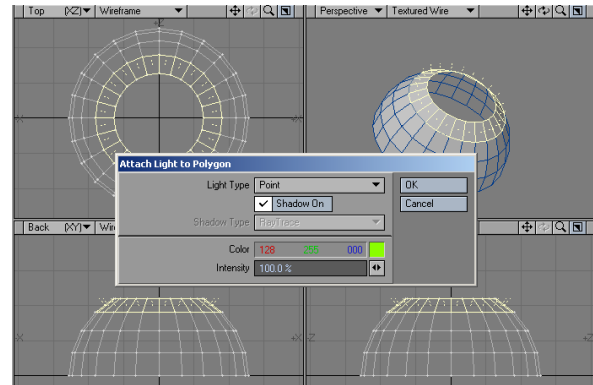


Note: If you are using the Node Editor with Volumetric Lights and using the Raytrace Node, or any node that samples the scene recursively, or a node that multi-samples, this can increase render times significantly.

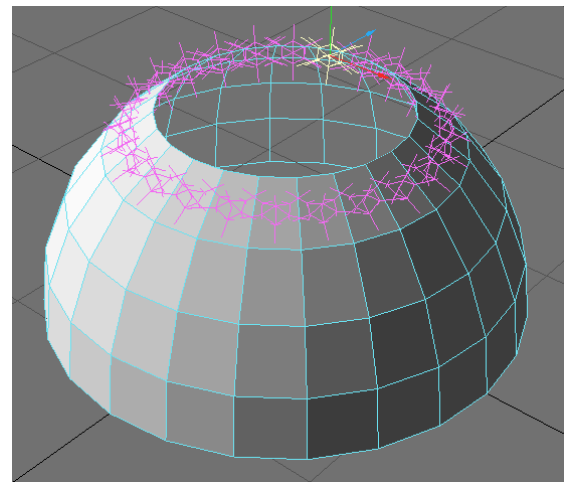
## Luxigons

Luxigons are very similar to Powergons, but are specialised just to add lights and set certain light properties. To use, in Modeler select the desired polygons and choose **Setup > Add Luxigon**.

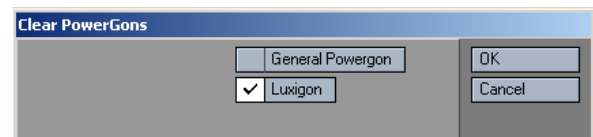
When the dialog appears, select the type of light you wish to add and set its properties.



Next, load the object into Layout and choose **Items > Lights > Convert Luxigons**. If you wish to use an existing light's properties, select it in the **Clone Item** pop-up menu. Enter the desired name of the new lights in the **Light Name** field. After you click **OK**, the defined lights will be created, positioned at the center of the appropriate polygons, aligned with the polygon's normal, and parented to the object.



You can clear Luxigons from selected polygons using the **Setup > Clear Powergons** command.







## Chapter 16: Scene Management





## Scene Management Introduction: Organizing your Scenes

Taking on a CG project can be a challenge, whether it is a simple scene with one object, one light, and one camera, or a complex scene with hundreds of objects, bones, surfaces, dynamics and everything else in-between. Keeping it all organized is key to a successful project and this chapter is all about the tools you can apply to keep your scenes manageable.

Starting with the basics of loading and saving items in a scene, the chapter then progresses from management tools like the List Manager onto more complex tools such as the Scene Editor and Graph Editor.

## File Menu

Layout's **File** menu contains common file management commands and tools.

### Clear Scene

Clear Scene — (**File** > **Clear Scene**) will remove all data from Layout and return Layout to its default setup. All unsaved data will be lost.

### Load Commands

The important difference between loading an object directly versus loading from a scene lies in what is stored in an object file versus what is stored in a scene file. The object file contains only the object geometry and surface settings — what the object looks like in its most basic state.

There is no movement information in an object file.

Movement information is saved in the scene file. However, this is not only movement in the normal sense, like moving an object from point A to point B, but also includes movement of the points in their positional relationship to each other.

Moving points can change the shape of an object, sometimes dramatically. However, generally, the number of points and polygons, as well as their relationship, remains the same.



**WARNING:** Understanding object and scene files is fundamental to understanding LightWave.



**NOTE:** Due to the increase of the Layout scene file format version to 5, the "Export" menu in Layout has been updated to accurately represent the scene file versions to their LightWave releases.

For example, say we had a flag object that was a flat-subdivided rectangle. In Layout, we could make the flag wave using something called a displacement map — essentially this animates the points in the object. The displacement map settings are saved in the scene file. The underlying flag geometry and surface settings are stored in the flag object file.

**Load Scene** (default keyboard shortcut **Ctrl O**)

### To load an existing scene:

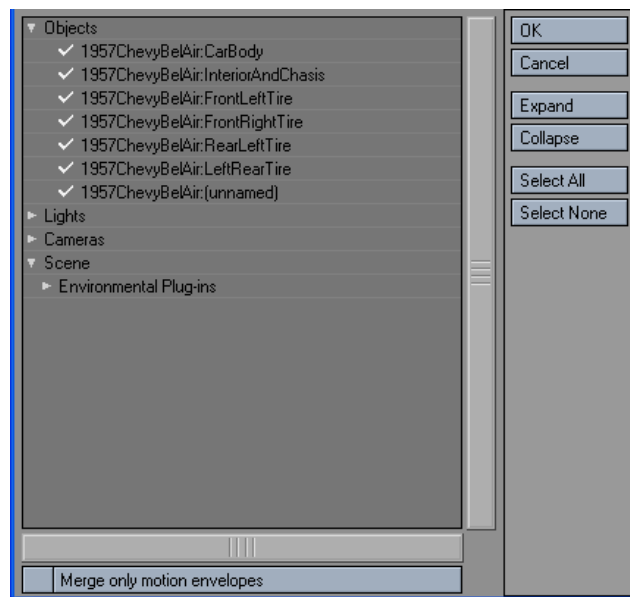
**Step 1:** Make sure the Content Directory is set properly.

**Step 2:** Choose **File > Load > Load Scene** and use the file dialog to navigate to the desired scene file. When the scene is loading, a progress dialog will appear. You may abort the load operation by clicking the Abort button; however, this may result in a partially loaded scene.

### Recent Scenes

Choose **File > Load > Recent Scenes**, and select a Scene file from the submenu.

### Load Items From Scene



Loading objects from a scene is accomplished using **File > Load > Load Items From Scene**. After you choose the scene, a panel will open up and you will have the option to choose which items from the scene you would like to load. Double-clicking on the root of an item type, for example Lights, will enable/disable all items of that type. The object layer number will be listed when loading a multi-layered object.

Expand and Collapse will perform the respective function on the root of the item.

Select all selects all items and Select None deselects all items.

Clicking on **Merge only motion envelopes** will load only the motion envelopes for the selected items.



**NOTE:** Be aware that when the first child in the chain becomes the new parent of the remainder of the chain, some transformation issues may arise as that child is no longer bound by the parent's transformations. These transformations will have to be cleaned up manually after the scene is loaded.

### Revert Scene to Last Saved

This option will close the current scene and re-open it to the last saved version.

**Load Object** (default keyboard shortcut +)

When you set up a new scene, you will need to load the appropriate objects into the scene.

**To load an object file into a scene:**

**Step 1:** Make sure your Content Directory is properly set. This will help with any images that are referenced in the object file.

**Step 2:** Select **File > Load > Load Object** and use the file dialog to navigate to the desired object file. (Note: Multiple objects may be loaded when more than one is selected in the file dialog.)



HINT: Alternatively, you can press the + (plus) key on your numeric keypad, which adds the appropriate item based on

the active edit mode selected at the bottom of the screen (i.e., Object, Bones, Lights or Cameras).

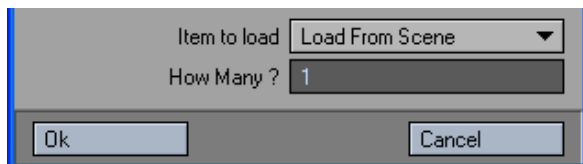
If the object contains multiple layers of geometry, each layer will be loaded as a separate editable item in Layout. An object is initially loaded so that its local Origin is at the global Origin and its local axes line up with the world axes.

**Load Object Layer**

Choose **File > Load > Load Object Layer** to load a specific layer from a multi-layer object. A dialog will appear allowing you to define which layer to load.

**Load Multiple**

The Multiloader command allows you to use the standard Load Object, Load From Scene, and Load Lights from Scene, with the added ability of selecting how many times you want to load the selected items.

**Load Motion File**

Load a saved motion file into the selected item by choosing **File > Load > Load Motion File**.



NOTE: Modeler's Path Extrude and Path Clone commands use these files to execute their operations.

## Save Commands

**Save Scene** (default keyboard shortcut S)

This option saves the current scene using its filename. If the scene has never been saved before, this is the same as using Save Scene As.

**Save Scene As** (default keyboard shortcut **Alt S**)

This option opens a file dialog letting you name (or rename) the scene before saving.

**Save Scene Copy**

Choosing File > Save > Save Scene Copy will allow you to save a copy of a scene using a new name.

**Save Scene Increment** (default keyboard shortcut **Shift S**)

This option gives you the ability to save a version of a scene by adding incremental numbers to the scene's name. This is handy when you are wanting to save your scene in various stages.



NOTE: An asterisk will appear next to the names (on the current pop-up menu) of scenes that have been modified since their last save.

**Save Current Object**

You should always save objects after you have altered their surface settings.

To save an object:

1. Make sure your Content Directory is properly set.
2. Select the object you want to save.
3. Select **File > Save > Save Current Object** to save... you guessed it... the current object.

**Save Object Copy**

Choosing **File > Save > Save Object Copy** will allow you to save a copy of an object using a new name, without affecting the object used by the scene.

**Save Object Increment**

This option gives you the ability to save a version of an object by adding incremental numbers to the object's name. This is handy when you are wanting to save your object in various stages.



NOTE: An asterisk will appear next to the names (on the current pop-up menu) of objects that have been modified since their last save.

**Save All Objects**

You can also use the **File > Save > Save All Objects** function to save all objects in the current scene.



WARNING: Use Save All Objects with caution. Be sure it is what you really want to do!



## Save Trans Object

Choose **File > Save > Save Transformed Object** to save an object with any changes of location, rotation, scale, morph, skeletal deformation, displacement map, and so on. The new object is also saved relative to the Origin. The current frame in Layout controls the state of the saved object.



NOTE: There is a unique quality to an object that uses displacement mapping with an image (planar, cylindrical, or spherical) as well as surface color mapped with an image. With the displacement map altering the object's shape, the color map is also bent to follow the contours of the transformed object. However, if you save this reshaped object with Save Transformed Object, it will now be permanently reshaped and the surface color mapping will no longer match.

## Save Endomorph

You can save the current "mix" of your morphs back into the endomorph by choosing **File > Save > Save Endomorph**. The morph will appear on a new "Miscellaneous" tab on MorphMixer. (Note that you may need to remove and then re-add MorphMixer.) Don't forget to save your object if you want to keep the new morph.

## Save Current Light

This command will save a scene file that contains the currently selected light and the lights properties.

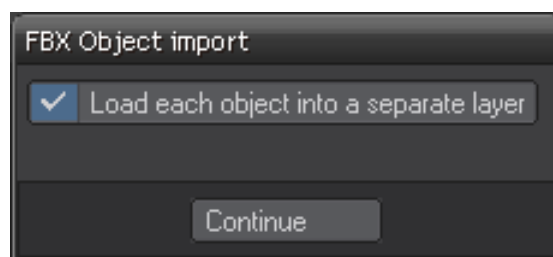
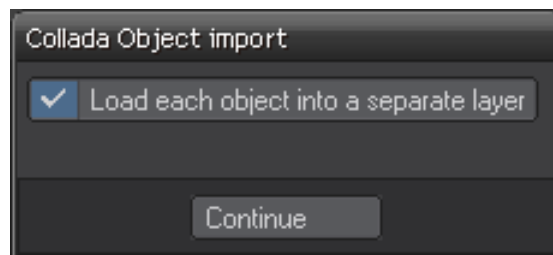
## Save Motion File

To save the selected item's motion path to a file, choose **File > Save > Save Motion File**.

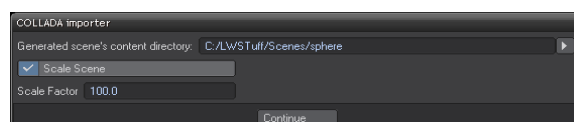
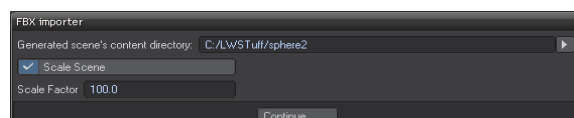
## Importing Items into Layout

### FBX and Collada Import Options

FBX and Collada imports just as a LightWave scene file, go to Load Scene/ Load Object, locate the file. in the load panel, then click Open.

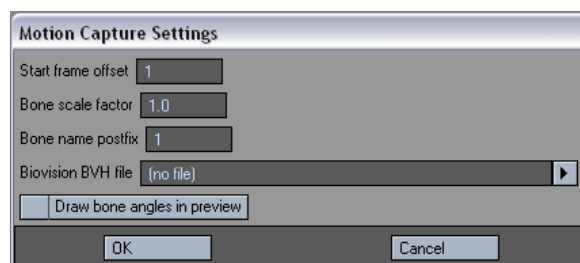


When importing FBX and Collada files as scene files, the following requesters open. The file path is where the objects associated with the imported scenes are stored.



### Mocap\_BVH\_Setup

LightWave provides a couple of plugins to support the BioVision (BVH) motion capture file format. (the other is The MoCap\_BVH\_Setup generic Layout plugin reads a BioVision BVH file, creates bones, and applies the motion capture data to them. This is one, the other is the plugin **Motion Capture Preview** attached to the geometry of a null in **Object Properties**.



Set Start frame offset to the frame you want the motion to begin. The Bone name postfix is simply a number appended to the end of all of the bone names (e.g., LeftKnee\_1).

After you run the plugin, replace the top null (in the created bone hierarchy) with the object to be animated. You could also use the Use Bones From Object feature on the Bones Properties panel.



If you need to change the initial resting position of bones, make sure you reset their rest positions (use the **r** key). You'll probably need to adjust some of the individual bone properties after you run the plugin.

### Lightgen2lw

This plugin will read in HDRShop data that has been generated with LightGen. For more information about LightGen visit: <http://www.ict.usc.edu/~jcohen/lightgen/lightgen.html>

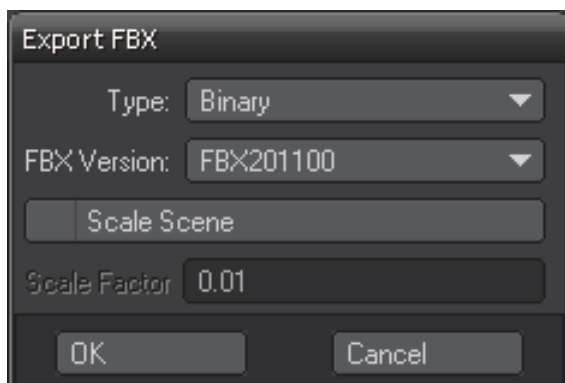
## Exporting Items from Layout

### Save 5.6 Scene

This command will save the scene data to the LightWave 5.6 scene format to ensure that older versions of LightWave can read the file.

### FBX and Collada Export Options

The exporter for Collada will save the file as .dae. The exporter for FBX has different options, such as Binary or ASCII types, and allows you to save as different FBX versions..



NOTE: The FBX and Collada exporters do not export information associated with plug-ins, including plug-ins native shipped with LightWave.

### Export Scene As VRML97

The **VRML97 Exporter** plugin (**File > Export > Export Scene as VRML97**) creates a VRML97 World based on the current scene. The VRML output complies with the ISO-VRML97 specification. The objects in the scene may be saved as separate files into the Content Directory or an alternate path. The following list shows some highlights:

- Accurate Translation
- Keyframed hierarchical animation
- Light intensity envelopes, including ambient
- Non-linear fog
- Color image texture mapping using projection or UV maps
- Solid, non-linear gradient and image backgrounds
- Support for SkyTracer warp image environments
- Particle animation with single-point-polygon object to PointSet node conversion
- Two-point-polygon object to IndexedLineSet node conversion
- SubPatch object morph capture (for capturing morph, displacement map and bone effects on SubPatch objects).

High-performance output

3D Sounds

Level-of-detail object replacement animation

Object instancing

Vertex color and lighting support

Multiple custom viewpoints

Custom VRML nodes

Touch activated behaviours

Viewer proximity activated behaviours

Object visibility activated behaviours

Objects output as prototypes (PROTO) definitions (optional)

Scene object ignore

Standard object viewpoints (optional)

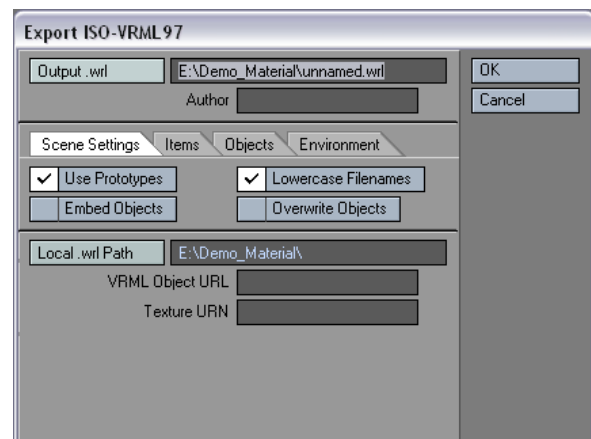
Optional embedded objects for single file scene output!

Optional lowercase conversion for embedded object/image filenames

Direct avatar navigation speed control

Improved compliance for export to VRML97 editors, including conversion of illegal VRML97 names (like 2Legs or My Light), and reflection of illegal negative scaling

### VRML Creation Settings



Output .wrl is the file path for the VRML97 World.

Author is comment text to embed in the file.

**Use Prototypes** is used to define and use objects in the VRML scene more efficiently. Some older importers may not like this, but it is required for morph capture.

When **Lowercase Filenames** is enabled, the filenames used in the file can all be converted to lowercase. This can be helpful on UNIX-based Web servers, where filenames are case sensitive, and links with mismatched cases will fail.

Use **Embed Objects** to include the geometry for all meshes in the main VRML97 World file. This may be more convenient, but for complex worlds, or reusing objects, it is less efficient. Using external object files allows the main world to load faster, and display bounding boxes while the objects are loaded. This option must be off for morphing objects as well as LoD objects - loading them all at once would defeat their purpose!

If **Overwrite Objects** is enabled and the **Embed Objects** option is not used, external object files will be created for objects in the scene. If

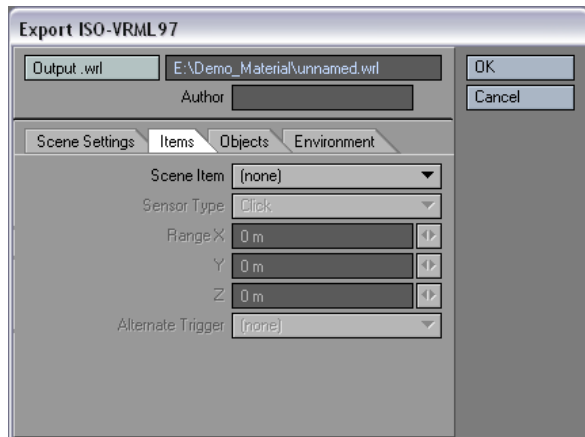


the objects already exist, this option must be enabled to overwrite the objects, thus updating surface or morph changes.

Local .wrlPath is the file path on your machine where external VRML objects will be found and/or saved. This will default to the current LightWaveContent Directory.

**VRML Object URL** is the URL where browsers will search for external objects, this should be the web equivalent of the local path (e.g., `http://www.someplace.net/vrml_objects/`).

Text entered into the **Texture URN** field will be pre-pended to texture map image filenames, as an alternate texture location. This should facilitate work with libraries like the Universal Media textures. This information, when specified, will appear in addition to the regular URL elements.

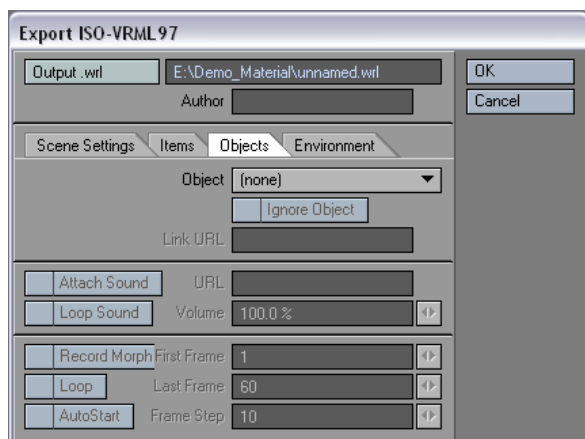


On the **Scene Item** pop-up menu, select a scene element to which you want to apply the settings on this tab.

The **Sensor Type** is the sensor used to start the item's animation.

For some sensor types, like **Proximity**, a distance range is required. When the viewer approaches the item within the **Range**, the animation is triggered.

**Alternate Trigger** is an alternate item to serve as the animation trigger for this item.



On the **Object** pop-up menu select an object whose settings will be edited on this tab.

The **Ignore Object** option will exclude a selected object and its children objects from export.

Use **Attach Sound** to add a sound effect, triggered with the selected

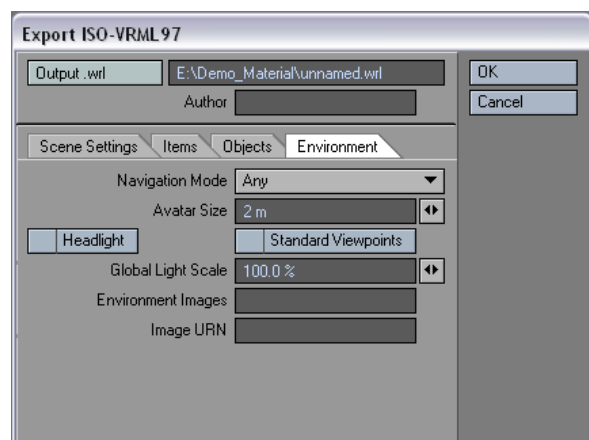
object's animation. Enter the URL for the audio file triggered in the URL field. You can also set the volume and whether the sound should be looped once it has started.

The **Record Morph** option saves a **Morph Object** — a special animated Proto object — in place of the standard external object files. This requires that the exporter step through the animation and capture the deformed mesh at different times. The deformed positions are used in a `CoordinateInterpolator` node hidden in the morph object. This currently works for `SubPatch` objects.

**First Frame** is the starting frame for the **Morph Object** animation. **Last Frame** is the final frame in the morph capture for this object. **Frame Step** is the number of frames between captured morph keys. Making this too small results in huge objects; making it too large results in an animation that is not smooth or points with motion that is too linear.

Enable **Loop** to repeat the morph animation, once it has been triggered.

Enable the **AutoStart** option to make the animation start as soon as the world is loaded.



Use the **Navigation Mode** pop-up menu to set the initial navigation mode for Web browsers.

Enable **Headlight** for good defaults in dark places.

**Standard Viewpoints** creates extra **ViewPoint** nodes (top, left, etc.) for scene and external objects.

**Avatar Size** lets the browser set appropriate movement for the dimensions of your world.

**Global Light Scale** globally scales all light intensities.

Environment Images are warp images generated by SkyTracer. These map nicely to VRML's idea of environment mapping. Enter only the basename portion of the image files. (For example, if you had `skyWarp__back.jpg`, `skyWarp__front.jpg`, etc., you would enter `skyWarp`.) Note that any panoramic images should be compatible, provided they are renamed to match the SkyTracer filename convention.

The text in the Image URN field will be pre-pended to the image filename and added to the list of URLs for the environment image.

## So What is VRML?

VRML, also known as ISO-VRML97 (ISO/IEC 14772-1:1997), stands for Virtual Reality Modeling Language. It is a standard for describing 3D objects and scenes via the Internet.

Like HTML-based web pages, VRML worlds can contain links to remote files. However, rather than using text or images for links, VRML uses 3D objects. As a result, the Web browser for VRML resembles





a 3D animation program or video game more than a word processing program.

VRML worlds can be embedded in HTML pages and vice versa. VRML models are based on either primitives, like spheres, cubes, and cones, or, more likely, sets of points and polygons. Since the latter is basically the approach used by LightWave3D's polygonal models, there is a pretty good match between LightWave scenes and VRML worlds.

Before you can view any of your VRML creations, you'll need to get a VRML 97 Browser. The VRML files produced by LightWave are text files that follow LightWave's style of separate object and scene files. This is not a requirement of VRML, but a powerful feature that lets a VRML scene include objects from different files, even from some remote library.

These external objects in the scene file consist of a file URL, a bounding box, and a set of position, rotation and scaling transformations. The bounding box information is used by browsers to render stand-ins while the objects are loaded.

VRML scenes also include multiple point lights, directional lights, and spot lights with adjustable cones. The VRML equivalent of the LightWave camera is a viewpoint. The exporter will add a named viewpoint for each camera in the LightWave scene, which browsers can use to jump between points of interest or standard views. In addition, VRML objects created by LightWave may include a set of standard viewpoints for the object.

## Animation

Objects in your LightWave scene that have keyframes in any motion channels will be given linear motion keys in the VRML file, through PositionInterpolator and OrientationInterpolator nodes.

The Pre Behaviour and Post Behaviour set for the channels in the LightWave motion has a critical influence on the VRML behaviour of an object. If the Pre behaviour is set to Repeat, the motion will begin when the world is loaded and keep on playing. Otherwise the motion will begin when the item is triggered. If the Post behaviour is set to Repeat, the animation will loop until re-triggered, otherwise it will stop after playing.

The default triggering is a click (TouchSensor) on the object that causes the animation to run from the beginning. Currently, the TouchSensor switch is placed on the highest-level animated object in a hierarchy, and triggers the animation of all the children simultaneously (as one would expect).

Morphing in VRML uses a CoordinateInterpolator node. The node is part of the Proto in the object file, if morph data has been captured. For this reason, Prototypes should be enabled and embedded objects disabled for morphing worlds.

## Surfaces

Double-sided surfaces are not supported in VRML97. Thus LightWave objects with polygons whose surfaces are double-sided are translated as if they weren't double-sided. VRML objects that seem to be missing polygons may actually have double-sided surfaces that need to be either flipped or aligned in Modeler. If the surface is truly meant to be double-sided, you will need to model the geometry with double-sided polygons.

If your model has a texture map image associated with it (color only, not diffuse, specular, etc.), there are a few tricks that can minimise the nuisance of hand-editing your VRML models. Since some browsers will have to load the image named in the object, that image name, saved in the LightWave object, is critical.

It pays to use LightWave's Content Directory system properly, so that the image path will be relative to that content directory (i.e., images\wood.jpg rather than C:\NewTek\images\wood.jpg). You may also want to move the image to the Content Directory so that the name in

the object will have no path, and browsers will seek the image in the same directory as the object.

In any event, wherever the VRML object finally resides, you will want a matching directory hierarchy where the browser will find the image or you can just edit the VRML file.

Another image issue is that of file format. JPEG and GIF images are almost universally supported on the Web, but the PNG format is gaining acceptance as a modernised, yet unencumbered, replacement for GIF. JPEG images are nice and small, and compression artifacts should be virtually invisible at Web/VRML resolutions. If you have nice high quality texture images for your rendering work and want VRML versions, make smaller JPEG versions of the images for the Web. Large textures may be limited by the browser's rendering engine in most cases anyway. When you install the VRML model, just use the smaller JPEG image or edit the VRML file.

## LightWave VRML Implementation

The organization of LightWave's VRML object output follows that of LightWave's own object format. A list of XYZ coordinate triples describe the vertices in the object. For each surface, there is also an IndexedFaceSet node that holds the polygons with that surface, described as a number for each point in the polygon, which refers to an entry in the main list of point coordinates. There may also be an IndexedLineSet node or a PointSet node containing any two-point and one-point polygons.

If the original LightWave object had a color texture map image, there will be an image file name and a set of texture coordinates. Texture coordinates, also known as UV coordinates, are 2D pixel positions in an image. They describe how the image lies on the 3D surface by pinning certain pixels to each polygon's vertices. These values can be calculated from LightWave's mapping and texture size settings.

In the case of planar UV mapping, U and V are simply x and y, (for Z-axis planar). Spherical UV mapping yields U and V coordinates somewhat analogous to longitude and latitude, with the U's all bunching up at the poles. Cylindrical mapping uses U's from the spherical case, then the V's are the coordinate lying along the texture axis. If the LightWave texturing is using UV mapping already, then these coordinates are used, since VRML texture coordinates are defined in a per-polygon fashion (i.e. discontinuous UVs).

The entire object may be embedded in a VRML Anchor, which makes it an active link on the Web. If you supply a URL for the object when you create it, then anytime that object appears in a scene, it will act as a clickable link to some other page. This should be used sparingly, as it can be quite annoying to keep jumping around the web when you're just inspecting an object.

The uses for URLs in your objects can range from booby traps or ads for your favorite Web site, to inventory data for some widget. A nice example is a VRML origami site, where each step in the folding of a paper menagerie has a simple model with a link to the next stage. This is similar to the VRML level-of-detail mode, where multiple models are grouped together and the viewer's distance determines which model, if any, is actually rendered.

## Performance Notes

Although the VRML format is capable of describing complex scenes, current 3D browsers are limited by the real-time rendering capabilities of their underlying computers. Thus, exquisitely crafted models with painstaking detail, suitable for those print-res close-ups, may fail painfully when they enter the realm of VRML renderers. To avoid the twin perils of long download times and slow rendering, remember: the first key to VRML success is efficient, low-polygon count Modeling.

Similarly, elaborate layers of diffuse, specular, and luminosity textures, whether images or algorithmic, will not survive any conversion to VRML. Don't even ask about bump maps, displacement maps, or surface shaders. Love it or leave it, VRML supports a single image map for a color texture, as well as diffuse, color, specular, and





transparency values. Since that texture image may very well have to fly through a modem, you'll probably want to keep it small.

Elaborate textures and lighting can be baked into a model's image map however, and lighting effects and coloring can also be baked into vertex color maps.

PointSetobjects are stored most efficiently if there is only one surface per object. Otherwise, duplicate references to the vertices are required. For large scenes, this could be significant.

## Scene Tags

Many of the VRML attributes set in the exporter UI are stored in the Lightwave scene file as comments. These comments can be viewed and edited on an item-by-item basis with the Comments (Layout Generic) plugin. These Comments should be formatted as `<Tag>=<value>` where the Tag is one of the following:

tem — Tag Name/Usage Description

URL — URL=<url> (URL="http://etc."  
Item URL, overrides object and children's URLs.

SOUND — SOUND=<url> [<volume> <loop?>]  
Sounds can be added to objects. Currently these are triggered with any animation.

TOUCH — TOUCH=  
Trigger when mouse is over object (mouse grope).

PROXIMITY — PROXIMITY=W H D  
Trigger if viewer enters active region (WxHxD).

VISIBILITY — VISIBILITY=  
Trigger when viewer sees object.

INCLUDE — INCLUDE=<filename>  
Dump contents of file directly into output.

IGNORE — IGNORE=  
Skip this object and its children.

TRIGGER — TRIGGER=<object>  
Other object for sensor.

VRML — VRML=nodeName{ node fields}  
Node creator, dump node from comment into file.

LABEL — LABEL=<text>  
Create text node.

MORPH — MORPH= <start> <end> <step> <loop?>  
Morph animation capture. Creates external MorphObject.

LOD — LOD=<objectfilename> [<range>]  
Level of Detail node. Use multiple tags in order of decreasing complexity (increasing range).

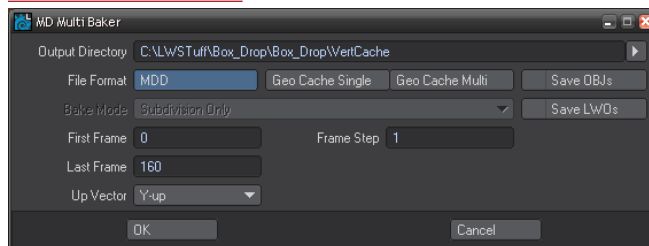
Camera — Tag Name/Usage Description (stored in first camera in scene)

NAVIGATE — NAVIGATE=<type> [<speed>]  
NavigationInfo Type is one of NONE, WALK, EXAMINE, FLY, or ANY. Browsers may restrict user navigation with this.

ENVIRONMENT — ENVIRONMENT=<basename>  
Override scene background image with URLs for front, back, left, right, and top images named like basename\_\_front.jpg, etc.

HEADLIGHT — HEADLIGHT=  
NavigationInfo Headlight on, if present.

## MD Multi Baker



Similar to MD\_Bake, MD Multi Baker contains the following options for exporting motion data and objects:

**Output Directory:** This sets the directory where the MDD files will be saved. The default directory is the Bin/VertCache folder where the Layout is started.

**File Format:** Determines the file type the motion data will be saved as.

**Bake Mode:** Sets what the motion data will record. Some file types only recognize data from specific mesh types.

**Cage Only:** The cage is the mesh before any subdivision is applied. Choosing this option will apply only motion data of the cage.

**Subdivision Only:** Subdivision occurs after the cage. Choosing this option will record only the subdivided mesh data.

**Cage and Subdivision:** This option will record both the cage and subdivided mesh data.

If you would like to export the objects as an OBJ file, select **Save OBJs**.

If you would like to export the objects as a LWO file, select **Save LWOs**.

**First Frame:** Sets the first frame to start recording on.

**Last Frame:** Sets the last frame for recording.

**Frame Step:** Sets the number of steps for recording. A setting of 1 will record the next frame, while a setting of 2 will record every other frame.

**Up Vector:** Sets the up vector. Consult your specific software package to determine which direction the Y-Axis is set but most packages, including LightWave set the Y-Axis to up.

One advantage for **MD Multi Baker** over MD\_Bake is that it can export multiple object's motion files at once. Select all of the objects you would like to export and activate MD Multi Baker. After selecting the setting you want select OK and MD Multi Baker will bake out all of the motion data for each object you have selected. Each motion file will have the name of the object from which it originated.



## Shockwave 3D

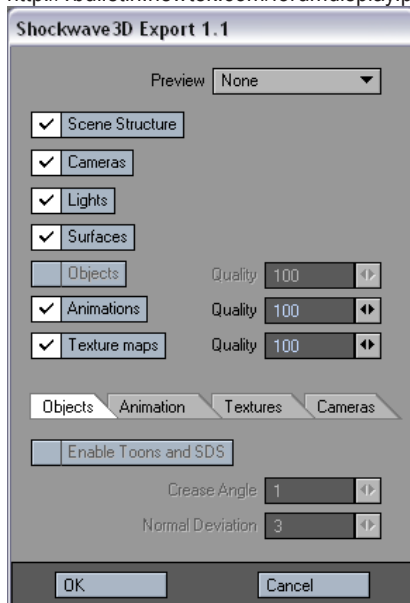
The Shockwave3D Exporter (**File> Export> Shockwave3D**) allows you to export Layout's current scene as a Adobe Shockwave® file (.w3d). This encapsulated file contains all of the information needed to recreate objects, surfaces, and animation, as well as the image files used in the scene. The exported file can then be integrated into Adobe Director® as a Cast Member. With this exporter, you can combine the content creation toolset of LightWave with the interactive functionality of Adobe Director, creating an integrated solution for distributing multimedia content.



NOTE: Although the exported file can be previewed in LightWave, you must have Adobe Director in order to create files for distribution. More information on Shockwave and Director can be found at Adobe's website: <http://www.adobe.com/products/director/>

Also, be sure to check out NewTek's Shockwave3D discussion in the NewTek discussion forum at

<http://vbulletin.newtek.com/forumdisplay.php?f=29>

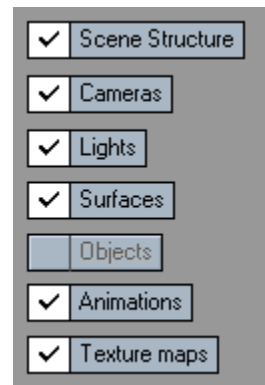


The top portion of the panel contains controls for the export selections, quality controls and preview options for the exported file. The options for these items can be found on the tabs located on the lower portion of the panel labelled **Objects**, **Animation**, **Textures**, and **Cameras**. The file location is determined when the **OK** button is pressed and the file dialog appears.

Some things should be considered when Modeling and animating a scene that will be exported with Shockwave3D. Because this media is meant for distributing large multimedia files, many of the exporter's features and options keep performance issues in mind. Please take a moment to read through this section and familiarise yourself on how LightWave and Shockwave3D are compatible with each other. It could save you a considerable amount of time on your next Shockwave3D project.

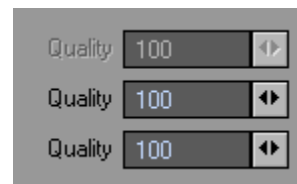
### Export Selection

This list of items determines which portion of the scene will be exported into the .w3d file. The **Scene Structure** option preserves the parenting hierarchy of your scene, while **Cameras**, **Lights**, **Surfaces**, **Objects**, **Animations**, and **Texture Maps** all control which elements of the scene will actually be exported. Simply activating these checkboxes will include these items in the file.



### Quality Controls

For three of the item export parameters, **Objects**, **Animations**, and **Texture Maps**, there are additional controls to adjust the quality of the assets used in the .w3d file. Since the .w3d file is encapsulated, these adjustments will have no effect on the items or settings within your LightWave scene. Only the data within the exported file will be affected. If you are concerned about download times, or performance issues, modify these parameters to create smaller or faster playing files.

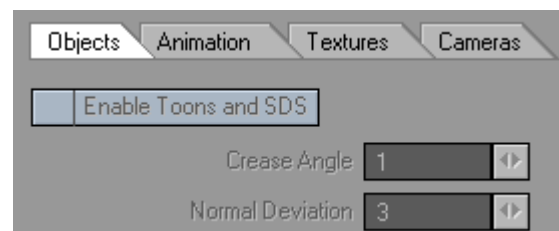


### Preview Options

On the Microsoft Windows platform, the Shockwave3D Exporter has the option of playing a Shockwave-enabled preview window. You can simply choose the resolution of the preview window, and on export, a separate window will open displaying the exported file. This representation of the exported scene is how it will appear as a Cast Member in Macromedia Director.

### Objects

The Shockwave3D Exporter exports polygonal models and frozen SubPatch objects from the scene into a .w3d file. To increase performance, Shockwave3D also creates multiple Level Of Detail (LoD) objects. These features are automatically enabled and their parameters can be controlled in Macromedia Director. Unfortunately, object morphing is not supported in Shockwave3D.





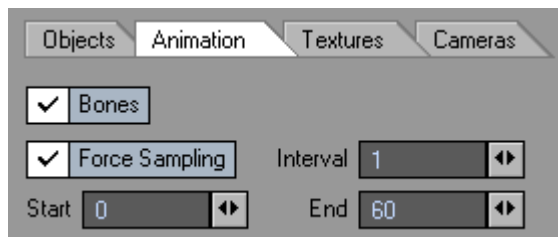
By activating the **Enable Toons and SDS** (cel shading and subdivision surfaces) feature, the objects are exported with the neighboring mesh information needed by these features in Director. Since this option increases the size of the exported file, deactivate **Enable Toons and SDS** if these features won't be utilised in Director. (Note that the SDS modifier must be added in Lingo.)

The **Crease Angle** determines how much of an angle is allowed between neighbouring normals before polygons are joined together. This parameter is used as a geometric smoothing angle when objects are being optimised. **Normal Deviation** sets a limit that normals can deviate from when the exporter creates any LoDs. The smaller the number, the less likely the reduced model's normals will be drastically different than the original object.

The Shockwave3D Exporter will remove very small triangles from your models and does not support 1 – or 2 – point polygons. Base vertices are points that will be removed last during any compression or polygon reduction during the export. To use the base vertices feature, create a Point Selection Set in Modeler named base verts. Any points included within this set will be given priority when the exporter does any of its optimisation functions, and will be reduced last.

## Animating Objects

All items (lights, cameras, objects, and surfaces) should have unique names. Also, only two of LightWave's scene items are capable of being animated with Shockwave3D, objects and bones. Thus, you cannot directly animate cameras or lights. However, you can animate a null object and parent the camera or light to the null object and achieve the same effect. Hierarchy controls exported from LightWave include both item parenting and pivot point manipulation.



You can select the range of the animation to be exported by defining the Start and End fields located on the Animation tab. By default, these initial values will match the start time and end time of the current scene. These options are not available if Force Sampling is not activated.

In LightWave, animation is created using keyframes. The exporter takes the motion defined by these keyframes and all the motion's modifiers from a single frame and creates what is called a Sample. This is much like freezing the motion curves of an item, but for that single frame.

These samples are then used in Shockwave as the keyframes of the animation. By adjusting the sample Interval, you control how often the exporter samples the animation. For example, a value of five will export a sample every fifth frame. This is very similar to LightWave's **Frame Step** parameter.

The smaller the Interval is, the more samples the exporter will create. In order to store this information, a larger file is created and will take longer to download and store in memory. Having the sample Interval set as high as possible will reduce the overhead needed to play the animation. However, having an incredibly high value won't solve all your problems either.

The Shockwave3D Exporter will rotate an item based on the fastest route from angle A to angle B. Obviously this is not how LightWave works. For this reason, it may be necessary to export a scene using

a much smaller sample Interval. This will give the Shockwave player more keyframes to rotate an item with and will match the LightWave scene more accurately.

## Animating Bones

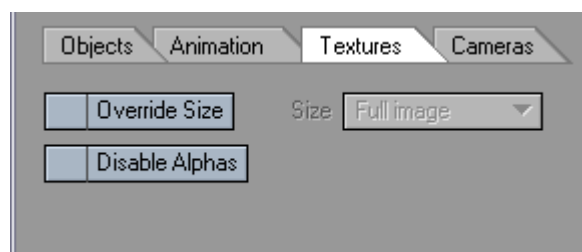
Animating bones is one of the more advanced features of the Shockwave3D Exporter. However, one thing to keep in mind is that Shockwave uses a different bones system than LightWave. The major difference between the two systems is how points get bone influences assigned to them. The animator should make all influence adjustments in weight maps as opposed to LightWave's bone setup options. For this reason, all bones must have a weight map assigned to them. Any point not assigned to a weight map is assigned to a null bone located at the root of the hierarchy. Also, unlike LightWave, a Bone's rest position is determined on frame 0.

## Surfacing and Texturing

Surfacing is converted from LightWave to Shockwave as closely as possible. This includes Color, Luminosity, Diffuse, Specularity, Glossiness, Reflection and Transparency. However, because the rendering methods (rendering vs. OpenGL) used in these two applications are setup differently, some tweaking may be needed to these settings in order to get the desired effect.

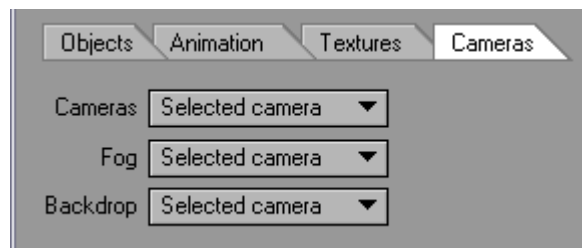
All surface mapping is converted to UV mapping when exported. However, Shockwave does not support the use of layered textures. Therefore, any surface channels that use layered textures must be flattened and turned into a single texture by using LightWave's surface baker. Texture mapping in the diffuse, glossiness, reflection, or specularity channels also works a little differently than LightWave. Any maps used in these channels are projected environmentally in Shockwave3D, thus ignoring any user-defined mapping.

Double-sided surfaces are not supported in Shockwave and must be created using actual double-sided geometry.



By overriding the texture size, you can force the Shockwave3D Exporter to globally reduce the image size of the textures in the .w3d file. For example, if you had 512 x 512 images, you could reduce them down to 64 x 64 by simply selecting Override Size and choosing 64. Disable Alphas simply disables the alpha channels on all the images.

## Cameras



The Camera pop-up menu lets you decide which camera should be exported. If All cameras is selected, the camera data for all the cameras will be embedded within the exported file. Switching between cameras is handled within Macromedia Director.



The Fog pop-up menu lets you choose which camera will have the fog settings attached to them. Fog settings exported include, Fog Type, Fog Color, and Fog Falloff.

The Backdrop drop-down menu lets you determine which camera will have the scene's backdrop color setting attached to them.



NOTE: Take note of the aforementioned animation limitation when animating Cameras.

## Lighting

Shockwave3D supports many of LightWave's internal light types, Distant, Spot, and Point, as well as ambient light. The lighting parameters supported by Shockwave3D include Light Color, Light Intensity, and Spot Angle.



NOTE: Take note of the aforementioned animation limitation when animating lights.

## Image List

This command will save a text file that contains a list of every image used in a scene and its location.

### Example:

Lightwave Image List

6769194: LW\_Box\_Front.tga E:\images\Endo\_Sliders\LW\_Box\_Front.tga

6769195: LW\_Box\_Back.tga E:\images\Endo\_Sliders\LW\_Box\_Back.tga

6769196: LW\_Box\_SideL.tga E:\images\Endo\_Sliders\LW\_Box\_SideL.tga

6769197: LW\_Box\_SideR.tga E:\images\Endo\_Sliders\LW\_Box\_SideR.tga

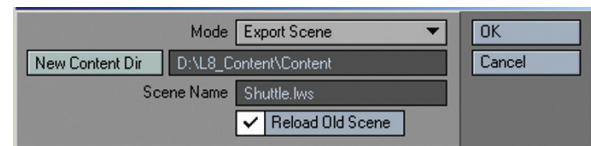
## Content Manager

Content Manager lets you collect a scene and its support files (i.e., object and images) and copy them to another location. It also has an option to just pull support files into the Content Directory. Note this tool has been moved to the Additional Plugins Menu in the Utility tab as Package Scene is a more functional replacement for it.

### Export Scene Mode

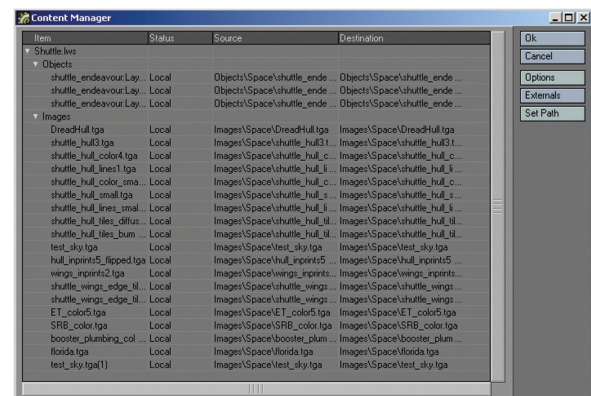
When the **Mode** pop-up menu is set to **Export Scene**, Content Manager will copy the scene and support files to the **New Content Dir.** You may edit the scene name in the **Scene Name** field if you want to change the name. You may reload the exported scene when the processing is finished by activating the **Load Exported Scene** option. Otherwise, the current scene will remain untouched.

### Export Scene mode



When you close the initial Options panel, the main interface will appear. The list window will show all of the objects and images in your scene. The Status column will show Local (the exported file will be within the content directory) or External (the exported file will be outside the content directory).

### Content Manager interface

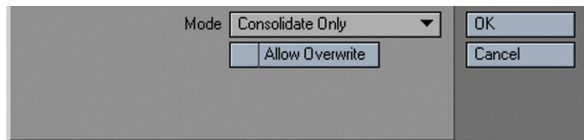


The Source column shows the location of the objects and images. If the files are local, only the path relative to the content directory (e.g., OBJECT\JOEY.LWO) is shown. If the files are external, the full path (e.g., C:\PICS\DC\PACEY.JPG) is shown.

The Destination column shows where those files will be copied. The destination and source are the same for local files. The destination will be blank for external files and needs to be specified by you. If desired, you can change the destination of local files.



## Consolidate Only Mode



The **Consolidate Only** mode moves files in the current scene into the current content directory. If you activate the **Allow Overwrite** option, Content Manager will automatically save the scene and objects.

### Consolidate Only mode

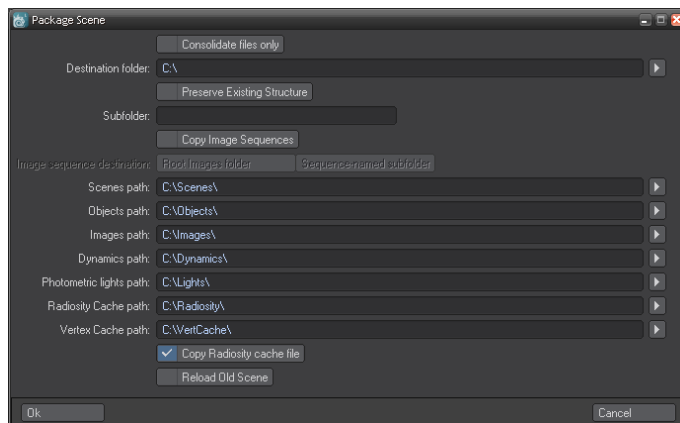
To change the destination:

- 1 Select the file(s) with your mouse. (Note: The **Externals** button will select all external files.)
- 2 Click the **Set Path** button.
- 3 Enter the path in the dialog or use the pop-up menu to select standard directory names.
- 4 The destination will be local and show a relative path.

Click the **OK** button to execute Content Manager based on your settings.

## Package Scene

Package Scene (File Menu) gathers information from your scene and locates the items into the specified directory.



The basics are that Package\_Scene is an LScript that uses only Lightwave/LScript commands to accomplish everything that Content Manager did, but also will work with .MDD, .BDD, and .PFX files, and works whether or not your content is actually in the current content structure. If a content element is currently loaded into your Lightwave scene, Package\_Scene should find it.

Select the destination folder and if you want a new level under that for each subdirectory, type the name in the Subdirectory box. The paths will be updated after tabbing or pressing the Enter key.

**Consolidate Files Only** will save all of the files in a single directory, specified with the Destination Folder.

After the interface in which you select the target directory for the export, the script:

1. Gets a list of all images and their paths.
2. Copies these images to the target directory.
3. Replaces all the images with their new location.
4. Changes the content directory to the target content directory.
5. Saves each object to the target location.
6. Saves the scene to a temp location.
7. Parses the scene file to find .MDD, .BDD, and .PFX references.
8. Finds and copies the dynamics files to their target location.
9. Rewrites the scene file, changing the respective dynamics paths, to the target location.
10. Loads the new scene file. This means at the end of operations, the user is now active in the newly exported and saved version of the scene, not the original scene from the original location.
11. If Reload Old Scene is checked, it will instead load the original scene.

## Quit

(default keyboard shortcut **Shift Q**)


Brings up a confirmation notice before the program actually quits in all cases, even if your scene and objects have just been saved.

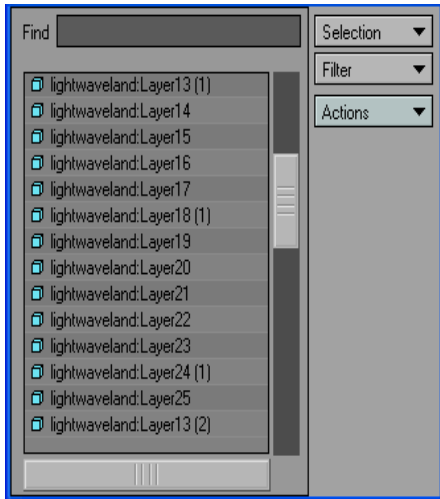




## Organizing Your Items: List Manager

Located directly to the right of Current Item pulldown menu, The List Manager allows for quick and easy management of the items in your scene.

 If you have a scene with a large number of items, you can use List Manager to organize your objects, cameras, lights, and bones into selection sets.



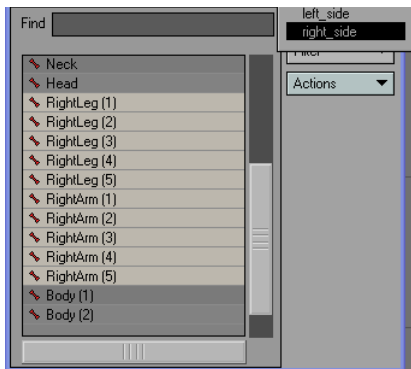
The default setting for List Manager will list all of the items of the Item Selection mode you have selected.

If you are looking for a particular item in a long string of items, you can use Find. The search is case-sensitive, so keep that in mind if an item is not showing up. After you type the name in, press enter and the first matching item will be selected. Pressing enter again will choose the next available item.

### Selection Menu

The Selection menu allows for the creation of selection sets for a group of objects. For example, you may have a group of bones where you want to change all of the setting of all the bones on the right side with the same settings.

You create selection set by first highlighting a group of items. You can select multiple items by holding down Shift or CTRL. Next, select Create Set in the Selection menu and give it a name. When you open the Selection menu again, you will see the named selection group.



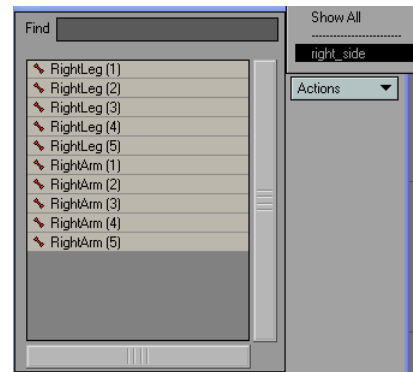
You can delete a selection set by choosing Remove Sets.



**Note:** If you have both the Bones and Objects List Manager panels open, when you select an object in the Object List Manager panel the corresponding set of bones for that object will show up in the Bones List Manager panel.

### Filter Menu

The Filter menu works like the Selection menu, except it takes the process further and will hide all other items not in the set.



You can delete a filter set by choosing Remove Sets.

If you would like to see all items after choosing a filter set, select Show All in the filter menu.

### Actions Menu

The Actions menu allows you to manage the list order and rename your items.

You can change the list order in ascending or descending order by Name, ID, or Selection, if you have created Selection groups.



**Hint:** If you want to manually change the order of items, simply left click on the item, or group of items, and move it up or down in the list. This will automatically update the list in List Manager and in the Current Item menu.





## Items Tab

### Load Scene

(default keyboard shortcut **Ctrl O**)

#### Scene

##### To load an existing scene:

**Step 1:** Make sure the Content Directory is set properly. (**Edit> General Options**)

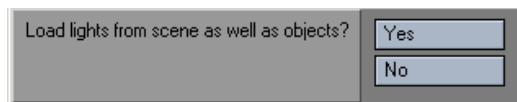
**Step 2:** Choose **Item > Load Scene** and use the file dialog to navigate to the desired scene file. When the scene is loading, a progress dialog will appear. You may abort the load operation by clicking the **Abort** button; however, this may result in a partially loaded scene.



NOTE: You can also Load a Scene file by choosing **File > Load > Load Scene**.

#### From Scene

Using Load From Scene (**Items> Load From Scene**) gives you the ability to load an existing scene into your current scene. You are given the option of loading the lights from a saved scene as well as objects.



This can be extremely useful when working with complex scenes that have many items. Having the ability to set up a character in one scene and load the entire setup into a different scene is just one example of how useful **Load From Scene** can be.



HINT: Build a collection of nice Lighting setups and quickly add them to your scene using **Load From Scene**.



NOTE: You can also Load Items From a Scene by choosing **File > Load > Load From Scene**.

#### Object (default keyboard shortcut +)

When you set up a new scene, you will need to load the appropriate objects into the scene. Use Load Object (**Items> Load Object**) to add objects to your scene.

##### To load an object file into a scene:

**Step 1:** Make sure your Content Directory is properly set. This will help with any images that are referenced in the object file. (**Edit> General Options**)

**Step 2:** Select **Items > Load Object** and use the file dialog to navigate to the desired object file. (Note: multiple objects may be loaded when more than one is selected in the file dialog.)

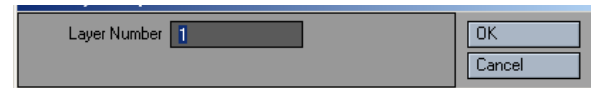


NOTE: You can also Load an Object into Layout by choosing **File > Load > Load Object**.

If the object contains multiple layers of geometry, each layer will be loaded as a separate editable item in Layout. An object is initially loaded so that its local Origin is at the global Origin and its local axes line up with the world axes.

#### Object Layer

Choose **Items > Load Object Layer** to load a specific layer from a multi-layer object. A dialog will appear allowing you to define which layer to load.



NOTE: You can also load an object's layers by choosing **File> Load> Load Object Layer**.



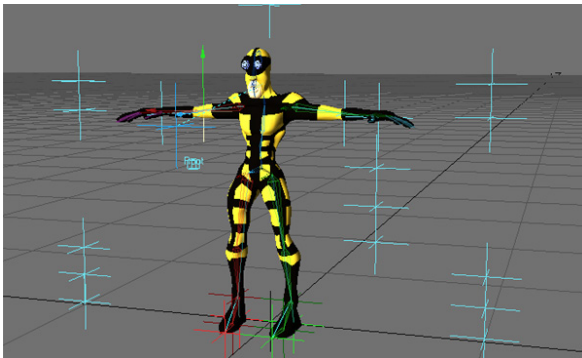
## Add

## Null

(default keyboard shortcut **CTRL N**)

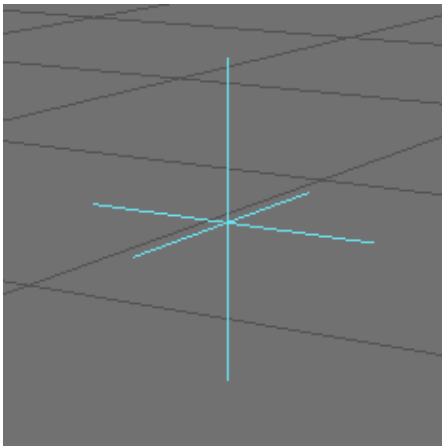
### The Null Object

Null objects are special objects that can be used for many different purposes in LightWave. Nulls can act as a target or parent item for lights or cameras, or perhaps a handle to help in manipulating a group of objects. They are often used with object/bone hierarchies to group major components and with Inverse Kinematics to act as goals for items to reach for.

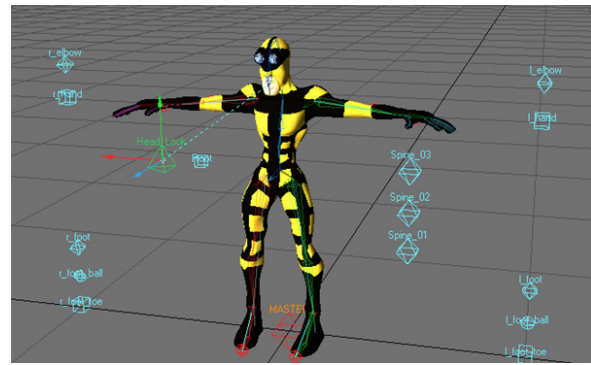


Example of a character using Null objects for Inverse Kinematics and Proxy Items.

Null objects appear in the Layout window as six-pointed stars or *jacks*; however, they never render in an image. Null objects, like the camera and lights, will appear to grow or shrink in relation to objects as the grid size is changed. They can be moved, rotated, scaled, and so on, just like any regular object.



**NOTE:** You can change the appearance of a Null object by applying "Item Shape" in the Add Custom Object list, within the Object Properties panel.



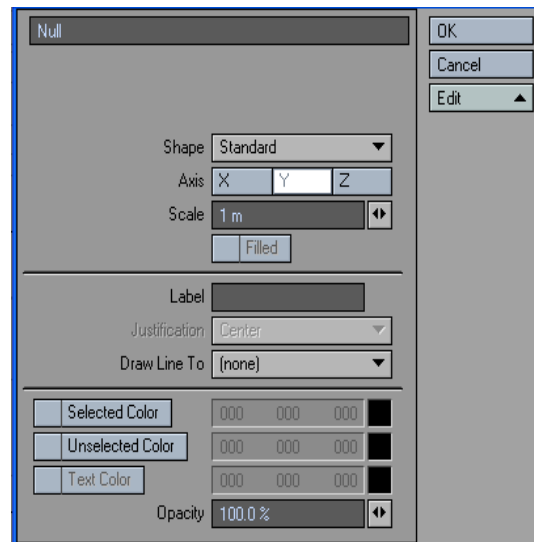
Null Objects with Item Shape Applied

To add a null object:

Step 1: Select **Items > Add Null**.

Step 2: Enter the name you wish to use for the null object in the dialog that appears or accept the default name of "Null."

Step 3: If you would like to change the properties of the null click on **Edit** and the menu will expand.



**NOTE:** When there are multiple items of the same type with the same name, a numerical suffix is added automatically (e.g., Null (1), Null (2), etc.).

**Shape:** Determines the shape of the null. Standard is the default Star null.

**Axis:** Determines the direction the null faces towards the positive axis.

**Filled:** If this option is available and you select it, the item shape will be a solid color.

**Label:** You can enter text which will be displayed in the viewports.

**Justification:** If you enter text in the Label section, you can specify where on the null where to display that text.

**Draw Line To:** Will draw a line from the null to the selected item.



**Selected Color:** When the null is selected, this will be the color of the null.

**Unselected Color:** When the null is not selected, this will be the color of the null.

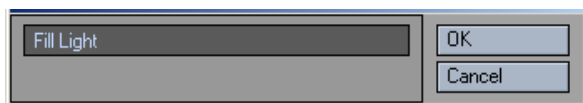
**Text Color:** If you have entered text, this will be the color of the text.

**Opacity:** Determines the opacity of the null.

Null objects are created by LightWave on a scene-by-scene basis and are not real objects that can be saved to disk. They remain part of the scene file, however, and will reload with the scene.

## Lights

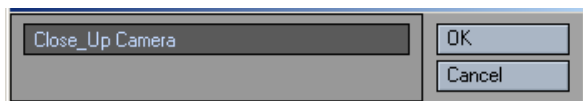
Choose **Items > Add: Lights** and select the desired light type from the submenu. This is where you would convert Luxigons as well. A dialog prompts you for a name when you add lights. Click OK to accept the default or type in the name you wish to use and click OK.



NOTE: For more information on Lights and Light Types see Chapter 17 - Light Properties, starting on page 536.

## Camera

You can add additional cameras by choosing **Items > Add: Camera**; however, only the **Current Camera** is used to render a scene. If you have more than one camera in the scene, use the **Current Camera** pop-up menu on the Camera **Properties Panel** (or Layout's **Current Item** pop-up menu) to select the currently active camera. A dialog prompts you for a name when you add a Camera. Click OK to accept the default or type in the name you wish to use and click OK.



NOTE: For more information on Cameras see Chapter 16 - Camera Basics, starting on page 520.

## Dynamic Object

One way to quickly add dynamic objects to your scene is to use the **Add Dynamic Object** drop down list (**Items > Add Dynamic Obj**). Choose between **Collision**, **Wind**, **Gravity**, and **Particle** from the drop down menu. You can add one or more of these controllers to your scene. Then, each can be set with its own **Parameter** settings to create just the effect you are looking for. A dialog prompts you for a name when you add a dynamic object. Click OK to accept the default or type in the name you wish to use and click OK.

**Collision** — The Collision controller lets you add an element for the particles and/or dynamic objects to bump into.

**Wind** — The Wind controller lets you add wind to blow your particles and/or dynamic objects around.

**Gravity** — Add a Gravity controller to add gravity-like effects to your particles and/or dynamic objects motions.

**Particle** — The Particle Emitter controller is the controller and the source for particles. There are two emitter selections: **HVEmitter** and **PolygonEmitter**.



469.

NOTE: For more information on Collision, Wind, Gravity, and Particles see the Dynamics Tab section in Chapter 16 - Object Properties, starting on page

## Convert Powergons

Powergons allow you to execute a short Layout command script, which is attached to selected polygons in Modeler. You can use this feature to quickly add, say, lights using polygons for positioning. Moreover, all of the lights properties can be defined as well.

### Executing the Commands

Powergon commands that are attached in Modeler can be loaded into Layout. Then choose **Items > Add: CVT Powergons**. This executes any command scripts attached to the selected object.



NOTE: For more information about Powergons see the Add Powergons section in Chapter 33 - Setup Tab.



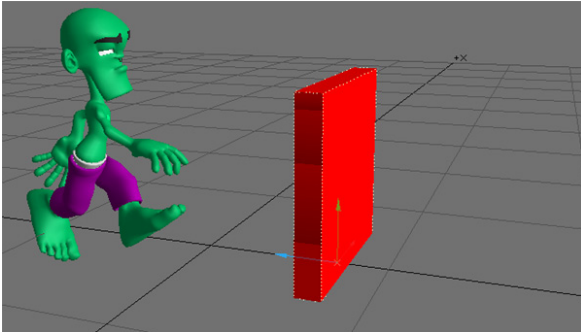
## Clone

(default keyboard shortcut **Ctrl C**)

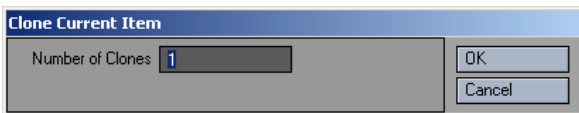
If you want to duplicate an item that is already in your scene, you can just choose **Items > Add: Clone**. The new item will inherit all of the source items properties and motion. As such, this can be a real time saver.

### Steps to cloning an item:

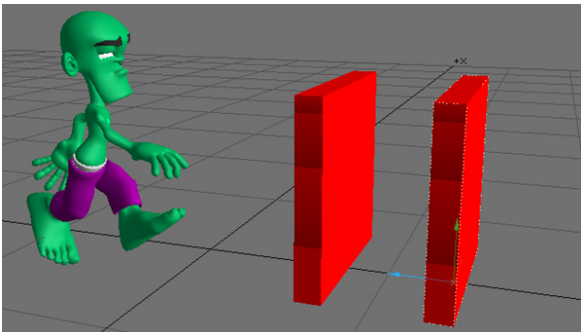
**Step 1:** Select the item you would like to clone.



**Step 2:** Choose Clone (**Items> Clone**) and input the number of clones you would like to create. Click OK.



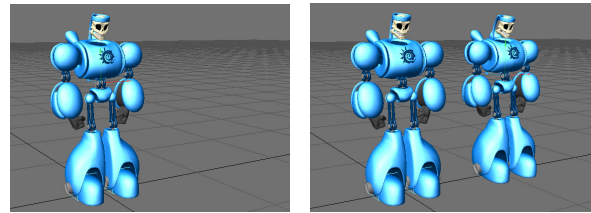
**Step 3:** You should now have clones of your original object.



NOTE: The newly created Clone(s) will be located at the exact coordinates as the original so you will need to move them.

## Clone Hierarchy

If you want to duplicate an item and all of its children in a hierarchy that is already in your scene, you can just choose **Items > Add: Clone Hierarchy**. The new items will inherit all of the source items properties and motion. As such, this can be a real time saver.



NOTE: The newly created Clone(s) will be located at the exact coordinates as the originals so you will need to move them.

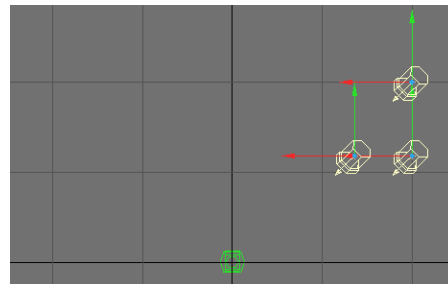
## Mirror

(default keyboard shortcut **Shift V**)

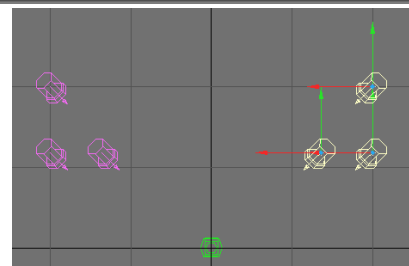
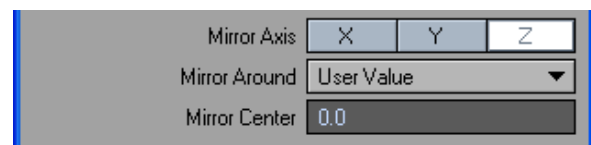
If you want to duplicate an item or group of items that are already in your scene, you can choose **Items > Add: Mirror**. This tool works in a very similar way to clone with one difference. Instead of duplicating the selected item(s) on top of themselves, Mirror gives you the option to mirror the selected items along the X, Y, or Z axis.

### Steps for using Mirror in Layout:

**Step 1:** Select the item or items you would like to duplicate.



**Step 2:** Choose Mirror (**Items> Mirror**) and a dialog box will appear. Choose the Axis to Mirror on.



**Step 3:** Click OK.



NOTE: Your original items will remain selected.



## Replace

---

### Rename

You can rename the current item, like cameras, bones, and lights, but not objects, using **Items > Replace: Rename**.

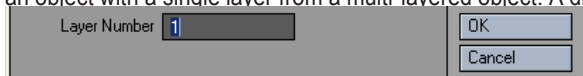
### Replace

The commands found in the **Items > Replace: Replace** drop down menu let you replace the current object with another object, a layer from another object or a null object. You might use this feature to animate a very complex object by inserting a stand-in object with a low-polygon count and then replacing it when it is time to render.

### Replacing a Multi-layer Object

When replacing a multi-layer object, select the *first* layer before replacing. The first layer will be replaced with the first layer from the new object file. Subsequent layers — belonging to the same original object — will be replaced if their layer numbers match those found in the new object file.

You can also use **Items > Replace > Replace > With Layer** to replace an object with a single layer from a multi-layered object. A dialog will



## Delete

---

### Clear Selected

*(default keyboard shortcut -)*

**Clear Selected (Items> Delete: Clear Selected)** will remove the selected item or items from a scene.

### Clear

You can use commands on the **Items > Delete: Clear** menu to remove selected items or all of the items of a specified type from a scene.

### Clear Drop Down:

Clear All Objects — Will remove all objects from the scene.

Clear All Bones — Will remove all bones from the selected object.

Clear All Lights — Will remove all but one of the Lights in the scene.

Clear All Cameras — Will remove all but one of the Cameras in the scene.

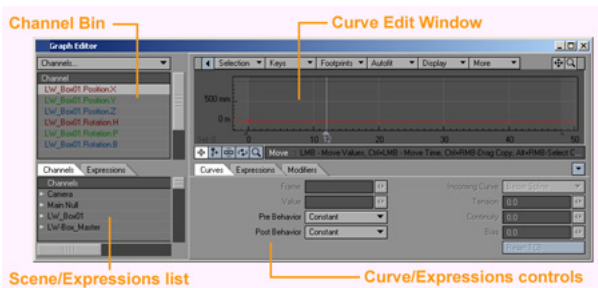


## Graph Editor

When you create keyframes for items, you specify a set of keys for the item's animation channels (generally position, rotation, and scale, but also light intensity, etc.). The **Graph Editor** provides both a more global and more detailed way to alter the settings that govern an item's animation channels. You have all of Layout's keyframe editing capabilities, plus some others like dragging keyframes to different frames, or graphically adjusting keyframe attributes. Use it to visually fine-tune or dramatically change an item's animation characteristics.



**NOTE:** The **Graph Editor** also controls all envelopes for options like light intensity, color, camera zoom, etc.

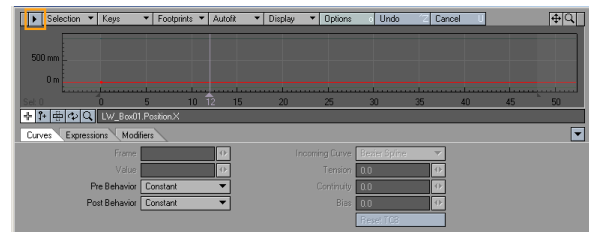
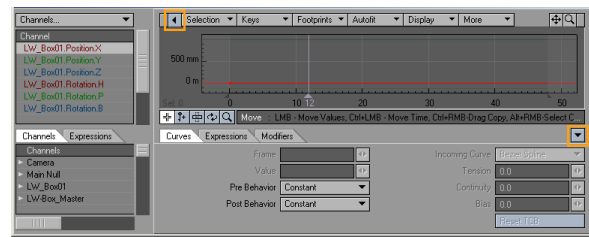


Each **Animation Channel** is displayed on a two-dimensional graph. Time is constant along the bottom and the **Channel** value or setting is equal to the vertical position. Because time is constant, you can visually judge things like where an item slows down or speeds up based on the slope of the curve. The **Graph Editor** is a great way to identify and fix those annoying hiccups, which can occur from time to time, in what should be a smooth animation.

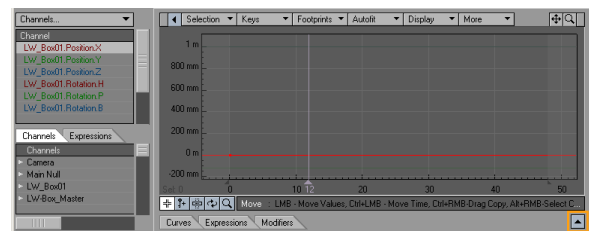
The **Graph Editor** has four major interface areas: the Channel (Curve) bin, the Curve Edit Window, the Curve/Expressions controls, and the Scene/Expressions list.

You can click the **Collapse** buttons to hide the left side and bottom areas of the panel. This will increase the screen real estate used by the Curve Edit window. When the left side is collapsed, the primary selected curve is shown in the information field.

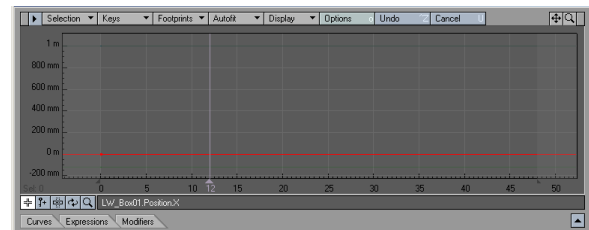
## Collapse buttons



*Collapse Trees Active*



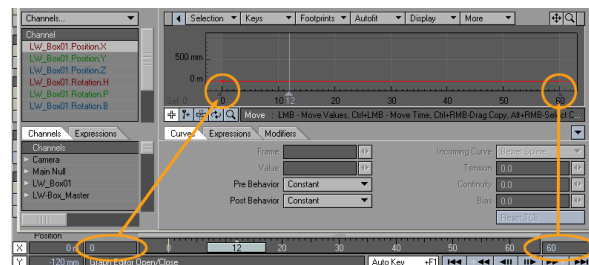
*Collapse Tabs Active*



*Collapse Tabs and Trees Active*

## Frame Range

If frames *outside* the range of frames in your scene (i.e., less than your first frame or greater than your last frame) are visible in the Curve Edit Window, those areas will be slightly darkened. There will also be a small handle at the very bottom of each border. You can drag these to interactively adjust your scene's first and last frame.



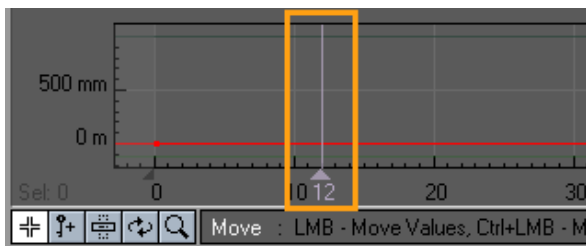
*First and Last Frame Handles and Corresponding Fields on the Main Layout Interface*





## The Time Slider

You can grab the time slider in the Curve Edit Window by its base and drag to change Layout's current frame. The current frame is displayed below the slider handle.



Clicking at the bottom of the graph — where the frame slider handle would be — moves the frame slider to that frame.

## Panel Layout Adjustments

You can drag the border between the Scene list and Channel Bin, and between the Channel Bin and Curve Edit Window to change the spacing between the two sides (if the **Graph Editor** is not at its minimum overall size).

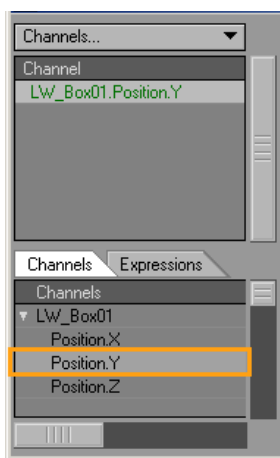
## Using the Channel Bin

When you first open the **Graph Editor**, the appropriate curves for the currently selected item are displayed in the Channel Bin. The Channel Bin is merely a repository to show you which curves are available for display and edit. Deleting curves from this window will not affect your scene. This system makes it very simple to edit multiple curves from several different items.

### To replace/add channels to the Channel Bin:

**Step 1:** Locate the desired item in the Scene list.

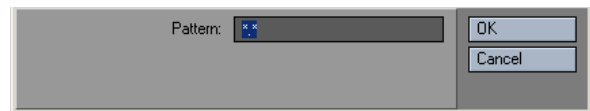
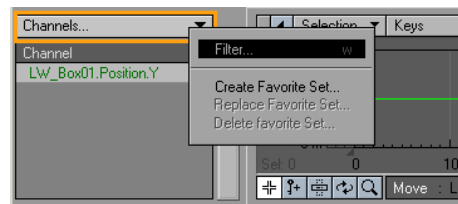
**Step 2:** Double-click on an individual channel or an item name (for all its channels). These will replace any existing channels in the bin. To add the channels to the Channel Bin hold the Shift key as you click or drag it into the bin. If you use the drag method, you can drag multiple channels/items into the bin by selecting multiple items in the Scene list (Shift key for range and Ctrl for non-contiguous selection).



Once you add channels to the bin, you can rearrange the order by dragging and dropping.

## The Channels Pop-up

The **Channels** pop-up menu, above the Channel Bin, has a few functions. You may select **Filter** to filter out channels from the Channel Bin that do not match a pattern. The pattern is case sensitive. \*.Position.\* would remove any channel that wasn't a Position channel. You could use \*.Y to show only Y channels.



Choose **Create Favorite Set** to manage curve sets. It lets you create sets of editable curves so that you can easily switch between different combinations of curves to manipulate during a session.

### To create a favorites entry:

**Step 1:** Make sure the desired curves appear in the Channel Bin.

**Step 2:** Select **Create Set** from the **Channels** pop-up menu.

**Step 3:** Enter an appropriate **Name** in the input field that appears and click **OK**.

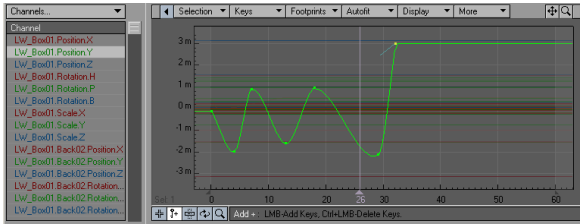
You can revert to any saved favorite set of curves by selecting it from the bottom of the **Channels** pop-up menu.

Choosing **Replace favorite set** lets you replace an existing favorite set with the curves currently in the bin. Choosing **Delete favorite set** lets you remove an existing favorite set from the **Channels** pop-up menu.



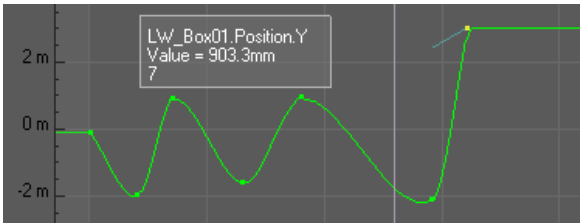
## Editing Curves

With the **Graph Editor**, you can easily edit multiple curves simultaneously or use curves from different items as references. Since you can mix curves of different types in the bin, you can do *interesting* things like compare the curve of light intensity with the X position of an object — any curve in LightWave can be compared or edited together!



You can interactively cut and paste key frames from one curve to another as well as simply replace an entire curve. This is also a good way to lock areas of curves together. By selecting multiple curves when you create keys the curves can be identical at those segments of the animation.

If you put your cursor over a key you will see a data label pop up to inform you of that point's Curve, Value, and Frame.



A Key's Data Label

## Edit Mode Selection

You can select from the edit modes by clicking one of the buttons beneath the **Curve Edit** window. From left to right, the mode buttons are **Move**, **Add**, **Stretch**, **Roll**, and **Zoom**. Pressing your **Spacebar** cycles you through the modes.

The information display, just to the right of the buttons, will display the name of the selected mode button, the shortcut key, and mouse operation information.



## To select curves for editing:

Click on the channels in the Channel Bin to select their curves for editing. (Hold the Shift key for range and Ctrl for non-contiguous selection.) Selected curves will become highlighted, but unselected curves will still be visible as a reference.

You can also **Alt+RMB** on a curve in the Curve Edit window to make it the (only) selected curve. Use **Alt+Shift+RMB** to add the curve to the current curve selection.

Your keyboard up and down arrow keys will cycle the selected key through the curves in the curve bin. Hold the **Shift** key as you press the cursor keys to increase or decrease the curves selected.

When multiple curves are selected, one will be the "primary" selected curve and its selection color will be a little brighter. Some commands, like Fit Values by Type, use the primary selected curve when more than one is selected.

## To select/unselect keys for editing:

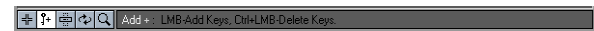
In the Move mode, select keys by clicking on them with your **LMB**. In any mode, except Roll, you can drag out a bounding box with your **RMB**. This will toggle the state of the bounded group of keys. To add to or subtract from a selection, hold the **Shift** key. To unselect all selected keys, just click in the graph border. To select all, **Shift+double-click**.

If only a single key is selected, you can use your keyboard left and right arrowkeys to cycle the selection of keys along curves.

## To add keys to curves:

**Step 1:** Select the channel curve(s).

**Step 2:** Click the **Add Mode** button.



**Step 3:** Click on the graph at the desired frame (horizontal) and value (vertical) position. Before you release the mouse button, you can drag up and down to adjust the value. Hold the **Ctrl** key to adjust the frame.

## To delete keys from curves:

**Step 1:** Select the channel curve(s) and then the key(s).

**Step 2:** Press the **Delete** key.

In the Add mode, you can delete keys with **Ctrl + LMB**.

## To change key value/time:

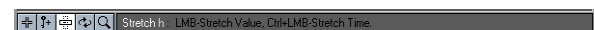
**Step 1:** Select the key(s) and click the **Move Mode** button.



**Step 2:** Drag with your **LMB** to change the value. Hold the **Ctrl** key to change time.

## To scale selected keys:

**Step 1:** Select the key(s) and click the **Stretch Mode** button.

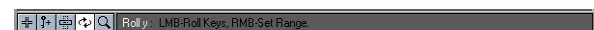


**Step 2:** Place your mouse pointer at the position you want to use as the center of the scaling.

**Step 3:** Drag using your **LMB** to scale value. Hold the **Ctrl** key to scale time.

## To roll a range of keys:

**Step 1:** Select the Roll mode.



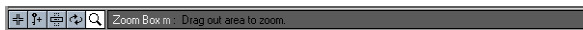
**Step 2:** Drag out the time range using your **RMB**.

**Step 3:** Drag your **LMB** to roll the keys—as keys go past the end of the range, they reappear at the other end of the range.

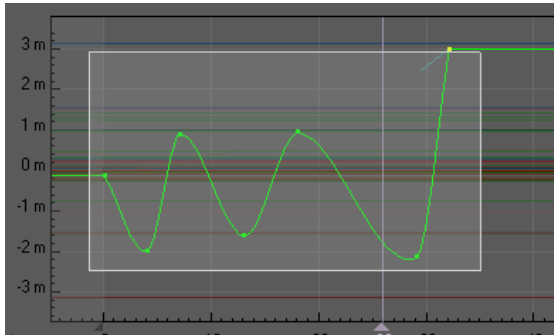


## To Zoom in and Out of the Graph Workspace:

**Step 1:** Select the **Zoom** tool



**Step 2:** Drag your Left Mouse button and drag a bounding box in the area you want to magnify.



**Step 3:** Right Click to Zoom out.

## Copying Keys

You may copy keys using your mouse or keyboard.

### To copy selected keys with a mouse:

**Step 1:** Select the keys for editing. Any mode except Roll can be used. Drag using **Ctrl+RMB**.

**Step 2a:** When you copy a single curve, an insertion point marker (and paste time display) will appear as your pointer moves over the curve. Release the mouse button to insert. Move your pointer off the curve to abort the paste.

**Step 2b:** When you copy multiple curves, color-coded insertion marks will appear. With your mouse pointer not directly over a curve, release to paste. Selected keys may be on different curves, but they can be copied only onto the same curve.



**NOTE:** Existing keys will be shifted over if you paste in more than one key (unless the Insert Overwrites Keys option is enabled).

See also the Copy Time Slice command (**Keys > Copy Time Slice**), discussed later.

## Weight Curve

Motions can now be assigned a Weight Curve which allow you to specify the influence a Motion has over time. These curves have the added benefit of allowing you to weight to any keyframed animation underlying the Motion in Layout. The weighting system works on a value of 1.0 indicating full influence and a weight value of 0.0 indicating no influence on the final animation. The difference with MotionMixer's weights is that as the weight approaches 0.0, and if there's underlying animation in Layout, then more of that underlying animation will be mixed into the final value:

Weight Value	Motion Influence	Layout Influence
1.0,	100%,	0%
0.75,	75%,	25%
0.5,	50%,	50%
0.25,	25%,	75%
0.0,	0%,	100%

If a Motion has no Weight Curve attached then it's treated as if it has a weight of 1.0. Weights can also be increased above 1.0. This is very useful for overlaying other Motions, the higher the weight value you apply the more influence that Motion will have over any others at that time (i.e., if you give a Motion a weight of 10.0, it will have ten times more influence over another Motion with a value of 1.0). When a Weight Curve has been applied, a small green bar is drawn under the Motion on the Timeline. Weight Curves can be added, disabled temporarily and opened in the **Graph Editor** either through the **Motion Properties Panel** or directly from the Timeline's context menus (see below).

## Curve Translation

Curves for Transitions, TimeWarps & Weights are now moved and scaled to match their positions on the Timeline. This makes working with the curves far more intuitive; the frame indicator in the **Graph Editor** now shows the position on the curve at the current time.

## Motion Instancing

Motion Instancing is now implemented. Any number of instances of a Motion can be placed on a Track and each can have their own independent attributes (i.e., Item/Channel states, in/out points etc.)

To create an instance of a Motion, select the source Motion from the Motion Menu (this can also be a previously created instance) and click Add Motion.

Freeing the "source" Motion also removes all instances.

## TimeWarp Curves

These can be attached to any Motion (or instance) and allow you to vary the timing of the animation, you can speed up or slow down sections, go backwards or freeze time and continue. A small red bar is drawn under the Motion to signify that a TimeWarp curve is attached. These can also be temporarily disabled.

Controls to add/remove/disable TimeWarp curves are in the **Motion Properties Panel** and on the context menus.



## Relative XZ Custom Offset Type

This is a hybrid of the normal Relative Offset and the Absolute Offset types. All channels are evaluated as relative offsets apart from the Y axis. This allows motions to be blended relatively but retain their Y heights (i.e., you don't end up with a follow-on Motion underground or floating above it).

## Character Custom Offset Type

As the name implies this offset type is specially designed for characters and makes it very simple to align Motions automatically. Say you have a Motion where a character turns a corner and ends up facing 90 degrees, you append a Motion of a walk in a straight line. MotionMixer can now rotate that Motion so that it continues in the direction that the last Motion ended on.

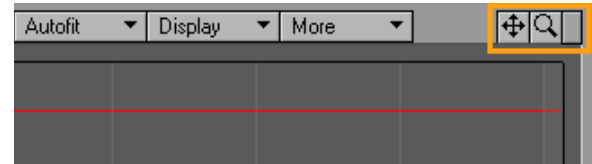
The **Adjust Alignment** control can be used to modify the calculated rotation value allowing you to fine tune the direction your character travels.

The **Compensate for Start Angle** checkbox allows you to compensate for a Motion that doesn't travel along the positive Z axis. This is very handy for motion capture where motions are often not axis aligned (i.e., in situations where the performer runs diagonally to make the best use of the capture area, or probably more commonly, where people don't create their forward motion along the positive Z axis).

To use the Character offset type set any items that control the characters movement (e.g., pelvis bone, IK Goals etc.) to CHAR in the Offset Editor (Custom Offset type).

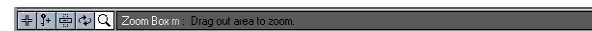
## Graph Editor: Adjusting the Curve Edit Window

Like other LightWave viewports, you can pan around the graph by holding the **Alt** key as you drag around the Curve Edit Window. Alternatively, you can use the drag-window drag button in the upper-right corner. You can change the zoom by holding **Ctrl+Alt** while you drag or by dragging on the zoom-window drag button in the upper-right corner.



## Zooming and Panning

The **Zoom Box** tool button is the right-most **Mode** button. Click it and you can drag out a box on the graph and zoom in on that area.



You can get a 2X zoom out by clicking your **RMB** with the **Zoom Box** tool selected. The zoom out will be centered at the point you click.

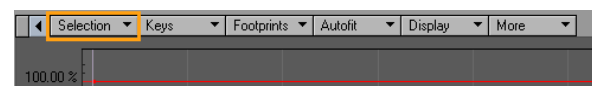
If you have a mouse wheel, scrolling it over the Curve Edit window will affect the zoom. Hold the **Ctrl** key to pan horizontally and use **Ctrl+Alt** to pan vertically.

## The Graph Editor Toolbar

The Toolbar contains a slew of commands for the **Graph Editor**. Many of these are also available on the various Curve Bin and Curve Edit Window pop-up menus, described later.

## Toolbar Selection Menu

The Selection menu contains commands that affect curves in the Curve Bin.



## Add Layout Selected (default keyboard shortcut **L**)

This command will *add* the channels for the currently selected item(s) in Layout to the Channel Bin.

## Get Layout Selected (default keyboard shortcut **Shift** **G**)

This command will replace the contents of the Channel Bin with the channels for the currently selected item(s) in Layout.

## Clear Unselected Channels (default keyboard shortcut **X**)

This command will remove any unselected curves from the Channel Bin.

## Clear Channel Bin (default keyboard shortcut **Shift** **X**)

This command will empty the Channel Bin.



## Remove Channel from Bin

(default keyboard shortcut **Shift D**)

This command will remove all selected curves from the Channel Bin.

## Invert Channel Section

(default keyboard shortcut **I**)

This command will invert the selection state of curves in the Channel Bin.

## Select All Curves in Bin

(default keyboard shortcut **Ctrl Up**)

This command selects all of the curves in the Channel Bin.

## Reset Bin Selection

(default keyboard shortcut **Ctrl Down**)

This command leaves only the first curve in the Channel Bin selected.

## Filter Curves

(default keyboard shortcut **W**)

Use this command to filter out channels from the Channel Bin that do not match a pattern. The pattern is case sensitive. **\*.Position.\*** would remove any channel that wasn't a Position channel. You could use **\*.Y** to show only Y channels.

## Filter Position Channels

(default keyboard shortcut **!**)

Use this to filter out everything but Position channels from the Channel Bin.

## Filter Rotation Channels

(default keyboard shortcut **@**)

Use this to filter out everything but Rotation channels from the Channel Bin.

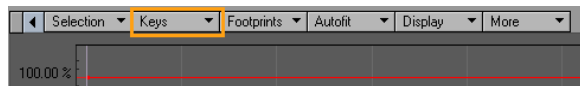
## Filter Scale Channels

(default keyboard shortcut **#**)

Use this to filter out everything but Scale channels from the Channel Bin.

## Toolbar Keys Menu

The **Keys** menu contains commands that manipulate the selection, creation, deletion, frame setting, and value of keys.



## Create Key

(default keyboard shortcut **Return**)

This allows you to create a key, Layout-style. A dialog will appear where you can enter the **Frame** and **Value**.

## Delete Selected Keys

(default keyboard shortcut **Delete**)

This command will delete any selected keys.

## Lock Selected Keys

(default keyboard shortcut **Shift L**)

This locks the selected keys so they are uneditable. Locked keys are grey.

## Unlock Selected Keys

(default keyboard shortcut **Shift K**)

Unlocks selected keys.

## Invert Selected Keys

(default keyboard shortcut **Shift I**)

This flips the order of selected keys in time.

## Snap Keys to Frames

(default keyboard shortcut **Q**)

This command causes every selected key that falls on a fractional frame to *snap* to the nearest whole frame.

## Set Key Values

(default keyboard shortcut **=**)

This command will bring up a dialog where you can enter a new **Value** for the selected keys.

## Bake Selected Curves

(default keyboard shortcut **B**)

This “bakes” the state of selected curves by creating keys at every frame. Curves do not necessarily need to be affected by a Modifier, but their effects will be taken into account.

## Copy Time Slice

(default keyboard shortcut **Ctrl C**)

You can copy values of selected curves at the current frame (even if there are no existing source keys) and paste them elsewhere. This command will copy the values. These values may be pasted at any frame with **Paste Time Slice**, but only onto the same curve(s).

## Copy Footprint Time Slice

(default keyboard shortcut **Ctrl X**)

This works like **Copy Time Slice**, but uses the value(s) from the curve's footprint instead of the actual curve. Use **Paste Time Slice** to paste.





## Paste Time Slice (default keyboard shortcut **Ctrl** V)

Pastes in values copied with **Copy Time Slice** or **Copy Footprint Time Slice** at the current frame. Keys will be created (or modified, if they already exist) at the new frame with the new values.

## Match Footprint Time Slice

(default keyboard shortcut **Ctrl** B)

This creates a key on the curve that matches the footprint value at the current time. Essentially, it is the same as doing a **Copy Footprint Time Slice** operation and immediate paste.

## Copy Selected Keys (default keyboard shortcut C)

This copies selected keys to a memory buffer. To paste the buffer, you must have your mouse pointer over a curve and use the **Paste Keys** command on the Curve Edit Window pop-up menu (**Ctrl+Shift+LMB**), discussed later.

## Add to Key Bin (default keyboard shortcut K)

This creates a named set of keys that you can insert into a curve later. To paste the set, you must have your mouse pointer over a curve and use the **Insert From Bin** item on the Curve Edit Window pop-up menu (**Ctrl+Shift+LMB**), discussed later.

## Numeric Move (default keyboard shortcut **Shift** T)

With this command you can shift the selected keys. **Frame Offset** is the number of frames to use for the shift. **Value Offset** is a number to add to or subtract from the value for each keyframe.

## Numeric Scale (default keyboard shortcut **Shift** H)

This command lets you scale the key times and values for selected keys. A **Time Scale Factor** of 1 means no change. A value of 2 would double the time and .5 would halve it. The **Time Scale Origin** is the center of the scaling. Thus, if you place this at one selected key, the scaling will happen around it and not affect that particular key. This setting uses the units displayed on the graph. **Value Scale Factor** and **Value Scale Origin** work similarly except they affect the selected key values.

## Roll Keys Left (default keyboard shortcut L)

Shifts the values of the selected keys to the left without affecting their time. This command works only with contiguously selected keys.

## Roll Keys Right (default keyboard shortcut J)

Shifts the values of the selected keys to the right without affecting their time. This command works only with contiguously selected keys.

## Reduce Keys (default keyboard shortcut -)

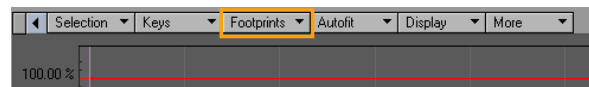
The Reduce functions let you remove consecutive keys that are within a certain threshold value of one another. The threshold is set by selecting **Set Key Reduction Threshold**.

There are two modes: **Reduce Keys** and **Reduce Keys (Recursive)**, whose default keyboard shortcut is **\_**. As an example, assume consecutive keys A B C D E all have values within the threshold. Choosing **Reduce Keys** would remove keys leaving A C E. Selecting **Reduce Keys** again would leave A E and a final **Reduce Keys** would leave only A. **Reduce Keys (Recursive)** would go directly from A B C D E to A in one step.

If you set the threshold to a negative number, **Reduce Keys** will eliminate every other key. **Reduce Keys (Recursive)** will remove all but the first key.

## Toolbar Footprints Menu

The Footprints feature lets you create an imprint of the current curve(s) to use as a reference and as a state you can revert back to.



## Leave Footprint (default keyboard shortcut **Shift** F)

Creates the imprint that will be visible as a shade of the real curve's color (once you make a change).

## Pickup Footprint (default keyboard shortcut **Shift** R)

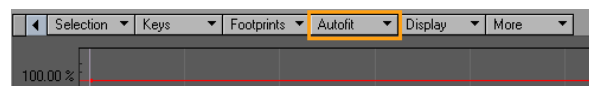
Erases the footprint for the current curve(s).

## Backtrack Footprint (default keyboard shortcut **Shift** B)

Will restore the curve to its footprinted state.

## Toolbar Autofit Menu

This group of commands affect the range of frames and values shown in the Curve Edit Window.



## Autofit (default keyboard shortcut A)

You can automatically scale the graph display to show all of the selected curves with this command.

## Autofit Selected (default keyboard shortcut **Shift** A)

You can automatically scale the graph display to show all of the selected keys with this command.

## Autofit By Type (default keyboard shortcut **Ctrl** A)

Autofit By Type will scale the graph to fit the values for the primary selected curve's type (i.e. position, rotation, scale, and so on).

## Fit Values By Type (default keyboard shortcut **Ctrl** F)

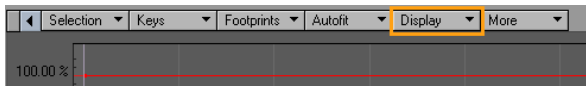
This works like Autofit By Type, but only scales the graph vertically, retaining the current frame range.





## Toolbar Display Menu

The Display menu contains commands that affect the graph display, as well as global options.



## Numeric Limits (default keyboard shortcut **Shift** **N**)

This will display a small dialog. The **Min Frame** and **Max Frame** values set the range of frames you want to see on the graph. The **Min Value** determines the value of the bottom boundary of the graph and the **Max Value** sets the upper boundary.

## Go To Frame (default keyboard shortcut **F**)

This action sets the current frame to the entered value. It also centers the Curve Edit Window around that frame.

## center Graph (default keyboard shortcut **G**)

Centers the graph on the current mouse position.

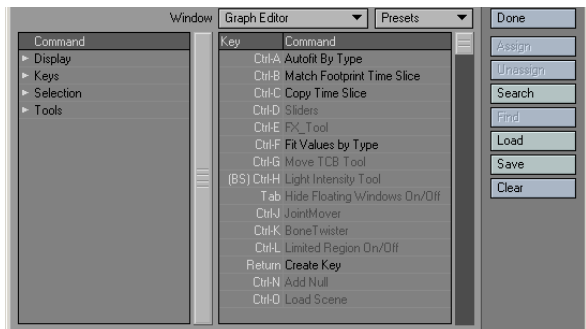
## Reset Graph (default keyboard shortcut **/**)

This simply resets the graph to default frame and value ranges.

## Edit Keyboard Shortcuts

(default keyboard shortcut **Alt** **F9**)

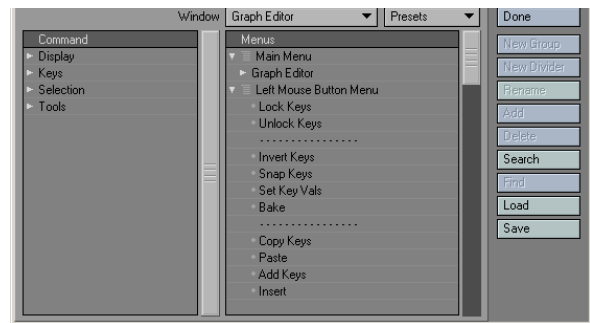
This command will bring up the standard **Configure Keys Panel**. However, it will list shortcuts for the **Graph Editor**. (Note that the **Window** pop-up menu will be set to **Graph Editor**.)



## Edit Menu Layout (default keyboard shortcut **Alt** **F10**)

The **Graph Editor** has its own set of menus. These can be customised using the normal **Configuring Menus Panel**. (Note that you are editing the **Graph Editor** menus when **Graph Editor** is selected in the **Window** pop-up menu on the **Configuring Menus Panel**.)

Beneath the Main Menu group is the **Graph Editor** group. This is the group used for the **Graph Editor**'s toolbar. It can contain its own group for pull-down style menus. In the Popup Menu group are the commands that will appear when you press **Ctrl+Shift+LMB** over the Curve Edit window.



## Insert Overwrites Keys (default keyboard shortcut **`**)

Normally, when you paste in more than one key, existing keys may be shifted over. Enable this option to replace the paste range and not shift over keys.

## Filter Static Envelopes (default keyboard shortcut **numpad 1**)

This option will keep envelopes that have fewer than two keys from being displayed in the Channel Bin when selecting groups (both from the scene list, as well as from other Layout panels.)

## Large Autosize Margins (default keyboard shortcut **numpad 2**)

This option will add an extra amount of outside space when you use autofit commands.

## Allow Fractional Keyframes (default keyboard shortcut **numpad 3**)

This option keeps frame adjustments to whole numbers. This is linked to Layout's **Allow Fractional Current Frame** setting (**General Options Tab** of the **Preferences Panel**).

## Lazy Layout Update (default keyboard shortcut **numpad 4**)

When active, the **Graph Editor** will not update in Layout until after you release the mouse button. This can allow smoother adjustments in complex situations. When inactive, Layout will update as you make adjustments.

## Track Layout Time (default keyboard shortcut **numpad 5**)

This option will scroll the Curve Edit window to keep the frame slider centered. This is useful for watching the curves scroll by as Layout is playing.

## Allow Passthrough Keys

(default keyboard shortcut **numpad 6**)

This option lets you drag keys *through* each other in time. Normally, you stop at a neighbouring key.



## Lock Motion Keys in Time

(default keyboard shortcut **numpad 7**)

This option causes keys to be created for all motion channels of the selected channel(s). Only Position, Rotation and Scale channels are affected. For example, Position.Y, Position.Z, Rotation.H, etc. would all be affected if Position.X was selected.)

## Move No Keys Sel

(default keyboard shortcut **numpad 8**)

By default, if no keys are selected, no editing will occur when using tools like Move or Stretch. Activate this option to change this so that all keys are considered selected when none are selected (like in Modeler).

## Track Item Selections

(default keyboard shortcut **numpad 9**)

This is a mode that will automatically bring the channels for the currently selected Layout item into the **Graph Editor**.

## Fit Values when Selected

(default keyboard shortcut **numpad 0**)

When this option is active, your view will automatically fit the values of the selected curve. This will not affect the visible range of frames, however.

## Show Modifiers

Activate this to always make the modified curve (i.e., after the effects of motion Modifier plugins) visible as a dotted line.

## Show Tangents

This command will activate or deactivate tangent handle display.

## Antialias Curves

This command will activate or deactivate the feature that smooths the display of curves.



HINT: If you are working with a lot of curves and keys, turning off the Antialias Curves and Show Tangents options can help speed up display refreshing.

## Show Key Info

This turns the pop-up display of key information off or on. This appears when your mouse pointer is directly over a key.

## Hide Background Curves

Normally, non-selected curves in the Curve Bin are visible in the graph. This display option will toggle their visibility state.

## Large Keyframe Points

Activate this option to increase the display size of keys.

## Custom Point Color

This turns on the user-defined (unselected) point color. Selected points are always yellow.

## Collapse/Show All

(default keyboard shortcut **F5**)

This collapses or shows the **Tabs** and **Trees** areas of the **Graph Editor**.

## Collapse/Show Tabs

(default keyboard shortcut **F6**)

This is the same as clicking the **Collapse Tabs** button.

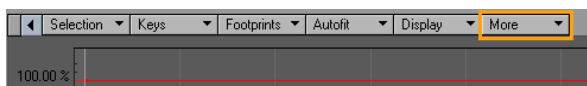
## Collapse/Show Trees

(default keyboard shortcut **F7**)

This is the same as clicking the **Collapse Trees** button.



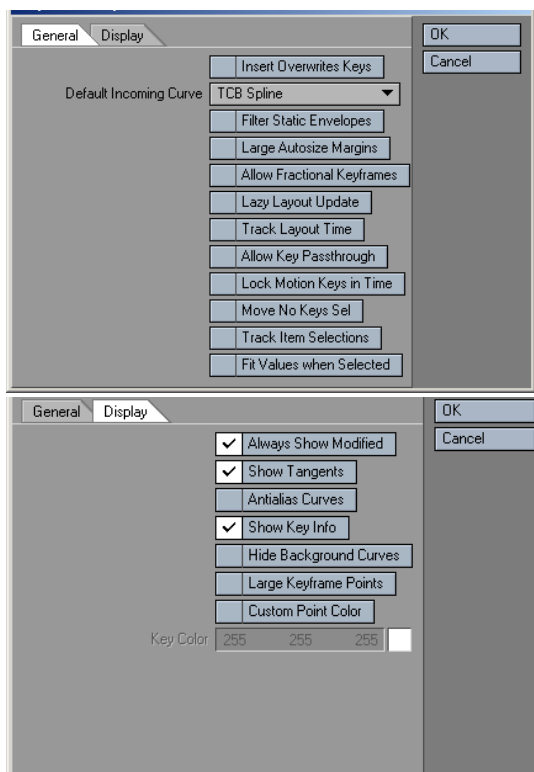
## Toolbar More Menu



## Graph Editor Options

(default keyboard shortcut **O**)

Most of the options on this panel can also be set from the Display menu and have been described previously. There are a few that can only be set here, however. On the **General Tab**, you can change the **Default Incoming Curve**. On the **Display Tab**, you can set the color used when the **Custom Point Color** option is active.



## Undo Last Action (default keyboard shortcut **Ctrl Z**)

Use this command for a single level undo/redo of the last edit.



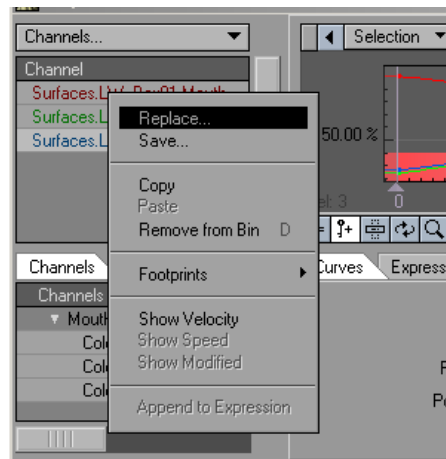
HINT: Also see the discussion on the Footprints feature, starting on page 189

## Cancel Changes (default keyboard shortcut **Shift U**)

This command restores all envelopes to their state at the time the **Graph Editor Panel** was last activated (i.e., making it the top window).

## Channel Bin Pop-up Menu

The Channel Bin has a pop-up menu that appears when you right-click a curve:



**Replace** — Used to load a curve from a file.

**Save** — Saves a single curve to a file.

**Copy** — Copies the curve under your mouse pointer when you open the menu to the curve memory buffer.

**Paste** — Pastes the curve stored in the memory buffer. Your mouse pointer must be over the curve you wish to replace before you open the menu.

**Remove from Bin** — Removes the selected curve(s) from the Channel Bin.

**Footprints** — See discussion on Footprints on page 189.

**Show Velocity** — Adds a non-editable background curve representing the *velocity* of the selected curve. Velocity is defined as the time rate of change for a single curve including a vector. In other words, a velocity curve defines how much the value of the current single channel changed at that time.

**Show Speed** — Adds a non-editable background curve representing the speed of the current curve. Speed is defined as the magnitude of the velocity vector. This means the Speed curve represents the time rate of change of all three position, rotation, or scale curves.

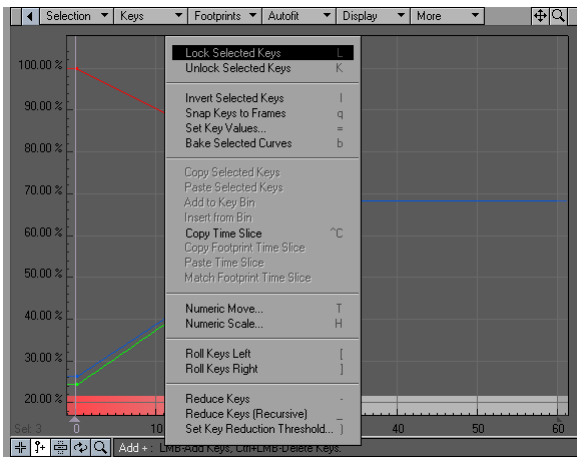
**Append to Expression** — This command will place the selected channel in the expression field. This is handy and will save you from having to type out channel names when creating expressions.

**Show Modified** — If the **Show Modifiers** option (Display menu) is active and your curve is being modified, from a modifier plugin for instance, this will let you see the actual modified motion curve.



## Curve Edit Window Pop-up Menu

Another pop-up menu is available when you work in the Curve Edit Window. It appears when you **Ctrl+Shift+LMB** click over the graph. This gives you quick access to many commonly used commands.



Most of the commands have been discussed; however, some only appear on this menu because they require your mouse pointer to be directly over a curve.

**Copy Selected Keys** — This copies selected keys to a memory buffer.

**Paste Keys** — Inserts the keys stored in a memory buffer with the **Copy Selected Keys** command— existing keys may be moved over if the buffer has multiple keys. Your mousepointer must be over the pasting point on a curve before opening the menu (pointer will highlight).

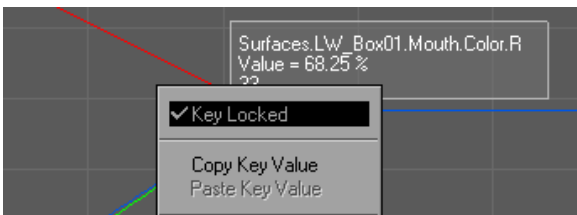
**Add to Key Bin** — Creates named set of keys that you can insert into a curve later.

**Insert** — Works like **Paste Keys**, but gets data from a specified key set.

**Options** — Displays the Graph Editor Options Panel.

## Key Pop-up Menu

Another contextual pop-up is available when you work in the Curve Edit Window. It appears when you right-click directly over a key: the operations affect that key only.



**Key Locked** — Locks/unlocks the key so it is uneditable; locked keys are grey.

**Copy Key Value** — Copies the key value to memory buffer.

**Paste Key Value** — Pastes the value stored in memory to the key.

**Ease In/Out** — Sets the **Incoming Curve** for the key to TCB Spline and sets the **Tension** to 1.

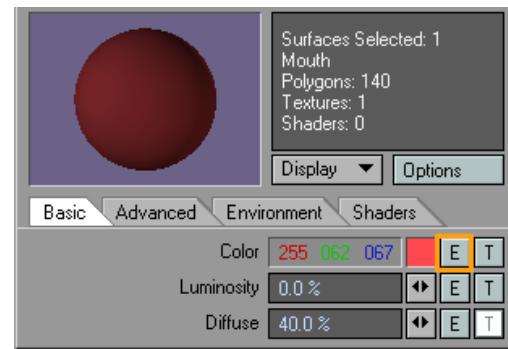
**Incoming Curves** — Quickly change the Incoming Curve type for the key by selecting it from the list at the bottom of the menu.



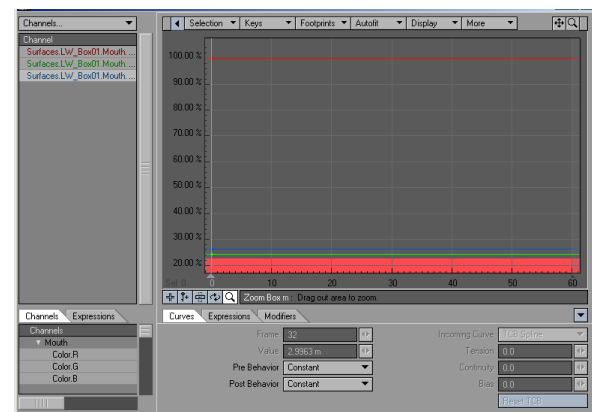
NOTE: Other customisable menus will appear when you **Ctrl+Shift+RMB** and **Ctrl+Shift+MMB**.

## Graph Editor: Editing Color Channels

You can also animate color channels. If you add an envelope for, say, a surface color, you can edit the related RGB channels on the **Graph Editor**.

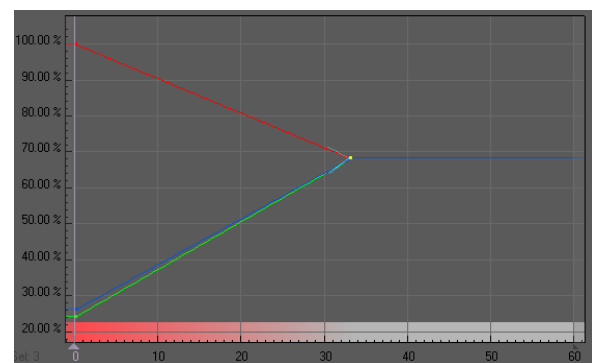


Clicking the Color Envelope Button



Color Envelope in the Graph Editor

When a color channel is selected in the Channel Bin, a color bar will appear at the bottom of the graph. The bar shows you the color for the **combined** red, green, and blue values at any particular point in time—even if you don't have all three color components in the bin.



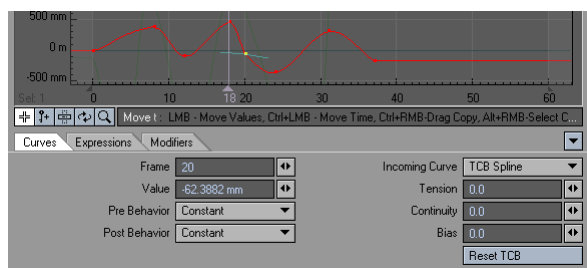
NOTE: You can adjust any color channel beyond the normal maximum and minimum — creating a high dynamic range color. This may have no visible effect on the color bar, but could affect how the color is interpreted by other factors.

You can use a color requester to set the key values by right-clicking on a key and selecting **Open Color Picker**. Note that the selected color will only set the color component for the selected channel(s). If other channels are selected, keys will be created as needed.



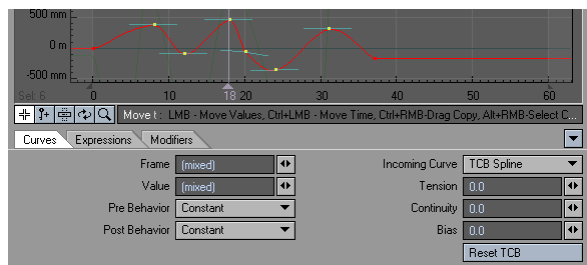
## Graph Editor: Curves Tab

The **Curves Tab** in the Curve Controls area contains specific values for a selected key(s). The **Frame** field contains the frame number and the **Value** field holds the related value.



### Multiple Values

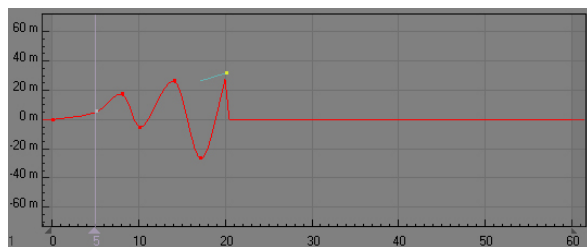
If you select multiple keys, input fields will show *(mixed)* if the keys have different settings. You can still edit the fields, which will change all selected keys to match the entered value. You can use this method to flatten a portion of a curve(s) or to match key times across multiple curves.



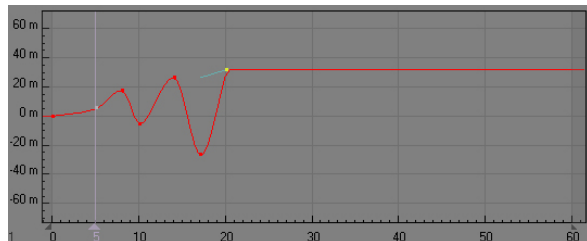
### Pre and Post Behaviours

The **Pre Behaviour** setting determines what happens before the first keyframe. The **Post Behaviour** determines what happens after the last keyframe. The available settings are:

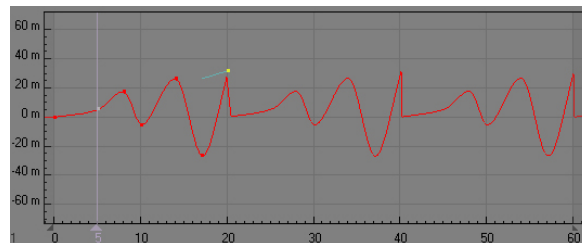
With **Reset**, the motion value is reset to zero.



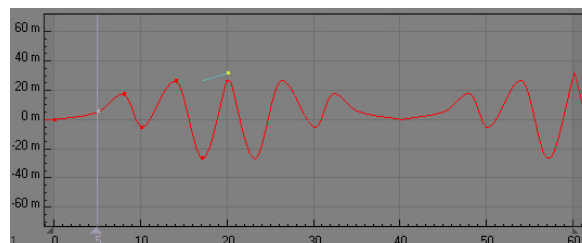
With **Constant**, the values beyond the ends are constant, that is, equal to the first or last keyframe value.



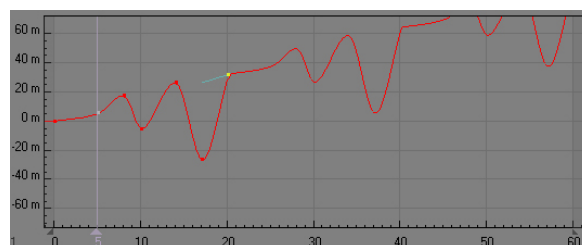
With **Repeat**, the motion repeats from the first to last keyframe.



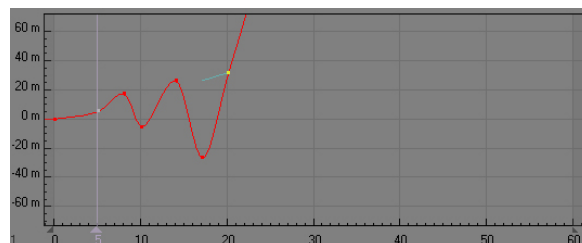
With **Oscillate**, the motion is mirrored over and over.



With **Offset Repeat**, the motion repeats, but it is offset by the difference between the first and last keyframe values.



With **Linear**, the curve receives a linear angle consistent with the angle at the start or end points.





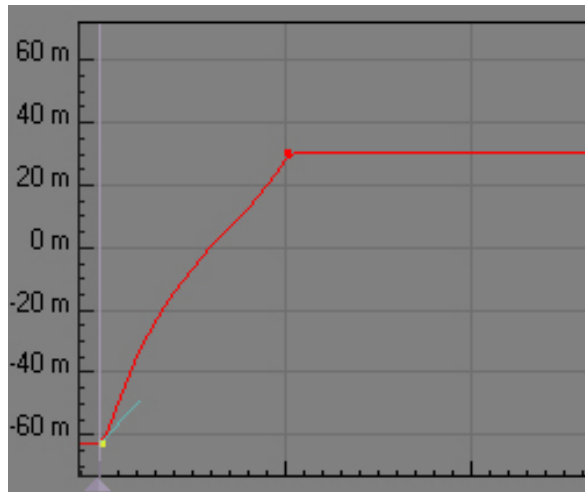
## Incoming Curves

The type of curve that precedes a key can be set using the **Incoming Curve** pop-up menu.

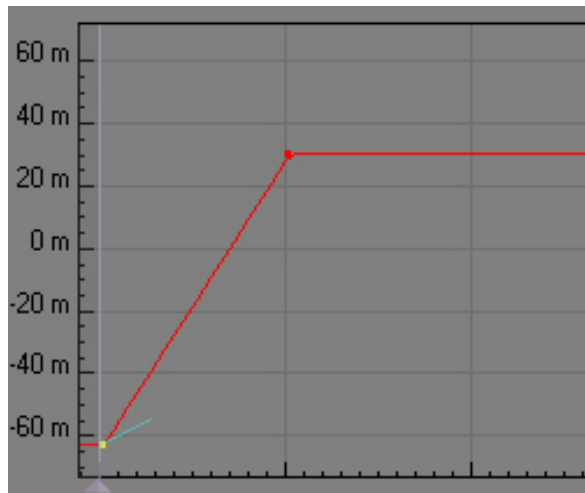
## TCB Spline

**TCB Spline** (tension, continuity, and bias) curves have three controls that determine the shape of a curve as it passes through a key.

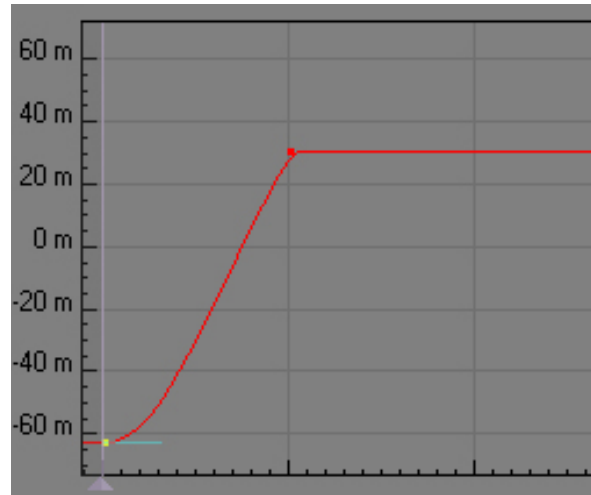
**Tension** causes an object in motion to slow down, or move a little bit less in each frame, as it nears the keyframe, and to accelerate as it passes the keyframe (-1 = low tension, 0 = normal tension, 1 = high tension). Without **Tension** (i.e., value of 0), the object would pass through the keyframe position at a constant speed. Positive values slow an item through a keyframe (*ease-in*) while negative values speed it up (*ease-out*).



Tension -1



Tension 0



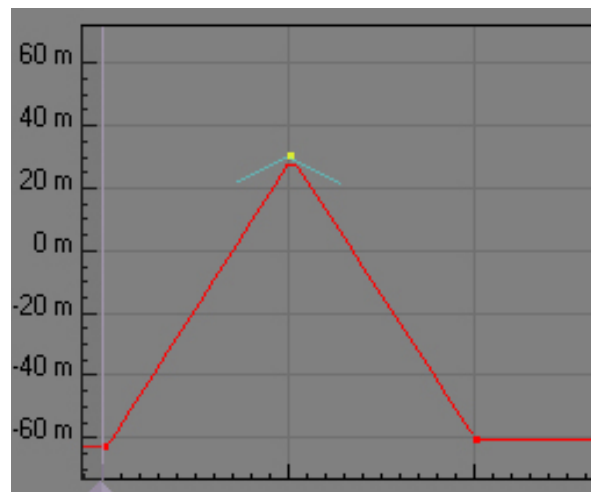
Tension 1

A high **Tension** value (1.0) is often used at the end of a flying logo move in order to make the logo come to a gradual stop. High **Tension** at the beginning of this move would make the logo start slowly, while a negative value would make the logo start quickly.



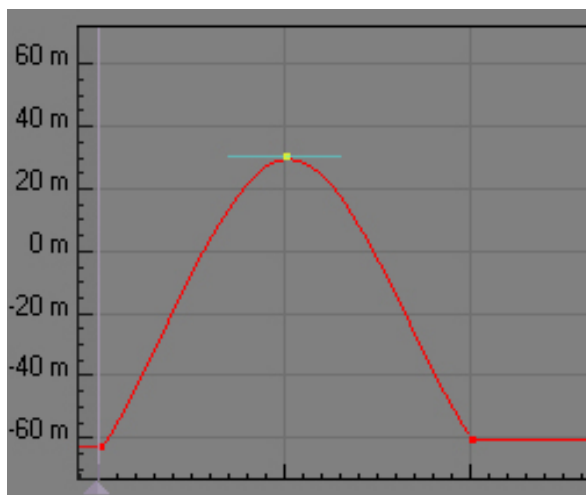
NOTE: If you right-click on a key, choosing *Ease In/Out* from the pop-up menu will set the Tension to 1 for all selected keys using TCB Spline.

**Continuity** accentuates a break or change in an object's graph (-1 = sharp, 0 = normal, 1 = smooth). Negative **Continuity** gives a sharper transition in the spline path at a keyframe, while positive **Continuity** gives a broader transition (sometimes *over-continuous*) through a keyframe. Negative **Continuity** is usually used to replicate a sharp change in motion such as that of a falling ball striking a floor and quickly reversing direction.

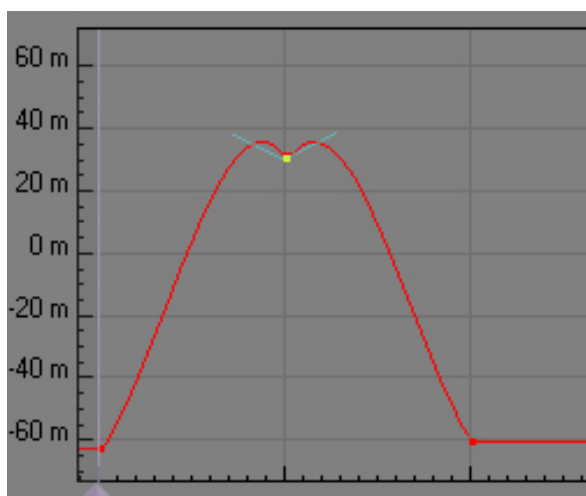


Continuity -1





Continuity 0

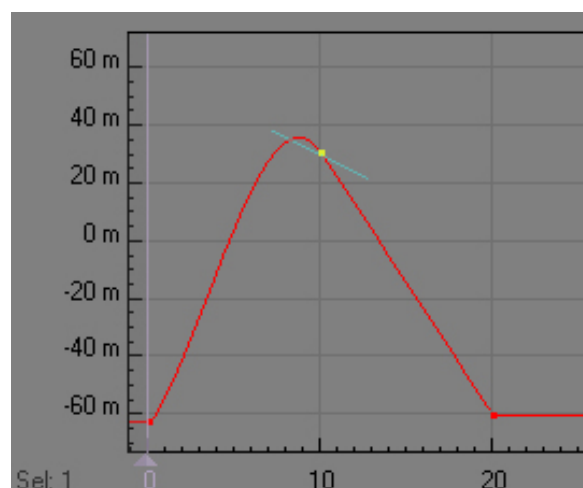


Continuity 1

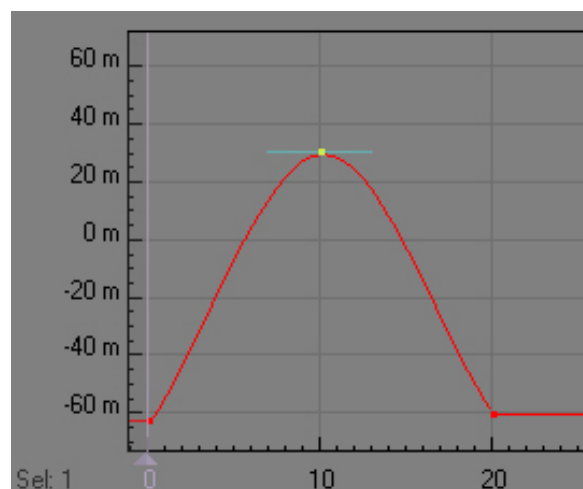
You would rarely want to use a positive continuity — this will cause an object to overcompensate as it passes through the keyframe and appear to stutter or *roller coaster* at the frame.

**Bias** determines whether an item's spline path *leans* to one side of a keyframe or the other (-1 = more slack incoming, 0 = equal slack, 1 = more slack outgoing). You accomplish this effect by moving the *slack* in the spline path to one side or the other of a keyframe. This serves to accentuate motion — the incoming motion by undershooting the keyframe and creating a feeling of anticipation, or the outgoing motion by overshooting the keyframe. For example, a racing car moving through a turn could use either a negative or a positive **Bias** setting to a) anticipate the turn with a negative **Bias**, or b) overshoot the turn with a positive **Bias**.

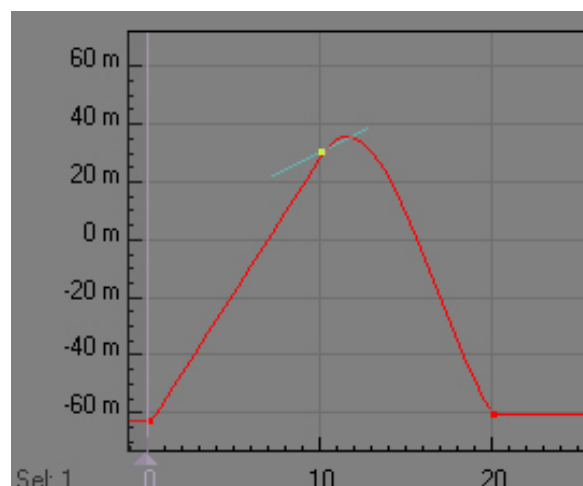
Negative **Bias** values place the slack before the keyframe while positive **Bias** values place it after the keyframe.



Bias -1



Bias 0



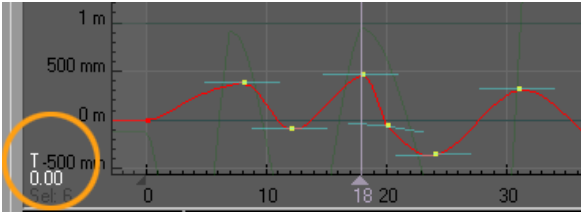
Bias 1



## Interactive TCB Adjustments

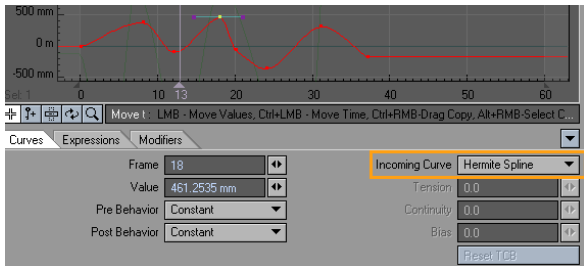
You can interactively adjust TCB keys with your mouse. Simply press F1 for tension, F2 for continuity, or F3 for bias and then drag your mouse left to decrease or right to increase the value. You will see a small indicator in the lower-left corner of the graph.

This only works in the **Move Edit Mode** and for the first mouse drag. The tool stops when you release the mouse button.



## Hermite Spline

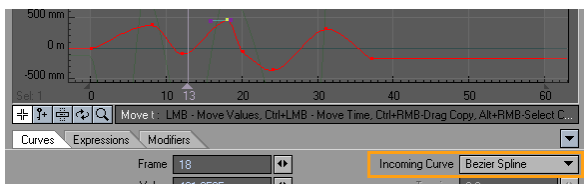
If you use **Hermite Spline**, a tangent control handle will appear, emanating left from the key. This type of curve is an extension of the standard **TCB Spline**, but allows a wider range of results. Drag the handle (at the end) up or down to change the angle of the tangent and thus the shape of the curve.



NOTE: Note that TCB Splines generally limit you to more realistic results.

## Bezier Spline

Splines using **Bezier Spline** operate like bezier curves do in many industry-standard paint and illustration packages. When you create a key, you must drag before releasing your mouse button to edit the handles. Otherwise, bezier keys have handles that coincide with the key. For an existing key, select it and then click-drag on it to pull out the handles.



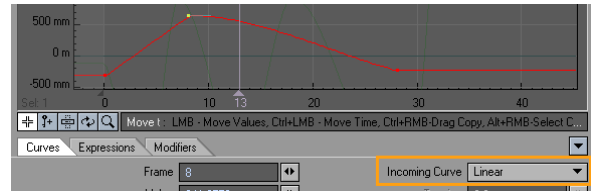
Drag the handles to change the angle of the tangent and thus the shape of the curve. If you are curious, Bezier splines are indeed a variant of Hermite splines, and thus the results will be very similar.



FUN FACT: Bezier curves were developed by Pierre Bézier for designing Renault automobile bodies.

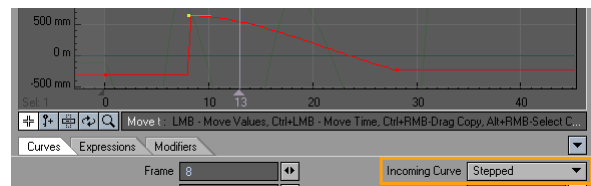
## Linear

**Linear** removes the gradual, smooth nature of a spline curve change and replaces it with more direct, sudden change. **Linear** affects the changes between the current keyframe to the previous keyframe only. By turning **Linear** on or off at different keyframes, graphs may contain both gradual and sudden changes.



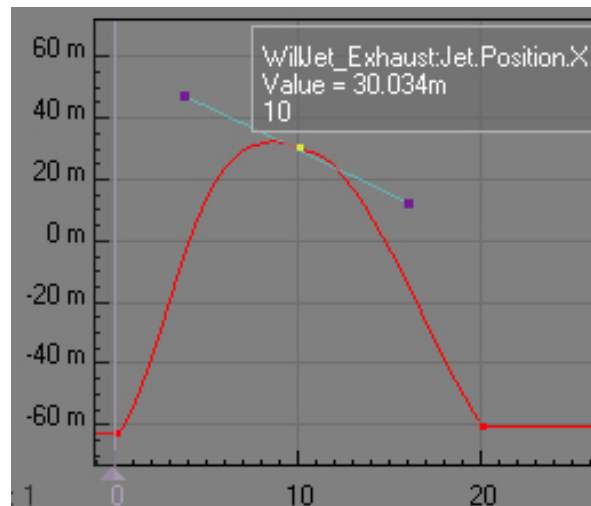
## Stepped Transition

**Stepped transition** holds the preceding keyframe value and then abruptly jumps to the next keyframe value at that frame.



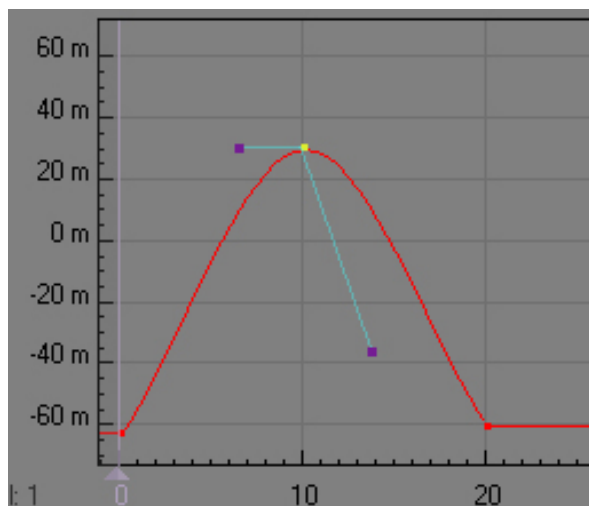
## Dual-handed Control Points

When a Hermite or Bezier (incoming curve) key is followed by another key with the same incoming curve type, the (outgoing) tangent on the right side will affect the outgoing curve as well. Normally, the outgoing tangent will be unified with the incoming tangent. That is, they will operate as though there is only one tangent.





You can break the tangents, that is, make them operate separately by holding the **Alt** key and then dragging either side.



To reunite the tangents, just double-click on either control handle. The opposite side will line back up.

## Expressions

*Expressions* are an advanced LightWave 3D feature that uses mathematical formulas to modify the value of any animation channel. Expressions let you make the motion of scene items dependent on other item motions or factors in a scene. You could, for example, force an object to stay between two other objects, keep feet from going through the floor, or even control the entire posture of a character based on its feet! The possibilities are endless.

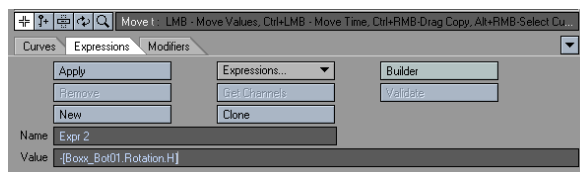
There are two types of expression, “Bracket” expressions and “LScript” expressions. Each has their advantages and disadvantages. **Bracket** expressions take their name from the fact that the channel they are reading is always enclosed in square brackets, “[” and “]”. The **LScript** expressions have a syntax similar to that of LScript. You can use much of the LScript documentation to help with the use of the functions available for using in LScript expressions.



NOTE: You can obtain a reference of available functions from the LScript documentation or from Expression Builder.

## Graph Editor: Expressions Tab

Expressions are built right into the **Expressions Tab** on the **Graph Editor**. This implementation has several advantages over the channel expression modifier. First, expressions are not an attribute for a single channel. Instead, the expressions stand alone and channels are attached to them. This allows you to attach multiple channels to a single expression! Moreover, you can save and load libraries of expressions you create.



NOTE: To get interactive updates in Layout for expressions, make sure you have Auto Key active.

### To create an expression:

**Step 1:** Click the **New** expression button. No channels need to exist in the Channel Bin nor does any channel need to be selected—expressions stand alone.

**Step 2:** Enter a name for your expression in the **Name** field.

**Step 3:** Enter your expression in the **Value** field.

You may also copy the selected expression by clicking the **Clone** button. This creates an independent copy that you can alter.



NOTE: The name of an expression on the Expressions... pop-up menu will list the number of channels attached to it in parentheses.

## Additive Expression

You can use the *Value* variable to make an *additive* expression. *Value* is equal to the (base) keyframed value. For example, if the camera's keyframed X position was 2m and the Light's keyframed X position was 3m, the expression `VALUE + [LIGHT.POSITION.X]`, placed on the camera's X position, would move it to 5m.

Note that editing an item in Layout with an additive expression attached (e.g., `VALUE + [NULL.POSITION.X]`) can cause recursive updates, since you're changing the *Value*. There are a few work-arounds:

**Step 1:** Do the edit in the **Graph Editor**;

**Step 2:** Turn off the expression in the **Graph Editor**. Perform the edit and then turn it back on; or

**Step 3:** Add a null and animate it the same way as the item itself (except without the expression). Then, replace *Value* in the expression with the null's corresponding channel (e.g., if *Value* was the item's Y position, then use the null's Y position instead).

## To rename an expression:

**Step 1:** Make sure the expression is selected. Its name will appear in the **Name** field. To select a different expression, choose it from the **Expressions...** pop-up menu.

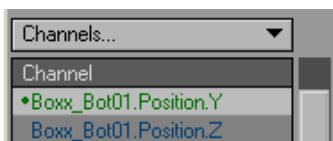
**Step 2:** Type a new name into the **Name** field. This will have no effect on any of the expressions channels.

## To attach an expression on a channel:

**Step 1:** Make sure the expression is selected. Its name will appear in the **Name** field. To select a different expression, choose it from the **Expressions...** pop-up menu.

**Step 2:** Select the channel(s) in the Channel Bin.

**Step 3:** Click the **Apply** button. The *modified dot* (•) will appear to the left of the channel name in the Channel Bin. The expression must be *legal*, of course.



NOTE: It is possible to attach multiple expressions to a single channel, but this is not recommended. The expressions will be evaluated in the order they were attached; however, there is no way to determine that order.

## To determine what channels are attached to an expression:

**Step 1:** Make sure the expression is selected. Its name will appear in the **Name** field. To select a different expression, choose it from the **Expressions...** pop-up menu.

**Step 2:** Click the **Get Channels** button. The contents of the Channel Bin will be replaced with the channels attached to the current expression.

## To remove a channel from an expression:

**Step 1:** Select the channel(s) in the Channel Bin.

**Step 2:** Click the **Remove** button.

## Libraries

You can save all of the existing expressions to a file on your hard drive by choosing **Expressions... > Save Library**. To load a previously saved library, choose **Expressions... > Load Library**. If an expression exists with the same name, it will be replaced. Otherwise, the library of expressions will be added to the list.

You can clear all unused expressions by choosing **Expressions... > Clear Unused**. This clears out any expressions that do not have any channels attached.

## Expression Syntax

LightWave supports two types of expression syntax. The first is identical to the Channel Expression syntax (e.g., the x position of a light at time *t* is given by `LIGHT.POS(TIME).X`). This works normally, as does all of the control syntax (`x < 5 ? y : -y`).

With integrated expressions, you can also use a *bracket notation* syntax to reference any channel in the system. By placing square brackets ([]) around a full channel name, you may access any channel in the system. This includes MorphMixer channels, envelopes, and so on (e.g., `[CAMERA.ROTATION.H]`). You can freely mix and match the two methods of referencing item information (`[LIGHT.POSITION.X]` and `LIGHT.POS(TIME).X`).

Channels referenced in this way will be evaluated in a *dependency-conscious* way. In other words, if channel X references channel Y, which has an expression that follows channel Z, then the bracket notation insures that the Y channel's expression (referencing Z) is evaluated before computing channel X.

Bracket notation expressions may also take an *optional* time argument. The syntax is `[CHANNEL,TIME_ARG]` where *TIME\_ARG* may be any legal expression, but cannot include anything using the bracket notation syntax. An example of a bracket notation expression that follows the camera's X position, but lags by half a second, would be `[CAMERA.POSITION.X,TIME - 0.5]`. To make it lag by four frames, it would be `[CAMERA.POSITION.X,FRAME - 4]`.



## To use bracket notation expressions:

**Step 1:** Add two null objects named *Control* and *Action* to an empty scene.

**Step 2:** Animate the Control null on its Y axis.

**Step 3:** Attach the expression **[Control.Position.Y]** to the Action null. This would lock the two together on the Y axis; wherever Control goes, Action follows.

**Step 4:** Change the expression to **[CONTROL.POSITION.Y,TIME]**. This has the exact same result. As such, unless you want to modify *Time*, you do not need to use the time argument.

**Step 5:** The expression **[Control.Position.Y,Time - 1]** would lock Action to Control, but give it a one second delay.

**Step 6:** The expression **[Control.Position.Y,Frame]** also has the same result as the original expression, but differs in that it feeds the time of the current frame to the expression. This lets you do something like: **[Control.Position.Y,Frame - 30]**. **[Control.Position.Y,Time - 1]** and **[Control.Position.Y,Frame - 30]** yield the exact same results (assuming 30 fps). One expression is working with seconds and the other is working with frames.



**NOTE:** Make sure you use spaces around math operators, like the minus sign used above. Not doing so may confuse the expressions parser, which allows some of those characters in scene item names. Loops are not allowed and the system will report an error if any are detected.

## Bad Expressions

If an integrated expression is erroneous, an error dialog will appear when you attempt to apply it or otherwise exit the input field (e.g., using **TAB** key, **ENTER** key, mouse click, etc.). Also, the **Apply** button will show **Uncompiled** instead of **Apply**.

## Subexpressions

An expression may reference another expression (*subexpression*). The format is identical to other bracket notation calls, except that in place of a channel name, you supply an expression name. *Time* may still be specified, just as if it was a channel reference.

So **[myCenter, Frame - 5]** would cause the system to evaluate the expression **myCenter** at the current **Frame - 5** and return that value.

If the subexpression contains a reference to the *Value* variable, then the current value of the channel — whose expression is using the subexpression — will be used. In other words, all expressions within an expression are using the same *Value* variable.

Also, subexpressions may not themselves contain subexpressions. There will be no error, but any *sub-subexpression* will always return 0.0 upon evaluation.

## Vector References

Bracket notation references to channels may also reference a vector for position, rotation, or scale. This works with the built-in expression functions that take vectors as parameters.

For example, this expression shows how you could find the center X coordinate of two items using scalar values:

```
((Left.Position.X) + [Right.Position.X]) / 2.0
```

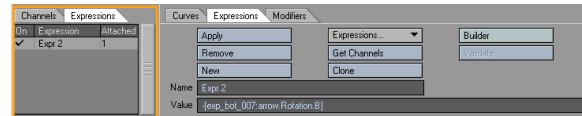
Here is an expression that performs the same feat, but using vectors:

```
center ([Left.Position], [Right.Position]).x
```

An advantage to the second method is that items with spaces in the names can now be referenced by vector functions.

## Expressions Tree

The Scene list area for the **Graph Editor** has an **Expressions Tab**. The Expressions tree shows all the expressions in the system. The first column reflects the active state of an expression. You may toggle the expression on and off by clicking in the On column. All attached channels will be affected.



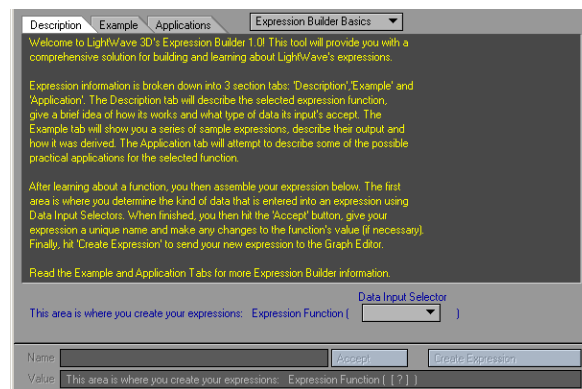
The Attached column shows the number of channels attached to the expression.

Double-click an expression name and the contents of the Channel Bin will be replaced with the channels attached to that expression. You will also select the expression. Hold the **Shift** key and double-click to add the channels without removing any.

If you right-click an expression name, a pop-up menu appears, allowing you to delete the expression. This will detach any channels that may be using it.

## Expression Builder

The Expression Builder is a “wizard-type” feature designed to help you write expressions. To access it, click the **Builder** button on the **Graph Editor's Expressions Tab**. This will guide you through setting up a single function as an expression.



To use, first select an expression function from the **Expression Builder Basics** pop-up menu at the top.

The **Description Tab** gives you a description of what the expression is. The **Example Tab** gives you an example of how to use the expression. The **Applications Tab** explains why the expression might be used.

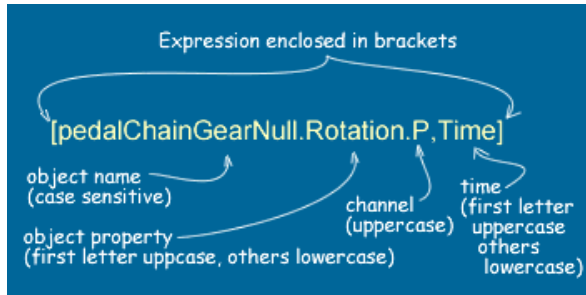
Once an expression “template” is chosen you will be presented with a composite of the expression below the info windows where you are presented with pop-up menus to set what values and/or channels are driving each of the components.

Once you've set the inputs, click **Accept**. You will be able to rename and modify the expression on the lines below. Click **Create Expression** to add the expression to the **Graph Editor's** list.



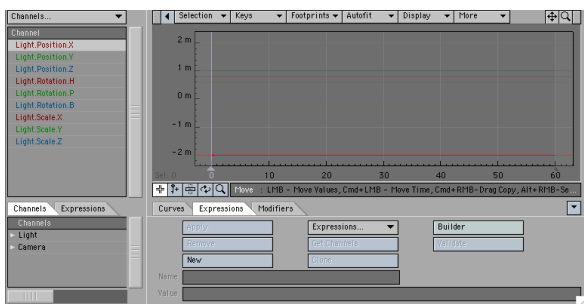
## Bracket Expressions — Syntax

Bracket expressions always have the channel reference enclosed in square brackets and have the general syntax as shown in the figure.



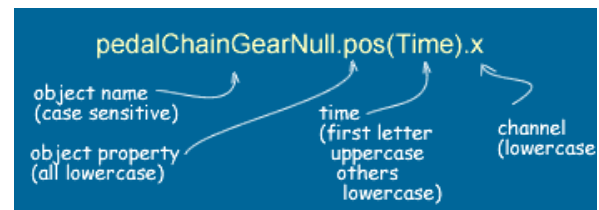
You can also use "Frame" in place of the "Time" and you can also do things like refer to earlier times e.g. [objectName.Position.X, Time - 0.1]

The primary disadvantage of Bracket expressions is that they may only be used via the **Expressions Tab** in the **Graph Editor**.

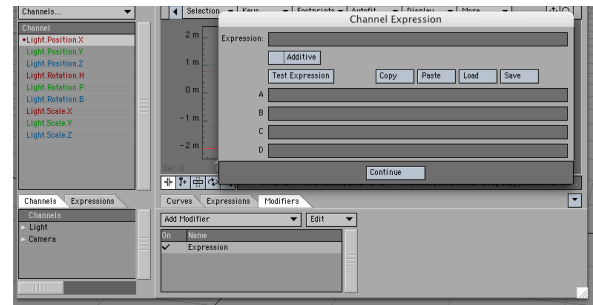


## LScript Expressions — Syntax

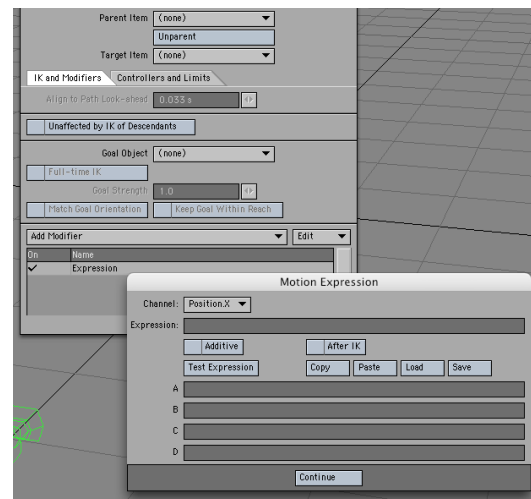
LScript expressions are not enclosed by square brackets. The object property is all lowercase, as is the channel — see the figure below.



LScript Expressions are generally used via the **Modifiers Tab** in the **Graph Editor**,



or, more typically, via the **Motions Options Panel**.



One advantage of LScript expressions applied in the **Motion Expressions Panel** is that they can act after an IK example.

LScript expressions can also be used to create expression-controlled displacements via the **Displacements Panel** where the expression can act before or after a bone deformation.





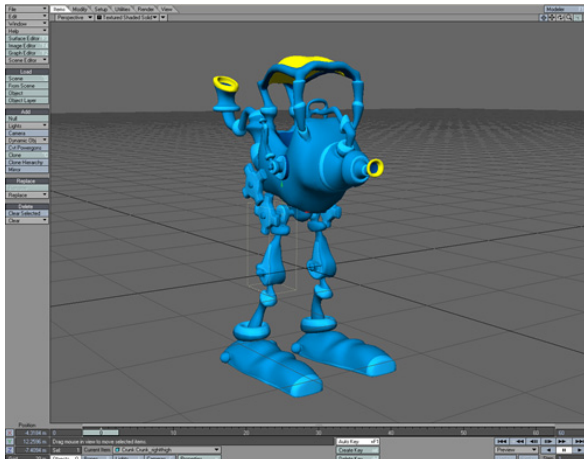
## Using Expressions — Examples

### Example 1: Expression Basics: Rotating a Gear

William Vaughan says: "I've been using LightWave for many years now and Expressions have always been something that I considered too technical, something that I'd give my math friends to work on. It wasn't until recently that I found out that it's quite simple to use expressions. This tutorial will walk you through setting up a basic expression that will help speed up animating gears on a Mech."

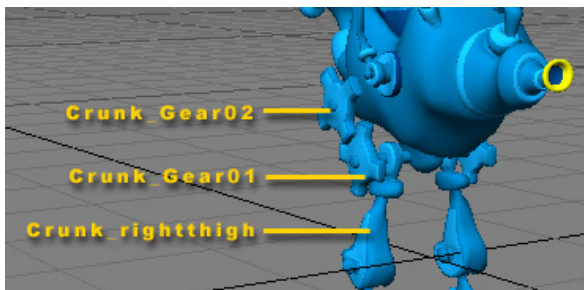
#### Creating an Expression:

**Step 1:** Load the Crunk Car from the Content CD.



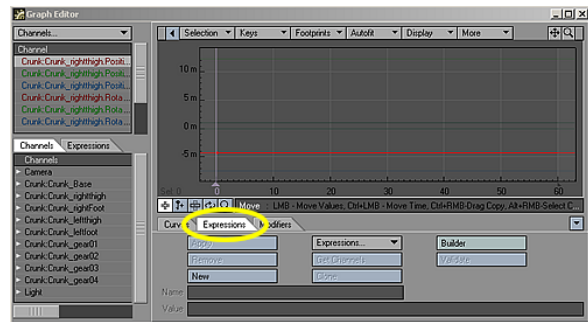
Our goal in this setup is to have the gears automatically rotate when the Thigh "Crunk\_rightthigh" is rotated. The lower gear "Crunk\_Gear01" is already setup to rotate by being parented to the thigh. We can do this because the lower gear's pivot point is in the same place as the Thigh. The upper gear "Crunk\_Gear02" doesn't share the same pivot point and needs to rotate in the opposite direction. You could manually animate the upper gear but using an expression will save you the hassle.

**Step 2:** Select "Crunk\_rightthigh" as the Current Object.

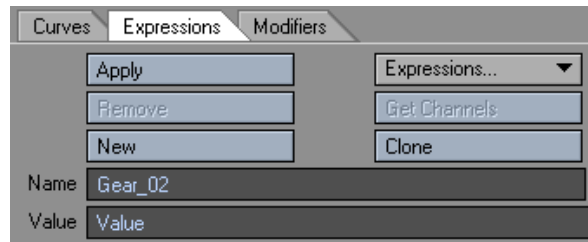


**Step 3:** Open the **Graph Editor** by clicking on the **Graph Editor** button, or use the keyboard shortcut of Ctrl F2.

**Step 4:** Click on the **Expressions Menu Tab** located under the graph display.



**Step 5:** Click the button labeled "New" and give it a name. I chose to name mine "Gear\_02".



**Step 6:** In the Value field, select "Value" and replace it with "-".



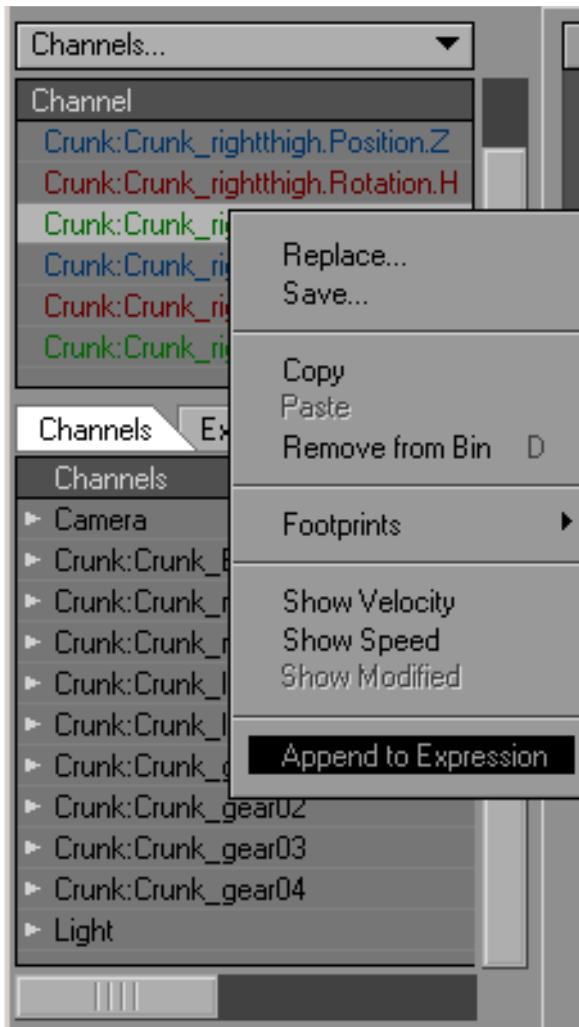
The "-" will make the gear spin the opposite way of the thigh by giving it a negative value.

**Step 7:** We want the Gear to spin the opposite way of the Thigh by giving it a negative value of the Thigh's Pitch. The "-" makes it negative, now all we need is the Pitch value of the Thigh. From the Channel List choose "Crunk:Crunk\_rightthigh.Rotation.P".





**Step 8:** Right-click and choose Append to Expression.



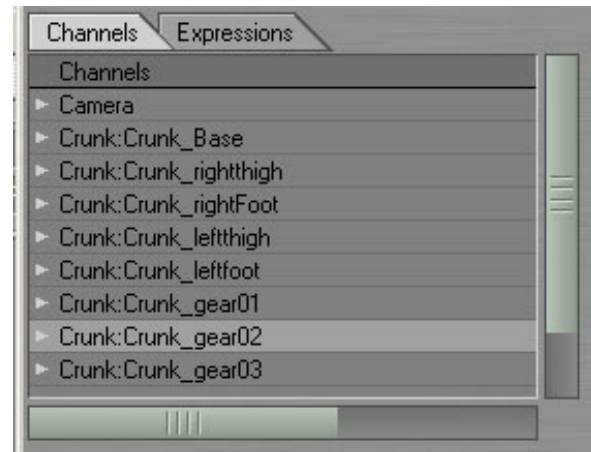
The Value Field should look like this:



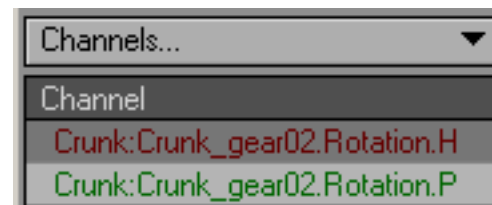
**Congratulations!** You have just written your first expression. Now that wasn't so hard was it? All we need to do now is apply this expression to "Crunk\_Gear02" and we'll be finished with the setup.

## Applying an Expression:

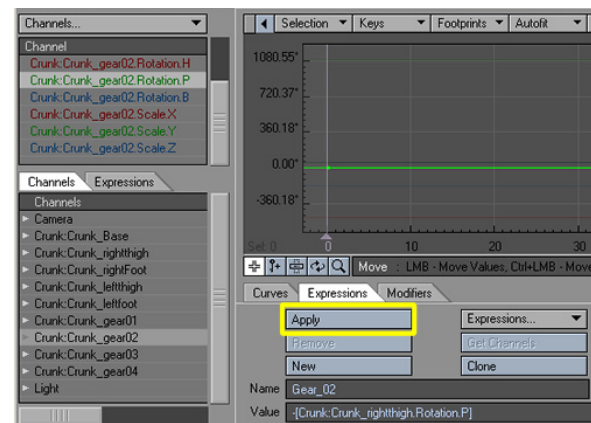
**Step 9:** Now that we have our Expression let's apply it to the Gear. From the Channels list select "Crunk\_Gear02".



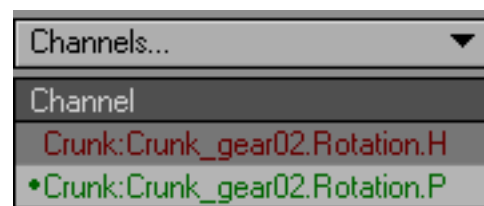
**Step 10:** From the Channel List choose "Crunk\_gear02.Rotation.P".



**Step 11:** Click Apply.



**NOTE:** A small dot is placed next to the channel to let you know that it has an expression attached to it.



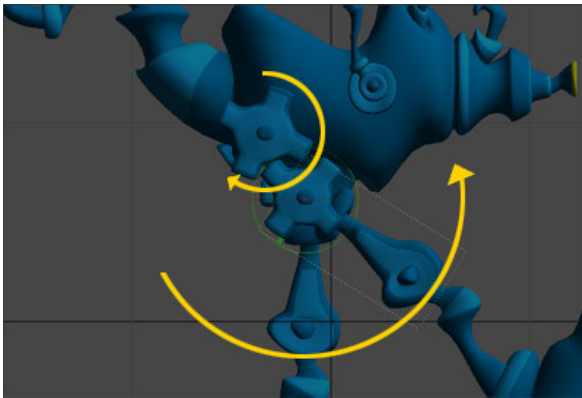
Let's take a look at what happens when we rotate the Thigh.

**Step 12:** Close the Graph Editor window and make sure that **Auto Key Create** is selected.

**Step 13:** Select "Crunk\_rightthigh" as the Current Object, and rotate its Pitch.



**Crunk\_Gear02** should rotate on its Pitch in the Opposite direction of **Crunk\_Gear01** and **Crunk\_rightthigh**.



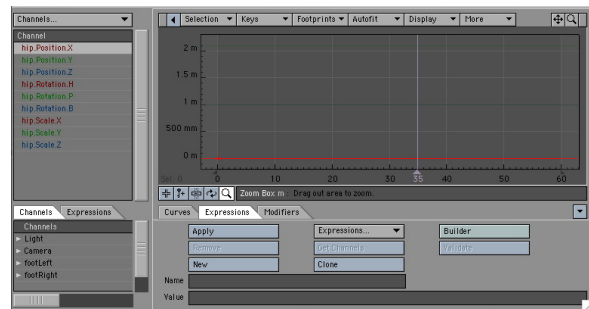
Well, there you have it. Your first expression applied and in action. We've only scratched the surface on the power of Expressions in LightWave.

## Example 2: Expression Builder

In this example we will use the **Expression Builder** to center the hips of a character between its two feet. This example assumes the character's hips are not determined by any form of IK calculation. If that is the case you will need to use LScript expressions (applied via the **Motion Options Panel**) as only LScript expressions can act after IK.

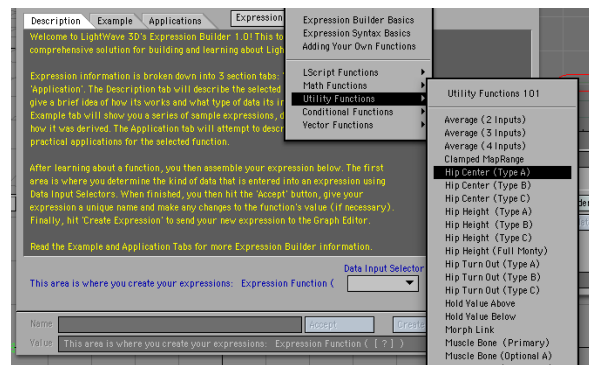
Assume the name of the character's two feet are "footLeft" and "footRight" and that the position of the hips are controlled by a single null called "hip". To center the hip null we will fix its X and Z coordinates (and leave the height, Y, to the animator).

Open the **Graph Editor** and load the hip channels. Click on the **Expressions Tab** — your panel should look like the figure.

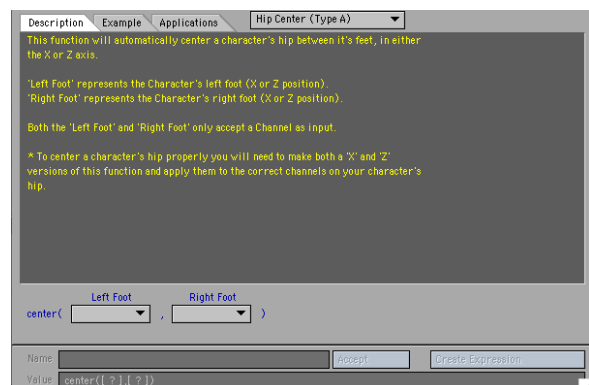


If you select your hip null before opening the **Graph Editor** the hip channels will automatically be added to the channel bin.

First we will create the bracket expression to center the X coordinate of the hip null. Click on the **Builder** button to open the **Expression Builder Panel**, click on the **Expression Template** button at the top of the panel and select the **Hip Center (Type A)** menu option as shown below.



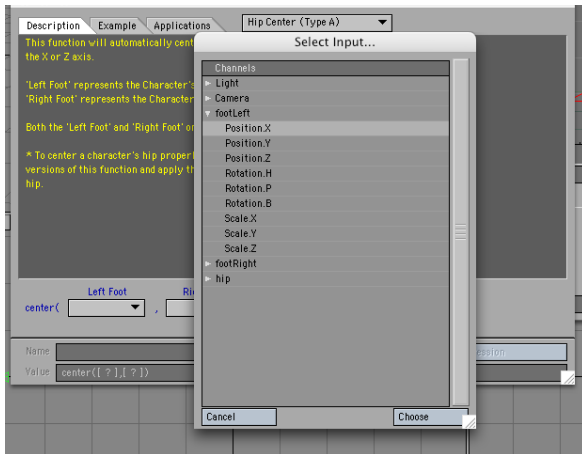
This brings up the expression template — all you have to do is say which channels you want to slot into the template. Expression builder will then create the full expression for you and add it to the **Graph Editor**. The template for hip centring can be seen below.



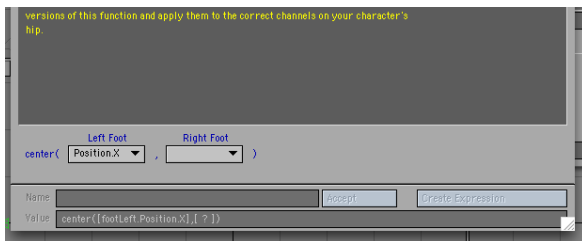


The good thing about the **Expression Builder** (apart from creating a template for you) is that it also gives an explanation for how to use the expression template. To center the hips we will use the “center” function. This function takes two arguments, the two coordinates between which you want to center the object. The important thing to understand is that the expression can only center in *one* dimension. To center in two dimensions (ie. the X and Z coordinates) we use the expression twice. Once to center the X coordinate and a second time, using the Z coordinates, to center in the Z direction.

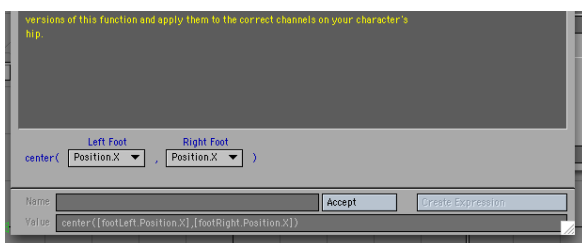
Back to the **Expression Builder** template First for the X direction. We need to enter the X position of the left foot and the right foot. To do this for the left foot click the **Left Foot** pop up menu button and select the **Channel** option to bring up the **Channel Selection Panel** as shown:



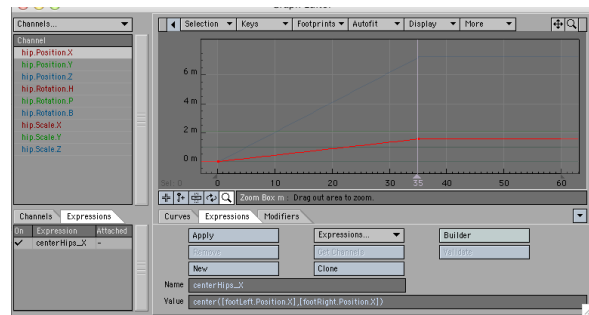
and then select the “Position.X” of the footLeft object. Click **Choose** and you should notice that the X position channel of the footLeft object has been entered into the expression (which is written in the “Value” field on the panel) as shown:



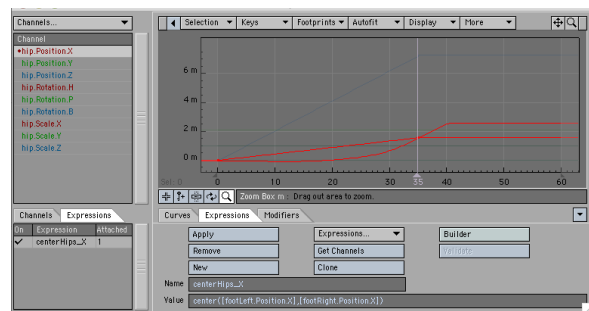
Do the same for the “Right Foot” pop up menu, but now select the Position.X channel for the rightFoot object. The **Expressions Builder Panel** should now look like this:



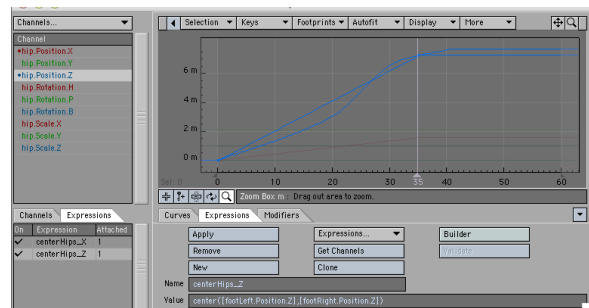
Now, click the **Accept** button to accept the expression. Expressions Builder gives the expression a name in the **Name** field — usually something like “newExp\_2” — delete this and replace it with something meaningful like “centerHips\_X”. Finally, click **Create Expression** — this tells **Expressions Builder** to enter the expression in the **Expression** fields in the **Graph Editor**. Close **Expressions Builder** and the **Graph Editor** should now contain your expression as shown below.



In order for the expression to actually affect the X coordinate of the hip object we need to “Apply” the expression the X channel of the hip object. To do this select the “hip.Position.X” channel in the channel bin and then click the **Apply** button. The hip.Position.X channel should then have a dot to the left of the name to show that that channel has an expression applied to it. If you have the **Always Show Modified** switched on in the **Graph Editor** options then as soon as you apply the channel the modified curve will appear in the curve window as shown:



We still have to center the Z coordinate of the hip object. To do this you could use the Expression builder again (choosing the Position.Z channels for the footLeft and footRight template slots), however, there is a quicker method. Since the expression for the Z coordinate is similar to the X coordinate we can create it by “Cloning” the centerHip\_X expression. You do this by clicking the **Clone** button (on the **Expression Tab** of the **Graph Editor**). This will create the same expression, but with the new name “centerHip\_X (Clone)”. All you need to do is edit the expression name – change it to something like “centerHip\_Z” and then change the X’s in the actual expression to Z’s. Then select the hip.Position.Z channel in the channel bin and click “Apply” to get:



Your hips will now remain centered between the character’s feet no matter what the feet do.

If you look at the Utility functions in Expression builder you will notice there are several variations of hip centring which are a little more complex to set up but they give you more control over your character.





### Example 3: Joystick Control for Morphs

In this more advanced example we will use Expression Builder to create the expressions for a joystick control. The joystick will be used to control the blinking of a character's eyes.

**Overview:** We begin by explaining what we are trying to do. The blinking of a character's eyes is modelled using four morphs: A left eye blink, a right eye blink, both eyes blinking and both eyes opening wide. We will control the left and right eye blinking by moving a null to the left and right respectively. The opening and closing of both eyes will be controlled by the *same* null, not with the left/right motion, but with the up/down motion of the null. We will call this null the "joystick" null. The big advantage of using this method is that if we move the null both sideways and up (or down) ie. diagonally, we will simultaneously combine all four morphs by using only one controller — the joystick null.

How do we achieve this with expressions? We will use four expressions:

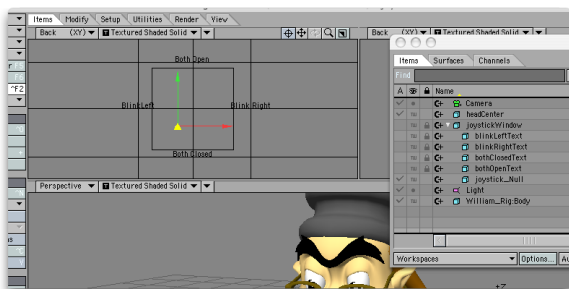
blinkLeft,  
blinkRight,  
blinkBoth and  
wideOpen.

Each expression will connect the joystick null to one of the four morph channels:

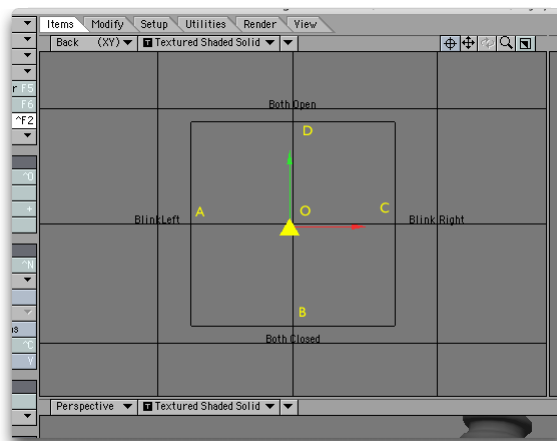
object.Body.Eyes.Blink\_Left,  
object.Body.Eyes.Blink\_Right,  
object.Body.Eyes.Blink\_Both and  
object.Body.Eyes.Wide\_Both.

(The actual object in the screenshots is called William\_Rig.) There remains one further hurdle: when the joystick null moves, it will move a certain distance (eg 50mm) which will need to be converted into a morph percentage. When using morphs, 100% is actually represented by the number 1 (and 0% by 0). Thus if we want a distance change of 50mm to correspond to a morph percentage change of 0% to 100% a function called "maprange" will convert this range of 0 to 1 for us.

The joystick is just a null which we will move in the XY plane to control the morphs. The utility of the joystick approach is greatly enhanced if there is some graphic on-screen indicating which directions correspond to which morph changes, and also, where 0% position is. For this example we have created all of these using five other nulls. Four nulls are used with an "Item Shape" custom object whose only purpose is to place some text in the viewport. The last null has a square as the custom shape showing the range for the joystick (and the center). Also, all the nulls are parented to the "square window" null. This allows us to position the whole joystick control setup anywhere in the scene. For convenience, the joystick is usually placed in a viewport by itself (or alongside any other such controls).



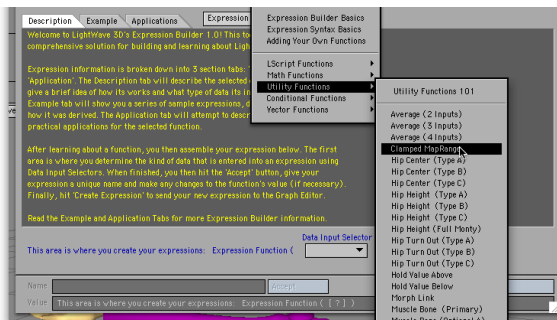
control the closing of the left eye as the joystick moves from O to A (see the figure).



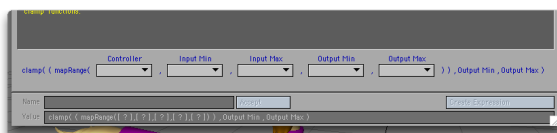
In terms of distance the null is moving from X=0 (at O) to X=-50mm (at A) the full range of the eye closing corresponds to a morph percentage change of 0 to 75% or, for the expression, a change from 0 to 0.75.



We will use Expression Builder (EB), so open the **Graph Editor**, click the **Expression Tab** and click the **Builder** button. Once the **EB** is open, use the pop-up menu to choose the **Clamped Range** utility function as shown:

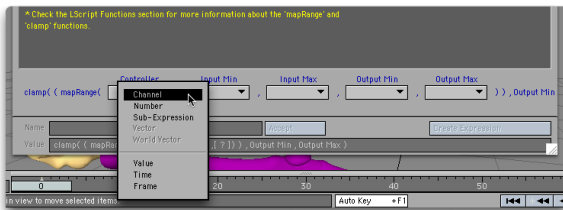


We now have an expression template with five inputs.

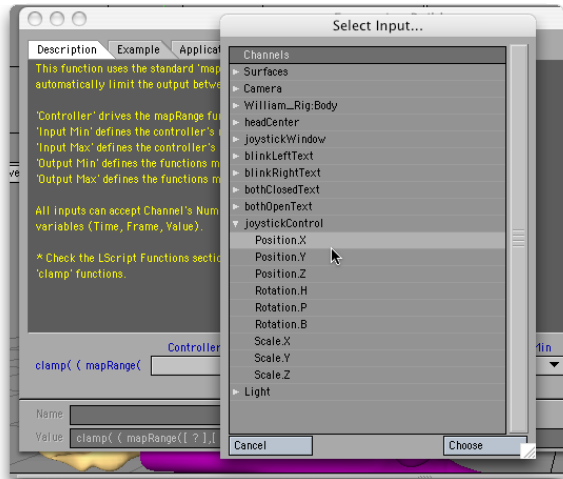


This is the function that "maps" the range of the joystick controller (ie. -50mm to 0mm) to the range of the morph (ie 0.75 to 0.0) To use it we need the joystick controller to go in the first input, so, select the "Channel" from the Controller input

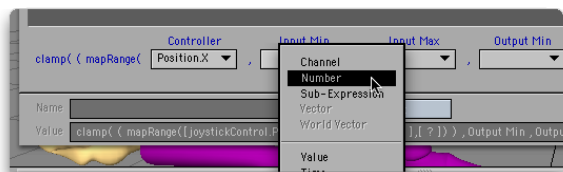
We will start by creating the BlinkLeft expression. This expression will



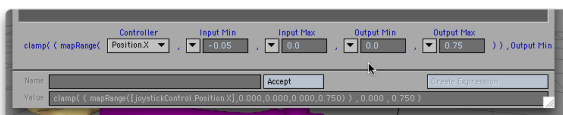
and then select the X position channel of the joystickControl object.



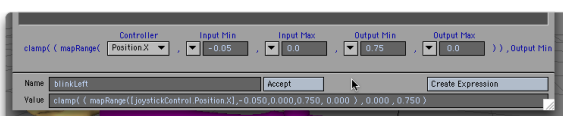
Now, the "Input Min" is the minimum value (as a number) that the joystick controller will take. In this case it is -50mm, or -0.05. Thus, choose the number option in the "Input Min" popup,



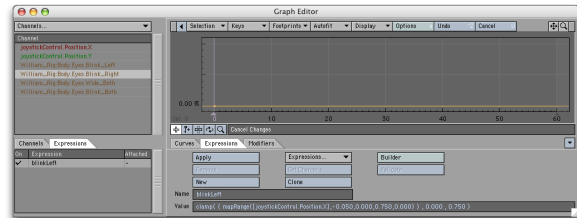
and enter -0.05. Similarly, for the remaining three entries: The "Input Max" is the maximum value we expect the X coordinate of the joystick to take (for controlling the left wink – it will actually move to +50mm for the right blink, but for the left wink we want an X value of 0 to correspond to the 0 morph percent. Thus the "Input Max" is 0. For the "Output Min" we want 0.75 – the morph percent that corresponds to -50mm (which isn't actually the smallest number), and for the "Output Max" we want 0 (giving a morph percent of 0%). Once all your numbers are entered, the template should look like this:



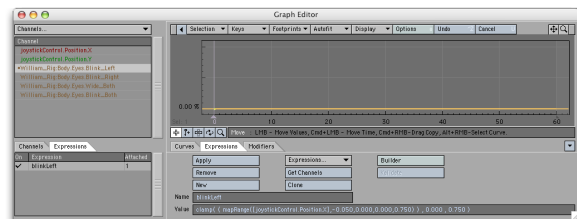
Now, click the **Accept** button, which tells **EB** to accept all your template entries. **EB** usually gives the expression some generic name; so delete it and type in "winkLeft" for the expression name.



Finally, click the **Create Expression** button. This tells **EB** to send the expression to LightWave. If you now open the **Graph Editor** and click on the **Expression Tab** you should see your expression!



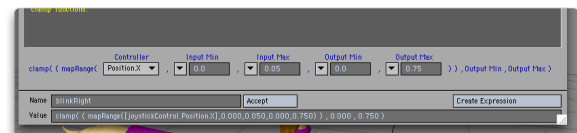
You can close the **EB Panel** as we won't need it until later. Back in the **Graph Editor** (GE), we now have to attach or apply the expression to the channel which it is going to control ie the left blink morph channel. To do this you need to get the channel into the channel bin (the list is on the top left of the GE). In our scene the morph channel is called William\_Rig:Body.Eyes.Wink\_Left". Once it is in the channel bin select it and then click the **Apply** button on the **Expression Tab**. A dot should appear next to the morph channels name telling you it now has an expression attached to it.



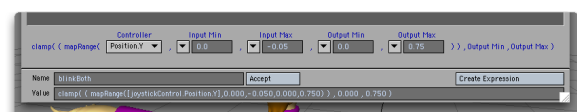
That's it for the left eye winking. If you move the joystick controller's X position from -50mm to 0 the left eye should wink.

We now have to repeat this three more times to create the expression to control the right wink, the wide open case and the both closed case. The process is exactly the same except that the numbers entered into the Expression builder need to be changed and, for the blinkboth and wideOpen cases, it is the Y position of the controller that is used. These three **EB Panels** are shown below:

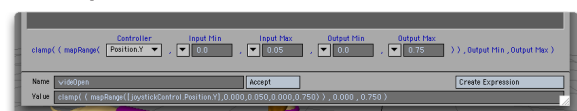
### Wink right



### Blink Both



### Wide Open



Once you have applied these three expressions to their respective channels your joystick controller should be fully functional.

One last thing. Why is the "clamp" function part of each expression? The maprange function can take values outside the range of the "output" min and max values i.e. it can give negative morph percents or percents greater than 100%. But the clamp function stops this from happening. It takes in the values of the maprange function and forces them to stay within the range of the very last two numbers in each expression (in this case, the range 0 to 0.75)





## Connecting LScripts to Expressions

You can write an Lscript function, or user defined function (UDF), using all the keywords available in LScript and then access the function in an expression.

Employing embedded LScript, users can now write their own functions to use with LightWave Expressions.

Expressions UDFs are used just like any built-in Expressions function. Parameter passing is limited to simple data types — strings, numbers and vectors. As long as an expression evaluates to one of these data types, it can be used as an argument. Expressions UDFs are stored in the LightWave->LScripts directory within their own directory called “expressions”. A default library of functions can be maintained within this directory called “library.ls”. This library of functions is automatically loaded into the Expressions engine when LightWave is initialised, and its defined functions are consequently available to any LightWave expression or Expressions UDFs that references them.

Additionally, individual Expressions UDFs can be stored in their own files in this same directory. The name of the file containing the UDF must exactly match that of the function name being referenced. The file may contain any number of other UDFs to support the main function, but must contain at least a UDF whose name and argument count matches that being referenced in the expression.

Data exchange between UDFs is not limited in their types. UDF-to-UDF calling is exactly the same as it is in LScript.

By way of example, assume the following files exist in the required directory:

```
LightWave->LScripts->expressions->library.ls
```

```
LightWave->LScripts->expressions->channelValue.ls
```

The “library.ls” file contains the following content:

```
locateChannel: fullchannel
\{
  parts = parse(".",fullchannel);
  group = ChannelGroup(parts[1]); // start with root channel
group
  lastgroup = group;
  subgroup = nil;
  x = 2;
  while(group)
  \{
    // scan sub-groups to match parts[x]
    // if a match can't be found, then it
    // is probably the start of the channel
    // name
    subgroup = ChannelGroup(group,subgroup);
    last if !subgroup;
    if(subgroup.name == parts[x])
    \{
      group = subgroup;
      lastgroup = group;
      subgroup = nil;
      ++x;
    \}
  \}
  if(!lastgroup) return(nil);
  // anything left in the parts[] array are the components
```

```
// of the channel name itself. put them together for
// channel searching
channelname = ""; // avoid creating an array
psize = parts.size();
while(x <= psize)
\{
  channelname += parts[x];
  if(x < psize) channelname += ".";
  ++x;
\}
// scan the defined channels in the final group to see
// if we can match the channel name
chchannel = lastgroup.firstChannel();
while(chchannel)
\{
  last if chchannel.name == channelname;
  chchannel = lastgroup.nextChannel();
\}
return(chchannel);
\}
// replace the built-in clamp() function
clamp: val, lower, upper
\{
  result = val;
  if(val < lower) result = lower;
  else if(val > upper) result = upper;
  return(result);
\}
```

While the “channelValue.ls” file contains the following content:

```
chan;
chanName;
channelValue: channel, time
\{
  if(chanName != channel) chan = nil;
  if(!chan)
  \{
    // cache the channel for speed
    chan = locateChannel(channel);
    if(!chan) return(0);
    chanName = channel;
  \}
  return(chan.value(time));
\}
```

In Layout, you might then enter an expression like:  
clamp(channelValue("WashLight.Intensity",  
Time),0.0,1.0)

This will invoke the channelValue() UDF, which then invokes the locateChannel() UDF (defined in the default library file “library.ls”) to resolve a string channel reference to an actual LScript Channel Object Agent. The channelValue() UDF returns the value of the specified Light Object’s intensity value at the current time. This value is then passed to the (script) clamp() UDF (also defined in “library.ls”) to keep it in a specified range.



Alternately, you could use the **Graph Editor's** direct channel reference syntax with the UDF call:

```
clamp([WashLight.Intensity,Time],0.0,1.0)
```

UDF references that have been loaded into the Expressions engine are automatically updated the next time they are evaluated when their respective files have been modified. For instance, if you have expressions referencing `channelValue()`, altering the last line of the function to read:

```
return(chan.value(time) + 1.0);
```

will instantly return new values the next time the expression is evaluated (e.g., on the next frame).

```
\end{document}
```

## Object References

Objects are referenced by their name. The system does not currently support space characters ( ' ') in object names, so cloned objects (i.e., "Null (1)", "Null (2)", etc.) cannot be used unless they are renamed.

The "Scene" object is the only pre-defined object in the system. All other object references must equate to an object in the current scene.

## Built-in Functions

double	<code>sqrt(double)</code>
double	<code>exp(double)</code>
double	<code>log(double)</code>
double	<code>sin(double)</code>
double	<code>cos(double)</code>
double	<code>tan(double)</code>
double	<code>asin(double)</code>
double	<code>acos(double)</code>
double	<code>atan(double)</code>
int	<code>random(int,int)</code>
double	<code>vmag(vector   double,double,double)</code>
(int   double   vector)	<code>abs(int   double   vector)</code>
double	<code>ceil(double   vector)</code>
double	<code>floor(double)</code>
double	<code>cosh(double)</code>
double	<code>sinh(double)</code>
double	<code>tanh(double)</code>
double	<code>mod(double   vector,double   vector)</code>
double	<code>pow(double,double)</code>
double	<code>rad(double   vector)</code>
double	<code>deg(double   vector)</code>
(double   int   vector)	<code>max(double   int   vector, double   int   vector)</code>
(double   int   vector)	<code>min(double   int   vector, double   int   vector)</code>
double range	<code>(double,double,double)</code>
double	<code>selector(double,double, double,double)</code>
double	<code>step(double,double,double)</code>
double	<code>round(double,int)</code>
double	<code>frac(double)</code>
double	<code>fac(double)</code>
double	<code>cot(double)</code>
double	<code>sec(double)</code>
double	<code>csc(double)</code>
vector	<code>cross3d(vector,vector)</code>
double	<code>dot3d(vector,vector)</code>
double	<code>cross2d(double,double,double, double)</code>
double	<code>dot2d(double,double,double, double)</code>
vector	<code>normalize(vector)</code>
vector	<code>center(vector,vector)</code>
vector	<code>extent(vector,vector)</code>
string	<code>parse(string,string)</code>
double	<code>number(string)</code>
vector	<code>vector(string)</code>
string	<code>string(double + int + s tring + vector)</code>
int	<code>integer(double   string)</code>
string	<code>strleft(string,int)</code>
string	<code>strright(string,int)</code>
string	<code>strsub(string,int,int)</code>
string	<code>strupper(string)</code>
string	<code>strlower(string)</code>
double	<code>randu([double   int])</code>
string	<code>hex(int[,int[,int   "true" ]])</code>
string	<code>octal(int[,int[,int   "true" ]])</code>
double	<code>angle(vector,vector,int)</code>



## Common Object Methods and Data (all objects respond)

vector	pos(double)
vector	position(double)
vector	rot(double)
vector	rotation(double)
vector	right(double)
vector	up(double)
vector	forward(double)
vector	pivot(double)
vector	wpos(double)
vector	wposition(double)
vector	wright(double)
vector	wup(double)
vector	wforward(double)
vector	limits.pos.min
vector	limits.pos.max
vector	limits.position.min
vector	limits.position.max
vector	limits.rot.min
vector	limits.rot.max
vector	limits.rotation.min
vector	limits.rotation.max
vector	limits.right.min
vector	limits.right.max
vector	limits.up.min
vector	limits.up.max
vector	limits.forward.min
vector	limits.forward.max
vector	limits.pivot.min
vector	limits.pivot.max
vector	limits.wpos.min
vector	limits.wpos.max
vector	limits.wposition.min
vector	limits.wposition.max
vector	limits.wright.min
vector	limits.wright.max
vector	limits.wup.min
vector	limits.wup.max
vector	limits.wforward.min
vector	limits.wforward.max

## Mesh Object Methods and Data

double	dissolve(double)
int	points
int	polygons

## Light Object Methods and Data

vector	color(double)
int	points
int	polygons
double	coneangle.rad
double	coneangle.radius
double	coneangle.edge

## Camera Object Methods and Data

double	zoom(double)
double	zoomfactor(double)
double	focallength(double)
double	focaldistance(double)
double	fstop(double)
double	blurlength(double)
double	fovhor(double)
double	fovhorizontal(double)
double	fovver(double)
double	fovvertical(double)

## Scene Object Methods and Data

int	points
int	polygons
int	renderstart
int	renderend
int	renderstep
double	fps
int	width
int	renderwidth
int	height
int	renderheight
double	aspect
double	pixelaspect
double	aspestratio
int	minspp
int	maxspp
int	recursion
int	maxrecurse
int	recursedepth
int	usingTraceShadows
int	usingTraceReflection
int	usingTraceRefraction
int	usingFields
int	usingEvenFields
int	usingMotionBlur
int	usingDOF
int	usingLR
int	usingLimitedRegion
int	lr.x1
int	lr.left
int	lr.x2
int	lr.right
int	lr.y1
int	lr.top
int	lr.y2
int	lr.bottom
int	limitedregion.x1
int	limitedregion.left
int	limitedregion.x2
int	limitedregion.right
int	limitedregion.y1
int	limitedregion.top
int	limitedregion.y2
int	limitedregion.bottom



## Selector/Converter items

double	x (selects the first element of a multiple-data type)
double	y (selects the second element of a multiple-data type)
double	z (selects the third element of a multiple-data type)
int	r (selects the first element of a multiple-data type)
int	g (selects the first element of a multiple-data type)
int	b (selects the first element of a multiple-data type)
vector	rgb (converts a vector data into color-normalized data)
string	asStr (converts int, double, vector to string)
string	asString
int	asInt (converts string, double to integer)
int	asInteger
double	asNum (converts string, int to double)
double	asNumber
vector	asVec (converts int, double, string to vector)
vector	asVector



NOTE: A vector is a group of related values. They could relate to position (X, Y, Z), rotation (H, P, B), color (R, G, B), etc. To get only one component, use a selector as demonstrated below.



NOTE: Expressions react to interactively moved items, even if Auto Key is turned off.



NOTE: You may use XS, YS, and ZS as aliases for Scale.X, Scale.Y, and Scale.Z.

## Sample Expressions

*HeadLight.rot(Time).h* returns the heading rotation value of the HeadLight item at the current time.

*Left.pos(Time).x + Right.pos(Time).x* returns the sum of the Left and Right items' positions on the x axis.

*<1,2,3>.y* returns 2

*<1,1,1>.rgb* returns <255,255,255>

*<.5,.25,1>.rgb* returns <127,63,255>

*BackLight.color(frame / Scene.fps).rgb* returns RGB vector value for color BackLight at a user-defined frame converted to a time index using the Scene object's fps setting. The frame variable is returned to the caller and can have its value explicitly set before each evaluation of the expression.

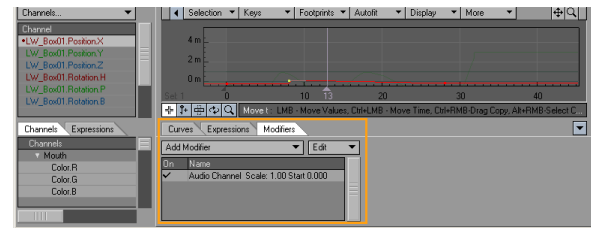
*2 \* "1 2 3".asVec.y* returns 4

*((Scene.usingLR ? (Scene.lf.right - Scene.lf.left) : Scene.width) / 2).asInt* finds the horizontal center of the frame.

## Graph Editor: Modifier Tab

### Channel Motion Modifiers

Channel motion modifiers are added on the **Modifiers Tab** of the **Graph Editor**. These modifiers control the motion at the channel level, as opposed to the (scene) item level, where motion is handled by item motion modifiers.



To use a channel motion modifier, select the target curve in the **Graph Editor's** Channel Bin. Then select the modifier from the **Add Modifier** pop-up menu on the **Modifier Tab**. Once added, click on its name in the list to access its settings, if any. Channels with modifiers will have a small dot to the left of their names in the Channel Bin.



NOTE: Modifiers that do not have an explicit additive option are generally additive in nature.



NOTE: Also see the MasterChannel Scene Master plugins. This lets you create custom user-defined channels.

## BoosterLink

IK Booster Link can be used to link a channel from one controller to a channel in another controller. The Linked controller can be driven by the referenced control. This is very similar to how expressions can be used without writing any expressions.



NOTE: See the IK Booster section for more information.

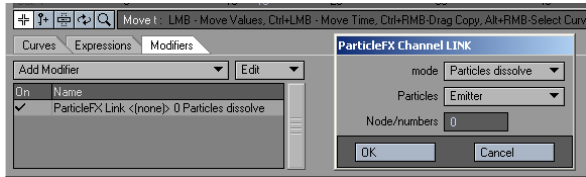
## FX\_Clink

These channel and motion modifiers let children of a parent, which uses FX\_Link, respect the parent's 'time shift' setting. This modifier has no interface.



## FX\_Link

The FX\_Link channel modifier has two functions. First, it can be used to dissolve out the object when the particles die.



### To dissolve an object based on particle life:

**Step 1:** Open the **Object Properties Panel** for the object to be dissolved. (You'll also need an existing emitter controller.)

**Step 2:** On the **Rendering Tab**, click the **Object Dissolve Envelope** button. This adds a Dissolve channel for the object and opens the **Graph Editor**.

**Step 3:** On the **Modifiers Tab** of the **Graph Editor**, add the FX\_Link modifier. Open its **Options Panel** by double-clicking its name in the list after it has been added.

**Step 4:** Set the **mode** to **Particles dissolve** and select the emitter from the **Particles** pop-up menu. Enter the particle number in the **Node/numbers** field; 0 is the first emitted.

**Step 5:** Click **OK**. When that particle dies, the object will be dissolved to 100%.

FX\_Link can also be used to vary a channel based on the number of particles at the emitter

### To control a channel based on the number of particles:

**Step 1:** Open the **Graph Editor** and select the desired channel in the Curve bin.

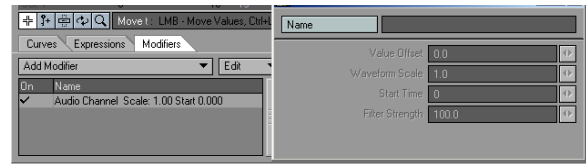
**Step 2:** On the **Modifiers Tab** of the **Graph Editor**, add the FX\_Link modifier. Open its **Options Panel** by double-clicking its name in the list after it has been added.

**Step 3:** Set the **mode** to **Particles numbers** and select the emitter from the **Particles** pop-up menu.

**Step 4:** Click OK to close the panel and play your scene. The channel you selected will be changed based on the number of particles in that particular frame. You can scale the value by entering a number in the **Node/numbers** field. The number of particles will be divided by this value.

## AudioChannel

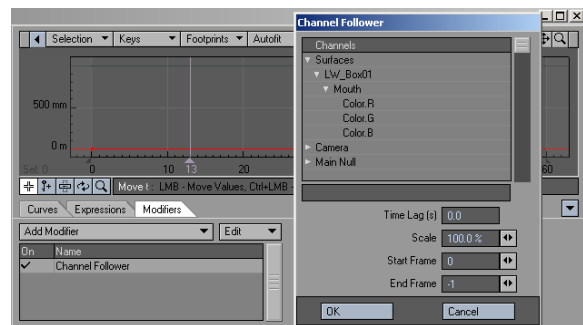
The AudioChannel modifier modifies a curve based on an audio file.



Click the **Name** button to select the audio file you want to use. The **Value Offset** lets you move the entire motion up or down. (The units are the ones used on the graph.) **Waveform Scale** is a multiplier. Thus, a value of 1 will have no effect. A value of 2 will double the values of the effect and .5 will halve it. Use the **Start Time** to enter a frame number when the audio should begin. The **Filter Strength** value will determine the sampling frequency used to convert the audio into a curve. A higher value will cause the curve to more closely follow the contours of the audio's sound wave.

## ChannelFollower

Using Channel Follower (aka Set Driven Key) is similar to parenting an object to another, except that you have control over which motion channels you wish to inherit. You can also modify and delay the inherited value. Moreover, the motion can be inherited from the camera, a light, a bone, or any object in the scene.



Select the channel you wish to follow from the **Channels** list window.

The amount of seconds entered into the **Time Lag** field is added to the current time. This number may be negative.

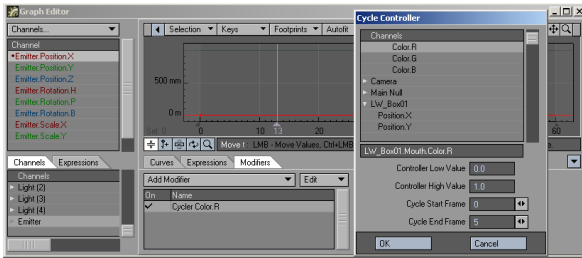
The value can be scaled by inputting a factor other than 100% in the corresponding **Scale** field.

The **Start Frame** and **End Frame** parameters specify when the modifier is applied.



## Cycler

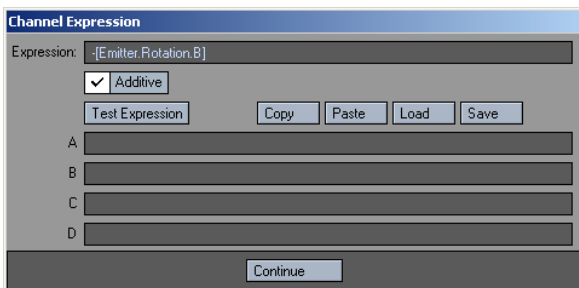
This is a channel-oriented version of the Cyclist item motion modifier.



Select the controlling channel in the **Channels** list window. The **Controller Low Value** and **Controller High Value** settings define how much change is required to equal one full animation cycle. The unit of measure for this parameter depends on the selected control channel.

## Expression

*Expressions* are an advanced LightWave 3D feature that uses mathematical formulas to modify the value of any animation channel. Expressions let you make the motion of scene items dependent on other item motions or factors in a scene. You could, for example, force an object to stay between two other objects, keep feet from going through the floor, or even control the entire posture of a character based on its feet! The possibilities are endless.



The **Expressions Panel** has four buttons: **Copy**, **Paste**, **Load** and **Save**. The **Copy** and **Paste** functions work on a per-screen basis — if you type in your full expression, you may click on **Copy**, open another channel up, and then click on **Paste** and the entire expression is pasted in, including scratch variables. Click on **Save** if you create an expression that you want to use again in the future. You will then be able to load it into other channels at your convenience.

The **A**, **B**, **C**, and **D** fields are scratch variables for the expressions. Each of these can have an expression, which is evaluated before the main expression. This means that the main expression can use the variable A, B, C, and D for some other calculated value. The scratch variables are evaluated in alphabetical order, so B can use A, C can refer to A and B, and the D expression can contain A, B, and C.

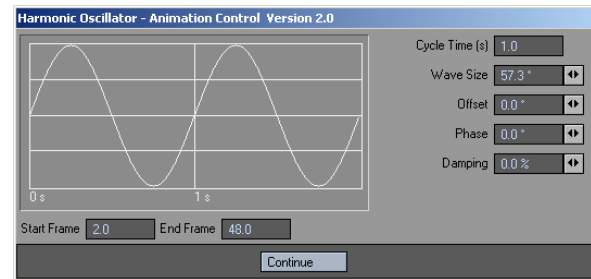
Scratch variables are useful both for breaking up massive expressions and for logically separating the functional elements of an expression. For example, driving a ball's pitch based on its Z-distance can simulate rolling if the ball rotates once for every  $\pi \times \text{diameter}$  it moves. This expression ( $\text{pitch} = \pi \times Z / \text{Diameter}$ ) fits on a line, but you need to include the model's diameter. If you apply it to another ball, the expression must change. If you include the diameter in A, then changing the expression is more obvious; when you scale up the ball, A can be more complex.

## Lscript and Lscript/RT

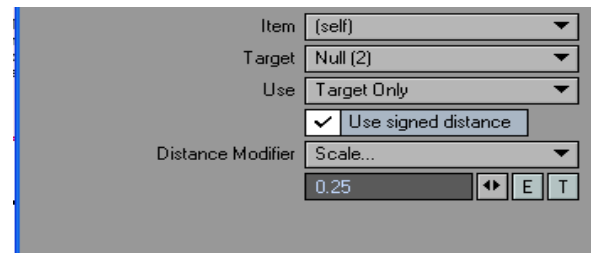
These commands allow you to load a channel-oriented Lscript.

## Oscillator

This is a channel-oriented version of the Oscillator item motion modifier. The channel is determined by which channel the modifier is added to in the **Graph Editor**. The effect is always additive.



## Proximity



The Proximity plugin is implemented as a channel modifier plugin. It can

be added as a channel modifier to any channel. This can be done either directly through the graph editor, or via an envelope for a parameter.

Controls:

Source(item the plugin is applied to):

Proximity computes the distance between the position of a source item and one or more target items or meshes. This setting is used to set the source item.

When set to "(self)", Proximity will use the item to which the channel being modified is attached. If the channel is not attached to an item, Proximity won't do anything if the source is set to "(self)".

Target:

Sets the target to which the distance from the source is measured. The target may be any item, with or without a mesh. If an item has a mesh, the shortest distance between the source position and the mesh will be computed. If an item does not have a mesh (such as a null, a camera, or a light) the position of the item's origin is used.

**Use:( How the channel will be effected)**

**Use:Target Only**

The distance from the source to the target is used. If the target has a mesh, the distance to the nearest point on the mesh is computed, otherwise the distance from the source to the target origin is used.

**Use:Target and Decendants**

Like "Target Only", but any items parented to the target (and the items parented to those items, etc.) are considered as well. The nearest distance found for all items is used.





## Use:Target Leaves Only

Like “Target & Descendents”, but only the leaf items are used. The leaf items are those items in the tree which have no children.

## Use:Target Leaves as Line

Like “Target Leaves Only”, but the positions of leaf items with a common parent are considered as the vertices of a polyline. The shortest distance between the source position and the polyline is used.

**Use Signed Distance:** When computing the distance from a source to a mesh, the distance can be either signed or unsigned. When signed distance is turned on, the distance will be negative if the source is below the surface of the mesh. Otherwise the distance will always be positive.

**Distance Modfyer:** remap the distance calculation

**Distance Modifier:None:** The channel value is set equal to the computed distance.

## Distance Modifier:Scale:

The distance is scaled by the given factor to give the channel value.

## Distance Modifier:Remap:

Arbitrarily change the distance to a channel value. Typically this is used in combination with a gradient texture or an expression.

All the distance modifiers can be enveloped, textured and whatnot for full control of the effect.

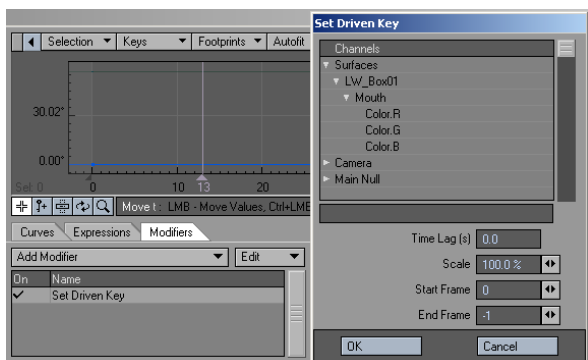
## Added Channels:

**Sticky.rMeshDistance:** Distance of the sticky item to the nearest point on a sticky surface

**Distance to mesh:** Distance of the sticky item to the nearest point on a sticky surface

## SetDrivenKey

Using SetDrivenKey (*aka* ChannelFollower) is similar to parenting an object to another, except you have control over which motion channels you wish to inherit. You can also modify and delay the inherited value. Moreover, the motion can be inherited from the camera, a light, a bone, or any object in the scene.



Select the channel you wish to follow in the **Channels** list window.

The amount of seconds entered into the **Time Lag** field is added to the current time. This number may be negative.

The value can be scaled by inputting a factor other than 100% in the corresponding **Scale** field.

The **Start Frame** and **End Frame** parameters specify when the modifier is applied.

## TextureChannel

This is a channel-oriented version of the **Texture Motion** item motion modifier. It works exactly the same except that you apply it directly to the channel you wish to modify.



Since textures are three dimensional, particularly procedurals, use the **Axis** setting to use the **X**, **Y** or **Z** of the texture. (Note: The differences between the **Axis** selection can be subtle.) You can also move the texture with the **Offset** setting and change the size of the texture using the **Scale** setting.

## MM (Motion Mixer) Channel Driver

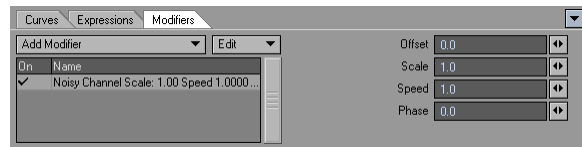
When an X Channel is added to a **Motion Mixer Actor**, this modifier is automatically added. Removing X Channels from an actor is achieved by either using the **Remove Items** entry in the **Actor Menu**, clearing the item from the scene, or manually removing the MM\_ChannelDriver modifier on the **Graph Editor**.



NOTE: For more information see **Motion Mixer** documentation, starting on page 277.

## NoisyChannel

This modifier randomises a channel.



The **Offset** lets you move the entire motion up or down. (The units are the ones used on the graph.) The **Scale** parameter multiplies the noise amount added in to the channel, so a factor of 1 will have a noise effect, a factor of 0 will have no effect, 2 will double the effect, .5 would halve it, and so on. **Speed** is the rate of change of the noise, basically like a texture velocity. **Phase** shifts the effect in time.

The formula is: channel value = old value + scale \* fractal noise(phase + speed\*time)

## RelChanneler

Open the Relativity Channeler Expression Module.

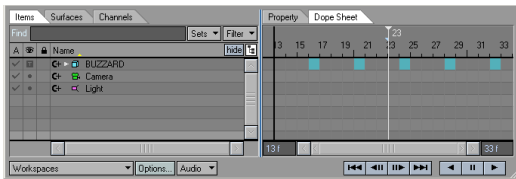


## Scene Editor

(default keyboard shortcut **Ctrl F1**)

The **Scene Editor** gives you the “big picture” of your scene. The **Scene Editor** is a data viewing and editing hub for many of the properties with Layout that pertain to items and surfaces. It does not intend to encompass all properties in Layout nor act as a complete replacement for those that it does support. But, it does intend to offer ease and speed when modifying data about multiple items or multiple surfaces. It also intends to allow users to have chosen data appear simultaneously within a single panel instead of having to jump around from panel to panel to make changes.

Not only are properties about items and surfaces presented, but also any channel data that is attached to these item types. “Items” refers to Layout Items that may consist of Objects, Bones, Cameras, or Lights. Surfaces refer to surfaces currently loaded within Layout. Since surfaces are always attached to Object Items, an object item must exist in Layout before a surface can exist.

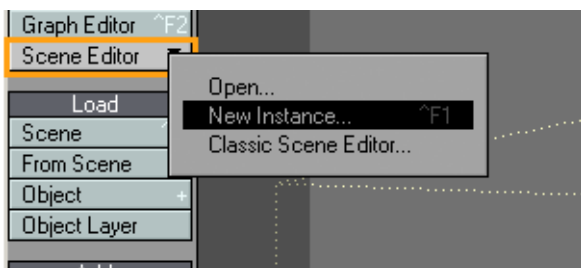


The New Scene Editor is a user configurable scene overview tool. It is used to see everything that exists in a given scene, and can be used to edit most attributes of any item or groups of items in the scene, all from a single location. Multiple instances of it may be opened at any time, and each can be configured independently of the others. This would allow you to configure one Scene Editor specifically for settings that relate to a character in a scene, while another can be set up with options relating to the environment. A third may also be added that configured specifically to control lighting, etc. Each Scene Editor stores its own set of configurations, so they’re “remembered” each time that particular scene is reloaded. Different configurations can be saved for different scenes. You have full control over how these can be set up.

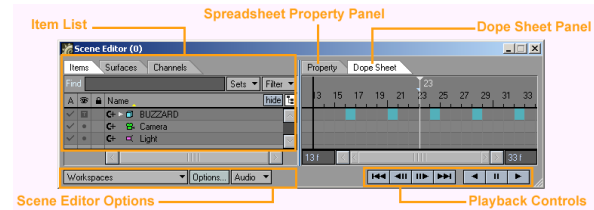
Since all instances of the Scene Editor are displaying the same scene information, changes made in one will automatically be updated in all other instances, so all Scene Editors are displaying the current state of all the scene properties at all times.”



To launch the new **Scene Editor**, click on the **Scene Editor** drop down menu and choose “New Instance”. You also have the option of choosing the **Classic Scene Editor** as well as multiple instances.



The Scene Editor has five main sections to it, each with a specific purpose. These areas are the Item List, Spreadsheet Property Panel, Dope Sheet Panel, Scene Playback Controls, and the Scene Editor Options.



## Item List

The Item List allows selection of various ‘items’ to work with. These include Objects, Lights, Cameras, Bones, Channels, Channel Groups, and Surfaces. Because of the similarities between Objects, Lights, Cameras, and Bones, they have been grouped together to form the **Item List** view. Surfaces have widely different characteristics and form the **Surface List** view. Each of these may contain Channels and Channel Groups and these channels and channel groups form the **Channel List**.



## Item List View: Items Tab

This view is used to view Objects, Lights, Cameras, Bones, and their associated channels, for modification within the Property and Dope Sheet views. It also allows manipulation of item hierarchies, names, existence, colors, lock states, selection status, channel selection, and more.

The item view consists of an item list with each row representing a single item. Four columns extend throughout the list. These are **Active**, **Visibility**, **Locked**, and **Name**.

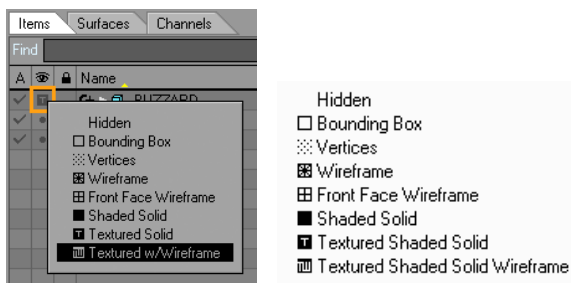


The **Active** column activates or deactivates items. Deactivating an object is like setting its **Object Dissolve (Object Properties)** to 100%, and deactivating a light is like setting its **Light Intensity (Light Properties)** to 0%. For a bone, this toggles its **Bone Active** state (**Bone Properties**).



NOTE: This only applies to Lights, Cameras, Objects, and Bones, not Channels, or Channel Groups.

The **Visibility** (eye) column determines how the item will be displayed. For objects, clicking on this icon will display a pop-up menu where you can select how the object is displayed. This can range from making the object hidden all the way up to showing it as a textured shaded solid. For Objects, the choices are Hidden, Bounding Box, Vertices, Wireframe, Front Face Wireframe, Solid Shaded, and Textured Solid Shaded. For Lights, Cameras, and Bones there is only an on/off state for visibility.



NOTE: You can override the **Visibility** setting to a certain extent by using the viewport's **Maximum Render Level** pop-up menu located on the top left edge of a viewport.



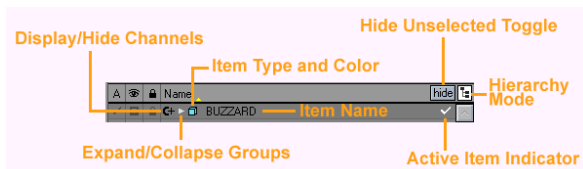
NOTE: The **Color** and **Visibility** options affect only the appearance of items in the Layout view. They do not affect the final rendered image.

Clicking in the **Lock State** column will toggle the locking function on for that item. Locked items cannot be selected in the viewports. However, not all methods for manipulating items obey this flag.

The lock icon will also appear on the **Current Item** pop-up menu (on the main interface) next to the item's name.

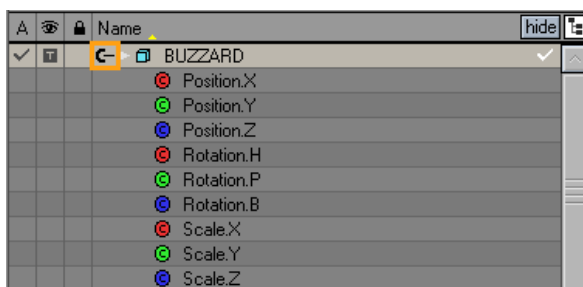


The **Name** column has more options than the other columns. It shows the item name, the type of item, item color. It also allows context menu options, hierarchy management, visibility (showing/hiding) of the hierarchy structure, channel visibility, shows and edits Layout item selection, and shows row selection.



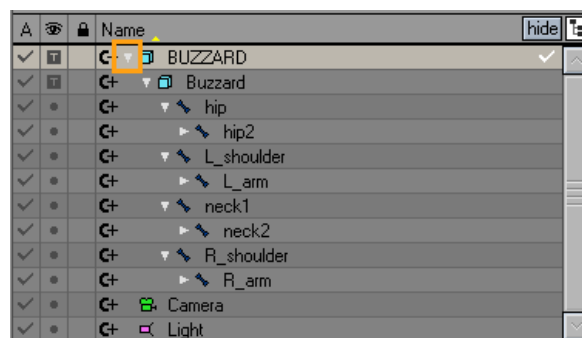
## Display/Hide Channels

The plus/minus sign icon will display or hide the individual channels for the item. Most of the time, these are the position, rotation, and scale channels; however, other enveloped channels can also be included, like light intensity.



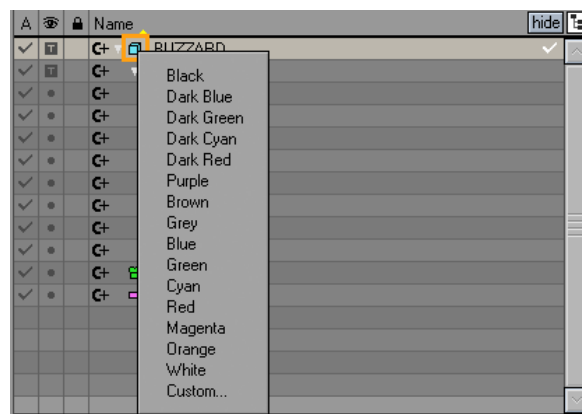
## Expand/Collapse Groups

The scene list is a standard LightWave list window. You can expand and collapse groups as well as subordinate items (e.g., child objects) by clicking the arrow icon that appears to the left of the item name.



## Item Type and Color

The item type icon indicates the type of item and the color used when the item appears in wireframe. You can change the color by clicking on the item and selecting a color from the pop-up menu. When working with complex scenes with overlapping objects, it can be beneficial to use different colors for certain items in the scene.



## Item Name

Simply displays the name of the item.

## Active Item Indicator

When an item is selected (active) a check mark will appear to the right of the item's name.

## Hide Unselected Toggle

This option allows you to hide any item that is not selected in the list. Clicking the button again will unhide all items. This can be very handy when working on complex scenes that contain many items.

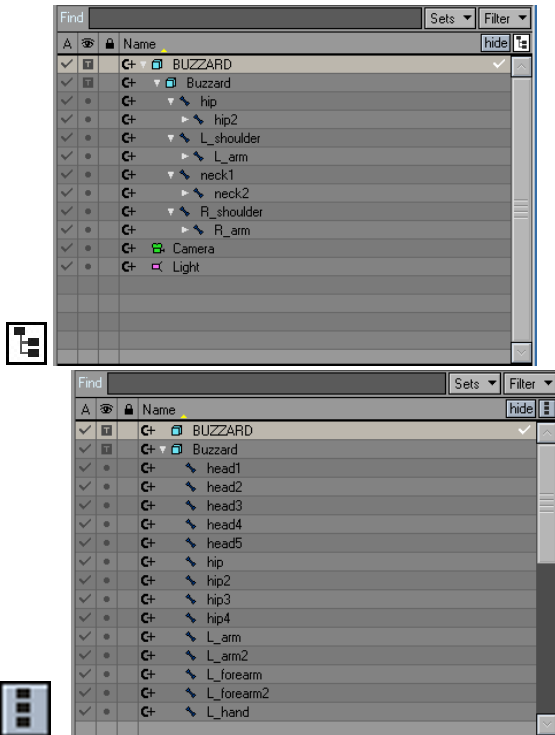


NOTE: This doesn't affect the visibility in the viewport, it only affects the visibility in the item list.



## Hierarchy Mode

At the top of the item list view is the **Hierarchy Mode** button that toggles the display of items hierarchically. When enabled, the items parenting information will be used to indent child items to the right of their parents. In addition, the ability exists to manipulate the parenting information via the mouse within the item list.



There are some rules that govern the hierarchical relationships:

**Object Item:** Can be child of Object Item, Light Item, Camera Item, (none)

**Bone Item:** Can be child of Bone Item, Object Item.

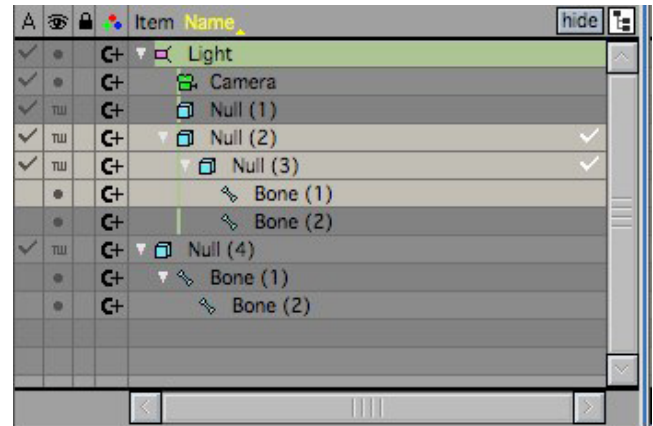
**Light Item:** Can be child of Object Item, Light Item, Camera Item, (none)

**Camera Item:** Can be child of Object Item, Light Item, Camera Item, (none)

**Channel:** Can be child of Object Item, Bone Item, Light Item, Camera Item, Channel Group

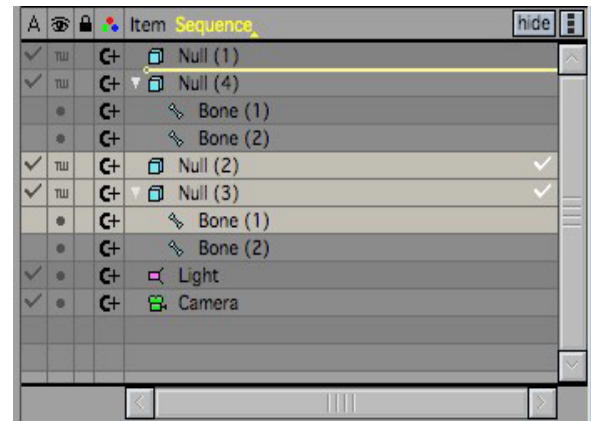
**Channel Group:** Can be child of Object Item, Bone Item, Light Item, Camera Item, Channel Group

**Re-Parenting:** Once some items are highlighted, use the mouse to drag and drop them onto an existing object, bone, light, or camera item to change the parent of the highlighted items. The mouse 'drop' point can also be below the desired parent item at the same indentation level of that parent. Visual markers appear in both cases. The visual aids are brightest when all the highlighted items are appropriate for the reparent operation and darkest when none are appropriate. As an example, trying to reparent a bone from it's owner object to another object is an inappropriate reparent operation. Reparenting can only be performed when the item view is in 'hierarchy display mode'.



The above image shows reparenting the highlighted items to 'Light'

**Re-ordering:** It is possible to change the internal storage sequence of lights, cameras, objects, and bones when a 'Sequence' sort modes is active. The sort modes are changed by mouse clicking on the item view's item column. The mouse is used to drag and drop a highlighted selection between existing items. It is important to note that the sequence is implicitly defined as: all objects (for each object, all bones in that object) followed by all lights followed by all cameras. As such, it is not possible to reposition an object between two lights, for example. The horizontal mouse position is important in determining where to actually reorder items. A visual aid shows the desired insertion point, even if that insertion point is not valid for the highlighted items.



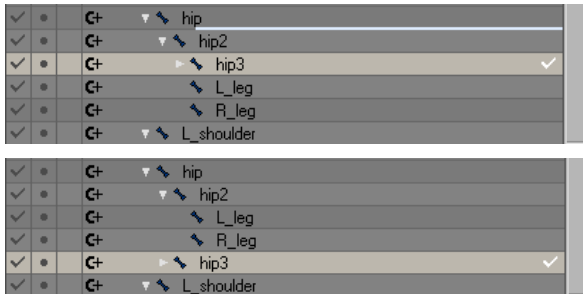
The above image shows inserting the highlighted items between 'Null (1)' and 'Null (4)'

Of important note is that the instance part of the name (the "(2)" in "Null (2)" for example) can change when saving and reloading the scene. This is because the instance value assignment is based on the item sequence at during certain item updates internal to Layout. Instance identification is not part of a scene reference to an item. To best ensure item name display consistency, each item name should be unique.



## Adjusting Hierarchy

You can drag item names up and down to change the order and hierarchy (i.e., parent/child relationships). As you drag, a light blue insert line will appear. You insert the item by releasing your mouse button at the line's position. The line will cycle between different lengths as you drag; the different lengths indicate different levels of hierarchy. The relative length indicates the level the line becomes when you release the mouse button.



### General Rules

Channels cannot be rearranged in either their hierarchy or top-down sequence.

When moving an item, it must not be an ancestor to any new parent.

An item cannot be moved to be a child of an item that it can have as a parent.

If multiple items are selected for rearrangement, the selected parents of any selected children will be used for rearrangement. The selected children will not be rearranged directly, but will follow their parent automatically.

Only the parenting/owner structure may be modified when any sort order display is used.

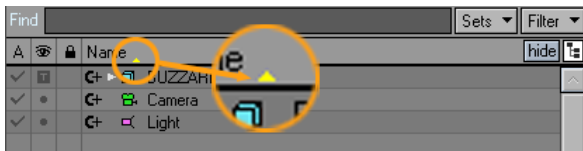
Sorting applies to each level of hierarchy individually.

Any child of a filtered item will also be filtered.

## Item Sorting

Clicking on each column heading can alter the order in which items in the item list are displayed.

The columns are: **Active**, **Visibility**, **Locked**, and **Name**.



Clicking on the **Active** column will sort based on the active state of the item. Sorting toggles between increasing and decreasing active state. The **Visibility** and **Locked** columns sort the same way.

The **Name** column uses alphabetical sorting and applies to item names and channel names. However, lights and cameras are always separated from objects. Therefore, within each hierarchy level, all bones are presented before objects, which are presented before lights, which are presented before cameras. Channel hierarchies are always presented just below the item that owns them. Bone hierarchies are also always presented below the objects that own them and always after the object's channel hierarchies.

There are really two hierarchies present here: one for items, and one for channels. They are managed separately and then incorporated into

the final item list view. Item sorting may also support the 'natural' and 'reverse natural' order. This sorts an item's ItemID.

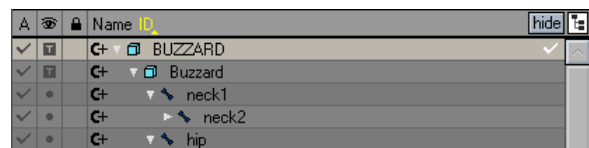
The **Active**, **Visibility**, and **Locked** columns are always displayed on the left of the **Name** column so that they do not keep shifting position as the **Scene Editor Panel** resizes or the item view resizes.

The column heading of the **Active** column is an icon that looks like an 'A'. If this column is used for sorting, an up or down arrow is displayed next to the column heading when increasing or decreasing sorting is used, respectively. When this column is not used for sorting, no such arrow appears.

The column heading of the **Visibility** column is an icon that looks like an eye. If this column is used for sorting, an up or down arrow is displayed next to the column heading when increasing or decreasing sorting is used, respectively. When this column is not used for sorting, no such arrow appears.

The column heading of the **Locked** column is an icon that looks like a pad lock. If this column is used for sorting, an up or down arrow is displayed next to the column heading when increasing or decreasing sorting is used, respectively. Not locked is lower than locked.

The column heading of the **Name** column is the word 'Name'. If this column is used for sorting, an up or down arrow is displayed next to the column heading when increasing or decreasing sorting is used, respectively. When this column is not used for sorting, no such arrow appears. In addition, the use of 'natural' sorting is supported, and is indicated with an arrow that also has a sort indicator of 'ID'.

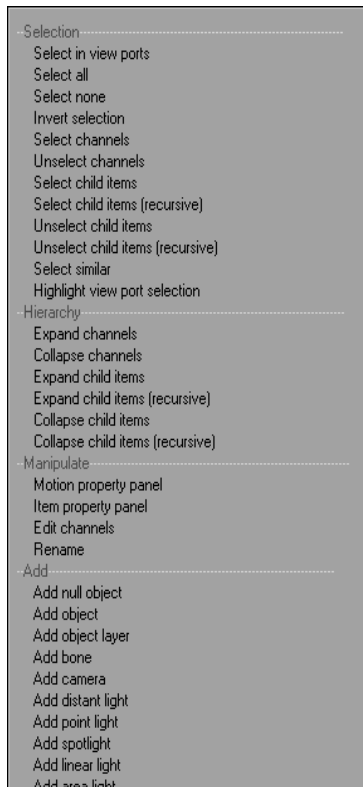






## Item List Right Click Menu

Context menu options are available by RMB-clicking in the item view rows. The row context may change depending on what type of item is in that row. When combining the contexts of multiple highlighted rows, only the options available to all highlighted items will be available.



### Selection:

Select in viewports — applies current selection to Layout item selection.

Select all — This selects all displayed items (after filtering).

Select none — This unselects all displayed items.

Invert selection — This will make all (displayed) items switch their selection state.

Select channels — selects the displayed channels of selected items.

Unselect channels — unselects all channels belonging to currently selected items.

Select child items — selects the displayed child of the currently selected items.

Select child items (recursive) — select all displayed children of currently selected items as well as their descendants.

Unselect child items — unselects all displayed children of currently selected items.

Unselect child items (recursive) — unselects all displayed children of currently selected items. This differs from a non-recursive form by affecting the expand state of deeper child levels as well as the selected items.

Select similar — selects all displayed items that have a similar item type and name as the item right-clicked on.

Highlight view port selection — all items selected in the viewport will become selected in the **Scene Editor**.

### Hierarchy:

**Expand channels** — This will show all channels for the selected items.

**Collapse channels** — This will hide all channels for the selected items.

**Expand child items** — This will make the children for all selected items visible.

**Expand child items (recursive)** — This will make all descendants of the selected item as well as their descendants visible.

**Collapse child items** — This will hide the children of all selected items.

**Collapse child items (recursive)** — This will collapse all descendants of the selected items. This differs from a non-recursive form by affecting the expand state of deeper child levels as well as the selected items.

### Manipulation:

**Motion property panel** — opens the Motion Options Panel for the selected item.

**Item property panel** — opens the Properties Panel for the selected item.

**Edit Channels** — opens the **Graph Editor Panel** with all the channels of the selected items ready to be edited.

**Rename:** Triggers a dialogue box to rename a single item. Renaming of multiple items at once is not supported.

**Add:**

**Add null object:** Activates the Null Object Name panel and adds a new null.

**Add object:** Activates the Load Object panel.

**Add object layer:** Activates the Load Object Layer panel.

**Add bone:** adds a single bone to either an object item or as a child to an existing bone item. The context determines which.

**Add camera:** allows adding a single camera to the scene.

**Add distant light:** Activates the Light Name panel and adds a single 'distant' type light to the scene.

**Add point light:** Activates the Light Name panel and adds a single 'point' type light to the scene.

**Add spotlight:** Activates the Light Name panel and adds a single 'spotlight' type light to the scene.

**Add linear light:** Activates the Light Name panel and adds a single 'linear' type light to the scene.

**Add area light:** Activates the Light Name panel and adds a single 'area' type light to the scene.

**Clone:** allows cloning of any item or selection of items

**Clear:**

**Clear>Remove/Clear:** Removes the selected items from the scene.

**Clear>Clear all objects:** Removes all objects from the scene.

**Clear>Clear all lights:** Removes all lights from the scene.

**Clear>Clear all cameras:** Removes all cameras from the scene.

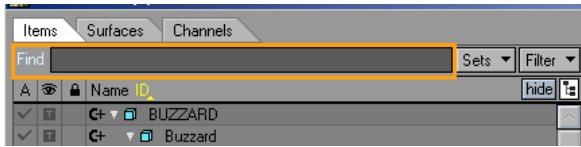
**Clear>Clear all bones:** Removes all bones from the scene.





## Item Searching

The **Find** tool at the top of the item view list is an option to perform item name and channel name searches. Enter a simple search string (the case does not matter), press the **Enter** or **Return** key, and the search begins.

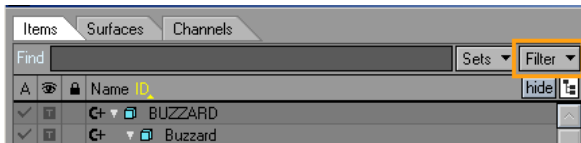


If a match is found, the item cursor is placed there and any list scrolling or item expanding necessary to view the matched item will be performed as well. It may also be useful to have the perspective camera (or a special camera) change to view that item.

That may be accomplished indirectly by selecting the found item while the object centring feature is enabled in layout, which puts the selected item into the perspective camera's view. The search iterates through each list item (regardless of what type it is) starting with the current item list cursor location, progressing to the list end, wrapping around to the list beginning, and winding up at the iteration starting point. The iteration is aborted once a string match is found. Filtered-out items do not get searched. If the search string used has more than one instance, all instances are selected.

### Filter

At the top of item view is a **Filter** option drop down button. These options allow the 'weeding out' of unwanted items and channels.



The drop down menu has the following options:

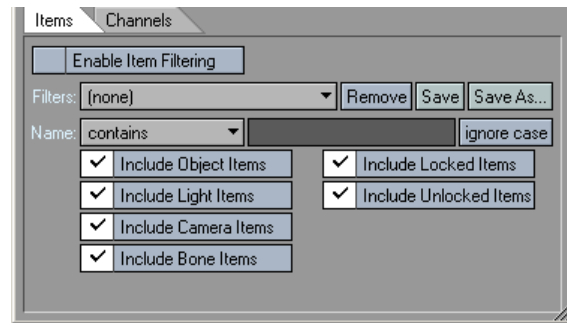
Options... — This brings up a separate, but modeless, **Options Panel** used for selecting the filtering criteria for both items and channels. With the **Options Panel** previously used filtering criteria can be recalled easily as well. Making changes to the criteria can be viewed in the item list as the criteria changes.

Enable/Disable Item Filter (Boolean) — Toggle to affect if item filtering is used on Object, Light, Camera, and Bone items.

Enable/Disable Channel Filter (Boolean) — Toggle to affect if channel filtering is used on Channel items.

The **Options Panel** has a section for item criteria and a section for channel criteria. Checkboxes that enable/disable the filtering are duplicated here so that the dropdown toggles may be used in addition to these.

### Filter Option Panel: Items Tab



For **Items** criteria, the following controls exist:

Enable Item Filtering (Boolean) — Toggle to affect if item filtering is used on Object, Light, Camera, and Bone items.

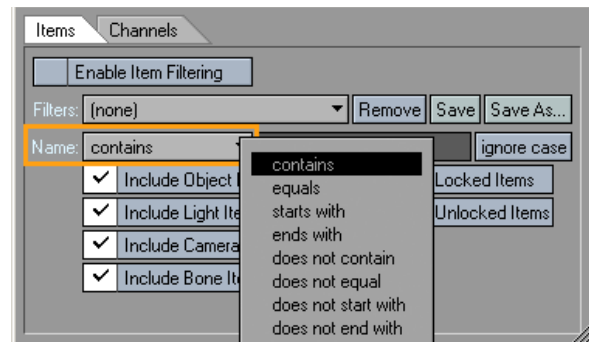
Stored filters dropdown — choose any stored filter. Once selected, it is loaded. You only need save filter settings you wish to use over and over again.

Remove — this will remove the filter that is currently selected, if one exists.

Save — this will overwrite the current filter choice with the current settings. If no filter is currently selected, this button acts as a 'Save As...' button.

Save As... — click to save the current filter criteria to a filter name. The filter name is requested after clicking this button.

Contains — This determines how to use the item name string. Choices include: contains, equals, starts with, ends with, does not contain, does not equal, does not start with, does not end with.



Item name string field — Any string can be entered to be used to match against all Layout items (not just those currently in the item list).

Ignore Case — enable and disable case sensitive compares.

Include Object Items — enable to allow object items

Include Bones Items — enable to allow bone items

Include Lights Items — enable to allow light items

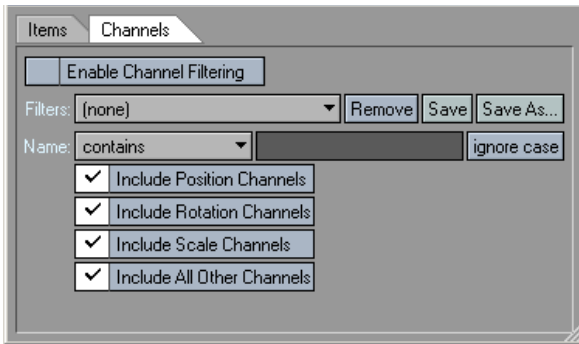
Include Cameras Items — enable to allow camera items

Include Locked Items — enable to allow locked items

Include Unlocked Items — enable to allow unlocked items



## Filter Option Panel: Channels Tab



For channel criteria, the following controls exist:

**Enable Channel Filtering (Boolean)** — Toggle to affect if channel filtering is used on Channel items.

**Stored filters dropdown** — choose any stored filter. Once selected, it is loaded. You only need **Save Filter** settings you wish to use over and over again.

**Remove** — this will remove the filter that is currently selected, if one exists.

**Save** — this will overwrite the current filter choice with the current settings. If no filter is currently selected, this button acts as a 'Save As...' button.

**Save As...** — click to save the current filter criteria to a filter name. The filter name is requested after clicking this button.

**Channel name string field** — Any string can be entered to be used to match against all channels (allowed)

**Contains** — This determines how to use the channel name string. Choices include: contains, equals, starts with, ends with, does not contain, does not equal, does not start with, and does not end with.

**Ignore** — This is used to enable case sensitive compares.

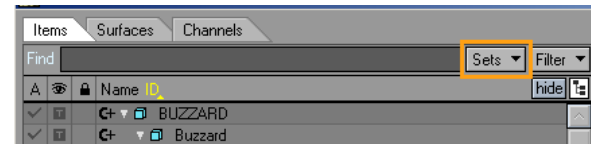
**Include Position Channels** — enable to allow position channels, which include 'Position.X', 'Position.Y', and 'Position.Z'

**Include Rotation Channels** — enable to allow orientation channels, which include 'Rotation.H', 'Rotation.P', 'Rotation.B'.

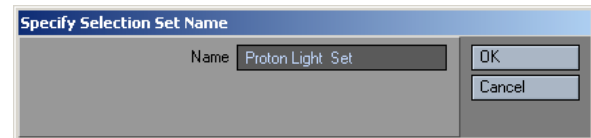
**Include Scale Channels** — enable to allow scaling channels, which include 'Scale.X', 'Scale.Y', 'Scale.Z'.

**Include Other Channels** — enable to include any channels that are not the standard position, rotation, or scaling channels

## Scene Editor Selection Sets



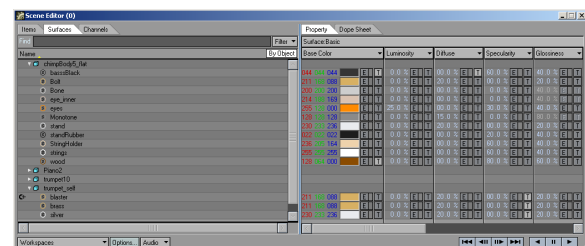
The currently selected items, which may include objects, lights, cameras, bones, channels, and groups of channels, can be remembered for later use via a **Selection Set**. This simply allows the current selection to be saved and later recalled. Selecting **Save Selection** will bring up the **Specify Selection Set Name Panel**.



Once a name has been entered, click OK and the selection set will be added to the **Sets** drop down menu. Select the set name from the list when you would like to recall that selection. Select **Remove Set** to delete a selection set from the list.

## Surface List View: Surfaces Tab

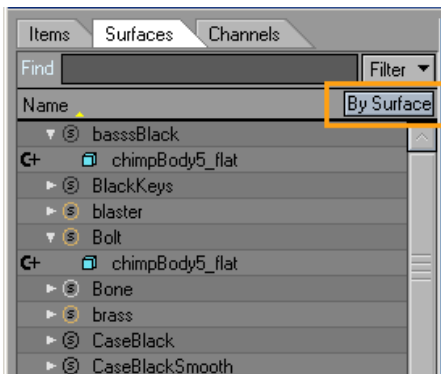
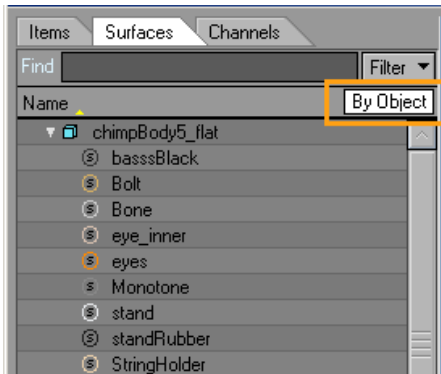
This view displays the names and ownership of surfaces currently loaded in Layout. Surfaces always belong to objects. Some of the same behaviours that apply to the **Item List View** will also apply here.





## Surface List Display

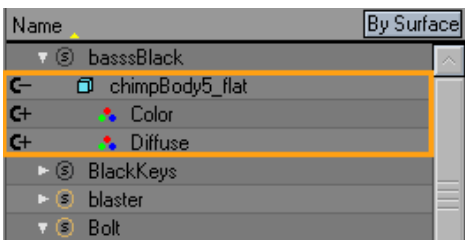
Surfaces may be displayed 'By object' or 'By surface'. **Display By Object** groups all surfaces belonging to a single object as children of that object. **Display By Surface** groups all objects that contain a surface with the exact same name as children of the surface.



Either way, highlighting a surface will also select that surface in the **Surface Editor** window.

The method of display is indicated within the column heading of the list as either '(By Object)' or '(By Surface)'.

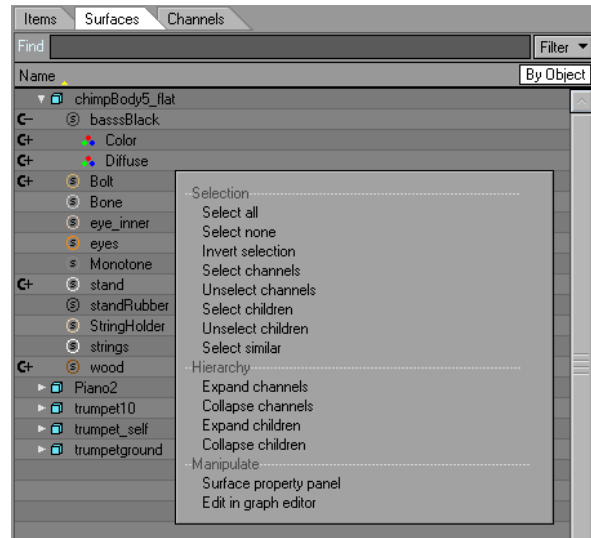
Surface items may also contain channels and groups of channels. For such surfaces, a channel expand icon will appear in that row and will operate as it does in the **Item List View**.



There is only one column in this list, the **Name** column. This column is sorted alphabetically by name, but can be reverse sorted as well by clicking on the column heading. A horizontal scroll bar located at the list bottom accommodates rows that are too wide to be fully seen.

## Surface List Right Click Menu

RMB-click on a surface row presents context menu options.



The options are the same as for:

### Selection:

Select all — This selects all displayed surfaces (after filtering).

Select none — This unselects all displayed surfaces.

Invert selection — This will make all (displayed) surfaces switch their selection state.

Select channels — selects the displayed channels of selected items.

Unselect channels — unselects all channels belonging to currently selected items.

Select children — selects all displayed children of currently selected surface groups.

Unselect children — unselects all displayed children of currently selected surface groups.

Select similar — selects all displayed surfaces that have a similar name as the item right-clicked on.

### Hierarchy:

Expand channels — This will show all channels for the selected surfaces.

Collapse channels — This will hide all channels for the selected surfaces.

Expand items — This will make the children for all selected surfaces visible.

Collapse items — This will hide the children of all selected surfaces.

### Manipulation:

Surface property panel — opens the Surface Editor.

Edit in graph editor — opens the **Graph Editor Panel** with all the channels of the selected surfaces ready to be edited.



## Filter Option Panel: Surfaces Tab

Surfaces and channels can be filtered much like they can for the **Item List View**. It is a way to prevent certain surface items from appearing in the list. The surface filtering is available via a drop down menu button at the top of the list:

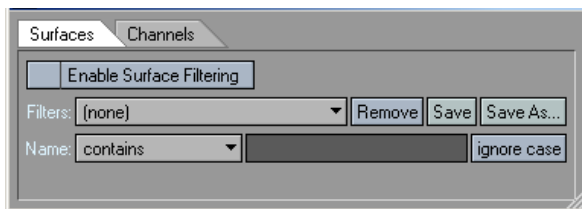
Options... — opens a panel used to select surface item and surface channel filtering options. With the **Options Panel** previously used, filtering criteria can be recalled easily as well. Making changes to the criteria can be viewed in the item list as the criteria changes.

Enable/Disable Surfaces Filter — Toggles filtering of surface items

Enable/Disable Channels Filter — Toggles filtering of surface channels

The **Options Panel** has a section for surface criteria and a section for channel criteria. Checkboxes that enable/disable the filtering are duplicated here so that the dropdown toggles may be used in addition to these.

## Surface Filter Options Panel: Surfaces Tab



For Surface criteria, the following controls exist:

Enable Surfaces Filtering — Toggles filtering of surface items.

Stored filters dropdown — choose any stored filter. Once selected, it is loaded.

Save — this will overwrite the current filter choice with the current settings. If no filter is currently selected, this button acts as a 'Save As...' button.

Save As — click to save the current filter criteria to a filter name. The filter name is requested after clicking this button.

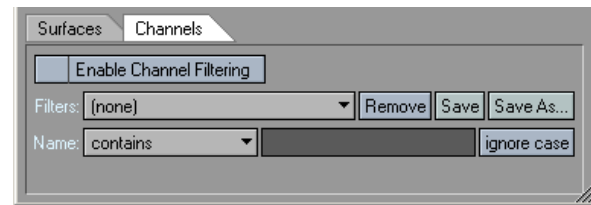
Remove current filter button — click to remove the currently selected filter button. A confirmation requester is presented first.

Contains — This determines how to use the string. Choices include: contains, equals, starts with, ends with, does not contain, does not equal, does not start with, does not end with.

Name String Field — Any string can be entered to be used to match against all Layout surfaces (not just those currently in the item list).

Ignore Case — This is used to enable case sensitive compares.

## Surface Filter Options Panel: Channels Tab



For channel criteria, the following controls exist:

Enable Channels Filter — Toggles filtering of surface channels.

Stored filters dropdown — choose any stored filter. Once selected, it is loaded.

Save — this will overwrite the current filter choice with the current settings. If no filter is currently selected, this button acts as a 'Save As...' button.

Save As — click to save the current filter criteria to a filter name. The filter name is requested after clicking this button.

Remove current filter button — click to remove the currently selected filter button. A confirmation requester is presented first.

Channel name string field — Any string can be entered to be used to match against all channels (allowed).

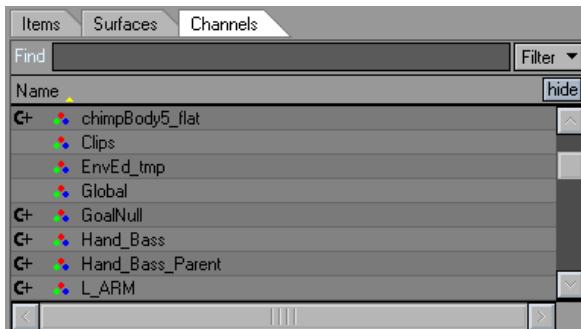
Contains — This determines how to use the channel name string. Choices include: contains, equals, starts with, ends with, does not contain, does not equal, does not start with, does not end with.

Ignore Case — This is used to enable case sensitive compares.



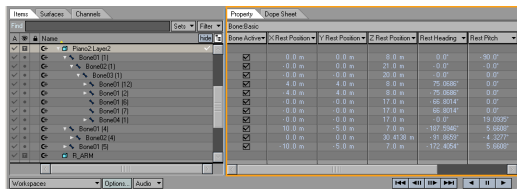
## Channels List View: Channels Tab

This view displays the names and ownership of channels used through the entire scene. This can include channels that are not normally visible by other means.



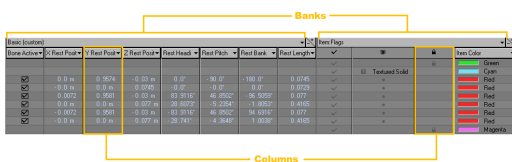
NOTE: The operation of this list is similar to that of the other list views. See Item List and Surface List View documentation above for more info.

## Spread Sheet Property Panel



Often you want to edit certain properties of an item, and luckily, changing the values and properties of items is simple in Layout. But what if you need to change the properties for 20 objects? The **Property Panel** organizes these properties and lets you edit a wide range of values quickly and easily.

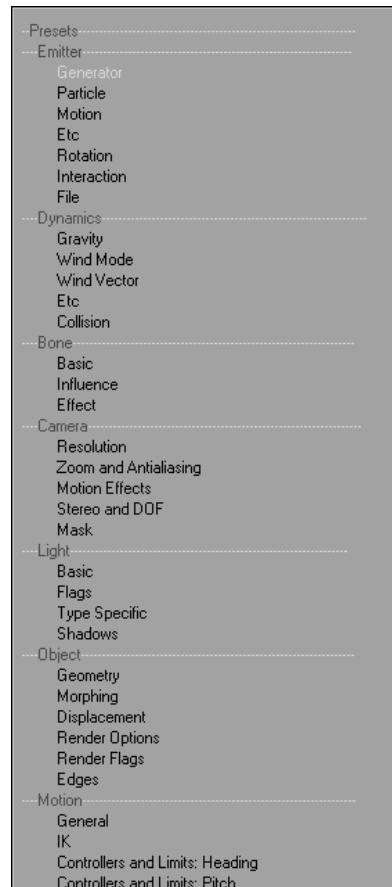
The **Spread Sheet Property Panel** consists of two sections, The **Bank** section and the **Column** section.



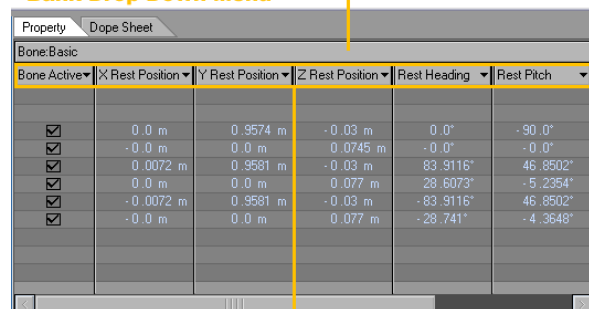
## Property Panel Banks

A collection of columns is called a **Bank**; and each bank can hold any number of columns. Some predefined banks exist that help to organize the vast multitude of properties available. Custom banks, which allow you to define which columns make up the bank, are also supported.

LightWave ships with a large collection of Bank Presets listed below:



### Bank Drop Down Menu

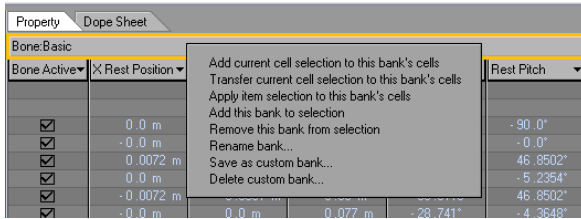


### Column Drop Down Menus



## Bank Options Menu

Right clicking on the **Bank** drop down menu will bring up the **Bank Options Menu**.



Add current cell selection to this bank's cells — This will use any selected cell position and apply it to the other columns in that bank for that same row. Only cells of matching type as already selected cells will be selected.

Transfer current cell selection to this bank's cells — this works like the 'add current cell...' except that transfer, clears out the current selection, so that the first type of cell encountered in the bank will be used.

Apply item selection to this bank's cells — This looks at highlighted items on the left side and tries to select cells in the same rows as those items in all the columns of the bank.

However, only columns that are of similar types can have cells selected. For example, a check box type cell and a numeric type cell cannot be edited at the same time.

A	Name	hide	X Rest Posit	Y Rest Posit	Z Rest Posit
✓	C+ Camera				
✓	C+ IKB_AnimeChic				
✓	C+ Body (3)		0.0 m	0.9574	-0.03 m
✓	C+ Body (4)		-0.0 m	0.0 m	0.0745
✓	C+ LeftLeg (4)		0.0072	0.0072	0.0072
✓	C+ LeftLeg (5)		0.0 m	0.0 m	0.077 m
✓	C+ RightLeg (4)		-0.0072	0.9581	-0.03 m
✓	C+ RightLeg (5)		-0.0 m	0.0 m	0.077 m

A	Name	hide	X Rest Posit	Y Rest Posit	Z Rest Posit
✓	C+ Camera				
✓	C+ IKB_AnimeChic				
✓	C+ Body (3)		0.0 m	0.9574	-0.03 m
✓	C+ Body (4)		-0.0 m	0.0 m	0.0745
✓	C+ LeftLeg (4)		0.0072	0.9581	-0.03 m
✓	C+ LeftLeg (5)		0.0 m	0.0 m	0.077 m
✓	C+ RightLeg (4)		-0.0072	0.9581	-0.03 m
✓	C+ RightLeg (5)		-0.0 m	0.0 m	0.077 m

Add this bank from selection — attempts to select all cells in the bank.

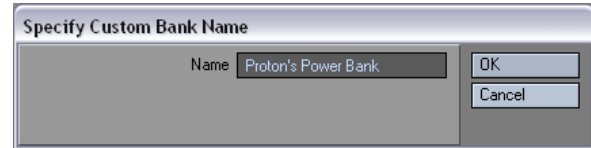
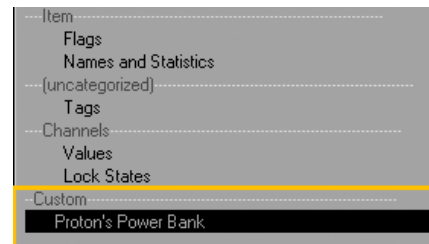


NOTE: The cell types must match.

Remove this bank from selection — Unselects all cells in the bank

Rename bank — This will rename the current bank for this instance of bank existence.

Save as custom bank — once any arrangement of columns (including resizing) are made, this 'custom' bank can be named and saved for future use.



Delete custom bank — deletes the custom bank preset. It will not remove a bank instance based off that custom preset, just the preset so that it will not show up as a bank preset.

## Sizing Banks

A bank can be resized using the right-most vertical line. In addition, using the upper portion of the vertical line (the part where the bank name row is) will proportionally size the columns in the bank, while it would normally just affect the last column.

## Adding and Removing Banks

To add a bank, simply click the **Add Bank** button (+) located at the end of the bank name. To remove a bank, click the **Remove Bank** button (-) located under the **Add Bank** button.



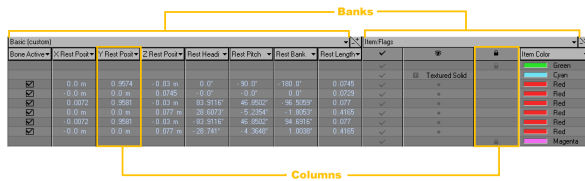
Use the slider-bar at the bottom of the **Property Cells** to adjust which bank(s) are currently displayed in the viewing area.





## Property Panel Columns

A **Column** consists of **Cells**. Each column in the **Scene Editor** is a property instance having a specific cell type. The Property cells contain the values for each of the properties, for all items. The values range from numerical values to file names, depending on which property you are editing.

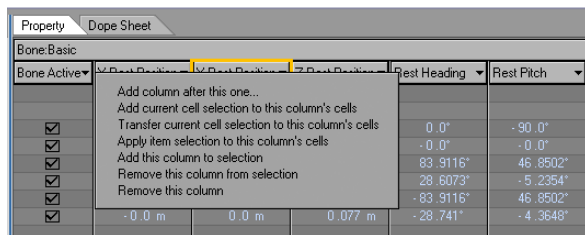


There are many column types available to choose from. Left click on the column header to select the Cell Type for that column.



## Column Options Menu

Right clicking on a **Column** drop down menu will bring up the **Column Options Menu**.



Add column after this one — This will create a new column to the right of the selected column.

Add current cell selection to this column's cells — This will select the same cells that are currently selected in another column and apply that selection to the selected column.

Transfer current cell selection to this column's cells — This will select the same cells that are currently selected in another column and apply that selection to the selected column. It will then remove the selected cells in the previous column.

X Rest Position	Y Rest Position	Z Rest Position
0.0 m	0.9574 m	-0.03 m <sup>1</sup>
-0.0 m	0.0 m	0.0745 m
0.0072 m	0.9581 m	-0.03 m <sup>2</sup>
0.0 m	0.0 m	0.077 m
-0.0072 m	0.9581 m	-0.03 m
-0.0 m	0.0 m	0.077 m <sup>3</sup>

X Rest Position	Y Rest Position	Z Rest Position
0.0 m <sup>1</sup>	0.9574 m	-0.03 m
-0.0 m	0.0 m	0.0745 m
0.0072 m <sup>2</sup>	0.9581 m	-0.03 m
0.0 m	0.0 m	0.077 m
-0.0072 m	0.9581 m	-0.03 m
-0.0 m <sup>3</sup>	0.0 m	0.077 m

Apply item selection to this column's cells — This will select the cells in the selected column based on the items selected.

A	Name	hide	Bone Active	X Rest Posit
✓	C+ Camera			
✓	C+ IKB_AnimeChic			
✓	C+ Body (3)		✓	0.0 m
✓	C+ Body (4)		✓	-0.0 m <sup>1</sup>
✓	C+ LeftLeg (4)		✓	0.0072
✓	C+ LeftLeg (5)		✓	0.0 m <sup>2</sup>
✓	C+ RightLeg (4)		✓	-0.0072 <sup>3</sup>
✓	C+ RightLeg (5)		✓	-0.0 m

A	Name	hide	Bone Active	X Rest Posit
✓	C+ Camera			
✓	C+ IKB_AnimeChic			
✓	C+ Body (3)		✓	0.0 m
✓	C+ Body (4)		✓	-0.0 m
✓	C+ LeftLeg (4)		✓	0.0072
✓	C+ LeftLeg (5)		✓	0.0 m
✓	C+ RightLeg (4)		✓	-0.0072
✓	C+ RightLeg (5)		✓	-0.0 m

Add this column to selection — This will select the cells that make up this column to add them to the selection.

Z Rest Posit	Rest Head	Rest Pitch
-0.03 m <sup>1</sup>	-1.7189 <sup>31</sup>	-1.7189 <sup>31</sup>
-0.03 m <sup>2</sup>	-1.7189 <sup>32</sup>	-1.7189 <sup>32</sup>
-0.03 m <sup>3</sup>	-1.7189 <sup>33</sup>	-1.7189 <sup>33</sup>
-0.03 m <sup>4</sup>	-1.7189 <sup>34</sup>	-1.7189 <sup>40</sup>
-0.03 m <sup>5</sup>	-1.7189 <sup>35</sup>	-1.7189 <sup>41</sup>
-0.03 m <sup>6</sup>	-1.7189 <sup>36</sup>	-1.7189 <sup>42</sup>

Z Rest Posit	Rest Head	Rest Pitch
-0.03 m <sup>1</sup>	0.0°	-1.7189 <sup>31</sup>
-0.03 m <sup>2</sup>	-0.0°	-1.7189 <sup>32</sup>
-0.03 m <sup>3</sup>	83.9116°	-1.7189 <sup>33</sup>
-0.03 m <sup>4</sup>	28.6073°	-1.7189 <sup>34</sup>
-0.03 m <sup>5</sup>	-83.9116°	-1.7189 <sup>35</sup>
-0.03 m <sup>6</sup>	-28.741°	-1.7189 <sup>36</sup>

Remove this column from selection — This will unselect the cells from the selected column.

Z Rest Posit	Rest Head	Rest Pitch
-0.03 m <sup>1</sup>	0.0°	-1.7189 <sup>31</sup>
-0.03 m <sup>2</sup>	-0.0°	-1.7189 <sup>32</sup>
-0.03 m <sup>3</sup>	83.9116°	-1.7189 <sup>33</sup>
-0.03 m <sup>4</sup>	28.6073°	-1.7189 <sup>34</sup>
-0.03 m <sup>5</sup>	-83.9116°	-1.7189 <sup>35</sup>
-0.03 m <sup>6</sup>	-28.741°	-1.7189 <sup>36</sup>



Z Rest Posit▼	Rest Headi▼	Rest Pitch▼
- 0.03 m <sup>1</sup>	- 1.7189 <sup>31</sup>	- 1.7189 <sup>37</sup>
- 0.03 m <sup>2</sup>	- 1.7189 <sup>32</sup>	- 1.7189 <sup>38</sup>
- 0.03 m <sup>3</sup>	- 1.7189 <sup>33</sup>	- 1.7189 <sup>39</sup>
- 0.03 m <sup>4</sup>	- 1.7189 <sup>34</sup>	- 1.7189 <sup>40</sup>
- 0.03 m <sup>5</sup>	- 1.7189 <sup>35</sup>	- 1.7189 <sup>41</sup>
- 0.03 m <sup>6</sup>	- 1.7189 <sup>36</sup>	- 1.7189 <sup>42</sup>

Remove this column — This will delete the selected column from the bank.

## Sizing Columns

Left click on the vertical dividers between columns to size a column.

## Property Panel Cells

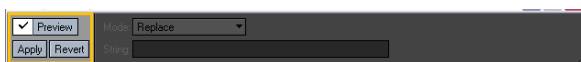
Each property has a cell and cell type depending on the type of data being manipulated. Each column in the **Scene Editor** is a property instance having a specific cell type. The rows are the items shown in the list view. Cells may be edited via a property editing panel, which allows greater manipulation flexibility when it comes to changing multiple cells at the same time.

Bone: Basic				
Bone Active▼	X Rest Position▼	Y Rest Position▼	Z Rest Position▼	Rest Heading▼
<input checked="" type="checkbox"/>	0.0 m	0.9574 m	- 0.03 m	0.0°
<input checked="" type="checkbox"/>	- 0.0 m	0.0 m	0.0745 m	- 0.0°
<input checked="" type="checkbox"/>	0.0072 m	0.9581 m	- 0.03 m	83.9116°

Cells

## Cell Types

Each cell type has its own unique **Adjust Properties Panel**, but they all share the **Preview**, **Apply**, and **Revert** options.



Preview — This option allows you to see updates to the **Scene Editor** in real time.



NOTE: This is a preview for the **Scene Editor Only**.

Apply — This will apply any changes made to the cell property.

Revert — This will undo the last change applied to a cell property.

## Unique Cell Type Adjust Properties:

Toggle — Check Box



Check — This will activate the option.

Uncheck — This will deactivate the option.

Toggle — This will toggle between active and inactive.



float.

Change Value To — Value that will be applied to the cell property.

Change Value Effect — Determines how the value will be applied.

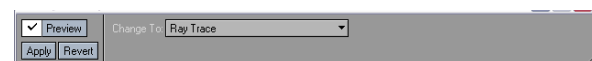
The **Replace** edit mode lets you make absolute changes to the cell's values. That is, when you apply the edit to the selected cells, the new value will replace the current values. For example, you select ten cells with various values in each cell, and in the **Change Value To** field you enter a value of 2.0. When the edit is **Previewed** or **Applied**, the cells will now all read 2.0.

The **Add**, **Subtract**, **Multiply**, and **Divide** functions are all relative editing modes. They adjust the cell's values relative to their original values.

Use Step — This will add the Step value to the primary value incrementally according to the order in which the cells have been selected.

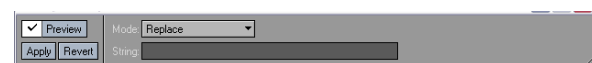
Step Effect — Determines how the Step Value will be applied.

List — Array of items like a drop down menu.



Change To — This will replace the cell property with the chosen option.

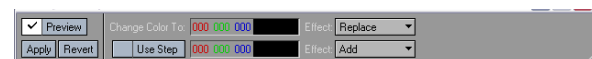
String — Strings used for item tag properties.



Mode — Determines how the string will be applied to the cell property.

String — The text or comment to be applied to the cell property.

Color — RGB color values where each component is 8 bits (not used for high dynamic lighting).



Change Color To — RGB value that will be applied to the cell property.

Change Color Effect — Determines how the RGB value will be applied.

The **Replace** edit mode lets you make absolute changes to the cell's values. That is, when you apply the edit to the selected cells, the new value will replace the current values. For example, you select ten cells with various values in each cell, and in the **Change Value To** field you



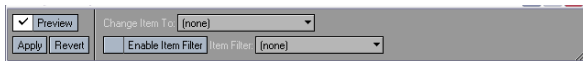
enter a value of 2.0. When the edit is **Previewed** or **Applied**, the cells will now all read 2.0.

The **Add**, **Subtract**, **Multiply**, and **Divide** functions are all relative editing modes. They adjust the cell's values relative to their original values.

**Use Step** — This will add the Step value to the primary value incrementally according to the order in which the cells have been selected

**Step Effect** — Determines how the Step Value will be applied.

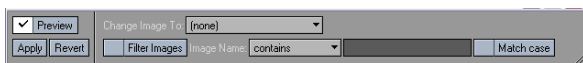
**Item** — layout item that is used for parent item, target item, etc...



**Change Item To** — The item that will be applied to the cell property.

**Enable Item Filter** — For scenes with many items, use the filter option to limit your selection options.

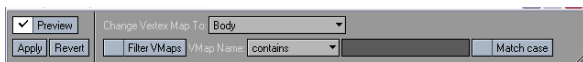
**Image** — Images used for items like Projection image.



**Change Image To** — The image that will be applied to the cell property.

**Enable Item Filter** — For scenes with many images, use the filter option to limit your selection options.

**Vertex Map** — Vertex Maps used for items like Bone Weights.

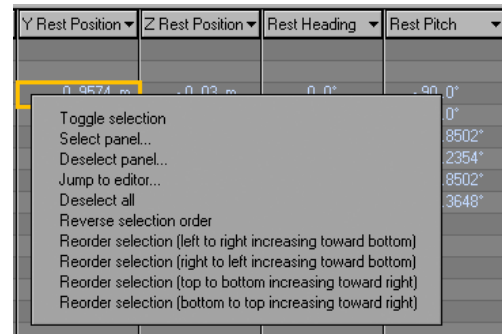


**Change Vertex Map To** — The **VMap** that will be applied to the cell property.

**Enable Item Filter** — For scenes with many **VMaps**, use the filter option to limit your selection options.

## Cell Options Menu

Right clicking on a cell brings up the **Cell Options Menu**.

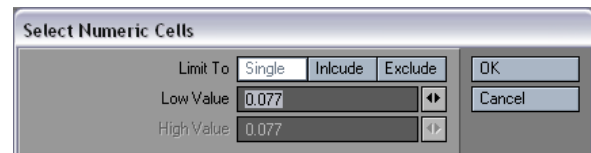


**Toggle Selection** — This will select and deselect the cell.

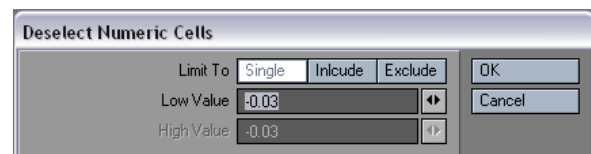


**NOTE:** **Ctrl + LMB** will toggle selection as well.

**Select Panel** — This panel gives you the ability to numerically select cells.



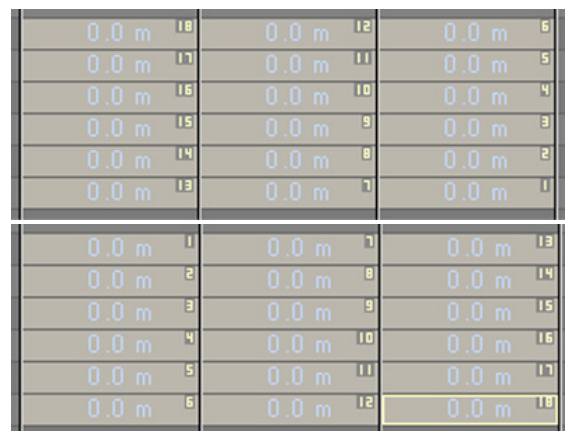
**Deselect Panel** — This panel gives you the ability to numerically deselect cells.



**Jump to Editor** — It is possible for a property cell to open a custom editor. The standard properties don't use this function but it is possible for third party tools to add properties that do.

**Deselect all** — This will deselect all cells.

**Reverse selection order** — This function will reverse the selection order of the cells that are currently selected.

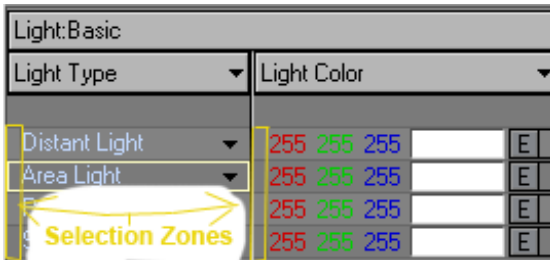


Other reorder options are **Reorder selection** (left to right increasing toward bottom), **Reorder selection** (right to left increasing toward bottom), **Reorder selection** (top to bottom increasing toward bottom), and **Reorder selection** (bottom to top increasing toward bottom).

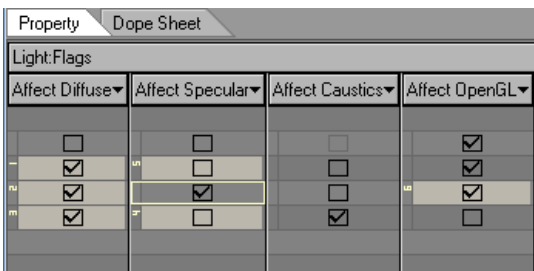


## Selection Zone

When a cell is available for multi-selection, it will be divided in two parts: on the left will be the Selection Zone and on the right is the Value Zone. Left clicking and dragging in the Selection Zone of a cell will add the cell to the selection set.



Shift-left clicking while selecting a cell will mark the corner of a selection box, expanding the selection box further out the more cells are selected. Control-left clicking will toggle on or off the selected cell. The cells are numbered in the order they are selected.

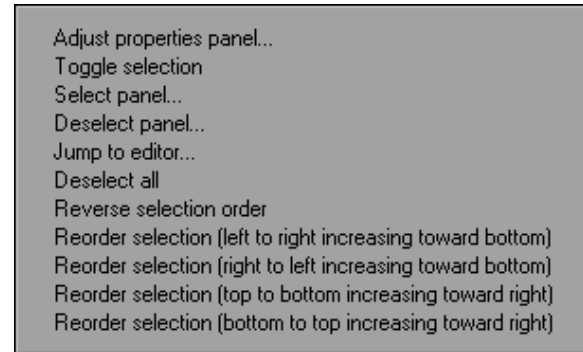


Note: If you are using the stepping features in the "Adjust Properties" panel, order of selection will be important. Also, Adjust Properties is defaulted with

Preview off to allow you to manipulate the values before seeing their effect on the cells. The Preview button will also be turned off after clicking the Apply button

## Selection Zone Right Click Menu

Right clicking in the Selection Zone will bring up a new menu. This menu has options for changing the selection order. The Adjust Properties panel is also accessible from this menu.



Note: String values cannot be added directly in the cell, but must be entered in the Adjust Properties Panel.



Hint: You can hide the Scene Editor by hitting the Tab key.



Note: If you have a Bank and Column selection that does not apply to the item type, the cell will not be selectable at all. The Bank and Column sections will now change the selected choice a different color instead of hiding it when the menu is activated.

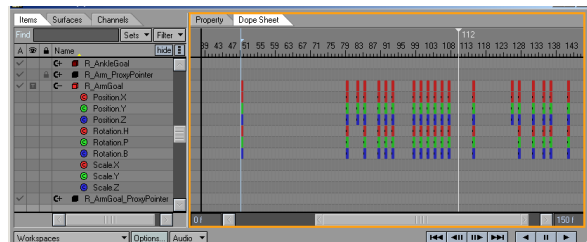


Hint: If you have multiple cells selected and would like to only change a value for one, hold the CTRL key while changing the value.



## Scene Editor: Dope Sheet

The **Dope Sheet** is a form of scratch sheet with which the animation of channel data (position, rotation, scale, etc.) can be manipulated without going into the details of what the exact channel values are. This gives an overview of the scene animation and is useful when the multiple items need to be viewed simultaneously so that their relative timing is correct. This view also allows easy manipulation of a large number of channels in one operation. It allows time offsetting and time scaling of channel data.



It can also accelerate the process of cutting out portions of motion or transferring motion from an item or channel to other items or channels. This view basically offers a method of manipulating key frame existence without worrying about the exact value of each key frame; one of **Graph Editor's** purposes in Layout.

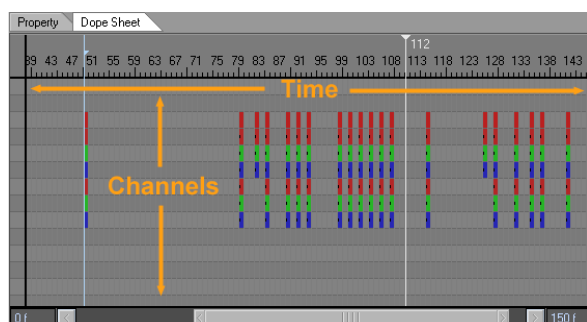


NOTE: The Dope Sheet does not modify key frame values.

The Dope Sheet is a grid, or 2D array, in appearance. The horizontal dimension is time measured in blocks. Each block represents a range of time, which is user adjustable in the **Options Panel**. The horizontal block visible width has a pixel width, which can be adjusted to see more time blocks or fewer within the visible portion of the sheet.

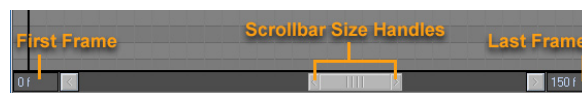


NOTE: There is a minimum pixel width of three pixels for each block.



The vertical dimension is measured in rows. Each row represents a single channel of key frame data, a group of channels, or nothing. The height of each row is predetermined and matches the row height of the **Item List View** rows and **Surface List View** rows. Each channel that contains at least one key frame will have its appropriate block colored differently. Similarly, an item that contains one or more channels (i.e. a channel group, but may be any row that has channel children) with a colored block will have its own block colored, differently from the color of the channel block.

The range of time that the dope sheet will allow viewing of is determined by the preview first and last endpoints set within Layout. The scrollbar will then allow viewing within that frame range. There are numeric controls within the dope sheet at the bottom, which also allow setting the preview first and last frames. These endpoints are shared with the ones in Layout.

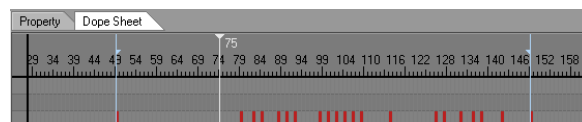


The scrollbar has sizing handles on both sides. These are used to adjust the block pixel width so that more or fewer blocks are visible at once. Minimum and maximum pixel widths exist as well.



NOTE: The horizontal scrollbar has a size limit so that it does not get too small to grab hold of.

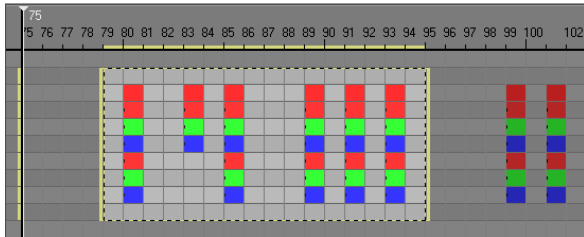
At the top of the sheet is a timeline tick marker display showing what each block represents. A time code is displayed for each block for which there is room to display a time code.





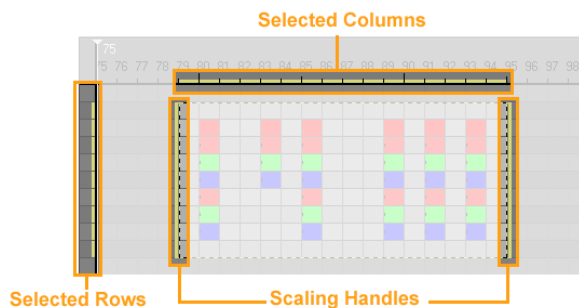
## Selecting Blocks

Blocks may be selected by using the Left mouse to drag a selection box around the blocks of interest. The selection box starts in one corner of a block and ends in a corner of a block. The box will snap to the nearest block boundaries and will indicate a selection of contiguous time and contiguous rows.

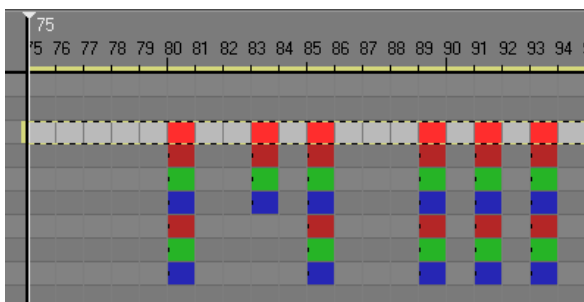


**NOTE:** During selection, the box may extend beyond the visible portions of the sheet; this will cause the dope sheet to scroll hidden areas into view.

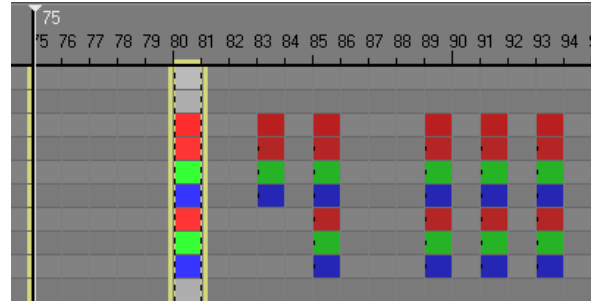
The left edge of the dope sheet contains an empty area to show which rows (items) are selected in the dope sheet. Rows that contain some selected blocks appear differently. The top of the dope sheet shows which columns have selected blocks. The selection box itself is an outline of the selected blocks. The left and right edges of this box have thin vertical handles used for time scaling operations.



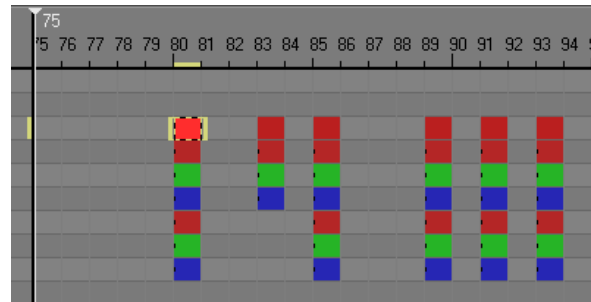
Clicking the **LMB** in the left edge of the Dope Sheet will select the entire row for all columns. It is important to note that the dope sheet block selection is independent of an item list selection or surface list selection.



Clicking the **LMB** in the top of the dope sheet (where the time line tick marks are) will select the blocks in that column for all rows shown. This does not include hidden rows or filtered items.



Clicking the **LMB** inside a single block will select that one block.



To add to an existing selection use the **Shift + LMB**.



**NOTE:** A new selection always cancels an existing selection.





## Manipulating Selections

Once a selection box has been formed, the contents may be manipulated. If a channel group is manipulated, all channels contained within (that are not filtered out) are manipulated even if the group is not expanded. When a range of time is manipulated, only the keys originally selected will be manipulated, even if the selection box now has moved as a result of an operation.

**LMB** and drag is used to move the time offset of a selection. Time offsets are applied block by block. If finer time resolution is needed, using the control-key modifier while dragging allows the keys to move in time with smaller increments.

**LMB** and dragging the left or right **Scale Handles** will allow time scaling. Using the left edge will use the selection box end time as a scale reference time. Using the right edge will use the selection box start time as a scale reference time.

The control key can be held down while mouse left-clicking. Doing so will allow the precise location of a key in time to be manipulated instead of whole block manipulations. This can be done when creating the selection box, moving a selection, and scaling a selection.

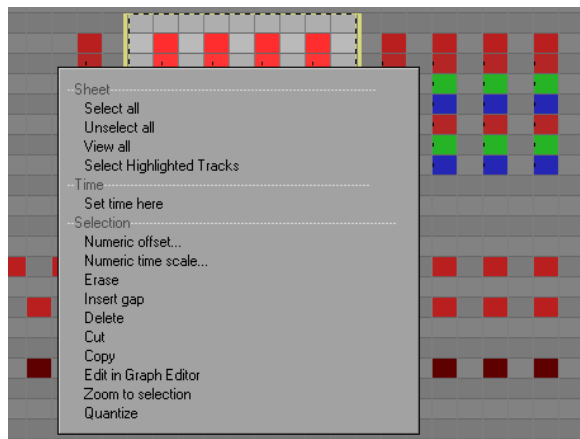


Holding the **Shift** key down while dragging, will 'reacquire' the keys to be manipulated. This means any key blocks that are currently highlighted AND inside the manipulation box

When you highlight some blocks and offset them so that now those original blocks are moved, only some keys may exist that are inside the highlighted blocks and the manipulation box. Hitting **Shift** while manipulating further will only affect those blocks that are both highlighted and inside that manipulation box.

## Selection Right Click Menu

Clicking the **RMB** inside the selection box will popup a context menu:



### Sheet:

Select all — select all blocks that the dope sheet will allow (all tracks all times within preview range)

Unselect all — unselect all blocks currently selected

View all — set the block pixel size such that the preview range can be seen in full.

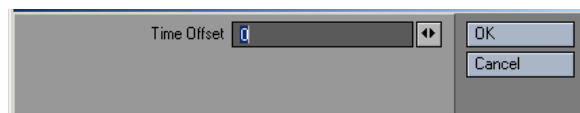
Select Highlighted Tracks — This will move the manipulation box to the highlighted blocks.

### Time:

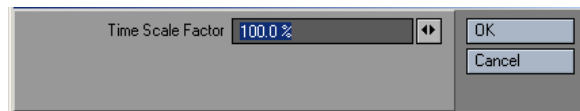
Set time here — set current time to the nearest frame to the current mouse position.

### Selection:

Numeric offset... — present dialog to offset the current selection in time.



Numeric time scale... — present dialog to scale the current selection in time.



Erase — Delete all keys within original selection. No keys After the selection will be shifted.

Insert Gap — Make the selection box a blank gap. Shift left edge to be right of right edge

Delete — destroys all keys within selection regardless of originally selected keys and shifts all keys after the selection right edge to the left.

Cut — Stores keys from original selection into a temporary buffer. Those keys are deleted. The selection box remains. Key times are stored relative to the selection left edge time. The time range is also stored so that the right edge of the buffer is not determined by the last key.



**Copy** — Works like cut without deleting the original keys.

**Paste Over** — Only available if there are keys stored in the temporary buffer. Places keys into the selection box starting with the left edge as a reference point. The selection box remains. Keys will not be pasted beyond the right edge of the selection box. Tracks are processed top to bottom regardless of where the stored keys came from. Existing keys in the selection box are left alone. Keys with the same time value will be overwritten. When pasting, it is possible that the number of rows copied will differ from the number of rows pasted to. When copy row count is less than the past row count, the copied rows are applied row by row and will cycle around again when the end is reached. In other words a copy count of two rows paste count of five will result in source row 1 being applied to target rows 1, 3, and 5; and source row 2 being applied to target rows 2 and 4. When the copy row count is greater than the paste row count, the excess copy rows will not be used.

**Paste insert** — like **Paste Over** except that all time from the left edge of the selection is moved forward in time in order to make room for the stored keys first. Then, the paste occurs ignoring the selection box left/right, but it does consider the top/bottom selection.

**Edit in graph editor** — opens the **Graph Editor** with the current selection box selected. This includes time range and tracks that may be selected.

**Zoom** — set the block pixel size such that the selection box time range occupies the visible portion of the sheet.

**Quantize** — This useful operation will make all keys move in time to the start of the block period they are contained within. This can help fix slight offset errors that may have crept in.

## Scene Editor Playback Controls

Besides dragging the frame slider to navigate through your scene, you can also use the transport controls located in the lower right corner of the **Scene Editor**. You can also play your scene backward (left arrow) or forward (right arrow) from the current frame using the playback controls.



## Keyboard Shortcuts

There are also some keyboard shortcuts that can be used:

Previous frame (**Left cursor** key)

Next frame (**Right cursor** key)

Previous keyframe (**Shift + Left cursor** key)

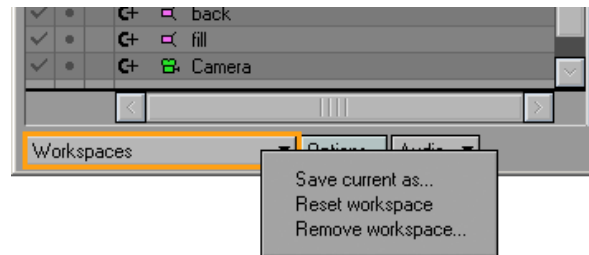
Next keyframe (**Shift + Right cursor** key)



**NOTE:** Your playback speed will vary depending on the complexity of your scene, object display mode, system capabilities and so on. Reducing the size of your Layout window can dramatically increase playback speed.

## Workspace

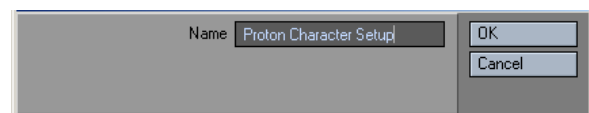
A **workspace** is a collection of user specific settings that can be stored and recalled quickly. Settings include: panel width and height and location, view dimensions within the panel, Item view selection, filter options, display options, Property view bank and custom bank choices, dope sheet settings, etc. These **Workspace** settings are stored on a per user basis. A user-specific configuration file is used. The initial workspace data to use is built-in as a default workspace.



The menu presented will show the existing workspaces available as well as options to save one, reset it, and remove one. Selecting a named workspace will transfer its settings to the current **Workspace** settings. Changes to the current workspace do not affect any stored workspaces unless that workspace is overwritten via the '**Save current as...**' option.

**Save current as...** — Allows user to pick a name for the workspace.

Using an existing workspace name will overwrite that workspace; otherwise, a new storage workspace is created.

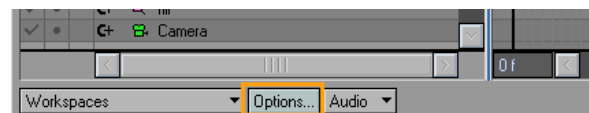


**Reset workspace** — The current **Workspace** settings (not any of the storage **Workspace** settings) are reset to their built-in values.

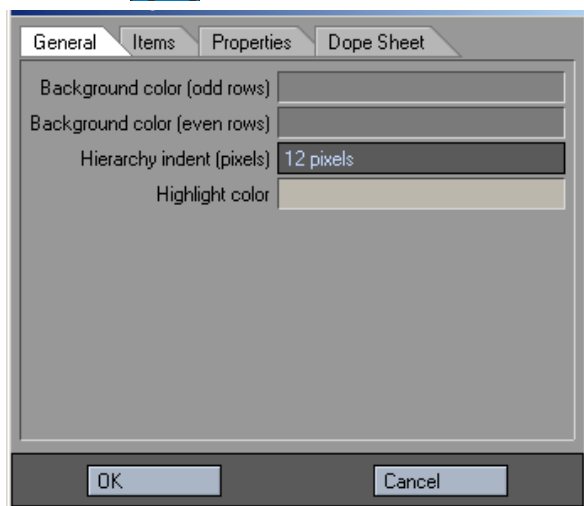
**Remove workspace...** — A pop-up menu displays the existing workspaces, which can be removed. Choosing one results in losing that stored workspace; this has no effect on the settings of the current workspace.

## Scene Editor Options

The **Scene Editor** options are categorised via the part of the GUI they affect: **General**, **Items**, **Properties**, and **Dope Sheet**. A separate panel is used to adjust these options, which are stored as part of the current workspace.



## General Options



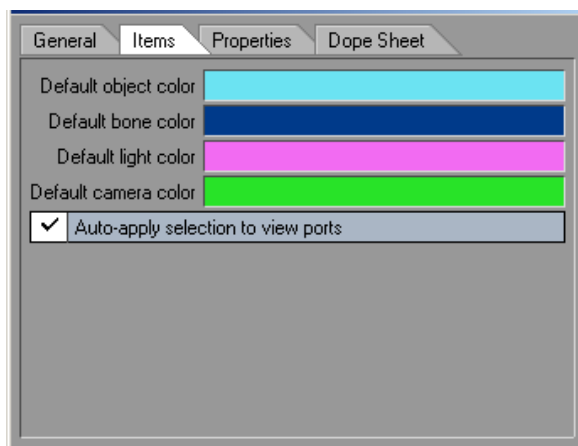
Background color (odd rows) — click to change color of odd numbered rows.

Background color (even rows) — click to change color of even numbered rows.

Hierarchy indent (pixels) — (unsigned short) set to number of pixels used as a gap for each additional child depth when displaying hierarchical data like layout items.

Highlight color (RGB) — color of background for highlighted items.

## Items Options



Default Object Color (RGB) — new objects are created in this color

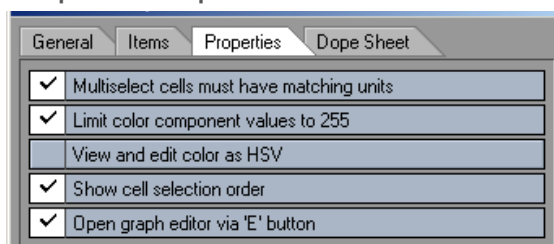
Default Bone Color (RGB) — new bones are created in this color

Default Light Color (RGB) — new lights are created in this color

Default Camera Color (RGB) — new cameras are created in this color

Auto-apply selection to viewports — This will select items in viewports as they are selected in the **Scene Editor**.

## Properties Options



Multiselect cells must have matching units — This is useful when selecting multiple channels and properties. When enabled, only those cells that share the same 'storage' units (e.g. length, angle, percentage) will be selectable. It may be useful to turn this off when you would like to select position, rotation, and/or scaling values simultaneously.

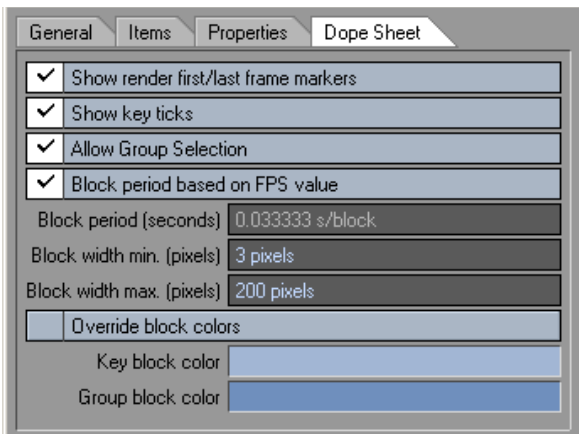
Limit color component values to 255 — When enabled, color values are limited to the 0-255 range. Some properties do interpret and benefit from color component values outside this range. (Light intensity for example)

View and edit color as HSV — This uses the Hue, Saturation, Value approach to displaying and editing colors within a cell.

Show cell selection order — When enabled, each cell will contain a small numeric indicator of what order that cell was selected. This can be useful when applying values that depend on the sequence of how the cells were selected in a multi-cell selection.

Open graph editor via 'E' button: (Boolean) — Normally, pressing on a cell's 'E' button will add an envelope and then open that envelope (channel) in the **Graph Editor**. When this option is disabled, the **Graph Editor** will not be altered from its current state. When working with many cells at once, it may become cumbersome to have the **Graph Editor** open up when you only wanted to add the envelope.

## Dope Sheet Options



**Show render first/last frame markers** — When enabled, the gadgets for the Render Start Frame and Render Stop Frame are presented and can be moved with the mouse by clicking on them with the left mouse button.

**Show Key ticks** — This will allow the precise location of a key in time to be displayed.

**Allow Group Selection** — determines if groups of channels can be selected and manipulated when collapsed. If disabled, only channels can be manipulated and never groups of channels. If enabled, a group of channels can be manipulated by simply manipulating the group blocks (as long as the group item is not expanded). When expanded, only the desired channels will be manipulated even if the parent group is also highlighted.

**Block period based on FPS value** — Uses the current frames per second value in layout to determine the block period (or amount of time a block represents). When disabled, you can specify the custom time amount that a block should represent.

For example, an animation may be intended for 30FPS rendering, while the creation of the animation may depend on a music rhythm based on a tempo of 96. So, you could set the block period so that each block represents a 1/4 note or 1/8 note.

**Block period** — This is how much time each block in a dope sheet should represent. Normally this is one frame but may be larger or even fractions of a frame. This is currently measured in seconds per block and defaults to 1/30<sup>th</sup> s/block.

**Block width min.** — This is the minimum pixel width that a block (that contains key frame data) should be.

**Block width max.** — This is the maximum pixel width that a block (that does not contain key frame data) should be.

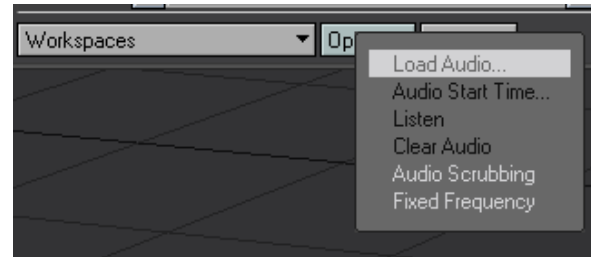
**Override block colors** — This setting will use the Key block color and Group block color instead of the default block colors.

**Key block color** — Choose a color for blocks that represent a single data channel that contains one or more keys.

**Group block color** — Choose a color for blocks that represent multiple data channels in which at least one of the data channels contains one or more keys.

## Scene Editor Audio Options

A **Menu** button is located at the bottom of the panel that controls some basic audio support features available in Layout.



**Play Audio** — This will play a loaded audio waveform from start to finish.

**Load Audio...** — presents a file requester to get an audio file to be loaded. The audio waveform is loaded and can be seen in the main Layout window within the timeline. Not much detail about it is available however.

**Audio Start Time...** — This requests the scene time when the audio should start.

**Audio Scrubbing** — When checked, the loaded audio will play whenever the timeline is scrubbed. If it is unchecked, no audio will play. The loaded audio file will record with previews.

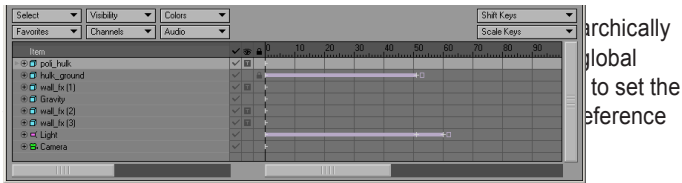
**Fixed Frequency** — This is a toggle option. When enabled, the audio will play at its intended sampled rate. If time scrubbing does not maintain real time playback rates, then gaps or skips in the audio may occur. When disabled, the pitch of the audio will vary as a result of the playback sample rate changing to accommodate the actual time scrubbing rate.

**Clear Audio** — clears out any audio currently loaded.



## Classic Scene Editor

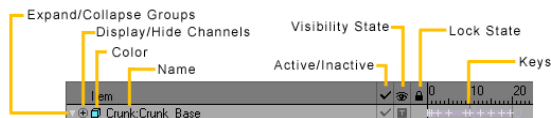
The **Classic Scene Editor** (**Scene Editor > Classic Scene Editor**) has been completely replaced with the new **Scene Editor** (**Scene Editor > New Instance**) and NewTek suggests that you use the new one over the classic one. The Classic remains to help the transition for our customers who have a history with the **Classic Scene Editor**.



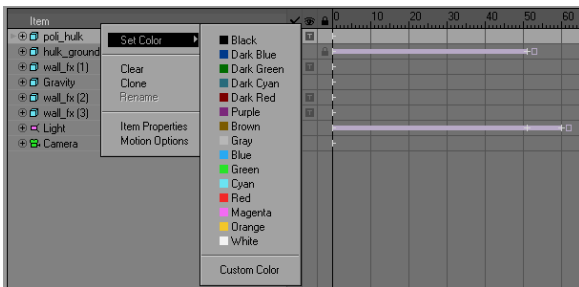
### Basic Functions

The scene list is a standard LightWave list window. You can expand and collapse groups as well as subordinate items (e.g., child objects) by clicking the arrow icon that appears to the left of the item name.

The plus/minus sign icon will display or hide the individual channels for the item. Most of the time, these are the position, rotation, and scale channels; however, other enveloped channels can also be included, like light intensity.

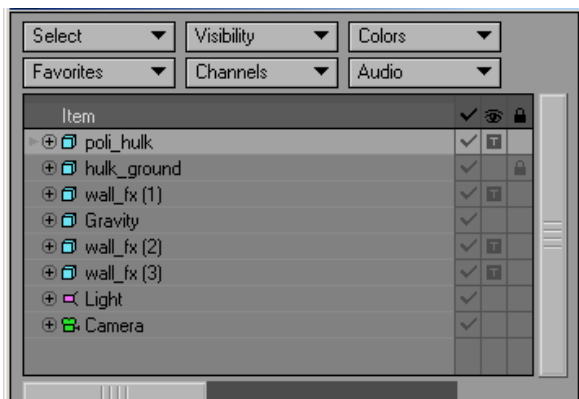


The item type icon indicates the type of item and the color used when the item appears in wireframe. You can change this by right-clicking on the item name and selecting a color from the pop-up menu. When working with complex scenes with overlapping objects, it can be beneficial to use different colors for certain items in the scene.



Double-clicking an item's name will bring up the related **Properties Panel**.

If you drag the right edge of the panel, you can shrink the size of the **Scene Editor** and hide the keys display. Use in this mode as a scene item picker!



The checkmark column activates or deactivates items. Deactivating an object is like setting its **Object Dissolve** (**Object Properties**) to 100%, and deactivating a light is like setting its **Light Intensity** (**Light Properties**) to 0%. For a bone, this toggles its **Bone Active** state (**Bone Properties**). This option has no effect on cameras.

The "eye" column is the visibility column. For objects, clicking on this icon will display a pop-up menu where you can select how the object is displayed. This can range from making the object hidden all the way up to showing it as a textured shaded solid.



The **Visibility** setting can dramatically affect not only how an *object* is displayed, but also how fast the display is updated. Moving a 200,000-polygon 100-surface spacecraft around the screen using a textured display surely requires greater computing power than a wireframe six-sided bounding box. There are other reasons as well. Often, in a very complex scene, you may need to play with object visibility options in order to concentrate on certain aspects of the scene.

Most of the **Visibility** settings are self-explanatory. **Front Face Wireframe** will show only polygons that face the camera. The **Textured Shaded Solid** adds image-mapped surface textures.

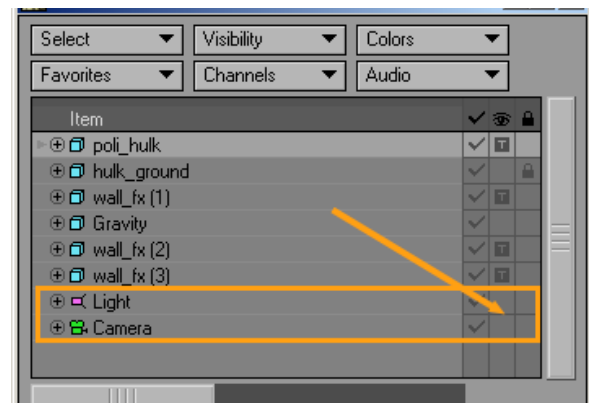


NOTE: You can override the **Visibility** setting to a certain extent by using the viewport's **Maximum Render Level** pop-up menu located on the top left edge of a viewport.



NOTE: The **Color** and **Visibility** options affect only the appearance of items in the Layout view. They do not affect the final rendered image.

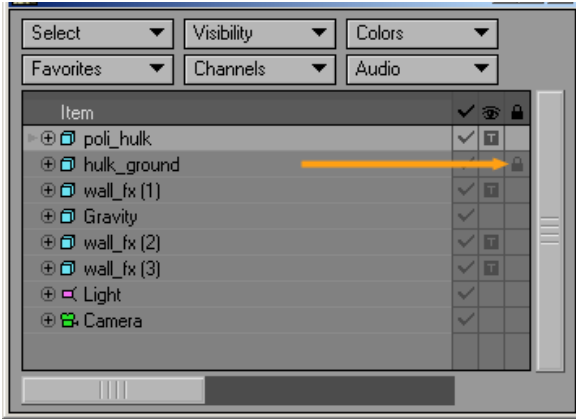
For lights, cameras, and bones, you can make them visible or hidden by clicking in the visibility column.





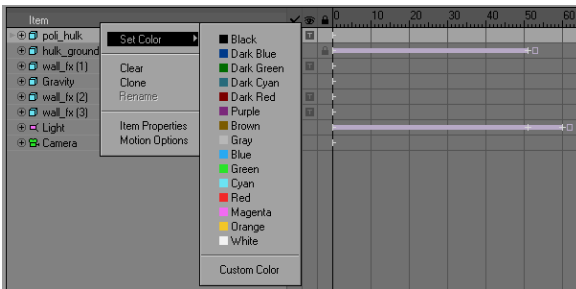


Clicking in the “lock” column will toggle the locking function on for that item. Locked items cannot be selected in the viewports. The lock icon will also appear on the **Current Item** pop-up menu (on the main interface) next to the item's name.



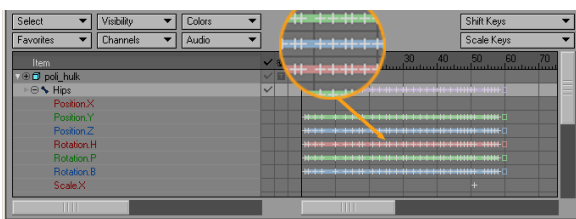
## Pop-up Menu

When you right-click an item, it displays a pop-up menu. This menu can set the item's wireframe color, clear the item, clone the item, rename the item, and open its **Properties** or **Motion Panels**.



## Keyframes

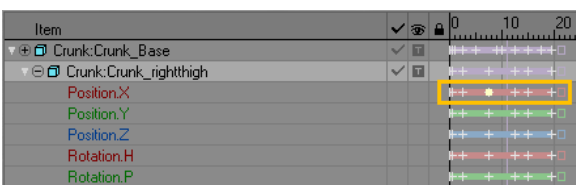
The colored bars to the right indicate the length of the keyframed channel. The bar starts at its first keyframe and ends at the last one. (Note that the ends may be past the end of the visible display area.) The plus signs indicate keyframes.



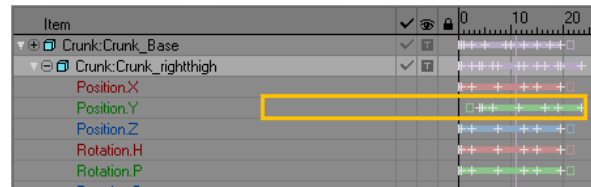
The line with the item's name is the master channel and will show a composite of all of the keyframes in any of the underlying channels.

## Adjusting Channels

You can move individual keyframes by dragging them with your mouse. The affected key is highlighted when you initially click on it. Dragging on the bar, off any key, will move the entire bar forward or backward in time.



*Key becomes highlighted as you drag one key*

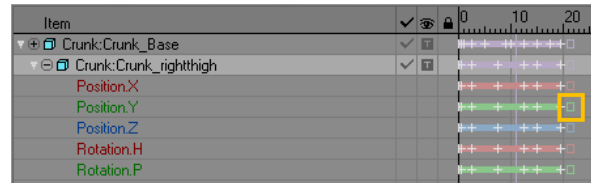


*Move all keys for channel by dragging bar*



NOTE: You can also drag keys on the master channel, which will affect only the appropriate underlying channels.

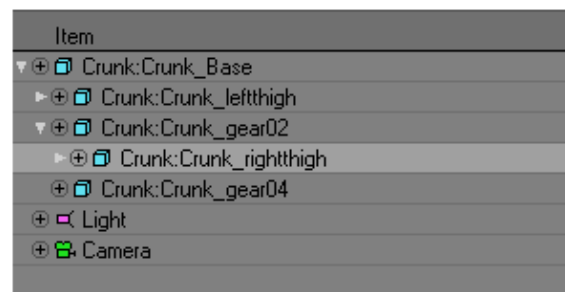
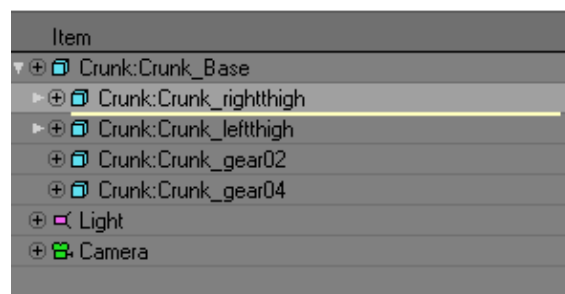
There are square handles at the beginning and end of every bar. You can drag these to scale all of the keys in the bar.



HINT: For more precise adjustments, use the **Shift Keys** and/or **Scale Keys** functions.

## Adjusting Hierarchy

You can drag item names up and down to change the order and hierarchy (i.e., parent/child relationships). As you drag, a yellow insert line will appear. You insert the item by releasing your mouse button at the line's position. The line will cycle between different lengths as you drag; the different lengths indicate different levels of hierarchy. The relative length indicates the level the line becomes when you release the mouse button.



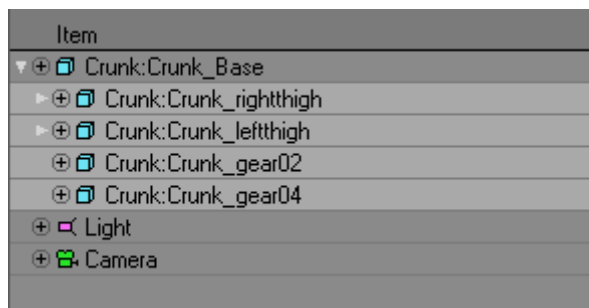
NOTE: An item inserted between a parent and child will always become another child.





## Effects of Selections

You can select a contiguous range of (same type) items by holding the **Shift** key down as you select. Hold the **Ctrl** key down to select/unselect non-contiguous items.



You can then drag the entire selected range of items to a new hierarchical position. Most of the editing functions can be limited to selected items only. These items will also be highlighted in the viewports.

## Adding Audio

You can sync your animation to sound: from the **Audio** pop-up menu, load a reference audio file (WAV format) that you can hear when you play the scene. A simple waveform is shown behind the time slider on the main interface. You can scrub through the audio by dragging the timeline slider or preview the audio by selecting **Play Audio**. Use the **Clear Audio** option to clear the audio from the scene.

The **Fixed Frequency** option keeps the audio from changing pitch when you scrub the frame slider.

You can delay the start time of a loaded audio file by selecting **Audio Start Time** from the **Audio** pop-up menu. The value you enter into the dialog is the delay amount in seconds (e.g., if **Frames Per Second**, on the **General Options Tab** of the **Preferences Panel**, is set at 30, entering 1.0 starts your audio at frame 30).

## Classic Scene Editor Buttons

Use the **Favorites** pop-up menu to create (or delete) selection sets for items that you want to access quickly. For example, if you always move the same five lights, you could multi-select them and make a *favorite*. Then, the next time you want to move them, you just select the favorites set you created and all five lights are selected automatically.

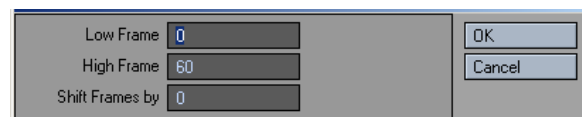
The **Select** pop-up menu lets you quickly select all items based on their type. To unselect a group of items, just click on any item. (One item is always selected.)

The **Visibility** pop-up menu will show/hide selected or all items.

The **Colors** pop-up menu will apply the selected color to all of the selected items. You can also set the default colors and apply them to the scene.

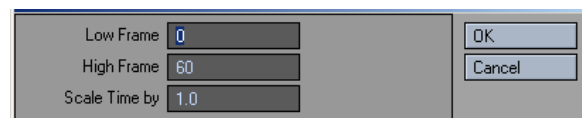
The **Channels** pop-up menu will expand/collapse selected or all items.

With **Shift Keys** you can shift keyframes for all or just selected items forward or backward in time. The **Low Frame** and **High Frame** values set the range of frames to be affected. This function lets you fine-tune the animation without making individual changes for each item in the scene (a potentially tedious task). Enter a negative **Shift Frames by** value to shift backwards in time.



Some operations can affect frames outside the specified range. For example, shifting a range of frames in the middle of a motion path will cause keyframes after the range to shift so they are not overlapped by the newly shifted keyframes.

With **Scale Keys** you can extend or shorten either the duration of all or just selected items. The **Low Frame** and **High Frame** values set the range of frames to be affected. The result is that events occur either more slowly or more quickly, as they have been scaled to take place over a longer or shorter period of time. **Scale Keys** enables you to fine-tune the animation, allowing certain events or the entire animation to take place within a specified time frame so that you do not need to alter specific keyframes manually. The **Scale Time by** value represents the scaling factor with 1 being equal to 100 percent.



**NOTE:** Time-related elements of the scene other than motions and envelopes will not change with the use of either Shift Keys or Scale Keys. Therefore, image sequence loop lengths and texture motion will not be affected.

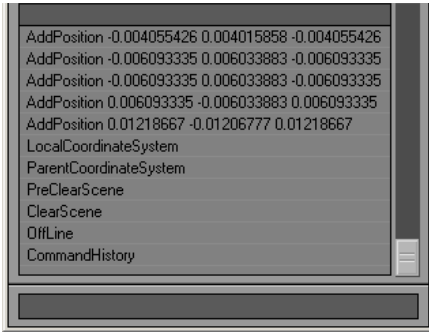


## Utilities Tab

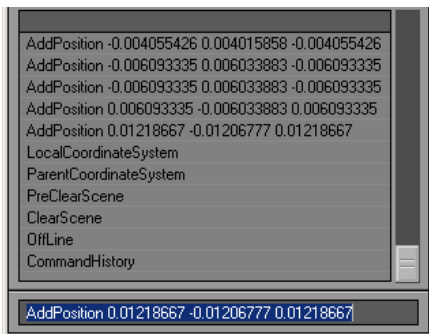
### Commands

#### Command History

The Layout interface is built on a *command system*. Buttons, keyboard shortcuts, plugins, etc., essentially submit commands to the underlying Layout engine, which performs the actual operation. You can view a list of the last commands that were executed by choosing **Utilities > Command History**.



You can execute a single command by entering it — with any required parameters — in the input field at the bottom of the **Command History Panel**. Clicking an entry in the list will automatically copy it to the input field — you can edit the command before hitting the **ENTER** key.



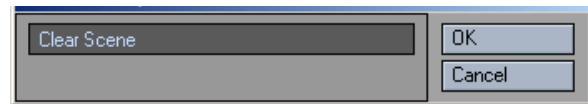
NOTE: To execute commands in a programmatic fashion, use LScript.



NOTE: You can save a list of commands to a file by choosing **Utilities > Save Command List**.

## Command Input

If you don't wish to use the Command History window, you can choose **Utilities > Command Input** and enter a command into the dialog that appears.



Obviously, using the user interface to execute commands is far easier, and most users do not need to use these functions. However, you might use the information from command history to reproduce or document the exact operations you made interactively. Moreover, some commands are not available to menus and keyboard shortcuts, thus, running them directly is the only way to execute them.



NOTE: To execute commands in a programmatic fashion, use LScript.

## Save Command List

You can save a list of commands to a file by choosing **Utilities > Save Command List**.

## LScript

Executes an un-compiled Lscript.



NOTE: The un-compiled Lscript file format is (\*.ls)

LScript is a high-level wrapper for the LightWave plugin Application Programming Interface (API). It encapsulates the complex underpinnings of the API away from the plug-in developer, allowing them to concentrate more fully on the task to be accomplished. LScript also provides added features not available in the plugin API, making plugin development faster.

Because LScript has its roots in the C language, the transition between scripting and native-language (binary) plugin development is eased a great deal. Scripts written in LScript can often be ported into C with far less effort. This makes it possible to use LScript as a rapid prototyping tool for plugin development.

Nearly all of the LightWave plugin architectures have scripting capabilities through LScript.

LScripts can be installed in the same way plugins are. The LScripts then become commands that can be added to menus or assigned to keyboard shortcuts.

LScript also provides a run-time system, allowing scripts to be compiled into an encrypted binary form that prevents modification or reverse engineering. Facilities for timed or counted execution are also provided by the run-time system.

Most important, LScript is a *virtual machine* system. LScripts are platform independent - scripts written on one platform should work directly and immediately on any other platform supported by LightWave. This differs from traditional plugin development in that each platform must have its own compiler, and each plugin must be compiled and maintained on that platform.



## LScript/RT

**LScript/RT (Run Time)** executes a compiled Lscript.



NOTE: The file format for compiled LScripts is \*.lsc

LScript is a high-level wrapper for the LightWave plugin Application Programming Interface (API). It encapsulates the complex underpinnings of the API away from the plug-in developer, allowing them to concentrate more fully on the task to be accomplished. LScript also provides added features not available in the plugin API, making plugin development faster.

Because LScript has its roots in the C language, the transition between scripting and native-language (binary) plugin development is eased a great deal. Scripts written in LScript can often be ported into C with far less effort. This makes it possible to use LScript as a rapid prototyping tool for plugin development.

Nearly all of the LightWave plugin architectures have scripting capabilities through LScript.

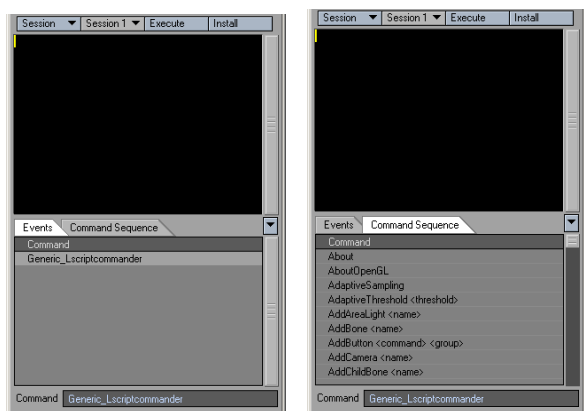
LScripts can be installed in the same way plugins are. The LScripts then become commands that can be added to menus or assigned to keyboard shortcuts.

LScript also provides a run-time system, allowing scripts to be compiled into an encrypted binary form that prevents modification or reverse engineering. Facilities for timed or counted execution are also provided by the run-time system.

Most important, LScript is a *virtual machine* system. LScripts are platform independent - scripts written on one platform should work directly and immediately on any other platform supported by LightWave. This differs from traditional plugin development in that each platform must have its own compiler, and each plugin must be compiled and maintained on that platform.

## LScript Commander

Lscript Commander (**Utilities> LScript: Lscript Commander**) can be used to create a command sequence or an actual LScript for Layout. A *command sequence* is merely a list of commands that can be executed in order from top to bottom. An LScript is similar, but more powerful. In an LScript, for example, you can have programming type commands like loops and so on.



The top black area is the session window and can contain either a command sequence or an LScript. You can type directly into this window or copy commands from either of the bottom two tabs.

The **Events Tab** contains a list of commands from operations that have occurred in Layout. The **Command Sequence Tab** contains a list of all available commands. You can copy commands from either list by selecting them (multiple selection is supported) and then right

clicking on the list. Once copied to the session window, you can edit the lines as needed.

You can execute single commands by entering them in the **Command** field.

You can define and use multiple sessions. The left **Session** pop-up menu contains controls to start a new session, load an existing one, save the current session to file, and close the current session. The **Clear Session** option erases all of the commands in the current session. Also on this menu are options to convert a command sequence into an LScript and convert an LScript into a command sequence.

The pop-up menu to the right is used to choose the current session, if there are more than one.

Click the **Execute** button to run the current command sequence or LScript.

Clicking the **Install** button will add the script to the Macros group on the **LScript Menu Tab**.

## Select Hierarchy

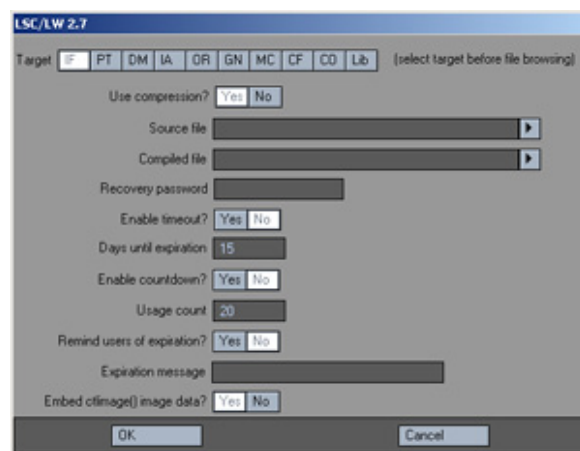
This LScript will select the entire hierarchy which the currently selected item belongs to.

## Select Children

This LScript will select all of the children (and their children) of the currently selected item.

## LS Compiler

This command will translate an un-compiled Lscript to a Binary compiled Lscript. Use the options on the panel to enable timeout, expiration message, and more.



Choose a **Target** type before file browsing. The Target is the architectural type of the script. If the script has a "@script" identifier in it, then you can specify it first, and the compiler will automatically select the architecture.

IF=Image Filter  
PT=Procedural Texture  
DM=Displacement Map  
IA=Item Animation (Item Motion)  
OR=Object Replacement  
GN=Generic  
MC=Master  
CF=Channel Filter  
CO=Custom Object  
Lib=Library file



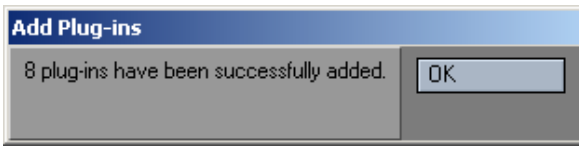
NOTE: un-compiled Lscript's file format is (\*.ls), The file format for compiled LScripts is \*.lsc.



## Plugins

### Add Plugins

To add plugins from Layout, choose **Utilities > Plugins: Add Plugins**. (Hold the Shift key to select a range. Hold Ctrl to select multiple non-contiguous files.) In Layout, an informational dialog will appear telling you how many plugin (commands) were added.



NOTE: Re-adding a plugin that is already added will not do any harm.



NOTE: A single plugin file can have many functions, some internal and not directly accessible by the user. Thus, when

you add one, it may report back that it has added more than one plugin. This is normal.

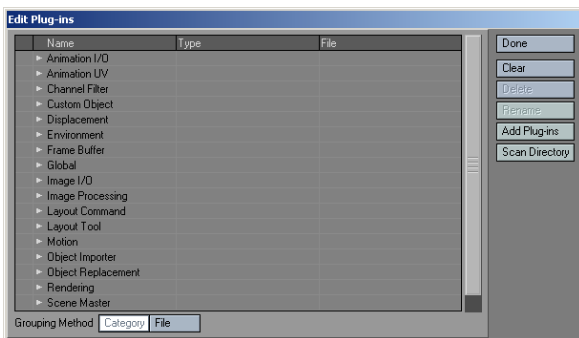
#### To add a directory of plugins:

You can quickly add all of the plugins in a directory (including subdirectories) using the **Scan Directory** button. (It may take a few seconds to scan all of your plugin files.

Be patient!) You can access the **Scan Directory** feature on the **Edit Plugins Panel**.

### Edit Plugins

To use the Edit Plugins command to review and delete plugins that have been added In Layout, choose **Utilities> Plugins: Edit Plugins**. You can also use the keyboard shortcut **Alt-F11**



#### To delete a plugin command:

You can delete a plugin command by selecting it and clicking the **Delete** button. (Other commands from the plugin will continue to exist.) Click **Clear** to delete all plugins listed. These operations have no effect on the actual files stored on your hard drives.

#### To rename a plugin command:

You can rename a plugin command's name by selecting it, clicking the **Rename** button and entering a new name.



NOTE: You can also add plugins on this panel. The **Add Plugins** and **Scan Directory** buttons will add an entire directory of plugins.

### File Grouping Method

If desired, you can list the plugin commands by their .p plugin filenames. Just select the **File Grouping Method** button at the bottom of the **Edit Plugins Panel**.

#### Last Plugin

The Last Plugin function (**Utilities> Plugins: Last Plugin**) will launch the window/panel of the last plugin used in the scene. This includes shaders, deformers and more.



NOTE: This function does nothing if no plugin has been used.

### Flush Plugins

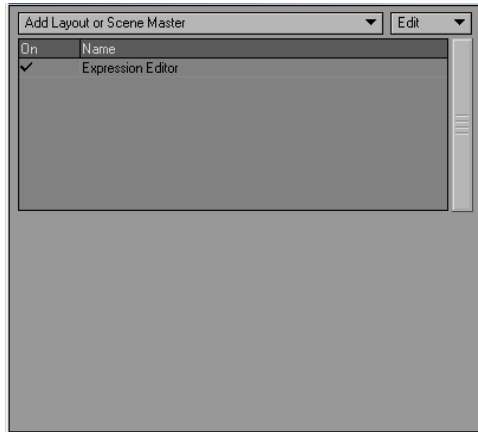
Choose Flush Plugins (**This option is unassigned.**) to tell the unused plugins to let go of their memory allocations.



## Master Plugins

### Scene Master Plugins

Choose **Utilities > Plugins: Master Plugins** to display the **Master Plugins Panel**. Use the **Add Layout or Scene Master** pop-up button to add global-type plugins.



NOTE: In some cases, plugins which are not meant for direct use, but are used internally, will be listed on the pop-up menu.



NOTE: Some of the Items in the **Master Plugins Panel** can be found on the interface and others can only be accessed through this panel.

#### *Here is a list to work from:*

Camera Selector — Only located in **Master Plugins Panel**.

Expression Editor — Only located in **Master Plugins Panel**.

Item Picker — Only located in **Master Plugins Panel**.

Lscript — Located under the **Utilities Tab**.

Lscript/RT — Located under the **Utilities Tab**.

Lscript Commander — Located under the **Utilities Tab**.

Master Channel — Located in **Utilities > Additional** drop down menu.

Lw\_macrorecorder — Only located in **Master Plugins Panel**.

Motion Mixer — Located under the **Windows** drop down menu.

Proxy Pick — Only located in **Master Plugins Panel**.

Scene Editor — Located in the **Top Toolbar** menu.

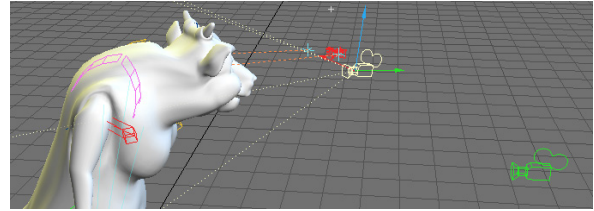
Setup Mode — Located under the **Setup Tab**.

**Spreadsheet Scene Manager** — Only located in **Master Plugins Panel**. This is a legacy tool. For a current **Spreadsheet** tool use the **Scene Editor**.

**Window Configure** — Located in the **Edit** drop down menu.

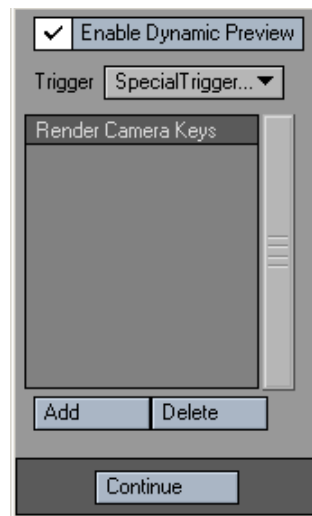
## Camera Selector

The **Camera Selector** master plugin lets you switch between multiple cameras. Note that this cannot be used to switch the camera during a render - the current camera is still used for that. However, you can use it to switch between cameras while playing a scene in Layout or when creating a preview anim.



To access, first bring up the **Master Plugins Panel (Utilities > Plugins: Master Plugins)**.

Then, add **CameraSelector**. Double-click the entry to bring up its setting panel.



The **Render Camera Keys** window lists a sequence of frames and the camera to switch to on those frames. Clicking the **Add** button adds the current camera at the current frame to the list. To delete an entry, just select it with your mouse and click Delete.

The **Enable Dynamic Preview** option turns this function on and off.

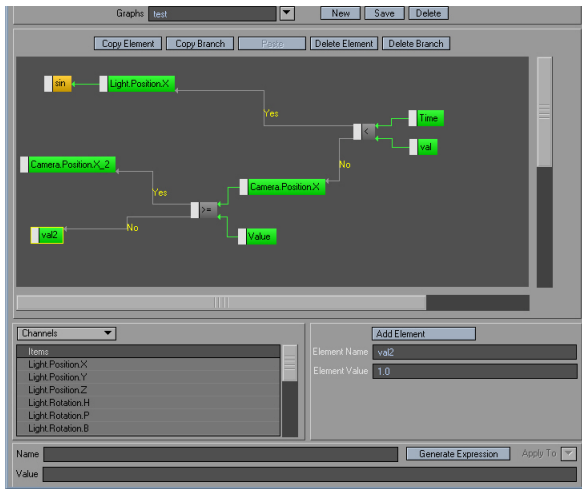
For CameraSelector to do its magic, it needs information not normally available to master plugins. As such, it automatically creates a null object called "SpecialTriggerNull" whose only purpose in life is to serve CameraSelector. If you wish to use a different object simply select it on the Trigger pop-up menu; however, normally, there is no reason to do so.



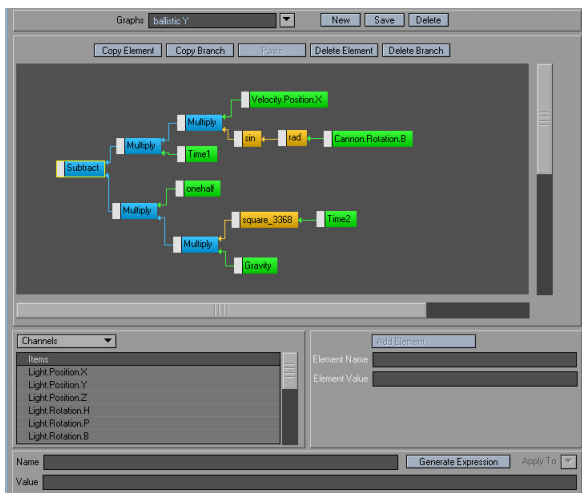


## Expression Editor

*Expressions* are an advanced LightWave 3D feature that uses mathematical formulas to modify the value of any animation channel. Expressions let you make the motion of scene items dependent on other item motions or factors in a scene. You could, for example, force an object to stay between two other objects, keep feet from going through the floor, or even control the entire posture of a character based on its feet! The possibilities are endless.



The **Expression Editor (Utilities> Plugins: Master Plugins> Expression Editor)** is a node-based expression generator. This allows you to build expressions for your scene in a graphical environment very similar to node-based compositors like Digital Fusion.

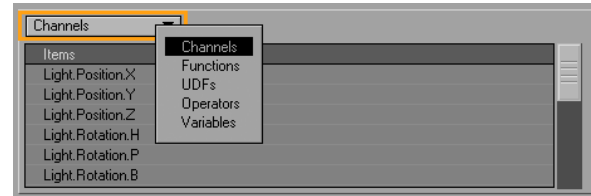


Use the **Name** field to give each expression you create a unique identity. By default the expression will be named "Expr1" when you generate the expression. The **Name** drop down menu, located next to the **Name** field, gives you the ability to change which expression is viewed in the workspace.



To start a new expression use the **New** button located at the top of the editor. Always remember to save your favorite expressions and delete unused or unwanted expressions using the **Save** and **Delete** buttons.

The **Elements** drop down menu gives you the following options:



**Channels** — All available animation channels will be listed in the **Items List**.

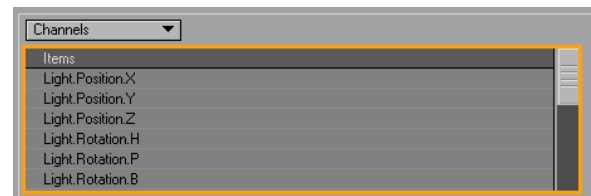
**Functions** — All available internal functions will be listed in the **Items List**.

**UDFs** — User-Defined Functions will be listed in the **Items List**.

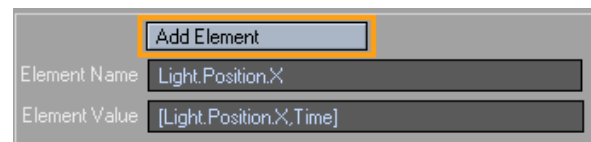
**Operators** — All available math and comparison operators will be listed in the **Items List**.

**Variables** — All available pre-defined editor variables will be listed in the **Items List**.

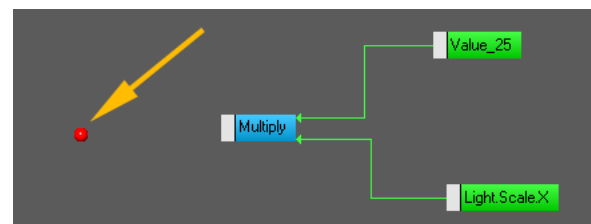
The elements that are available for each of these options will be displayed in the **Items List** located directly below the **Elements** drop down menu.



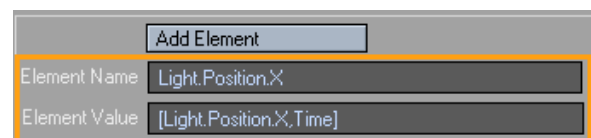
Once you have selected the element you want to use, click the **Add Element** button located to the left of the **Item list** to add it to the **Expression Editor** workspace.



Left click in the workspace and a small red dot will appear. Use this to determine where the next element will be added. By default the new element will appear in the upper left hand corner of the workspace.



Use the **Element Name** field to change an element's name, and the **Element Value** field to change the element's value. Both of these fields can be found under the workspace area. These fields will also display the Name and Value information of the selected element.

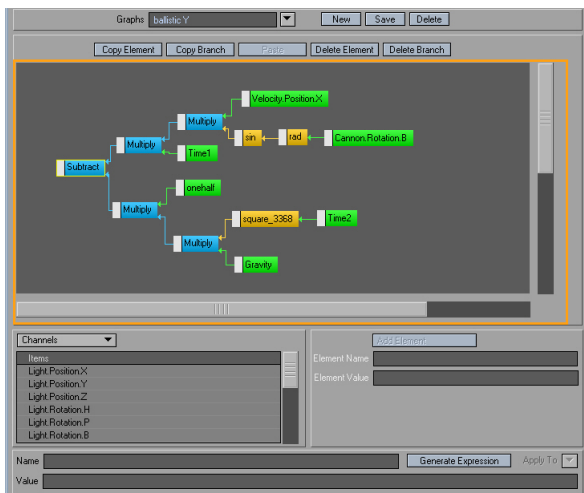


These fields are not used for some elements.



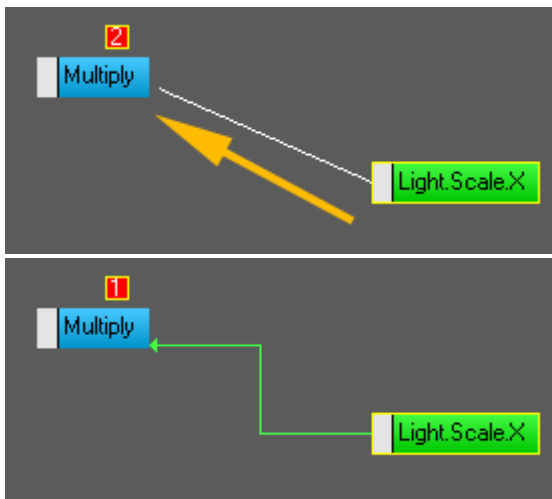


The **Workspace** is the area where all the elements will be brought together and organized in the **Expression Editor**. Use the Horizontal and Vertical scroll bars to move around within the workspace.

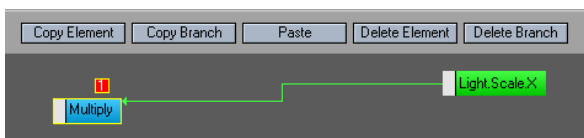


## Linking Elements in the workspace

To link elements together, **Left-Click** on the “white” end of an element and drag it to another element. To remove a link, **Left-Click** in the white area and drag away from the element.



To copy an existing element, select the element you would like to copy and use the **Copy Element** function located at the top of the **Expression Editor**. Use the **Copy Branch** function to create a copy of the entire hierarchy of the selected element. Once copied, use the **Paste** function to paste it from the clipboard.



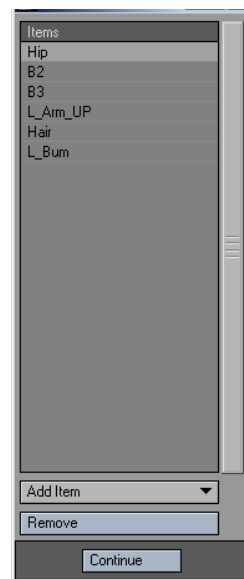
Use the **Delete** and **Delete branch** functions to remove elements from the workspace.

Once you have created an expression use the **Generate Expression** function to compile the node-based workspace into a written expression. Use the **Apply To** drop down menu to select a channel to use the expression on.



## Item Picker

Item Picker (**Utilities> Plugins: Master Plugins> Item Picker**) displays the **Quick Pick Panel**. You can instantly select frequently used items in your scene by just clicking on the item's name in the list.



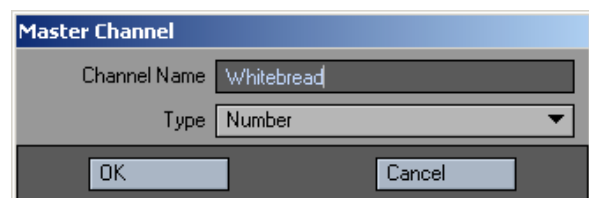
Use the Add Item pop-up menu to add an item from the scene to the list. To delete an entry, select it and click Remove.



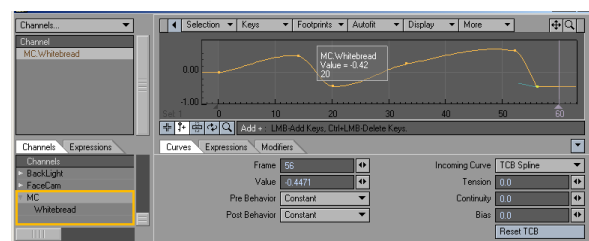
NOTE: The **Scene Editor** can also be used as an item picker.

## Master Channel

The **Master Channel** (**Utilities> Plugins: Master Plugins> Master Channel**) tool lets you create a user-defined channel, which will appear in the Scene list of the **Graph Editor** under the entry **MC**. You can keyframe the channel and use it as you would any channel.



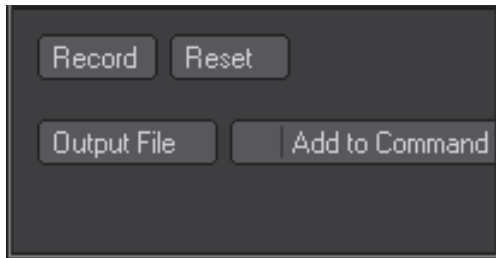
To create the master channel, add the plugin and enter a name into the **Channel Name** field of its options dialog. The **Type** setting determines the units of measure for the channel. You can add the plugin more than once to create multiple master channels.





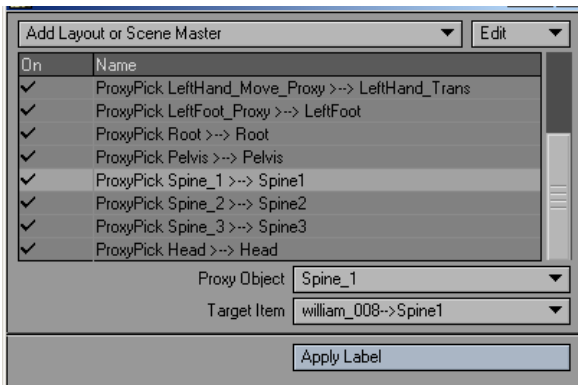
## LightWave Macro Recorder

This tool will monitor the stream of Command Sequence events that take place in Layout, and will generate a v2.0 Generic LScript file that will reproduce the sequence of commands, including their timings.

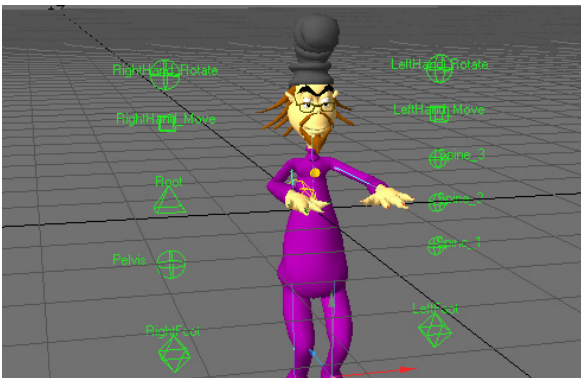


## Proxy Pick

Proxy Pick (**Utilities > Plugins: Master Plugins > Proxy Pick**) translates the *selection* of one object, known as the “proxy”, into the selection of another, known as the “target.” This is useful for picking small but crucial items out of complex, crowded scenes.



The **Apply Label** button uses the selected **Proxy Object** and **Target Item** to apply the ItemShape custom object to the proxy. The name of the target is used in ItemShape’s **Label** setting and its **Draw Line To** option is set to the target. If a previous proxy was used, this button will clean up the settings. This is an optional step.



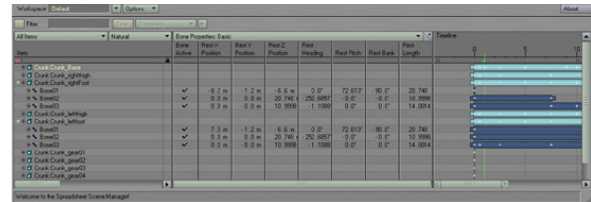
NOTE: Remember to disable Proxy Pick when you want to actually select the proxy rather than the target, as it is not yet telepathic.

## Spread Sheet Scene Manager



NOTE: This is a legacy tool. For a current Spreadsheet tool use the **Scene Editor**.

Often you want to edit certain properties of an item, and luckily, changing the values and properties of items is simple in Layout. But what if you need to change the properties for 20 objects? The Spreadsheet Scene Manager (**Utilities > Plugins: Master Plugins > Spreadsheet**) organizes these properties and lets you edit a wide range of values quickly and easily.



The Spreadsheet Scene Manager has six sections: Workspaces, Filters, Items, Properties, Timeline, and the Edit Controls. Many of these areas have Expand and Collapse controls, so you can adjust which sections are currently displayed. A message bar above the Edit Controls informs you about the tasks performed.

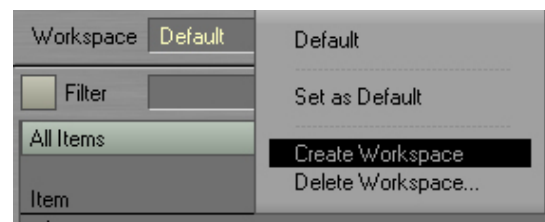
### Workspaces

You can use the Spreadsheet to customise settings and options that are displayed in the Manager’s panel.

Whether creating a workspace makes editing easier for you or saves you time, you can configure almost every aspect of the display with this plugin.



Settings are defined in the Spreadsheet as a Workspace. You create a library of commonly-used displays by using the **Create Workspace** and **Delete Workspace** commands. These two commands are found in the drop-list beside the Workspace name.





## Creating a workspace:

**Step 1:** Customise the Workspace to fit your display needs.

**Step 2:** Apply a Name to the Workspace by entering text in the **Workspace** name field.

**Step 3:** Select the **Create Workspace** command in the drop-list menu.

Once a Workspace is created, the name of the Workspace appears in the top portion of this drop-list menu. To revert to the settings saved in this Workspace, simply select its name from the drop-list. This will load the settings for the selected Workspace and reconfigure the interface to match the saved parameters.

To remove the current Workspace from the drop-list, select the **Delete Workspace** command from the drop-list. A confirmation dialog will ask whether you want to delete the Workspace from the scene file. Once you delete it, the Workspace name is also removed from the drop-list.



NOTE: Workspaces are saved with the scene file.

## Options

The **Spreadsheet Scene Manager** has options for customising how the **Spreadsheet** tools function and how information is displayed. A check-list contains a list of **General**, **Display**, and **Workspace** options for the **Spreadsheet**.



You can use this check-list (shown above), or you can access the tabbed interface by selecting the first item in the list, labelled **Options**.

## Filters

A complex animation might have dozens, if not hundreds, of items within the scene. Searching through this almost endless list to find a particular group of items can often be frustrating.

Using the Spreadsheet's filtering options, you can narrow the list down to make it easier to navigate, so you can get to your items faster.



You activate the **Spreadsheet**'s filtering tools with the **Filter** check box. Then you enter the characters for your search in a text field so that you can search through the item names in your scene. The filter parameters determine how the search is conducted.

## To use a Filter:

**Step 1:** Create a scene with three nulls named leftLeg, leftArm, and rightArm.

**Step 2:** Select Scene > Generics > Spreadsheet.

**Step 3:** Activate the **Filter** option.

**Step 4:** Enter the word "Left" in the text field.

**Step 5:** Select the **Containing** filter parameter.

The Manager then searches through the items listed in your scene, looking for names with the word "Left." The resulting list of items reads, leftLeg and leftArm. Because the item rightArm doesn't contain the word "Left," it is omitted from the filtered list.

Activate the **Case** option to refine the search parameters; **Case** includes the upper or lowercase characters of your entry in the search. In the example above, activating the Case option results in no items in the filtered list because the search characters contain an uppercase "L" in the word "Left." The item names use a lowercase "l" and since they don't match, they are omitted from the filtered list.



NOTE: The filtering tools are not available in the Hierarchy listing mode.

After you create a working filter, you can add it to a favorites list by selecting the **Create Favorite...** command in the Favorites drop-list.

This command prompts a dialog window that requests a name for your search parameters. Later, you can quickly reselect the filter and its settings by name. The new filter is placed at the top of the drop-list, above the Favorite commands.

Replacing and Deleting Favorites from this menu is as easy as selecting a filter name from a list.

This list of Favorite filters is saved within the animation's scene file.



NOTE: The section of the Spreadsheet Scene Manager that contains the filter controls can be minimised by using the **Expand** and **Collapse** button.

## Mouse Functions

The mouse has several different functions in the **Spreadsheet Scene Manager**.

**LMB:** Selects and highlights a field.

**SHIFT + LMB:** Range selection.

**CTRL + LMB:** Non-continuous selection.

**RMB:** Opens the cell's **Options** menu.

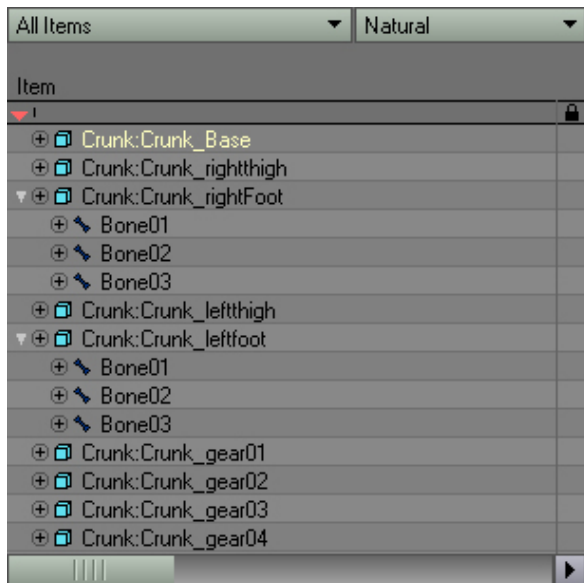
In the **Items List**:

**Double-click:** Selects the object in Layout.



## The Item List

Items in a scene are displayed in the Spreadsheet Scene Manager's **Item** list window.



Items in this list are determined by any active filters and by the current Display Mode. Use the plus (+) and minus (-) icons to open and close an item's channel lists.

By selecting a Display Mode you specify which items the Spreadsheet will include and also how the Spreadsheet is displayed. The **Hierarchy** display mode lists the items by creating collapsible trees of parent and sibling items. The **All Items** mode lists every item in the scene, while the **Objects Only**, **Bones Only**, **Lights Only** and **Cameras Only** modes refine the list to include only certain types of items.

The **Sort Order** mode determines how the items are actually listed in the window.

The **Natural** and **Reverse Natural** modes list the items in either the order or the reverse order they were loaded or created in the scene. **Alphabetical** and **Reverse Alphabetical** will display the items in a fashion that will make it easy to locate individual items.

In the **Selected Items** display mode, you have two additional sorting modes: **Selected** and **Reverse Selected**. These modes refer to how the items were selected in Layout.

## The Property Cells

The **Property Editor** is where you find most of the Spreadsheet's functionality. Here you can select and edit an item's individual value or groups of values for multiple objects. This list of cells is organised using rows and columns, much like a spreadsheet used in business applications. That is why this plugin is called the Spreadsheet Scene Manager.

Bone Properties: Basic							
Bone Active	Rest X Position	Rest Y Position	Rest Z Position	Rest Heading	Rest Pitch	Rest Bank	Rest Length
✓	-8.2 m	-1.2 m	-6.6 m	0.0°	72.613°	-90.0°	20.748
✓	0.0 m	0.0 m	20.748 m	-252.6857°	-0.0°	-0.0°	10.9998
✓	0.0 m	-0.0 m	10.9998	-1.1088°	0.0°	0.0°	14.0014
✓	7.3 m	-1.2 m	-6.6 m	0.0°	72.613°	-90.0°	20.748
✓	0.0 m	0.0 m	20.748 m	-252.6857°	-0.0°	-0.0°	10.9998
✓	0.0 m	-0.0 m	10.9998	-1.1088°	0.0°	0.0°	14.0014

Listed in the columns, left to right, are the item properties supported by the Spreadsheet Scene Manager. These properties are organized in lists of related attributes called Banks. You can select which banks are currently displayed by selecting them from the **Banks** drop-list.

You can view more than one Bank at a time. You can add and delete different Banks from the **Property Cells** by using the plus (+) and minus (-) button located by the Banks drop-list. Use the slider-bar at the bottom of the **Property Cells** to adjust which bank(s) are currently displayed in the viewing area.

## Editing Cells

The **Property** cells contain the values for each of the properties, for all items. The values range from numerical values to file names, depending on which property you are editing. Because the Spreadsheet is a non-modal interface, when a value displayed in a cell is changed, the item's property is updated instantly. This lets you interactively change the properties of each item and see their results in Layout's viewport.

You can define this level of interaction from the **Options** check-list. Enable **Apply Quick Edits Immediately** to make changes as soon as the mouse button is released. Otherwise changes are delayed until you hit the **Apply** button.

There are two ways to edit the values displayed in an item's property cell. The first method is to edit the values directly in the cell.

**Step 1:** Select a cell or group of cells to edit. This can include cells from multiple columns, as long as they are of the same type. The interface will not allow you to mix value types.

**Step 2:** Using the **LMB**, edit a selected cell. Once you let go of the mouse button, the values for all selected cells will adjust accordingly.

The second method is to use **Edit** controls.

**Step 1:** Select a cell as you did above.

**Step 2:** Use the **Edit** controls located at the bottom of the interface to adjust the values in the cells. These controls reflect the type of value you are editing, so they vary from cell to cell. The image below shows numerical data (integers, percents, distances, angles, time and floats) but it gives you an idea of how the editing controls work.



## Controls for numerical data:



The **Edit** controls allow you to edit the property cell's values in absolute or relative mode. You enter a value in the **Change Value To** field and choose an edit effect from the **Effect** drop-list. The effect selects how the cell's value will be altered when the edit is applied.

The **Replace** edit mode lets you make absolute changes to the cell's values. That is, when you apply the edit to the selected cells, the new value will replace the current values. For example, you select ten cells with various values in each cell, and in the **Change Value To** field you enter a value of 2.0. When the edit is **Previewed** or **Applied**, the cells will now all read 2.0.

The **Add**, **Subtract**, **Multiply**, and **Divide** functions are all relative editing modes. They adjust the cell's values relative to their original values. In the example above, if you select the **Add** edit mode, all cells will add the value 2.0 to their original values when you apply the effect.

By pressing the **Apply** button, you commit to the edit. You cannot undo these edits. For that reason, you can use the **Preview** and **Reset** buttons to view the effect of the edit without actually applying the edit. If you like the results of the previewed changes, simply press the **Apply** button and the changes are made to the item's properties.



**NOTE:** The Spreadsheet Scene Manager will notify you when you are editing in Preview mode by highlighting the **Items**, **Property Cells**, and **Timeline**

**Panels.**

The **Use Step** field lets you increase the values in the selected cells by an exact increment. Simply put the increment in the field, select an edit mode, and **Preview** or **Apply** the changes.

## Envelopes

The **Spreadsheet Scene Manager** also lets you edit any parameters that can support envelopes. Like in Layout, these parameters are indicated by a small **E** button in the cell itself.

All Items	Natural	Channel Values
Item		All Channels
+	Bone02	
+	Bone03	
+	Crunk:Crunk_gear01	
+	Crunk:Crunk_gear02	
+	Crunk:Crunk_gear03	
+	Crunk:Crunk_gear04	
+	Light	
-	Camera	
	Position.X	- 2.8409 <b>E</b>
	Position.Y	12.7522 <b>E</b>
	Position.Z	- 106.4855 <b>E</b>
	Rotation.H	0.0 <b>E</b>
	Rotation.P	0.0 <b>E</b>
	Rotation.B	0.0 <b>E</b>
	ZoomFactor	3.5077 <b>E</b>

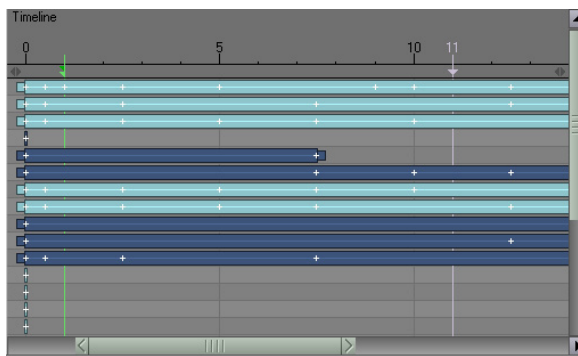
You can add envelopes by group selecting the cells and pressing the **E** button. You can also use the **Shift E** combination to remove envelopes from properties.

If the cell has envelope information stored in the channel, a keyframe divot appears on the left edge. If a keyframe is located at the current frame, the divot turns yellow, and the value becomes editable. Otherwise this value is ghosted.

Clicking this divot will create a key, and Shift clicking will remove a key.

## Timeline

The Spreadsheet's timeline has much of the same functionality as LightWave's **Scene Editor**. You can edit an object's motions by simply adjusting an animation bar located on the timeline.



The long bars let you adjust the first and last frame for both the **Render** and **Preview** ranges.

The fifth bar in the image above shows the position of the current frame. By sliding these bars around, the Spreadsheet updates the settings in Layout dynamically.

You can adjust the **Start Time** and **End Time** by dragging the black arrows beneath the frame hash marks.

The different colored bars display the frame range of the item's motion. Small plus signs (+) within the bars represent the keyframes for motion. To change the start and end time of the motion, you must select the middle of the bar; you can then slide the bar to the left and right. To scale the motion, you drag the left or right handles of the bar.

The slider located at the bottom of the Timeline both zooms and scrolls the viewable area.

## Column Sorting

You can change a column's sort order by selecting the area under the column's header name.

As you select columns, numbered flags appear that reflect the sorting order. By defining this order, you can sort the item list according to its properties. For example, you can sort by an object's Subpatch level, which will put all the items at level three at the top, and everything at level one at the bottom.



---

## Chapter 17: Object Properties

---

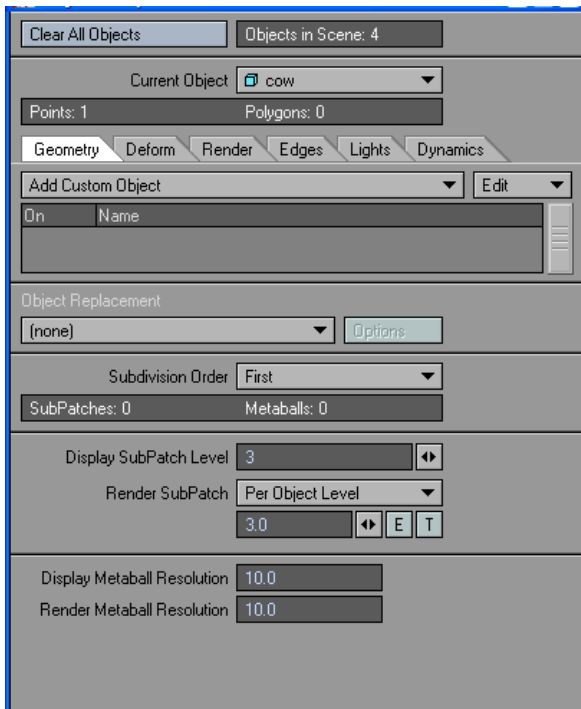




## Object Properties

### Object Properties Panel Introduction

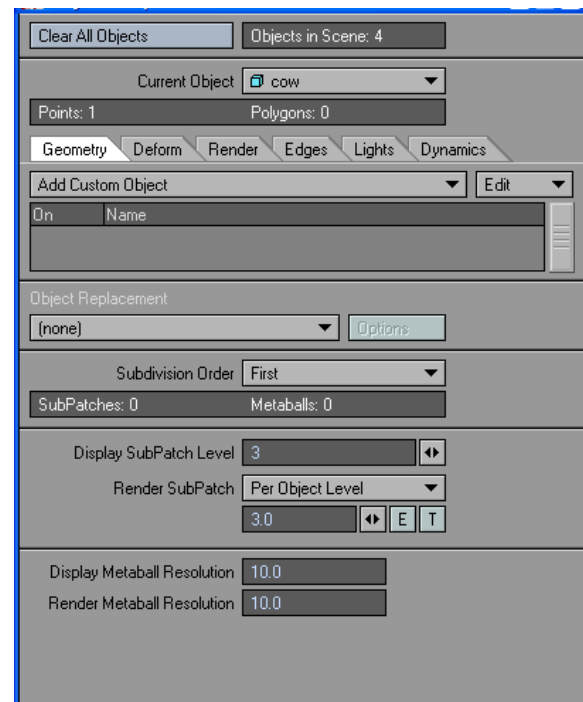
The **Object Properties Panel** controls the settings for the current object—the object last selected in Layout. It will be shown on the **Current Object** pop-up menu on the **Object Properties Panel** and changed here as well.



NOTE: The Properties Panel for the current editing mode (i.e., Objects, Bones, Lights, etc.) can be displayed by clicking the Item Properties button on the main Layout interface.

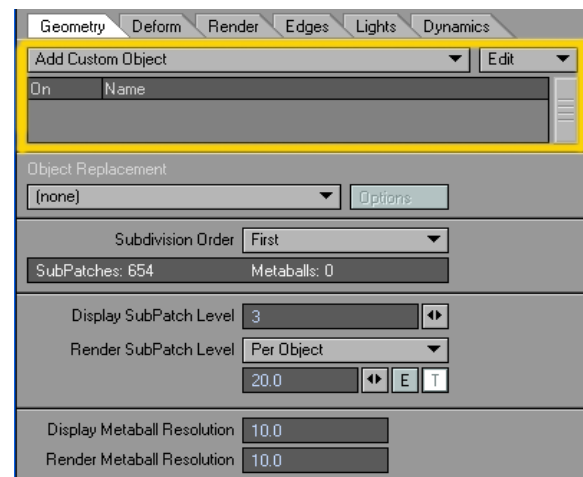


## Object Properties: Geometry Tab



### Add Custom Object

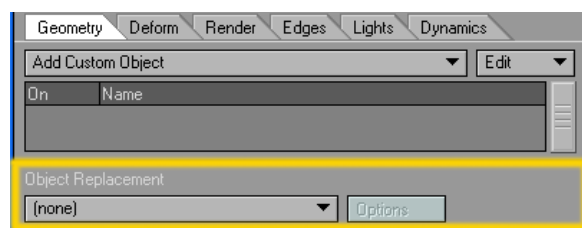
Use the **Add Custom Object** pop-up menu on the **Geometry Tab** of the **Object Properties Panel** to apply a Custom Object plugin to the current object. Custom objects are usually used to change the look of a null and give you additional visual feedback features. Like null objects, custom objects do not render.





## Object Replacement

You can use special plugins to replace objects during the course of an animation.

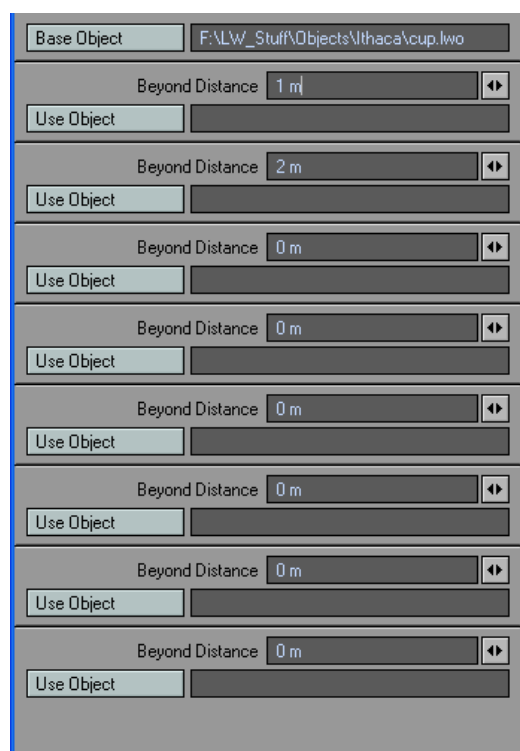


## Lscript and Lscript/RT

Use these two options to run Lscripts for object replacement.

## Level-of-Detail Object Replacement

Adding **Level-Of-Detail Object Replacement** lets you replace the object with another object based on its distance from the camera. This can be a great time-saver for objects that are sometimes close to the camera and sometimes far, during your animation. There is no reason to waste rendering power on a gazillion-polygon battle cruiser that is too small to make out any of its details. If it's far enough away, you might be able to get away with a simple box or sphere!



From top to bottom, the entries must be in near-to-far order, so the furthest entry is last. When the distance from the camera to the object is within each range, the corresponding object is loaded. The **Base Object** setting defaults to the object the plugin is added to; however, you can define a different one if you like.

## ObjList

ObjList replaces the current object with those listed in a text file. The file is defined using a file requester that appears when you click the **Options** button. ObjList works much like ObjectSequence, discussed later, but the object replacement list lets you use objects in different directories or even across a network. The file must be formatted as follows:

```
#LW Object Replacement List
<replacement frame number>
<replacement object file>
<replacement frame number>
<replacement object file>
<replacement frame number>
<replacement object file>
```

The replacement object information is defined in pairs of lines. The first line of the pair is the frame number at which the object is to be replaced. The second line of the pair is the filename of the replacement object, using a full path.

For example, to sequence through a series of box objects, this list would be used:

```
#LW Object Replacement List
0
c:\Newtek\Objects\terrihendrix.lwo
10
c:\Newtek\Objects\lloydmaines.lwo
20
c:\Newtek\Objects\willoryfarm.lwo
```

The sequence of objects does not need to be similar copies of the same thing. You can replace the cow with a chrome teapot if you want to.

The first paired lines must be the object you want at frame 0. LightWave will assume this even if you enter a different frame number. As such, if you used:

```
#LW Object Replacement List
35
c:\Newtek\Objects\box2.lwo
```

The Box2 object would still be loaded at frame 0.

## ObjectSequence

ObjectSequence replaces an object with another one at a certain frame of the animation. This replacement is like choosing **Items > Replace With Object File** except that it happens during the animation. ObjectSequence is to objects, what an image sequence is to images.

To perform an object replacement, you must have multiple objects with names that differ only by a three-digit number. For example, if you want to change between a series of box objects, you would name the first object box000.lwo. If you want that object replaced by a second object at frame 10, name the second object box010.lwo. This would be replaced by box027.lwo at frame 27, and so on. These objects don't need to have anything in common except their names.

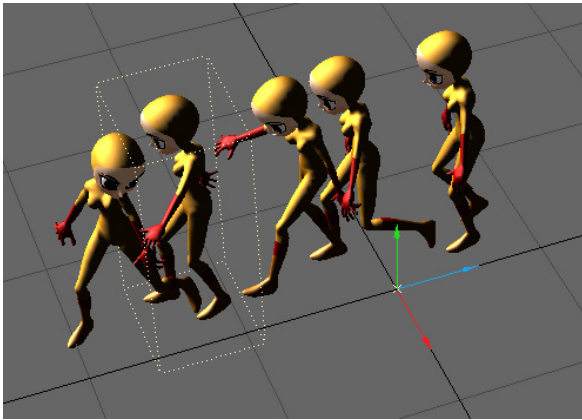
To use ObjectSequence, load the first object normally. Then, select ObjectSequence as the Object Replacement plugin. All of the object files must be in the same subdirectory.

Object geometry is normally created in Modeler and object animation is accomplished in Layout. However, you can also animate object geometry, that is, change the relative positioning of points in an object and thus change its shape. A simple example of this is sending a ripple through a flag, which might be a simple segmented flat box. LightWave's bones



feature lets you bend and distort object geometry using an object skeleton. Often the bones are set up in a hierarchy using Inverse Kinematics (IK) to help animate a complex structure, but IK can just as easily be applied to a hierarchy of objects. Geometry can also be influenced by morphing.

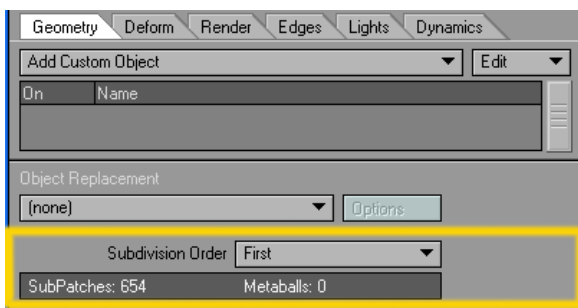
To create such object sequences, you can use the **Save Transformed Object** command (**File > Save > Save Trans Object**) to save the stages of transformation for an animated object.



Sequence of objects created using Save Transformed Object to save object in various stages of animation. Note that the location as well as the shape of the geometry is recorded.

## Using SubPatch Objects in Layout

The **Subdivision Order** setting controls the order in which LightWave *meshes* and deforms a SubPatch object (using bones, **Endomorphs**, **Displacement Maps**, etc.). This can have a huge impact on what the rendered object will look like. LightWave subdivides the SubPatch object, converting it into a polygon mesh (i.e., meshing) on the fly. **First** is the default and should be used whenever possible since it allows any deformation to affect an object *in detail*. For example, you can't add displacement wrinkles to skin with **Last**, since the wrinkles would affect only the low-resolution cage. Moreover, choices other than **First** take twice as long. This is because LightWave actually does the meshing computation first regardless, to get the undistorted point positions that are later used for surface texture mapping purposes. Thus, choices other than **First** require a second meshing computation.



However, if the meshing is performed first and then bones considerably bend the object, you may get unwanted *pinching* in your object. In this case, **Last** or After Bones may work better since the conversion is not done until after the SubPatch cage is deformed. Also, if you are morphing a SubPatch object, you will want meshing to occur after morphing (using **After Morphing**, **Last**, etc.). This is because the SubPatch object is really a control-point cage. If you mesh before the morph, you are actually changing the shape of the control cage and unexpected results will likely occur.

If you use a **Displacement Map** on a SubPatch object, you probably want the subdivision to occur before the displacement (using **First**, etc.). This will give the displacement more points to displace.

If you need to choose one of the *in-between* settings, here is the order in which LightWave performs object deformations:

- Morphing
- Before-bones plugin displacements
- Bones
- Object coordinate plugin and built-in displacements
- Motion (scale, rotate, move)
- World coordinate plugin and built-in displacements

This is the order of all the mesh operations:

- First subdivision
- Morphing
- After Morphing nodal displacement
- After Morphing bump displacement
- After Morphing displacement map
- After Morphing displacement plugins
- After Morphing subdivision
- Before Bones nodal displacement
- Before Bones bump displacement
- Before Bones displacement map
- Before Bones displacement plugins
- Bone deformation
- After Bones subdivision
- Before Local nodal displacement
- Before Local bump displacement
- Before Local displacement map
- Local displacement plugins
- After Displacement subdivision
- Local to world motion transform
- After Motion subdivision
- Before World nodal displacement
- Before World bump displacement
- Before World displacement map
- World displacement plugins
- Last subdivision

Before all those mesh operations are done, the motion of all the items in the scene is computed. Keyframing, IK, motion controllers, motion plugins, etc. That creates a transformation from each item's local coordinate system to the world coordinate system. It includes everything: translation, rotation, scaling, parenting. (The transformation is actually a matrix; there is no explicit rotation from local to world for example).

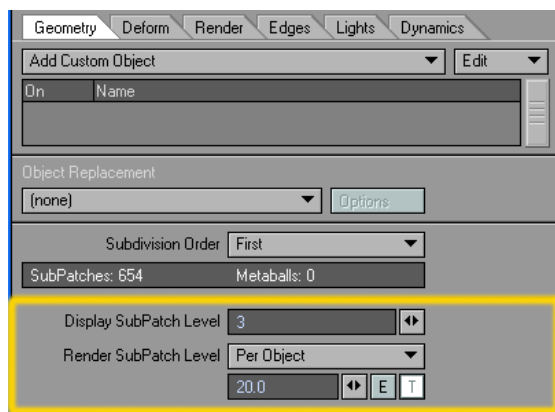
The local to world motion transform (step 19 on the list) applies that transform to each vertex of the mesh item. Before that step only the vertex positions in the item's local coordinate system are known (without the effect of item motion). After that step the positions of the vertices are in world coordinates (with item motion applied).

The "Before World" steps are slightly misleading in name. It actually means "Before World displacement plugins". As you can see they are in fact applied after the item motion is applied, i.e. to the vertices in world coordinates.



## SubPatch Display and Render Levels

SubPatch objects may be used in Layout as any other object. In Layout, the **Object Properties Panel** includes two settings that are important when you use SubPatch objects. The **Display SubPatch Level** and **Render SubPatch Level** settings determine the level of subdivision smoothing needed for display and rendering purposes, respectively. These values, generally, have an effect similar to Modeler's **Patch Divisions (General Options Panel)** setting, discussed next.



The SubPatch levels may be set to zero. This is similar to a level of one, except that the patches are not continually *remeshed*, even if **Subdivision Order** is not set to **First**. In other words, a level of zero makes patches act like normal polygons, and this allows faster interaction.



Note: Adaptive Pixel Subdivision (APS) is a new system of subdividing an object. With APS an object can be subdivided per polygon, by a texture, or even by pixels. It works by computing the subpatch level for each polygon, and storing those levels in a VMap.

### Additional Settings for Render Subpatch Level

**Per Object:** The classic method of having the same subdivision level for the entire mesh.

**Per Polygon:** Sets the Subdivision level per Cage Mesh polygon independently (commonly referred to as variable subdivision)

**Pixels Per Polygon:** Ties the amount of pixels of the output frame to the subdivision level of the cage polygons. For example, a setting of 4 means that a given polygon will have an area no larger than 4 pixels.



Note: When you make an item inactive in the Scene Editor, the mesh will still remain in the scene. This is necessary for a number of reasons, plugins and other parts of LightWave may still need to reference the item. APS effectively gives the mesh a setting of 0 for a subdivision level. This is for both Catmull-Clark and subpatch subdivision.

### Value modifiers:

**Numerical Input:** Set the subdivision level directly

**Envelope Control:** Animate the Subdivision level, and can also use expression, motion modifiers to control the level here. Use this setting for manual control over the subdivision level, use this if you know an object will not need more geometry than a given level for a given timeframe.



Hint: When using a black and white texture, usually an Alpha Map, you will need to tell APS which values are black and white. You do this by setting the Gradient to "Previous Layer" and the gradient settings to the subdivision levels.



Note: If an item is made inactive through the Scene Editor, that item will not have APS applied to it during rendering.

**Texture Control:** Use any of the standard LW texture controls to drive subdivision(texture, procedural or gradients):

Standard Gradients:

- Previous layer
- Distance to Camera,
- Distance to Object,
- X Distance to Object,
- Y distance to Object,
- Z Distance to Object

APS specific gradients:

**Angle to Camera Z axis:** Uses the angle between the position of the object/polygon in camera coordinates and the camera Z axis. 0 == object/poly in center of view, 90 == beside the camera

**Weight Map:** Using a weightmap to control the subdivision level, an averaging method is used to get a uniform weight on a per cage polygon level.

**Incidence Angle:** Uses the angle between the surface normal and the ray from polygon center to camera.

**Face Angle:** Uses the angle of the polygon normal in camera coordinates. 0 == facing the camera XY plane.

**Projected Size:** Uses the projected size of the polygon, in square meters. Basically, it calculates how big the polygon would look if it were held 1 meter away from the camera.



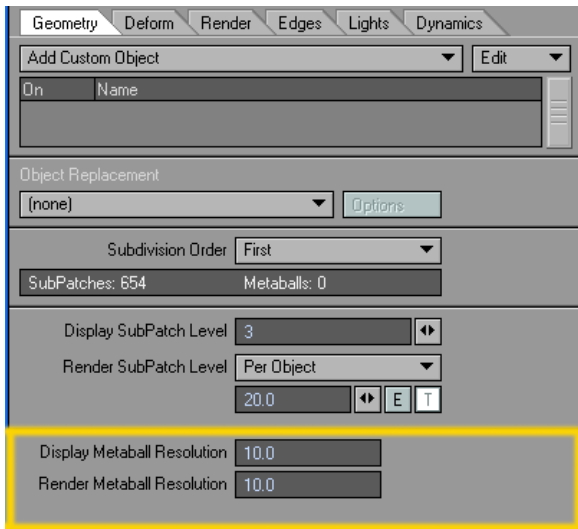
Hint: Currently, to use the distance gradients you need to set the subdivision order to after motion. As you need world coordinates to evaluate items that are not parented to the object itself, you can only get to world coordinates after motion of the object have been applied.

Treat Subpatch objects just like polygonal objects. You can use all of the normal Layout features, like bones, to animate and deform the SubPatch object.

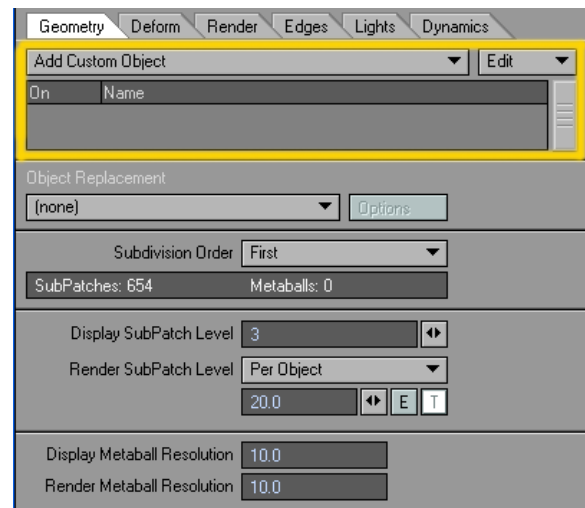


## Meta-Primitive Display and Render Levels

The **Display Metaball Resolution** value represents the number of subdivisions. To get a smoother surface, increase the value. There is no limit. A companion setting, **Render Metaball Resolution**, controls the resolution used for rendering, which can be different.

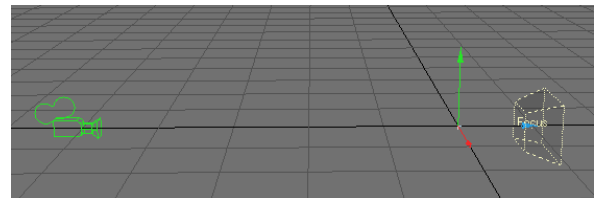


## Object Properties: Custom Objects



## Depth of Field Display

The **Depth of Field Display** custom object is a handy tool for seeing your **Focal Distance** and **Lens F-Stop Depth of Field** settings found in the **Camera Properties Panel**. Anything that is located in the **Focus** shape will be in focus and anything outside of the **Focus** shape will be out of focus.



### Steps for applying the DOF Display custom object:

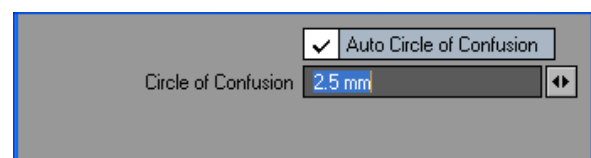
**Step 1:** Add a Null to your scene.

**Step 2:** Parent the Null to the camera.

**Step 3:** Activate Depth of Field in the Camera Properties Panel and adjust the Focal Distance and Lens F-Stop settings.

**Step 4:** Add the Depth of Field Display Custom Object from the **Add Custom Object** drop down menu.

Use the numeric options to set the **Circle of Confusion** setting. This setting means the size of a point that is recognized as a single point. The smaller the value, the smaller the area will be in which objects are defined as 'sharp'.





## HyperVoxels Drawing (HV Custom Object)

This custom object is automatically placed in the Add Custom Object list when **HyperVoxels** are applied to an object.

## IK Booster

IK Booster is discussed in the **IK Booster** section of Chapter 18. One option of applying IK Booster is to do it here in the Add custom Objects list. The preferred way is to add it with the **IK Boost** tool (**Modify> IK Boost**).

## Camera Mask

It is sometimes necessary to use a textured object for the background instead of using a **Background Image (Effects Panel, Compositing Tab)** for effects like casting shadows onto the background or when you want to move the background around. The Camera Mask custom object can be used to compute the exact Z distance needed to fill your camera view.

### To use Camera Mask:

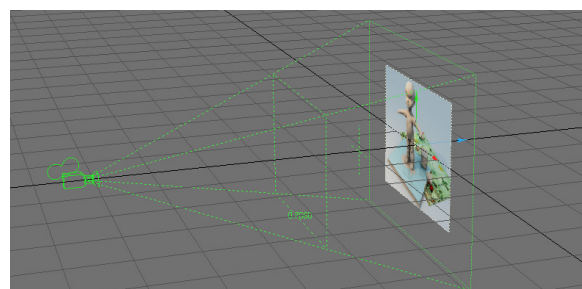
**Step 1:** Model a flat Z-facing rectangle object with the proper aspect ratio. For example, 640 mm x 480 mm for a 4:3 aspect ratio. Apply your surface texture.

**Step 2:** Load the object into Layout and parent it to the camera.

**Step 3:** Add the **Camera Mask** custom object plugin to the object. Enter the object's size into its options dialog and then close it. You will then see a rectangle that represents the exact Z position where the object would fill the camera view. The numeric value of this *magic distance* is also displayed.



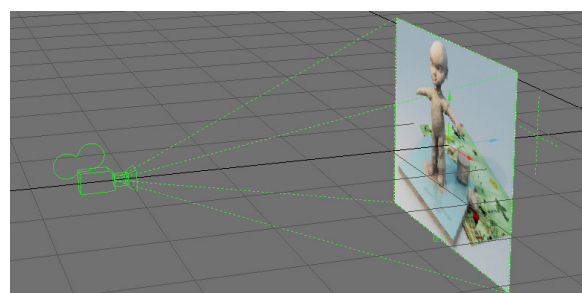
**Step 4:** Now, adjust the Z position of the object. You will see a representation of the camera view extend from the object with your textured rectangle attached to the end. If you have a **Camera Mask** set on the **Camera Properties Panel**, it will be visible.



Note

that when the object is selected, the camera's mask will appear as dotted lines. Otherwise, the mask will be the mask's set color. (Note: You may see OpenGL display errors when the mask is solid.)

**Step 5:** Adjust the object's Z Position to the magic distance. The object will now fill the camera view.



You

might also use a background plane with parts cut out, in conjunction with a Background Image, so you can position things between the plane and Background Image.

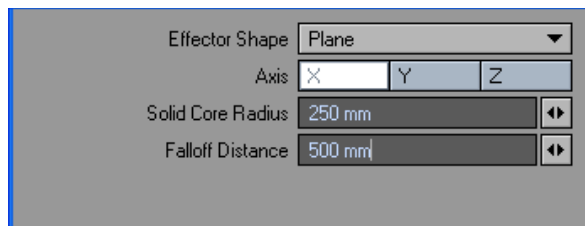
A cross (+) also appears, which marks the camera's **Focal Distance** setting from the **Camera Properties Panel**.



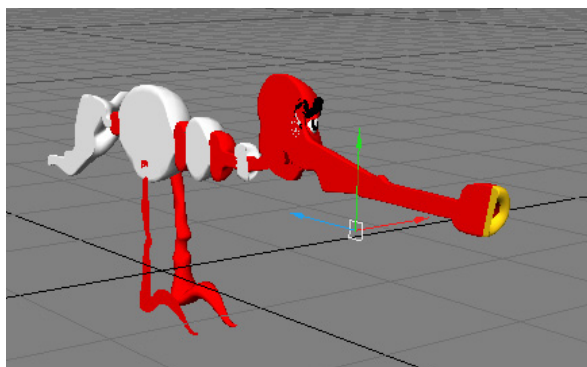
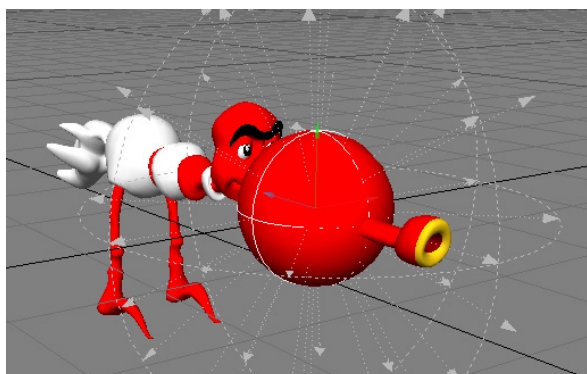


## Effector

The Effector custom object can be used with the **Effector Displacement plugin**. This custom object is designed to give you better visual feedback of your effector in Layout. Note that it does *not* directly communicate with the related Displacement plugin, so all settings must be set manually.



It has two modes to match the shape of the effector: **Point** and **Plane**.

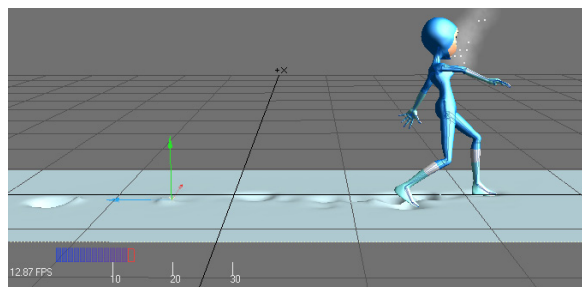


When **Effector Shape** is set to **Point**, set the **Solid Core Radius** and **Falloff Distance** to match those settings on the Displacement plugin's panel. The arrows and dotted-lines indicate the falloff area. The solid-line ball in the center is the solid core.

When **Effector Shape** is set to **Plane**, the effector will look like a four-sided plane. The **Axis** settings will become available, and you can then set them accordingly. The other settings have no effect in this mode.

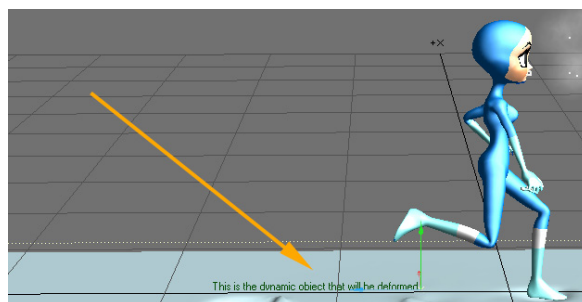
## Frame Rate Meter

Adding **Frame Rate Meter** to a null object will display frames-per-second information when you play a scene or drag the frame slider.

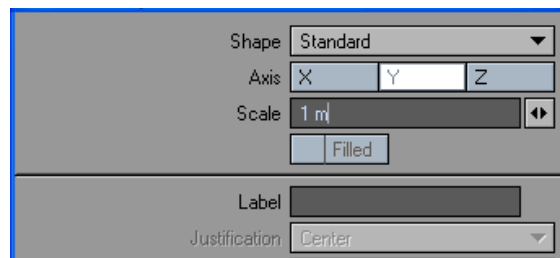


## Object Comments

If you add a comment to a scene item using the **Items Comments plugin** (**Utilities > Additional > Comments**), you can add the **Item Comment Display** custom object plugin and see the comment in your viewports.



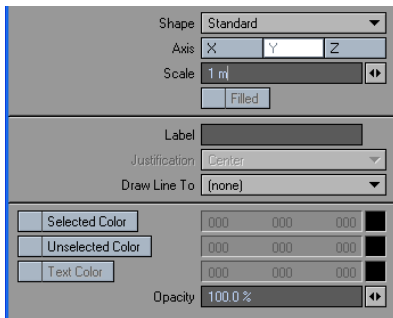
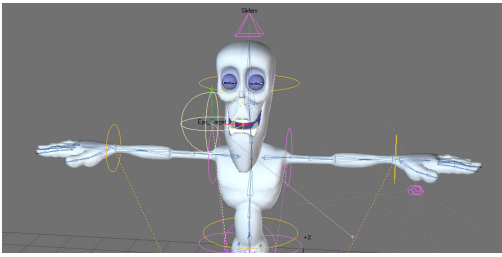
The **Item** setting determines which scene item gets its comment displayed. It does not have to be the item you add the custom object plugin to. You also have control over the text **Color**, density (**Alpha**), and **Justification**.





## Item Shape

Item Shape lets you specify the shape and look of your custom object. This is a great way to give each null in your scene a unique identity.



**Shape** — Choose the actual shape that will be displayed. Your choices are Standard (Standard Null), Box, Ball, Pyramid, Diamond, Tetra, Ring, Grid, and None.

**Axis** — Which axis the shape is facing. This is best represented with the Ring and Grid shape.

**Scale** — Size of the shape.

**Filled** — This checkbox will determine whether the shape is displayed in wire or solid.

**Label** — Places a Label for the Item Shape.

**Justification** — Label Placement with the standard choices Left, Center, Right.

**Draw Line To** — Allows you to draw a dotted line to any item in the scene.

**Selected Color** — Color the item shape will be when it is selected.

**Unselected Color** — Color the item shape will be when it is not selected.

**Text Color** — Color of the Label text.

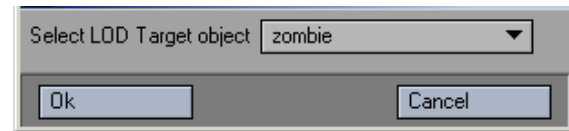
**Opacity** — Determines how opaque the item shape will appear.

## Lscript and Lscript/RT

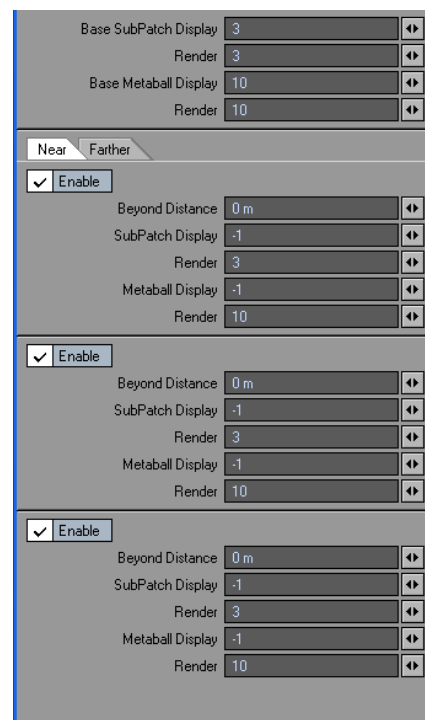
These two items will allow you to apply a Custom Object Lscript.

## Level-of-Detail Mesh Refinement

Adding the **Level-Of-Detail Mesh Refinement** custom object tool allows you to independently change the display and rendering resolution of SubPatch and Meta-primitives based on their distance from the camera. This can save you rendering time when those types of objects are sometimes close to the camera and sometimes far, during your animation.

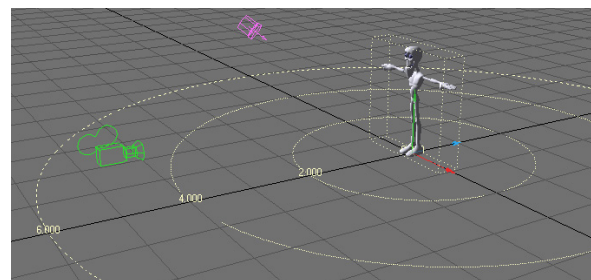


From top to bottom, the entry groups must be in near-to-far order, so the furthest group is last. When the distance from the camera to the object is within each range, the corresponding **Display/Render** settings are used.



A value of -1 disables that (display/render) parameter. It is like deactivating the **Enable** option for a group, but lets you control it differently for individual items in that group.

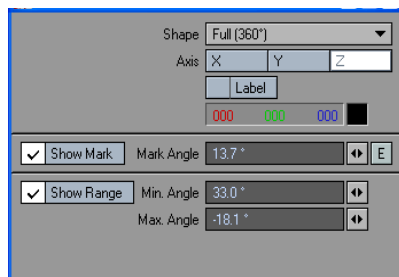
In your viewport, concentric rings are displayed around the object showing the defined distance ranges.





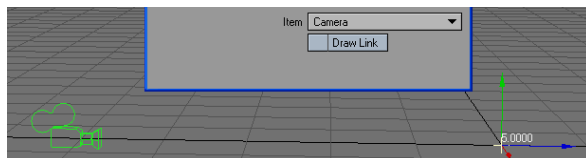
## Protractor

**Protractor** sets up a custom object you can use to measure angles. With the **Shape** setting you can make it a **Full** or **Half** circle. The **Label** option places numerical labels at set intervals. Use the **Show Mark** option to place a tick mark at a set angle. Use **Show Range** to highlight a set range.



## Range Finder

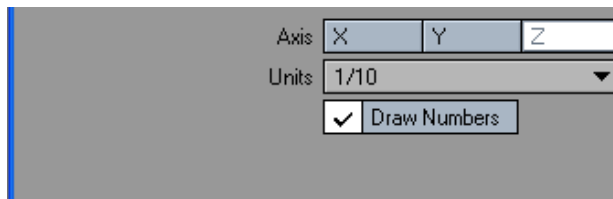
Adding **Range Finder** to an object will display the distance in meters from the selected item in the scene. The **Draw Link** option will draw a dotted line between the items.



Null objects work best. However, if you add this to a regular object, you may want to use the **Bounding Box** rendering level (selected on the viewport's titlebar). Otherwise, the object's surface may obscure the numeric display.

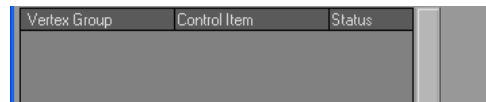
## Ruler

Adding **Ruler** to a null object provides a measurement device. You can change the length of the ruler by stretching the object along the selected **Axis**. The units of measure can be **1/10** of a meter or **feet/inches**. The **Draw Numbers** option controls the display of the numbers.



## SockMonkey

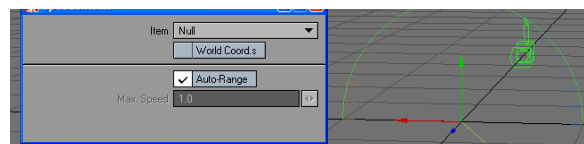
The **SockMonkey** custom object plugin draws bounding boxes for links created in the main SockMonkey displacement plugin. However, if you use the **Auto-add Control Item** button on the main interface, this custom object plugin is automatically added to the created control item, so you don't need to worry about adding this manually. If instead, you use the **Add Relationship** option - where you manually define the **Control Item** - you can add this custom object plugin to get the bounding box.



The **Parent Object** is the main SockMonkey object. The **Pending Relationship** is the related **Vertex Group** defined on the main interface. Once assigned, this dialog will no longer appear when you try to access the options. If not assigned to a link, adding this plugin does nothing.

## Speedometer

Adding **Speedometer** to a null object allows you to measure the speed of an item in meters per second. Choose the item whose speed you wish to measure with the **Item** pop-up menu. Activate the **World Coords** option to measure the actual speed based on world coordinates. (You'll probably want to do this if the item is parented to moving item.)

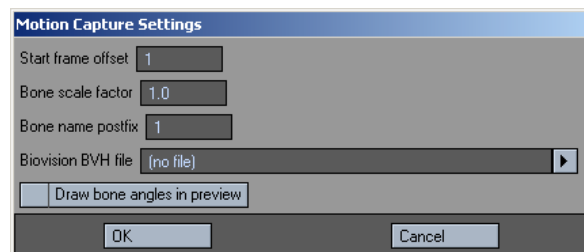


The **Max Speed** setting determines the value when the pointer is *pegged* all the way to the right. Select **Auto-Range** to have the plugin determine the maximum. Note that the pointer can go past the maximum if the speed exceeds it.

## Motion Capture Preview

This custom object plugin can be used to preview BioVision motion data. The preview is fast and accurate. Use it to determine if there were any errors in the motion conversion.

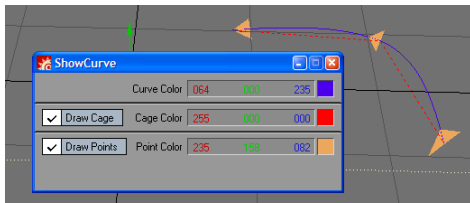
Simply add the MotionCapturePreview custom object to a null object on the **Object Properties Panel**. (If you run the MoCap\_BVH\_Setup generic Layout plugin, a null called "MotionCapturePreviewNull" will automatically be added to the scene with the custom object plugin already applied.)





## ShowCurve

**ShowCurve** displays an object curve in Layout — normally curve objects are a Modeling tool and cannot be seen in Layout. (If there is more than one curve in the object, the first curve is used.)



The color of the curve can be adjusted using the **Curve Color** setting on the **Options Panel**. The **Draw Cage** option, when active, displays the “cage” of the curve by connecting the vertices with dashed lines in a color of your choice. The **Draw Points** option, when active, adds arrowheads to each of the vertices to indicate the direction of the curve.



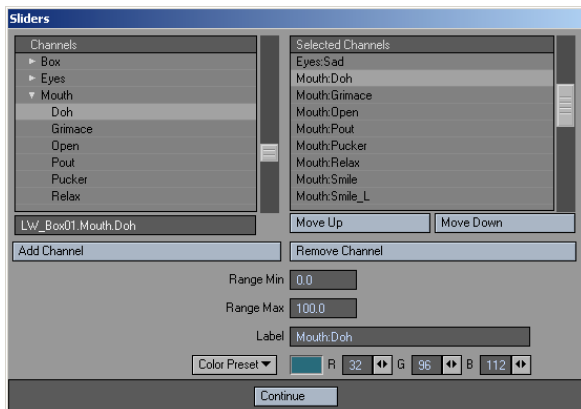
NOTE: ShowCurve was designed to be used with the CurveConform displacement plugin.

## Sliders

Sliders (**Modify> Sliders**) are slider gadgets that are displayed over viewports. An individual slider is tied to a specific animation channel. A slider will indicate the current value of a channel and also let you interactively adjust that channel value.

### To configure your sliders:

Open the **Sliders** custom object settings dialog from the **Object Properties Panel**.



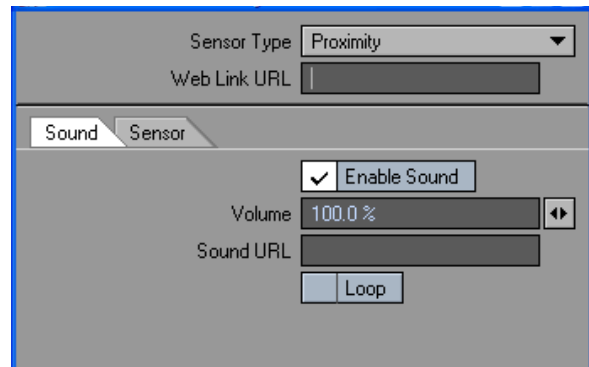
The left window (**Channels List**) will list all of the channels in the scene. To attach a slider to a channel, simply select the channel in the left window and click **Add Channel**. **Selected channels** in the right window can be removed with the **Remove Channel** button.

The **Range Min** and **Range Max** settings define the interactive range of the slider. The underlying channel can go beyond these values, but the slider’s range of control and feedback will be limited to this range. If the underlying channel goes outside of the range, the slider value will turn red. Clicking on the slider handle will immediately change the channel to the slider’s corresponding value.

The description **Label** will default to the channel name, but you may edit that if you desire. You can also set the color used for the slider with the **Color preset** pop-up menu or specific **RGB values**.

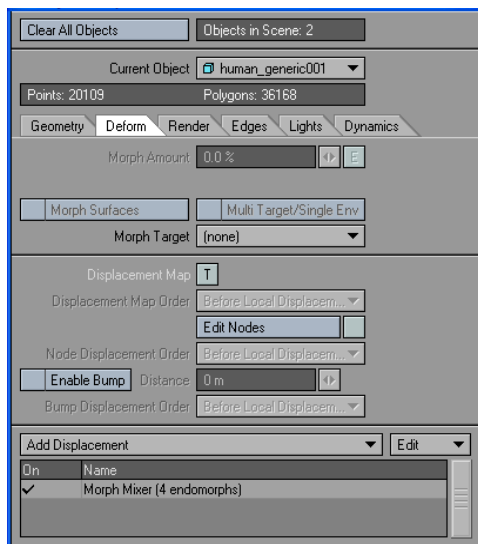
## VRML97 Custom Object

The VRML97 custom object can be applied to an object to make its VRML attributes visible in Layout. It will display URLs, the Proximity-Sensor bounding box, LOD ranges, sound nodes, and links to alternate triggers.



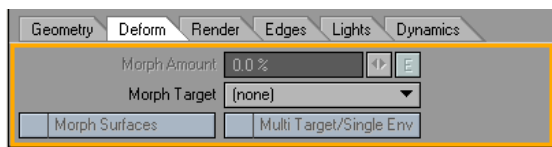


## Object Properties: Deform Tab



### Morph Targets

*Metamorphosing, or Morphing*, causes a 3D metamorphosis from one object into the shape of another object. Morphing requires a minimum of two objects: a beginning object and a target object. These controls are on the **Deformations Tab** of the **Object Properties Panel**.



To be able to morph successfully between objects, the number of points and their order must be identical.



NOTE: Using Endomorph objects simplifies the process by keeping all target data within a single object file. This guarantees the same number of points and helps to maintain point order.

When you enter a **Morph Amount**, the object will be transformed by that percentage into the **Morph Target** object. However, you will almost always animate the amount of morphing over time using a standard LightWave envelope to control the morphing of an object's shape, surface colors, or both, during an animation.



NOTE: If you use an envelope, the Morph Amount will have no effect on the result.

The **Morph Target** is the destination object (the object that the current object will morph into). The target object itself can have its own target and you may create a chain of up to forty targets. (Any number of objects may be morphing within a scene.)

With **Morph Surfaces** you can cause the surface attributes (color, texture, etc.) of the first object in a morph chain to convert to the surface attributes of the second object. Even if additional objects are morphing, only the first and second objects may use **Morph Surfaces**.

## Multiple Target/Single Envelope

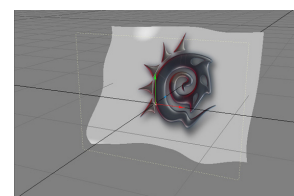
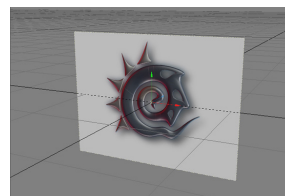
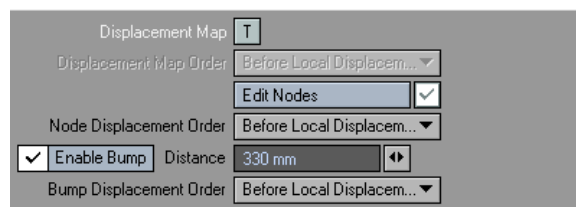
To morph a single object through a chain of multiple targets using only one envelope, you can use the **Multi Target/Single Env** option. The morph chain should be set so that object A has target B, object B has target C, object C has target D, and so on. Once that is set up a morph value from 0 to 100 morphs the A object into the B object. Morph values from 101 to 200 will morph object A into object C. Values of 201 to 300 will morph to D, and so on.



NOTE: Multi Target/Single Env with surface morphing will reflect only a surface change during the first morph.

## Displacement Maps

**Displacement Maps** are similar to **Surface Maps**, which add color and texture to object surfaces. However, instead of affecting the way an object's surface looks, **Displacement Maps** move the points in an object, which changes its shape. The change can be subtle or dramatic, making it look like a totally different object.



Left: Without Displacement, Right: With Displacement



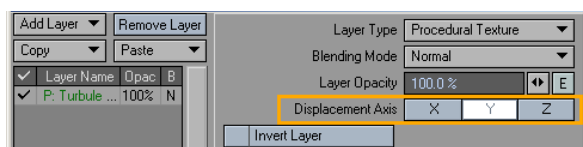
NOTE: Remember that Displacement Mapping is saved in the scene file not in the object file. If you want to load an object with its Displacement Map information, choose File > Load > Load Items From Scene.

You can easily make blowing curtains, rippling water surfaces, and bumpy terrain by applying a **Displacement Map** to an object. Although the object's points move around, the polygon relationships remain and, thus, surfacing information follows the displaced polygons.

### Differences from Surface Textures

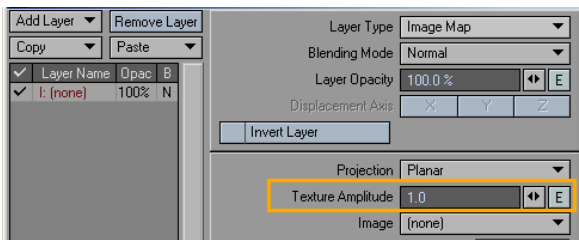
For the most part, textures for **Displacement Maps** are set up in the same way as textures for surfaces. You will, however, notice a few differences.

One difference is that 3D textures (i.e., procedural and gradients) have an option that lets you select the **Displacement Axis**.



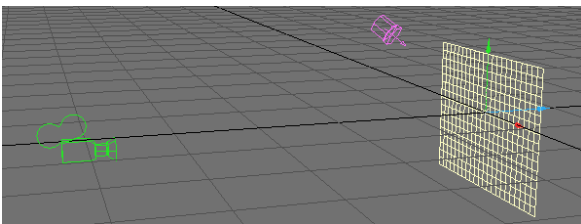


You can also *factor* the effect you would normally get from an **Image Map** displacement using the **Texture Amplitude** setting. Use a value less than 1 to lessen the effect or more than 1 to increase it.



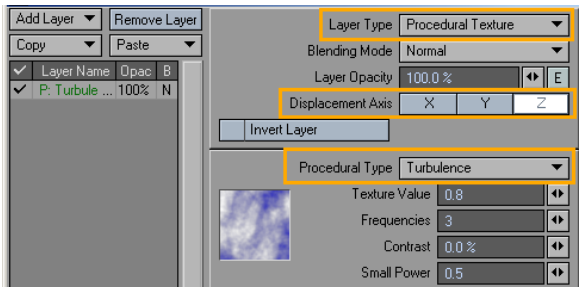
### Exercise: Displacement Map

**Step 1:** Create a plane with about 23 segments in both directions at a size of about 400 mm high and 1m wide and load it into Layout.

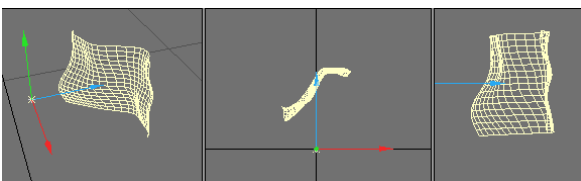


**Step 2:** Press **P** to open the **Object Properties Panel**.

**Step 3:** Click the **Displacement Map Texture** button on the **Deformations Tab**. When the **Texture Editor** appears, change the default **Layer Type** to **Procedural Texture**. Select **Turbulence** as the **Procedural Type** and set the **Displacement Axis** to **Z** since we want the flag to move along that axis.

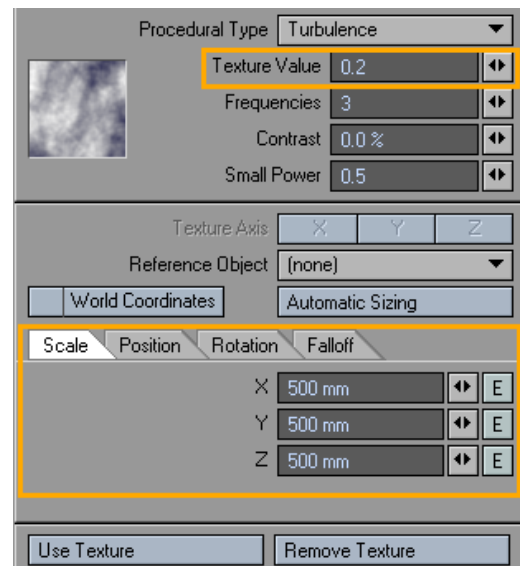


**Step 4:** Study the effect of the **Displacement Map** on the object. Interesting, but a little too exaggerated. Essentially, the size and amount of the displacement is out of proportion for our object. It's like dropping a boulder into a padding pool. What we really need is a small stone.

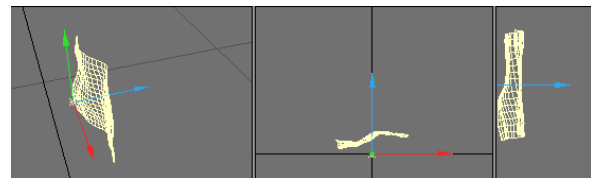


**Step 5:** Go back to the **Texture Editor** and change the **Texture Value** to .2. This will reduce the general amount of displacement.

**Step 6:** Now, this object is only about 400 mm wide. To get smaller ripples, change the texture's **Scale** to 500mm for all axes.



This tones down the wave in the object.

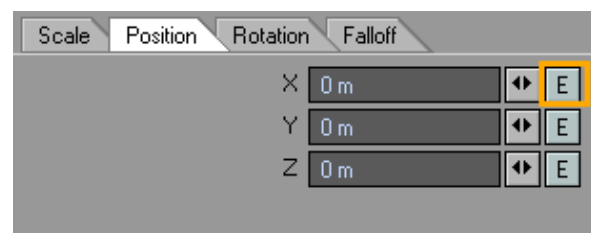


**Step 7:** Click the **Play** button in the lower right corner of the main interface. The display updates automatically and we get better feedback as we animate the texture. (Note: the flag doesn't move yet.)



**NOTE:** If you don't see anything, stop playback and choose **Display > Options: Display Options** to open the **Display Options Tab** of the **Preferences Panel**. Make sure the **Bounding Box Threshold** is at least 5000 (a little more than the number of polygons in the object). Start Play again.

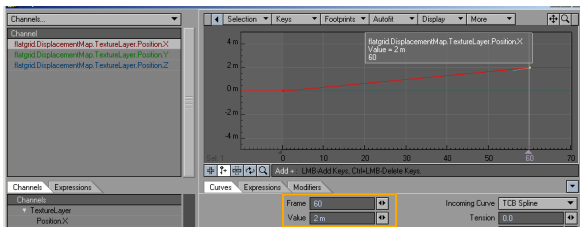
**Step 8:** You can animate the wave by animating the texture's **Position**, which is the center of the texture. Click on the **Envelope** button for the **X** position field.







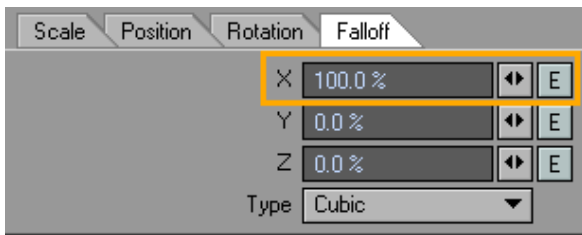
**Step 9:** The **Graph Editor** will appear. Select only the **Position X** for the texture and add a key at frame 60 (the last frame in the scene) with a value of about 2m. (You may need to adjust the graph zoom a little. Just drag on the magnifying glass icon.)



if your scene is still playing, you should see your flag waving. We are moving the texture two meters over 60 frames.

**Step 10:** Now of course, if this were a real flag, it would be attached to a pole and the left side would not wave as much as the right. We can simulate this by moving the center of the texture and applying falloff to the effect.

Set the texture's **Falloff** to **X = 100%**. This sets the amount to reduce the displacement per default unit, which should be one meter (**General Options Tab** of the **Preferences Panel**). Now, the effect is in full force at its center and reduced 100% at one meter from the center (the left edge, since the flag is 1 meter wide). If you want a little more movement towards the left side of the flag, reduce the **Falloff** value.



**Step 11:** The flag is one meter wide, but its local Origin is at its center — you could figure this out by looking at it in Modeler or where it loads by default in Layout. Thus, the right edge is at 500mm. This should be the starting point for the texture's X Position.

Back in the **Graph Editor**, switch to the **Move Mode** and drag the first key (at frame 0) up. As you drag, look at the layout viewport. You should see the texture move to the right (along the X axis). Since you have falloff, as you move the key, the falloff will begin at a different point — eventually the right edge of the flag — and fall off completely at the left.



**NOTE:** When you animate the Position values, the Falloff center is based on the Position at frame 0.

**Step 12:** To add a little more variance to the wave, you could add keys to the Z position of the **Displacement Map**.

## Displacement Map Order Menu

Determines the sequence of Displacement Map evaluation order:

- After Morphing
- Before Bones
- Before Local Displacement
- Before World Displacement

## Displacement Mapped Objects

Generally, when you construct objects in Modeler, you try to minimise the number of polygons that are used. However, if you know you will use the object with a **Displacement Map**, you may need to subdivide certain or all areas of an object into a greater number of polygons to create more bendable areas. All polygons should also be triangles. **Displacement Mapping** can cause four-plus sided polygons to become non-planar, which may result in rendering errors.

## Displacement Mapping Versus Bump Mapping

**Displacement Mapping** is different from **Surface Bump Mapping** in that the object's geometry actually changes, where **Bump Mapping** takes the change with shading.

If you use an **Image Map** as a **Displacement Map** (instead of a procedural texture), pure white areas will displace an object's points 100 percent of the value of the **Texture Amplitude**. Pure black areas will not displace at all, and in-between values will be applied relatively.

Note that the **Texture Amplitude** value has an enveloping option, which can animate a displacement over time.



**HINT:** The Ripples displacement texture will not actually raise the surface, but rather it spreads the points out across the surface. If you need to create water ripples that appear to rise when you view the surface edge closely, try using the Fractal Noise texture instead. Fractal Noise will actually displace points out from the surface.

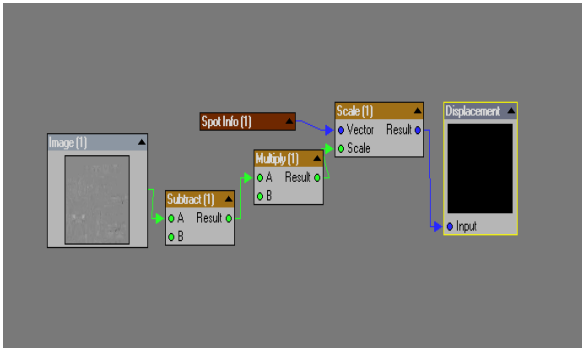


**HINT:** It is recommended to have mipmapping turned off when using displacements.



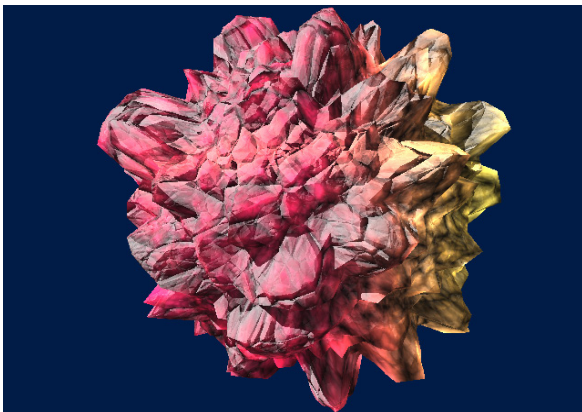
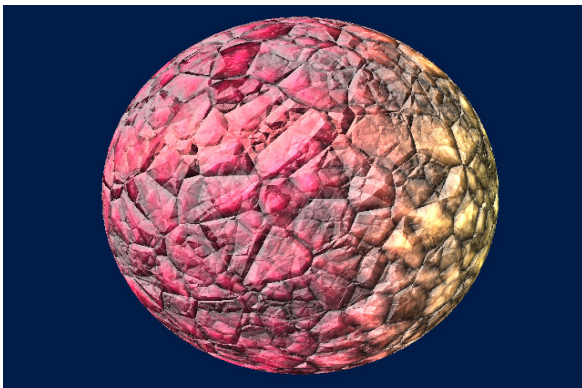
## Edit Nodes

When checked on, you can use the Node Editor to displace your object..  
When you open the Node Editor you will notice a new node, Displacement, and this is the hook for your displacements.

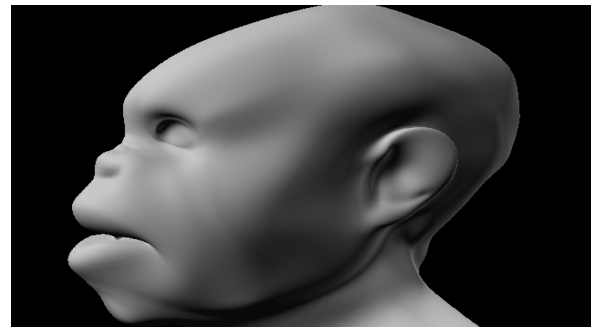


Example of Node Editor with Displacement Map

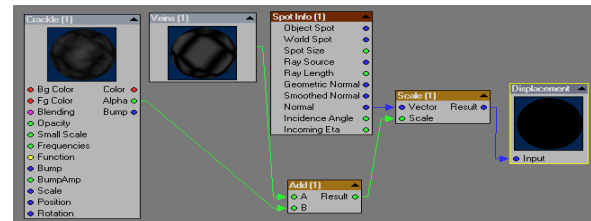
Examples of Displacement Node:



Nondisplaced ball; Bottom, Displaced ball



Top: Nondisplaced object; Bottom: Displaced object



Note: In the image above, there is a subtract node, then a multiply node. Subtract - In LightWave black is usually used as no displacement and white as full displacement, but ZBrush is different. ZBrush uses black as a negative displacement (displace in the opposite direction of white), mid grey as no displacement, and grey as full displacement. To do this subtract 0.5 so that instead of the image going from 0 to 0.5 to 1, it goes from -0.5 to 0 to 0.5. Multiply - Instead of making the image go from -0.5 to 0 to 0.5, we would really prefer it to go from -1 to 1, so multiply the image by 2. However this node also has the added bonus on controlling how far we want to displace the vertices, especially because we will displacing them by a default distance of 1 meter!

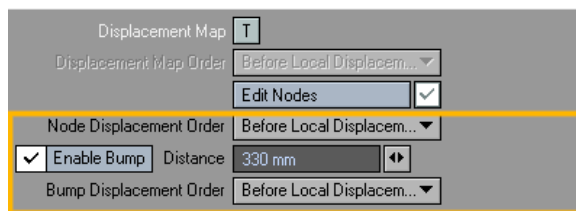


NOTE: In order for the Displacement node to function as expected, use a Vector as the input. Otherwise you may get unexpected results. You should also use the Spot Info node with the Normal output.



## Bump Displacement

The **Bump Displacement** option uses the bump texture on a polygon and vertex, and applies it as a displacement texture. The direction of the displacement is set by the vertex normal, and the amount of displacement is set by the **Dist** value.



Bump Displacement is great because it creates actual geometry deformations from bump shading. (Remember, bump shading by itself does not affect the geometry.) The result is better looking bump contours, shading, and shadows.

This feature is especially good on SubPatch objects because their detailed geometry allows the displacement to more closely match the bumps and surface contours, particularly when Render SubPatch Level (Object Properties) is set to high values.

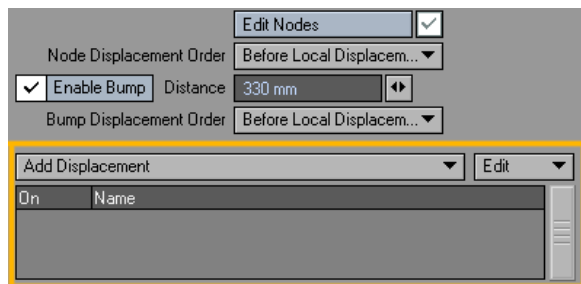
### Bump Displacement Order Menu

Determines the sequence of Bump Displacement evaluation order:

- After Morphing
- Before Bones
- Before Local Displacement
- Before World Displacement

## Add Displacement

Use the **Add Displacement** pop-up menu on the **Deform Tab** of the **Object Properties Panel** to apply a displacement plugin to the current object.



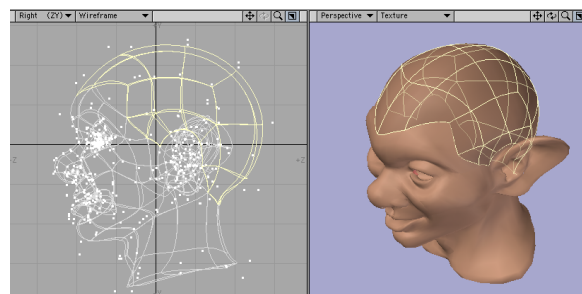
## Sasquatch Lite

**Sasquatch Lite** is one of the hair and fibre tools inside of LightWave. There are two kinds of **SasLite** effects: **fur** and **long hair**. Fur is simple, but limited in scope. Long Hair is more controllable, since you make guide chains (2-point polys) to tell it where to go, but it takes a lot more setup time.

Rendering **SasLite** is a two step process, no matter which kind of fibres you pick, because it's really a separate rendering. It uses its own render engine which is applied to the LightWave render as a **Pixel Filter**. We'll go into that below, but be sure you don't skip that step, or you won't be able to render any fibres at all.

We'll cover **Fur** first. **SasLite** applies its fur effects to surfaces; no maps of any kind are required. This lacks subtlety, but it does have one advantage; you don't have to **UV Map** your models. But you do have to plan ahead when you are making them, and tailor them to **SasLite**.

If you don't want fur to completely cover your model, you can easily accomplish that by choosing the area where you want the fibres to grow, and giving them a different surface name. Then, in **SasLite**, you can specify which surfaces fibres will grow on.



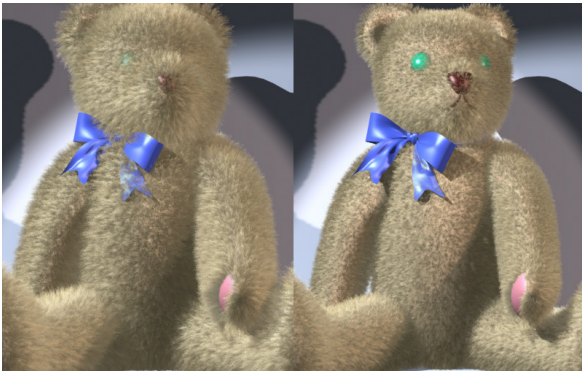
Select the scalp polys, and give them a "Hair" surface name.

As you do this, you need to be aware that the fur effect won't stop cold at the edge of the surface; it will leak out a bit into the surfaces that bound it. How much it leaks out depends on how long the fibres are, the Clump Size, and so on. The longer you grow them, or the larger the clumps, the more fibres will begin in "non-fibre" surfaces. So, if you know you are going to have short fur, you'll want to choose polys that are close to the desired hairline. If you are choosing longer fur, and you have the polys to work with, move back from it just a bit.



Notice the long hairs growing from the Troll's face! (Density and Coarseness have also been changed, to compensate for the change in Length.)

Secondly, the fibres grow right out of the surface that you have chosen for fur, without paying any attention to surrounding geometry surfaces. So, if part of your model is resting on a surface that's getting fibres, they'll grow straight through it. The work-arounds are to use short enough fibres so that the interference is minimal, or to put geometry with non-fur-bearing surfaces behind those parts.

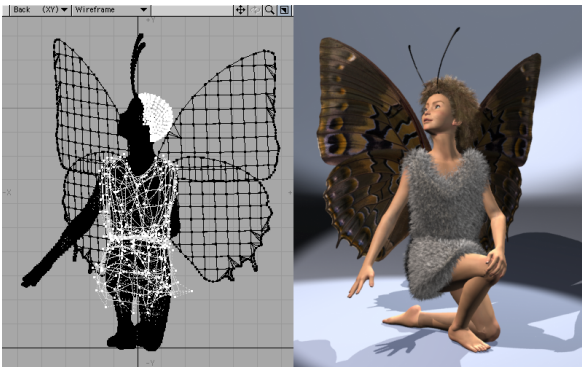


*Fur that's too long obscures the features and bow. (Note fur-less shadows.)*

There are other built-in limits, too. For instance, **SasLite** will not put fur or hair on objects with more than 25,001 points. (**SasLite** will warn you if you overrun the limit when it attempts to add the Pixel Filter, and fails.)

If you are using subpatches in the object **SasLite** is objecting to, try reducing the Render SubPatch level.

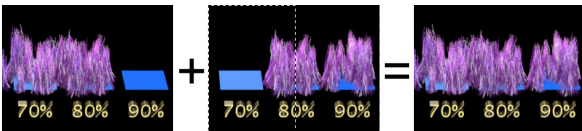
If you want to use an object with more points, cut and copy the polys for the wig, beard, back hair, or whatever you want to use, and put them in a child layer. Make them 1% or so smaller than the original, so the hair will grow out through the original, not float over it. Then, in Layout, set the **SasLite** Displacement on the Fur/Hair layer.



*This DAZ model has too many points to use SasLite; but a separate Scalp layer makes it possible.*

You will also find that you are limited to no more than 8 separate instances of fur or hair in a scene. If you need more, you'll have to make several renders, and then composite them in Post. Since the calculations for each surface will result in identical fibres every time that surface is rendered, this isn't as difficult as it sounds, even if two fur-bearing surfaces are in contact with each other.

Simply render the middle fur when you render each end. Then in your Post program draw the mask right through the middle surface. Everything will match perfectly.



*It's easy to match surfaces if you render the middle one, as I did for many of these illustrations.*

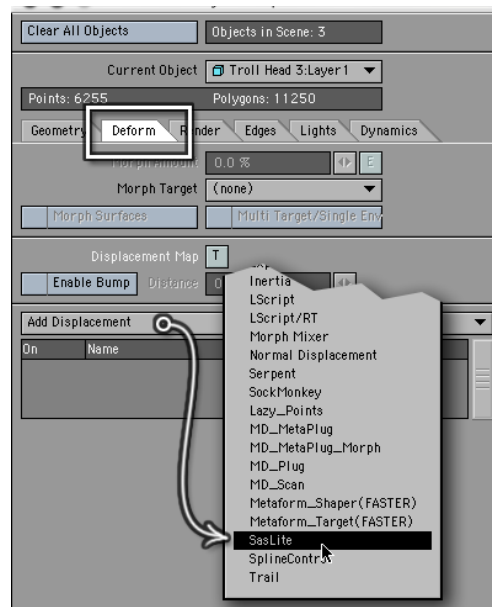
It also helps that the fibres show up perfectly in the Alpha channel; so you can make the underlying surfaces transparent in Alpha, and easily isolate the fibre effects.



*Fur and Hair are perfectly rendered in the Alpha Channel.*

The actual effect is applied to the model in Layout. To do this, go to **Object Properties** (click on the **Objects** button at the bottom of the screen, then click on **Item Properties** button, and choose the object you want to apply the effect to from the list: or just hit **Shift + O** then **P** and then choose your item.)

Choose the **Deform Tab** in the **Object Properties** dialog, click on the **Add Displacement** button and choose **SasLite** from the list. Double click on the name to open the **Options Panel**.

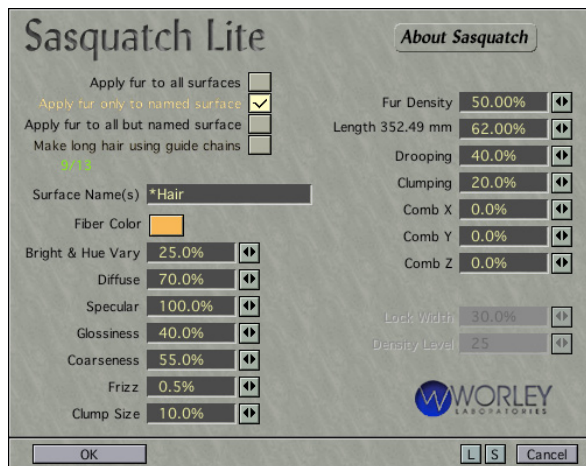






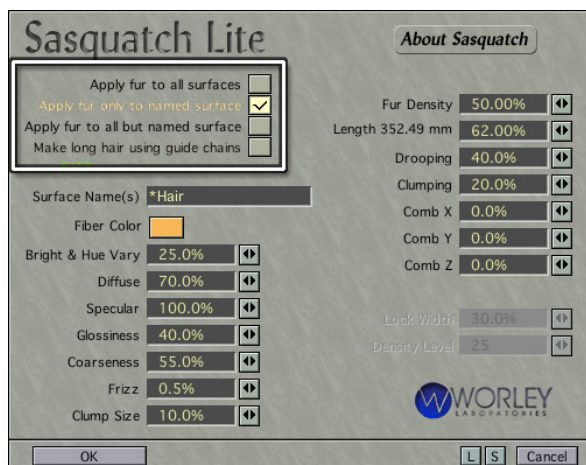
Choose **SasLite** from the **Add Displacement** menu of the **Deform Tab** in the **Object Properties** dialog.

That will open the **SasLite Options** dialog.

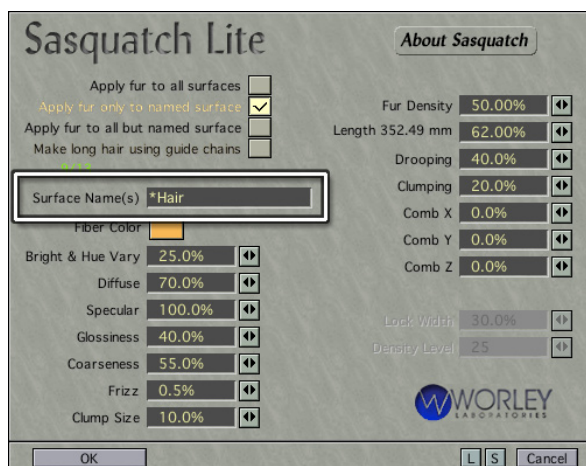


### The SasLite Options Dialog Box

The first thing in the panel is a group of radio buttons that allow you to add fur to all surfaces, only to named surfaces, to all but named surfaces, or to make long hair using guide chains. (They look like checkboxes; but choosing one disables any others.)



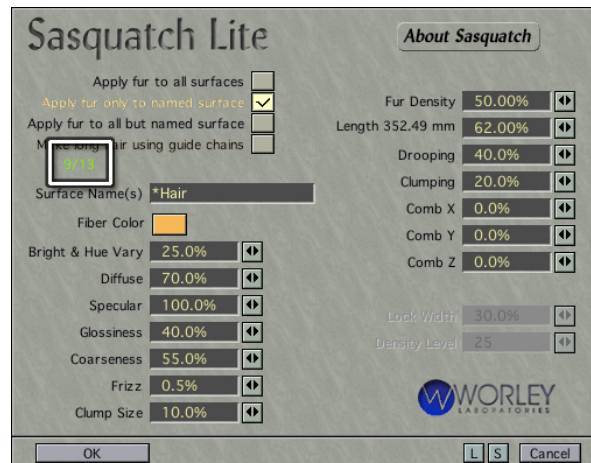
Choose the one you want to use; they are self-explanatory. If you choose anything except "Apply fur to all surfaces," you'll have to type the name of the surface into the Surface Name(s) Checkbox. Be warned; there is no drop down list, and once this dialog is open you can't move it or do anything else until you close it; so you'll have to remember what you called the surfaces. (Or open the **Surface Editor** before you open this dialog, so you can see them.)



Type in the Name of the surface you want covered with fibres. Because of this, it's probably best to get in the habit of calling them all with variations of the same name; for instance HeadHair, ToeHair, BackHair etc.

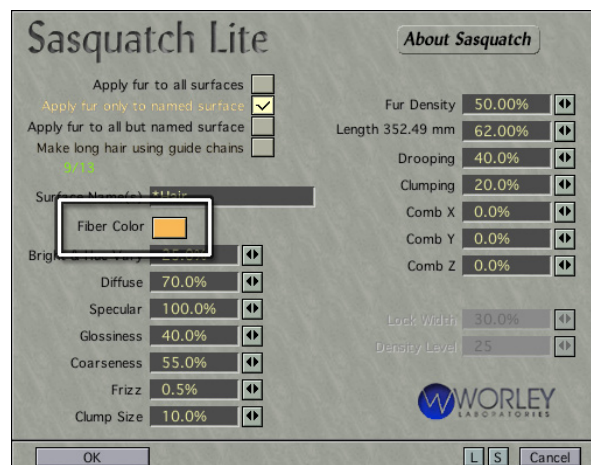
Separate the names with commas when you type them in, or use the asterisk (\*) wild card character to catch all the combinations at once. For instance, '\*Hair' would put fur on all the surfaces named above. (It is case sensitive)

If you look carefully, you'll see a fraction above that text field. That shows you the number of surfaces **SasLite** is going to apply this effect to, over the total number of surfaces in the object. It's good to get in the habit of checking it, just to make sure that you haven't misspelled something. If **SasLite** doesn't find a surface name you typed, it will just skip it. So, for instance, if you forgot to type the asterisk in the example above, the fraction would show 0/12 surfaces, not 3/12. Getting in the habit of checking can save time.



Check this fraction; it will tell you how many surfaces are getting the effect.

Below that is a color swatch, so you can choose the fibre color. There can be only one; but if you really need two or more, you can add other instances. Just don't give them exactly the same parameters, or the fibres will overlap and be invisible. (There is no random seed in **SasLite**.) If that does not work, simply add another set of polys, and put the second fur on them. It will grow through both, of course.

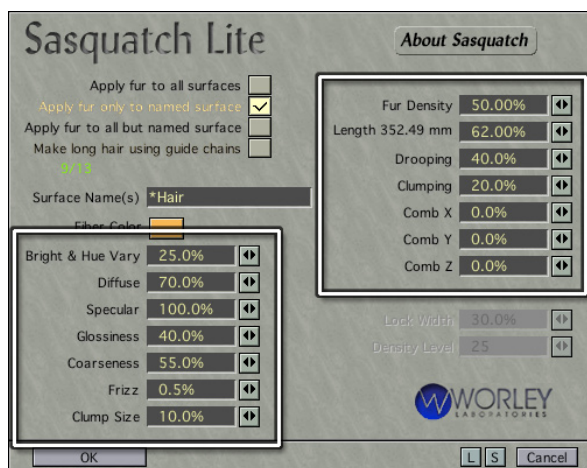


Choose a color. It generally works best if the color you choose is brighter than you actually want the fibres to be. Between the lowered diffuse needed for glossy hair, the white highlights, self-shadowing, etc. you're likely to find the hair looks darker and less saturated than the color you picked.

Click on the swatch to open the color picker for your system, and pick a color.



Now we come to the list of percentages.

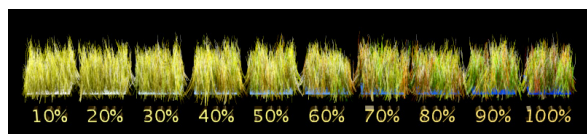


The rest of the dialog is percentages of various attributes. The first thing you need to understand is that many of these percentages influence the others. For instance, longer fibres are automatically coarser and so appear denser than shorter ones. So, if you like the coarseness and density of your fibres, but want them longer, you can't just change the length and get the effect you're after. You need to also reduce the coarseness and increase the density to compensate for the increased length. By how much is dependent on the size of your model and its complexity. You'll just have to play with it until you get a feel for it (Fortunately, that's not as hard as it sounds.)

The bright side of this interdependence is that, if you get a setting you like, it's going to look about the same on all the models you apply it to, no matter what size they are. This is because the smaller models will automatically have the shorter, finer hair they need to get just the same effect as the larger ones. It just takes some getting used to.

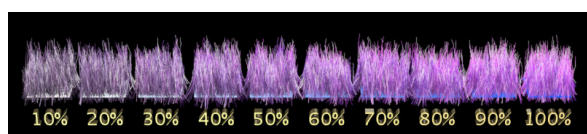
So, that being said, let's take a look at the options.

First is **Bright & Hue Vary**. This setting makes up for the single color available in **SasLite** by allowing you to specify a percentage variation in the Hue and Value of your fibres. It's fairly self-explanatory; but you do need to remember that the Saturation doesn't vary at all. If you need varying degrees of Saturation, you'll have to use multiple instances (and change the values in some of the fields so the fibres aren't superimposed in the render.) You might also want to know that even 100% variation doesn't really give you all the hues in the spectrum; only about half of them. So, for instance, if you choose yellow for the fibre color, 100% Brightness and Hue variation will give you fibres that are various shades of yellow, green, orange, and red; but not blue or purple.



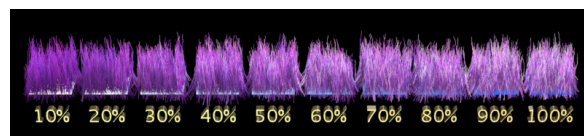
Variations in Bright & Hue Value

The next option is **Diffuse** value. This works just the way it does in the rest of LightWave. The higher the value, the more the lights in your scene will affect the fibres. Decreasing this value will make the highlights on the fibres appear brighter, but will also make the colors look darker and less saturated. This is one of the percentages that you can push way past 100%, if you need the fibres to be more reactive to light.



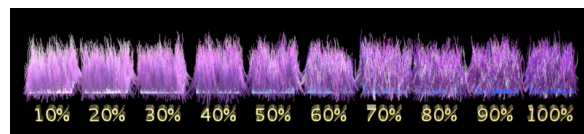
Variations in Diffuse Value

**Specular**, of course, gives you brighter highlights, just as it does in LightWave itself. The higher the number, the more white highlight is applied to the fibres; so, naturally, lower numbers give color that looks deeper and more saturated. Like anything else, it's a tradeoff. This value can exceed 100% as well.



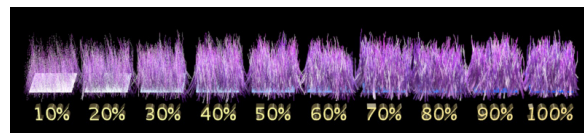
Variations in Specular Value

**Glossiness** makes the highlights more concentrated, so the fibres appear to be shinier. Because of this, the control will be disabled if the Specular value is 0%. The lower the value, the more diffuse the highlight, the more matte the fibres will appear to be, and the lighter the color looks. The higher the value, the sharper the highlights, and the more the color appears to be the one you chose. This can go over 100% as well.



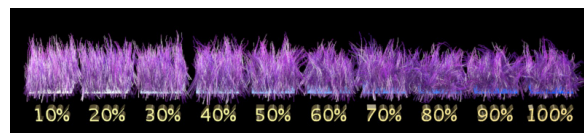
Variations in Gloss Value

**Coarseness** allows you to decide how thick the individual fibres are going to be. This lets you make them fine enough for velvet or a collie dog, or thick enough for grass stems. As mentioned earlier, it's dependent on the length of the fibres, as well as on the size of the model you are growing the fibres on. Be careful; low values (fine fibres) can dramatically increase render time. You are better off starting with the highest value you think you can get away with, and lowering it if necessary. You should also be aware that the finer the fibres are, the less dense they appear; so if you are decreasing the Coarseness, you'll want to increase the density to maintain the same apparent coverage, and vice versa. You can push the Coarseness value to 400%.



Variations in Coarseness Value

**Frizz** bends the fibres at all angles. (It's more "totally wild" than "frizz.") The amount applied to each fibre clump, and to a lesser extent to the fibres within each clump, varies a bit, so they'll look more realistic. The direction of the bends also varies; so high values can make the fibres look like an extreme case of "bed head."

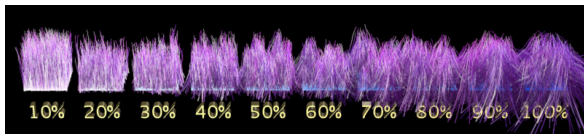


Variations in Frizz Value



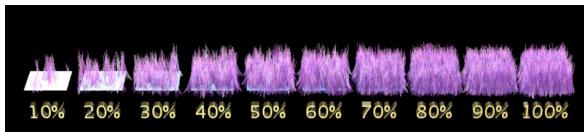


**Clump Size** determines how wide a clump of fibres will be. (The number of fibres in the clump is determined by the Clumping control, on the other side of the panel. If the Clumping is 0%, Clump Size is disabled.) As you can see from the image, it also is one of the things that determines how far from the surface the effect will spread. If you have a large Clump Size, **SasLite** will get the fibres to make it so, even if it has to pull them from thin air!



Variations in Clump Size Value

**Fur Density** determines how thick the fibres will grow on the surface. Once again, it's interdependent with length and with the size of your model. Higher percentages yield more fibres, and lower ones less. High values also add to render time, so it's a good idea to use the lowest number that gives you decent coverage. Density can be pushed to 250%.



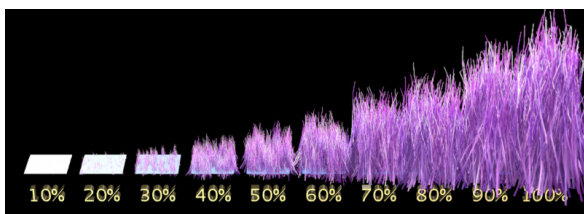
Variations in Fur Density Value

The other thing you have to understand about Density (and Length) is that these percentages are not linear. So, if you choose 50% of these effects, that's not twice as much as 25%; it's closer to four times as much.

It's easiest to see this in the Length field (on the right) because you are given a hard number as well as a percentage. If you try it, you'll find that if 1% is, say, .06 mm, then 25% will be 40.34 mm; 50% will be 161.38 mm; 75% will be 363.10 mm, and so on.

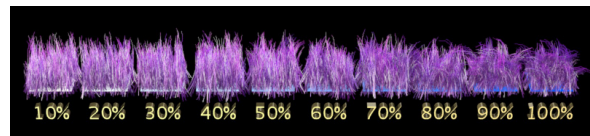
What it means is that the higher the numbers in these fields, the greater the difference will be if you increase the number. The difference between 10% and 15% in the above example is only 8.06 mm. But the difference between 70% and 75% is 46.8 mm. Bear that in mind as you choose percentages, and you won't have to make nearly as many test renders.

**Length** determines the length of the longest fibres - once again, it's variable. It also determines the coarseness, the apparent density, and how far they spread from the surface. All of this is clearly visible in the image shown here. Notice how much longer the long fibres are below the surface, as well. These are all plain single polys; but this might be important if you are growing hair on a thin surface. Also notice the curve in the progression. (There are fibres growing on the 10% surface; but they are just too short, fine, and sparse to see.)



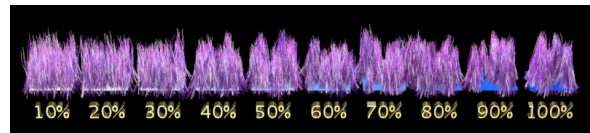
Variations in Length Value

**Drooping** determines how much the fibres droop towards the negative Y axis. Higher values droop more than lower ones.



Variations in Drooping Value

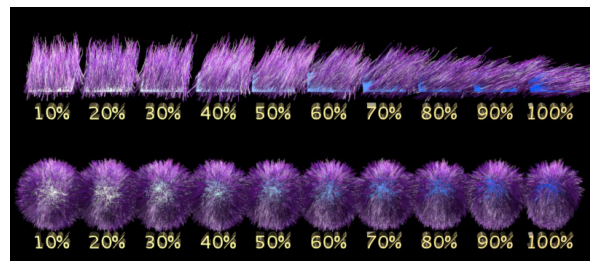
**Clumping** determines how many fibres will be in a clump. Once again, this appears to be internally variable. In other words, the clumps won't be uniform; which is good, of course. The higher the value, the more the hairs are gathered together, and so the more room there appears to be between clumps.



Variations in Clumping Value

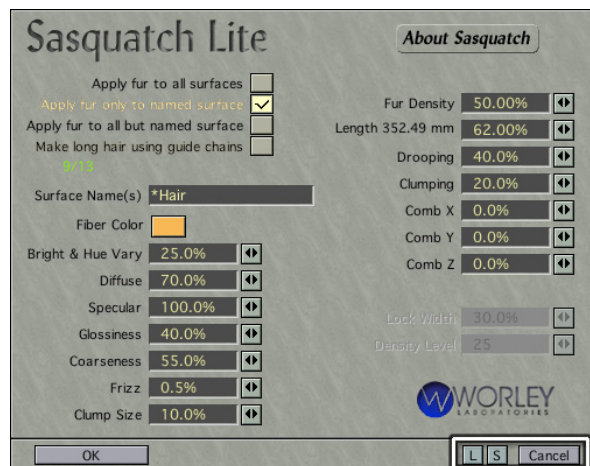
**Comb X, Y or Z** slants the fibres towards the model's own X, Y or Z axis (not the axes of the world, so the appearance of combed models isn't affected by rotation in Layout.)

However, it won't do this if the normals of the polys the fibres are growing from are too closely aligned with the axis you've chosen; so be aware of that when you try to use it. Also, of course, it matters whether you choose positive or negative values here. To comb the fibres downwards, for instance, use a negative percentage in the Comb Y field. (These three are the only fields that will accept negative percentages.)



Variations in Combing Value, showing the X and -Y axes.

In the bottom right hand corner of the Dialog Box, you'll find two tiny buttons labeled L and S. Those are to Load saved parameters, and to Save those parameters in the first place. When you click on them, you'll get a normal browser for your system that lets you Save a file and Load files you have previously saved.



Buttons to Load and Save parameters, and to Cancel changes.



When you Save a set of parameters, you are allowed to choose both a name and an extension for it. On a Mac, you can safely ignore the extension. On a PC, you might want to choose .txt, since these are really just text files with lists of the parameters and values chosen. If you open one in a text editor, they're quite clear; you can edit them by hand if you are so inclined.

When you load one, all the parameters are loaded, including whether you want fibres on all surfaces, only named surfaces, etc., and the name of the named surfaces if you are using them. Yet another good reason to be consistent when naming surfaces you intend to use fibres on!

You can click on Cancel if you decide not to change the parameters after all. The **OK** button, which allows to accept the changes and return to Layout, is in the lower left corner.

Unlike some of the Worley dialogs, you move from one field to another by tapping the Enter key. If you don't tap it, **SasLite** won't keep the value. This is consistent with the other requesters and dialogs in LightWave.

As mentioned before, you can put up to eight instances of **SasLite**, or eight different sets of fur and hair parameters, in a single scene. Besides allowing you to have different types of fibres on different objects in the scene, this allows you to get some very interesting effects by putting different fibres on the same object. For instance, you can easily make a fur with a soft, dense undercoat and long silky top fibres, as seen on this troll.

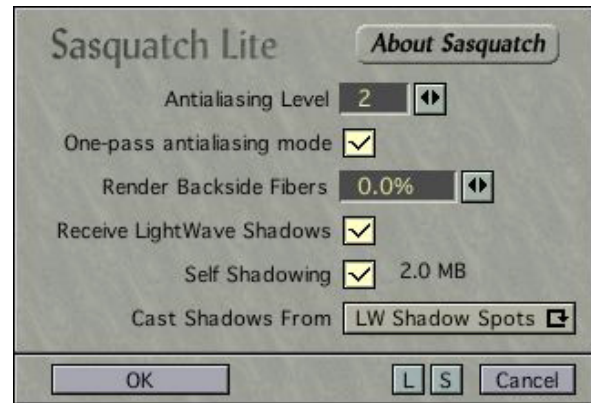


It's easy to make complex fur using multiple instance of **SasLite** on the same object.

Once you have all that set up, you need to tell Layout to render the fibres. As mentioned before, all the fibre rendering takes place in **Sasquatch Lite**'s own render engine, and is applied to the finished LightWave render as a **Pixel Filter**.

So you need to open the **Image Processing** dialog (type **Ctrl+F8** or choose **Window > Image Processing**) and choose **SasLite** from the **Pixel Filter** drop down menu.

Double click on its name in the list to open the **Sasquatch Lite** dialog for the Pixel Filter.

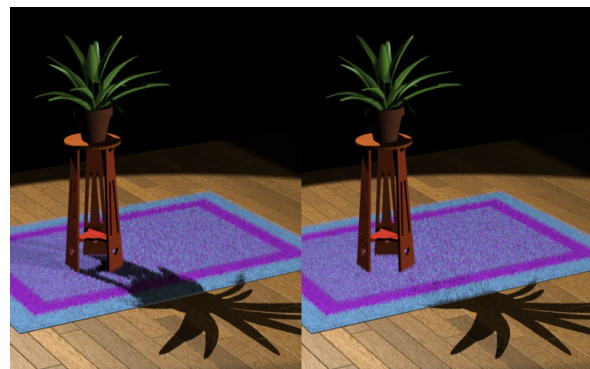


The Sasquatch Lite Pixel Filter Options Dialog

The first option is the **Antialiasing Level**, which is completely independent of Layout's antialiasing. (It will work even if you have AA turned off for normal renders, so be careful. If you don't want to take the time to AA the fibres, you have to turn it off here, not there.) The default number is 2, which is probably right for most uses.

Below that is a checkbox for **One-pass antialiasing mode**. If it's checked, **SasLite** will only calculate the fibres once, and then just reapply them after each antialiasing pass the LightWave render engine performs. This can save a lot of time, since **SasLite** will just make exactly the same calculations, with exactly the same results, each time. Leave it checked since **SasLite** calculations may take several minutes. If you have One-Pass antialiasing checked, though, fur will not receive any motion blur, which LightWave calculates by making multiple passes..

**Render Backside Fibers** allows you to render fibres on surfaces not facing the camera. This becomes important when you are rendering fibres that go around a curved surface, like someone's head. In real life, you can see the fibres that extend beyond the curve. **SasLite** allows you to choose how many of those to render, so you can balance your image between the realism of seeing enough of those fibres to believe they are all there, and the time it would take to actually calculate them all. The next option allows the fibres to receive LightWave **Shadows**. This can add to the realism of your scene, but you need to be aware that only shadows cast by Spotlights will be used. This is because **SasLite** uses **Shadow Maps** to render the shadows. It makes these itself, whether the spotlights are set for **Shadow Maps** or **Raytraced Shadows**. So, when you are setting up a render with **SasLite**, it pays to tighten the beam of the spotlight so you don't waste any of the map, just as you would normally when using **Shadow Maps**.



Receive Shadows enabled and disabled.





**Self Shadowing** allows the fibres to cast shadows on themselves. However, it doesn't allow the fibres to cast shadows on other surfaces that use **SasLite**, so it's not a good way to get fibre shadows on other objects (If you want that, you really need to get the full version of Sasquatch). If this is disabled, the fibres will appear to be much lighter. Also, if this control is disabled, and "Receive Shadows" is enabled, it's possible to get strange terminator lines on your fur where the shadows occur, as shown near the troll's face in this illustration.



You can choose whether to have the fibres cast shadows on themselves or not.

The final control, **Cast Shadows From**, tells **SasLite** whether to cast self-shadows from All Spotlights, or only those which have shadows enabled (LightWave Shadow Spots.) This control has no effect on the ability of the fibres to receive LightWave shadows (that's determined by the **Shadow** setting for the Light in question) and is disabled if **Self Shadowing** is not checked. Generally, you can leave it at LightWave **Shadow Spots**, because every light calculation adds some time; but you may want to change it to **All Spotlights** if you feel you need more definition in the fibres (and you are using some spotlights with shadows disabled.).

We've now covered fur in detail; but what if you want your character to have long, flowing, styled tresses instead of an unruly mop or a buzz-cut?

In that case, you have to use **Long Hair Guides**.

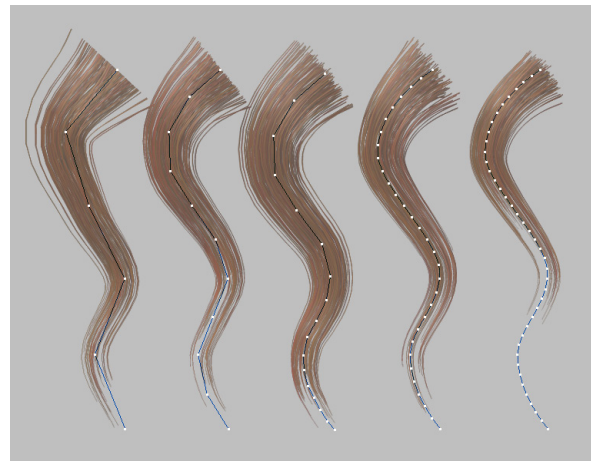
Setting them up isn't difficult, (although it can be tedious.) All you have to remember are a few rules.

1. Long hair guides are 2-point poly chains.
2. The end the hair is growing from must be attached to a surface with a different name.
3. The guide chains should be on their own layer(s) so you can choose not to render them. (The fibres will render, even though the chains are set to "Unseen by Camera" or 100% dissolve.)

The poly chains are quite easy to make; you can either make points, and then connect them or use **Rail Extrude** on a 1-point poly.

You may have heard that the root point of the poly chain must be named "root." but you can actually name it "LeftScalp," or "HindLegRoot" or "George." It just has to be a different name than the rest of the chain. It also doesn't have to be a point. You can grow your chain right out of the scalp, if you are so inclined!

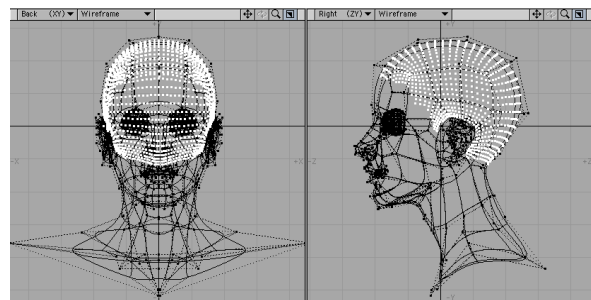
Also, you need to bear in mind that the hair that **SasLite** grows will always be shorter than the hair guide you are making. How much shorter depends on the number and distribution of the points in the guide. The more there are close to the end (away from the root,) the longer (and thicker) the hair will be. But **SasLite** won't render anything beyond 30 polys. It won't warn you, it just won't render. So, if your hair is much shorter than you expected, you might want to count those polys. (Just select one guide chain, and check the Polygon Statistics)



The length of the fibres is dependent on the points in the guide chain.

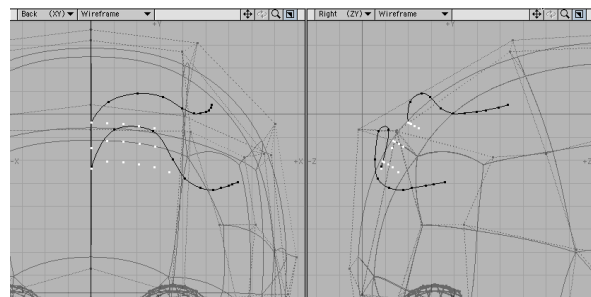
Constructing the chains is really Modeling, so we will just gloss over it here.

We will use a variation of the method shown by Mike Green at <http://www.newtek.com/products/lightwave/tutorials/modeling/saslite/>. Basically, you need to copy most of the head into another layer, freeze it, and kill the polys (k). This will give you a cap of points to work with. You then need to delete all the points that fall outside the hairline, and you're ready to extrude.



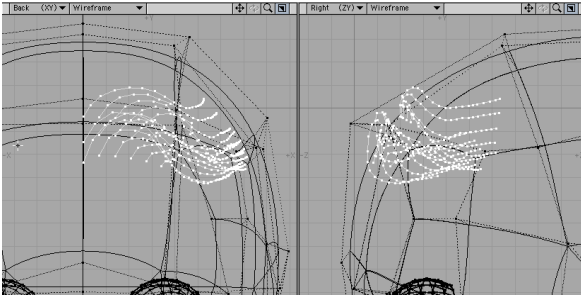
Freezing the scalp and killing the polys yields a nice set of hair roots.

Taking a section of the head at a time, depending on the hairstyle, you can copy a group of the points into a new layer, use Points to Polys to make them into 1-point polygons, and give them the "root" surface. In a different layer, you can make curves where you want the hairs on the edges of this group to lie, using all 3 dimensions, and as many curves as it takes. (Usually 2 or 3, but sometimes more.)



Use curves to define the hairs on the outside of the group.

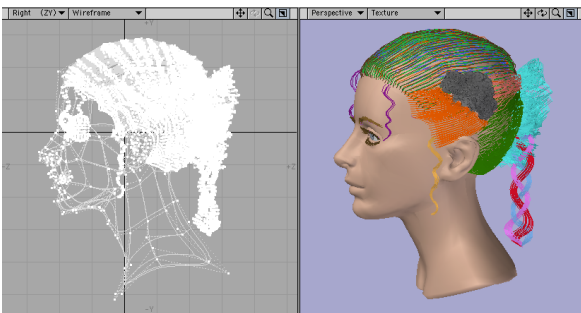
Then you need to put that layer in the background, the Root Polys layer in the foreground, copy the Root Polys to the clipboard, and Rail Extrude the hairs. That should give you a thick bunch of hair guides. You can give them whatever surface you want for that part of the hair by pasting the Root Polys back in, and Merge Points (**M**) making sure that "Keep 1 Point Polys" is checked.



The result of using Rail Extrude.

If this method doesn't give you the expected results, you can select a row of points from the Point Scalp layer, copy and paste them into a new layer, make a curve, and clone a single hair along that curve using Rail Clone (**Multiply > Duplicate > Rail Clone**). You can then adjust each hair if necessary. This will take a lot longer, but also allows more control.

If necessary, you can use various **Modify** tools to tweak the hairs so they look just the way you want.

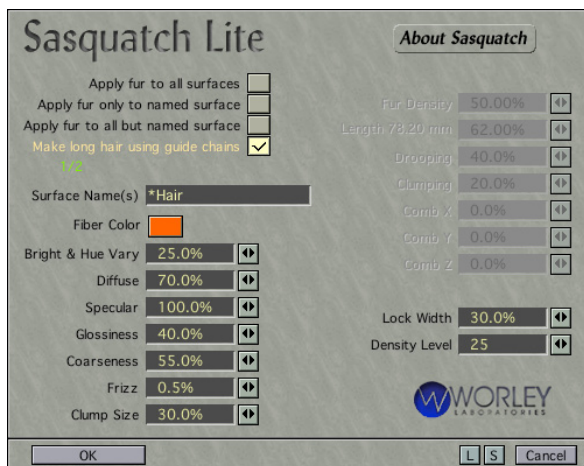


This elaborate hairstyle used several surfaces, and two layers of Hair Guides (to stay within the 25,001 point limit.)

It's ready to render.

Begin the same way you do for Fur; that is, open the **Object Properties**, and apply the **SasLite Displacement** from the **Deform Tab**. Double click to open the **Options**, and you'll see the very same dialog as before.

But this time, click the **Make long hair using guide chains** radio box.



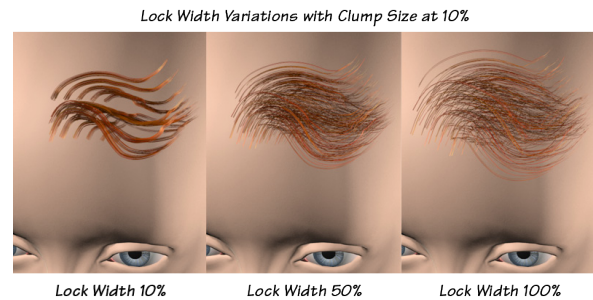
Sasquatch Lite options when using Guide Chains.

You'll have to use a surface name, so that **SasLite** can tell which is the "other" surface, where the hair growth starts. Once again, you can use asterisks as wild cards, it is case sensitive and if you check the dim green fraction, you can see if you remembered the name (It will indicate the number of surfaces the effect will be applied to out of the number in that object.)

The column of options on the left works exactly the same way as before. Choose the Fiber Color, the Bright & Hue Vary, the amount of Diffuse, Specular, Glossiness, etc. If you've played with Fur, this will all be familiar to you.

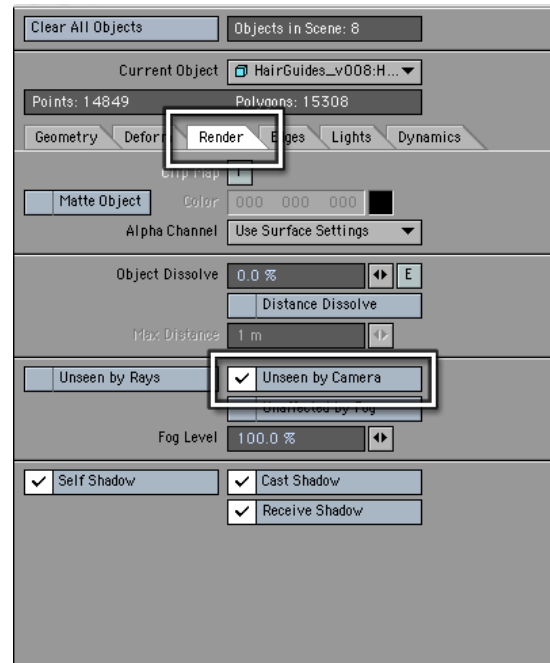
On the right though, you'll find that most of the column is dimmed, and there are two new options.

The first is **Lock Width**. This option tells **SasLite** how much room each group of hairs will take around each of the Guide Chains you constructed. It doesn't put more fibres there; it just spreads them out or collects them more tightly.



Variations in Clump Size Value when using Guide Chains.

Once you have all of that set up, go to the **Render Tab** of the **Object Properties**, and check "**Unseen by Camera**." You don't want to render the guide chains, just the hair!



*Don't forget to make the Guide Chains Unseen by Camera!*

Rendering works in the same way. In fact both Fur and Hair use the same Pixel Filter, and you can use both of them on the same object if you like. (The limit, of course, is 8 instances total, mix and match!)

**SasLite** fibres don't cast shadows on anything, as previously mentioned. But the guide chains will, if you are using **Shadow Maps**, and leave **Unseen by Rays** unchecked. This can be either good or bad. Sometimes, shadows are nice; but since the guides are longer (and fewer) than the fibres, they can be annoying and distracting. As always, it's up to you!



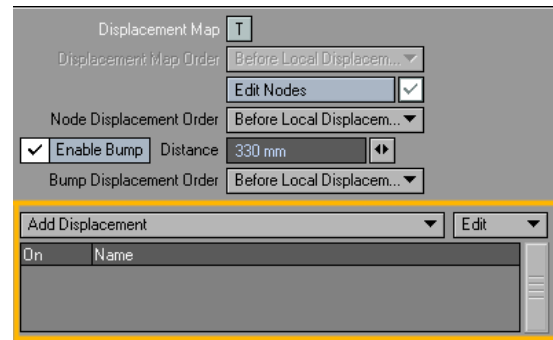
If you use **Shadow Maps**, you can render shadows of the Guide Chains.

Be aware when you are rendering, whether you are using Fur or Long Hair that the fibres are more light-sensitive than you might think. Also, although they will only make shadows (either self-shadowing or received shadows) from spotlights, they are affected by all the lights in the scene; and you cannot exclude them from a light, even if you exclude the guide chains. Fortunately, you can decrease the Diffuse value of the fibres if this is a real problem.

In addition, don't forget that **SasLite** is a Pixel Filter. That means that, besides not casting shadows, the fibres won't show up in reflections, or appear behind transparent surfaces. If you need any of those effects, plan to do them in **Post**.

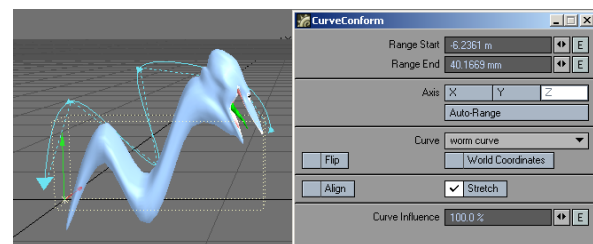
With a little practice, you'll be able to use SasLite to enhance your pictures. If you like SasLite, you'll love the full version which allows you to vary the length of fur fibres using maps, use maps to change the fur colors, render guide chains with any number of segments, allow the fibres themselves to cast shadows, and much, much more.

## Object Properties: Displacement Plugins



## Curve Conform

Curve Conform uses a curve object to deform an object's mesh. In order to determine what part of the curve applies to what part of the mesh, you must define an axis and a range of distances along that axis. The axis is the direction in the mesh, which will be transformed to lie along the curve. The **Range Start** and **Range End** values define where the beginning and end of the curve match up with the selected **Axis**.



When Curve Conform is first added, it scans the mesh to compute a bounding box. The range and axis are set to match the longest side of this bounding box. The **Auto-Range** button can be used to rescan the mesh and set the range based on the bounds of the currently selected axis.

The curve object is set on the **Curve** pop-up. If there is more than one curve in the object, the first curve is used.

The **Flip** option reverses the influence direction of the curve. **World Coordinates** leaves the curve's position fixed in 3D space, deforming only that part of the mesh that moves into its range.

**Stretch** alters the mapping of the range to the curve's arc length (the length of the curve), so that the entire range exactly fits into the length of the curve. This can cause stretching or compression of the mesh along its axis.

When **Align** is enabled, the vertices are rotated, as well as translated, so the mesh's "thickness" along the axis is preserved, like a bend operation. This mode maps the range directly to the curve, so the **Flip** option has the effect of flipping the mesh, but leaving the basic shape the same.

The **Curve Influence** percentage blends the deformed shape with the original un-deformed shape.

**Show Curve (Custom Object)** displays an object curve in Layout—normally curve objects are a modeling tool and cannot be seen in Layout. (If there is more than one curve in the object, the first curve is used.)



## FX\_Hardlink

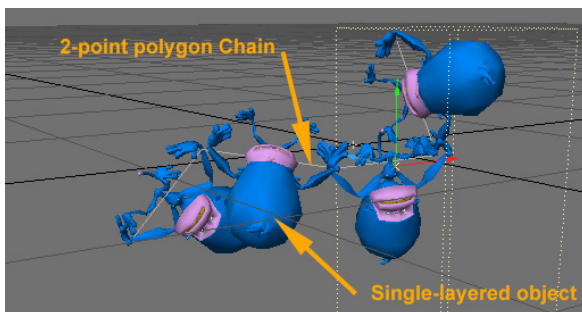
**FX\_Hardlink** is a very powerful tool that allows you to use the dynamic motion from one object and apply it to another. In most cases the dynamic object contains very few polygons (2-point polygons) while the “HardLink” object has a higher polygon count.

In the example below, **Cloth Dynamics** is applied to a 2-point polygon chain made up of (x4) 2-point polygons. It is the parent of a single layered object that is made up of four monkeys. When you use **FX\_Hardlink**, the monkeys take on the motion of the 2-point polygon chain.

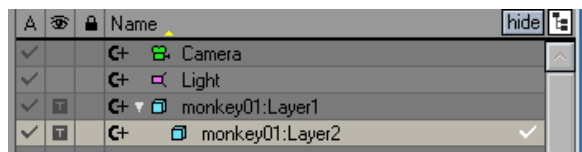
**Hardlink** keeps the geometry rigid and will displace it based on the **Piece Mode** setting.

In the example below, **Cloth Dynamics** is applied to a 2-point polygon chain made up of (x4) 2-point polygons. It is the parent of a single layered object that is made up of 4 monkeys. When you use **FX\_Hardlink**, the monkeys take on the motion of the 2-point polygon chain.

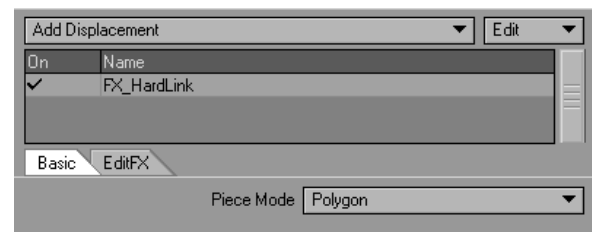
**Hardlink** keeps the geometry rigid and will displace it based on the **Piece Mode** setting.



NOTE: The HardLink Object must be parented to the Dynamic object in order for FX\_Hardlink to work.

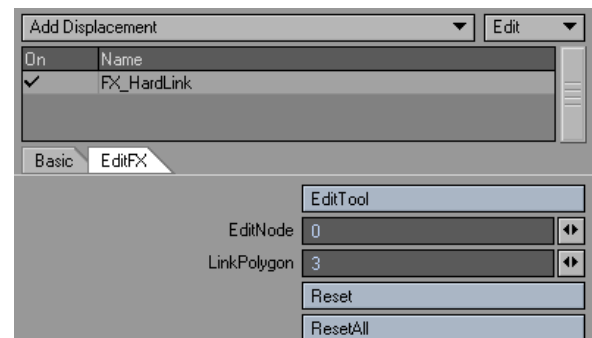


## FX\_Hardlink Properties



Under the **Basic Tab** you can choose what **Piece Mode** you would like to use. **Polygon** treats each segment as its own object. **1 Piece** will make the entire object a solid and considers it one piece. **Point Set** will read selection sets that you could create in Modeler.

The **Edit FX Tab** allows you to change what pieces are linked to what nodes. This is handy if the tool doesn't quite link what you thought it would on complex objects.







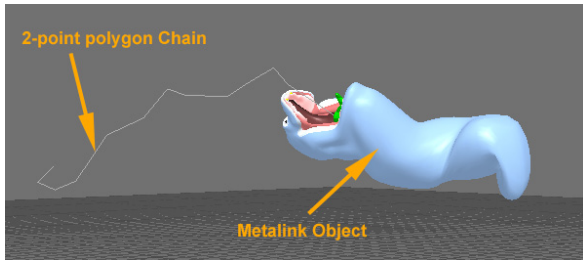
## FX\_MetalLink

**FX\_Metalink** is a very powerful tool that allows you to use the dynamic motion from one object and apply it to another. In most cases the dynamic object contains very few polygons (2-point polygons) while the "Metalink" object has a higher polygon count.



NOTE: You may run into problems if the Dynamic Object is too rough. Subdividing the Dynamic Object may help.

In the example below, a 2-point polygon chain with Cloth dynamics applied to it is the parent of the Worm (Metalink Object). The Work takes on the dynamic motion of the 2-point polygon chain with zero calculation time.



NOTE: The Metalink Object must be parented to the Dynamic object in order for FX\_Metalink to work.



The **Smoothing** option attempts to smooth the reshaping of the Metalink object. If unchecked, the reshaping can pass through the vertex of the cage object (dynamic Object).



NOTE: If you want to use other Displacement Maps and not have them be ignored by Metalink also apply Metalink\_Morph.

## FX Metalink Morph

**FX\_Metalink\_Morph** is a displacement plugin that enhances the functions of **FX\_Metalink**. By itself, **FX\_Metalink** cannot use normal morphing information because it ignores bones, **Morph Mapping**, and **Displacement Maps**. However, if you use **FX\_Metalink** with **FX\_Metalink\_Morph**, you can use normal morphing data.



NOTE: The FX\_Metalink\_Morph plugin can be added before or after FX\_Metalink.

**FX\_Metalink\_Morph** has one pop-up menu called **Morph Mode**. Set this to **One time morph** to execute morphing only one time. This mode is appropriate when the morphing is from **Morph Mapping**. Use **Every time morph** to execute morphing for each displacement process. This mode is appropriate when the morphing varies, like the **Displacement Map** of waves. The **Non morph** setting simply disables this plugin.



# HVDeform

## 1. Introduction

HVDeform gives you new levels of control over hypervoxel particles. With HVDeform you can:

- Create more realistic looking pools of water
- Change the orientation of hypervoxel particles arbitrarily
- Stretch and squash hypervoxels based on the properties of individual particles
- Alter the size of hypervoxels as a function of time, space, or distance to an object

More precisely, HVDeform allows you to change the size, thickness (or flatness), and orientation of particles as a function of distance to object surfaces, particle properties, or any arbitrary envelope.

## 2. Plugin basics

HVDeform is split into two displacement handler plugins:

- HVDeform surface
- HVDeform particle deformer

### 2.1. HVDeform surface plugin

The HVDeform surface plugin is used to tag the items whose surfaces can be used to deform particles. Only the items tagged with this plugin will be considered when computing the distance between a particle and the nearest surface.

Important: to compute the distance to a surface, and any other properties associated with it, HVDeform always uses the mesh generated with the Display SubPatch Level, not the Render SubPatch Level, even when rendering. Generally, the HVDeform plugin does not require very high tessellations in order to produce good results. The time taken by the HVDeform plugin to find the nearest surface point is proportional to the number of polygons, edges, and points in the mesh. Keeping the Display SubPatch Level low will keep HVDeform operating at interactive rates in the preview, and not take an unnecessary long time during rendering.

A scene does not need to have any items tagged with the HVDeform surface plugin.

Tip: interesting effects can be achieved by adding the HVDeform surface plugin to an item which is invisible.

### 2.2. HVDeform particle deformer

The HVDeform particle deformer is where it all happens. The plugin alters the size, thickness, and orientation of hypervoxel particles. The particles are taken from all particle systems attached to the object to which the plugin has been applied.

The plugin affects all particle systems attached to the object, and those attached to the objects children, and their children and so on down. A child may have a HVDeform particle deformer attached as well, in which case the settings of that instance are used for the child and its children.

## 3. Plugin usage

The HVDeform particle deformer plugin is typically added to a particle emitter. If particles need to be deformed as a function of distance to an object, those objects should have the HVDeform surface plugin applied.

Important: in order for HVDeform to operate correctly, the geometry settings for the hypervoxels have to be set in a particular way:

- “Stretch Direction” must be set to Velocity
- “Stretch Amount” must be set to -100%
- “Maintain Volume” switched off
- “Align To Path” switched on

Tip: the HVDeform settings interface shows a reminder about these settings by clicking on the “i” button.

The way hypervoxel particles are deformed is governed by a set of rules. Rules specify what sort of deformations to apply under what circumstances. The results of the rules are combined according to how applicable they are to a particle's situation.

A particle can be altered by changing its size, thickness, and orientation. The size of a particle is the radius of the particle assuming that it is a perfect sphere. This sphere can be squashed or stretched into an ellipsoid by altering its thickness. The thickness is the radius of the ellipsoid in the direction of its orientation.

## 4. HVDeform particle deformer interface

The interface is split into three areas:

- General settings
- Rule and group setup
- Rule and group settings

### 5.1. General settings

#### 5.1.1. Load and Save Preset

HVDeform settings can be loaded from and saved to the Preset shelf. When loading from the Preset shelf, the option is given to load all the settings, only append the rules from the preset, or replace the rules with those in the preset.

#### 5.1.2. Assumed HV size

The HVDeform plugin needs to know what size the hypervoxels are. This should be set to the same as the Particle Size setting in the HyperVoxels Geometry tab.

Remember: the particle size is the radius of the HV particle, not the diameter.

Tip: if Particle Size in the HyperVoxels settings is an envelope, then the Assumed HV size can be set to an expression which sets it equal to the HyperVoxels Particle Size channel.

#### 5.1.3. HV surface multiplier

The apparent surface of a hypervoxel may not be at the boundary of a particle. This setting gives the fraction of the radius at which the surface of the hypervoxel is assumed to be.

Tip: for surface type hypervoxels, the multiplier is usually 0.5. For volume type hypervoxels, a multiplier of 1.0 is typical.

#### 5.1.4. Maximum surface distance

This is primarily used for optimisation. If there are objects with HVDeform surface plugins applied, HVDeform computes the minimum distance from each particle to any of the surfaces. The earlier parts of the scene can be



rejected as being too far, the quicker this search will take. Setting this value too small may result in rules not having any effect when they should. If set too large, the particles may take longer to compute.

#### Infinite

Maximum surface distance is set to infinity. If there are HVDeform surfaces in the scene, rules will always be applied to the particles. If there are no HVDeform surfaces, no rules are applied.

#### Ignore

Maximum surface distance is set to infinity. The rules are always applied, even if there are no HVDeform surfaces in the scene.

#### Custom...

Uses the value or envelope supplied for the maximum distance. If there are HVDeform surfaces in the scene, rules will be applied to the particles if they are nearer than the given distance to a HVDeform surface. If there are no HVDeform surfaces, no rules are applied.

#### 5.1.5. Target pool depth

Value used by some deformations. It indicates to what thickness particles should be squashed to when on a surface to produce a pool of water or other stuff. But you can abuse it to mean anything else as well.

#### 5.1.6. Target stretch length

Value used by some deformations. It indicates to what thickness particles should be stretched to when falling off a surface. But you can abuse it to mean anything else as well.

#### 5.2. Rule and group setup

Rules tell HVDeform how to deform hypervoxel particles. Rules can be grouped together in groups. Rules and groups are displayed in the form of a standard tree control. Rules can be children of groups. Rules and groups can not be children of other rules. Rules and groups can be added, removed, and reordered. The names of rules and groups should be unique.

To add a rule or group, just enter the name in the appropriate entry field. To remove a rule or group, select it in the tree control and click the appropriate delete button.

As a matter of convention in this manual, group names are capitalised, while rule names are all lower case.

Note that the order of the groups and rules is important, as will be explained below.

#### 5.3. Rule and group settings

These settings are only available if a rule or group has been selected in the rule tree. Some (actually, most) of the settings here are only available for rules.

##### 5.3.1. Name

The name of a rule or group can be changed by giving a different name. Names should be unique.

Note: if there are any envelopes in use in the scene which reference channels associated with the old rule or group name, those envelopes will be deleted.

##### 5.3.2. Fraction

The fractions determine how much a rule or group contributes to a hypervoxel particle deformation. The contributions of rules and groups are computed hierarchically. The contribution of a rule in a group is the product of the contribution of the rule within the group with the contribution of the group. So if a group does not contribute to the final result, none of its rules will either.

The contribution of a group or a rule within a group is computed by summing up the fractions of all the groups or rules with the same parent up to a maximum value of 1. The contributions are then the percentage that each group or rule contributed to the sum.

For example, consider a setup with two group and some rules as follows:

Group/Rule

Fraction

On SurfaceflattenoscillateOff Surfacestretchsquashshrink

0.60.10.10.40.30.80.6

(Note that more typically the fractions would be given as expressions rather than constants. The actual values used are computed by evaluating the expressions for each particle at a given time).

The sum of the fractions of the groups is 1. The On Surface group contributes  $0.6/1 = 0.6$ , and the Off Surface group contributes the remaining 0.4.

The sum of the fractions of the rules in the On Surface group is 0.2. The flatten rule contributes  $0.1/0.2 = 0.5$  to the group, as does the oscillate rule. The contribution of the rules to the final result is their contributions within the group multiplied by the contribution of the group. Hence flatten contributes  $0.5*0.6 = 0.3$  to the final result, and ditto for the oscillate rule.

The fractions of the rules in the Off Surface group add up to 1.7, which is clipped to 1. The stretch rule contributes 0.3, leaving 0.7 for the remaining rules. The squash rule eats it all up, as it has a fraction of 0.8. With no more contributions to fill, the remaining 0.1 of the squash rule is ignored, as well as the entire shrink rule. The contributions of the rules to the final result are  $0.3*0.4=0.12$ ,  $0.7*0.4=0.28$ , and  $0.0*0.4=0.00$  respectively.

In summary, the rules are mixed for the final result as follows:

Group/Rule

Contribution

On SurfaceflattenoscillateOff Surfacestretchsquashshrink

(0.6)0.300.30(0.4)0.120.280.00

Another example (check tooltips on contributions for details):

Group/Rule

Fraction

Contribution

tumbleFlyinggoing upcoastinggoing downOn Groundrestingwobble

0.40.50.31.00.41.00.20.1

0.4(0.5)0.150.350.00(0.1)0.06670.0333



There are several pre-defined fraction computations:

#### On surface

Sets fraction to 1 if particle is no more than the assumed HV size from a surface, 0 otherwise.

#### On surface transition

Like "On surface", but with a smooth transition from 1 to 0 as the particle's distance from the surface increases from 0 to the assumed HV size.

#### Off surface

The complement of "On surface". 1 if the particle is more than the assumed HV size from a surface, 0 otherwise.

#### Off surface transition

The complement of "On surface transition". Smoothly transitions from 0 to 1 as the particle's distance from the surface increases from 0 to the assumed HV size.

#### Towards surface

Sets the fraction to the cosine of the angle of incidence, limited to a minimum of 0. (The angle of incidence is found by computing the dot product between the surface normal at the surface point nearest the particle, and the opposite of the particle's velocity vector). A value of 1 means that the particle is approaching the surface head-on. A value of 0 means that the particle is not moving towards the surface. In-between values give an indication of how steep the particle is approaching the surface.

#### Away from surface

Sets the fraction to the negative of the cosine of the angle of incidence, limited to a minimum of 0. This is like "Towards surface", except that it goes the other way. A value of 0 would mean that the particle is not moving away from the surface.

#### Remainder

The fraction is set to 1, which effectively makes the rule use up any remaining free contribution.

#### Custom...

Uses the given value or envelope.

#### 5.3.3. Size

The new particle size.

#### Assumed HV size

Use the assumed HV size setting.

#### Custom...

Set the particle size to the given value of envelope.

#### 5.3.4. Thickness

Sets the thickness of the particle. The thickness is the size of the particle along its orientation axis.

Same as size

Use the Size setting.

#### Assumed HV size

Use the Assumed HV size setting.

#### Squash

Reduces the thickness of the particle to the Target pool depth setting as it gets closer to the surface.

#### Stretch

Stretches the particle thickness from the Target pool depth value to the Target stretch length value and back to the Size setting value as the distance from the surface increases.

#### Custom...

Set the particle thickness to the given value of envelope.

#### 5.3.5. Orientation

Sets the direction along which the particle is oriented. This is the direction along which the particle's thickness is changed.

#### Surface normal

Align the particle with the surface normal at the point nearest the particle.

#### Velocity vector

Align the particle with the direction in which it is moving.

#### Velocity&Normal transition

Interpolates between velocity vector and surface normal. If the particle is off the surface, the velocity vector is used for the alignment. When the distance between the particle and the surface is zero, the normal vector is used. In between, the alignment is interpolated between the velocity and the normal vector. The interpolation takes into account whether the particle is moving towards or away from the surface.

#### Custom...

Set a custom particle orientation. The given vector need not be normalised. It is normalised by the HVDeform plugin.

Tip: unlike in many other places in Lightwave, the three components of the vector can be enveloped individually.

#### 5.3.6. Preserve Volume

When changing the thickness of a particle, HVDeform can change the size of the particle so as to preserve the volume of the undistorted particle.

When set to 100%, volume is preserved completely. When set to 0%, no attempt to preserve volume is made. The size of the particle remains unchanged if the thickness is changed. Values in between 0 and 100 cause volume preserved to various degrees.

Tip: when set to 0, the Size and Thickness settings are effectively decoupled. The sizes given will be the sizes rendered.



## 6. HVDeform Surface Settings

The interface for a HVDeform surface can be opened by double-clicking the HVDeformSurface deformation instance.

### 6.1. Smooth mesh

When enabled, HVDeform will interpolate polygon normals to give the appearance of a smooth mesh. This mostly affects particles which are aligned with the surface normal.

If mesh smoothing is turned off, the polygons are treated as perfectly flat surfaces. When a normal-aligned particle crosses an edge between two polygons, it will appear to abruptly change orientation from one polygon normal to another.

### 6.2. Smoothing angle

The smoothing angle is used to determine when a part of the mesh is treated as being flat or smoothed. Basically, larger angles causes adjoining polygons with larger angles to be smoothed. Smaller angles makes the mesh less smooth.

## 7. HVDeform channels

The HVDeform plugin exposes a number of information channels attached to the emitter on which the plugin has been applied. These channels give information about various HVDeform settings, and properties of the particle being evaluated.

Note: in general, the values in the HVDeform channels are only valid during the evaluation of a particle as part of a layout refresh or render. The use of these channels outside of the HVDeform plugin is unlikely to give the results you expect.

The channels are in the HVDeform channel group. Any envelopes for rules and groups appear as sub-groups with name equal to the name of the rule and group (hence it is important that rule and group names are unique, in order to avoid confusion).

Channels which give information about the HVDeform plugins general settings, and channels containing particle information, are read-only. You can not assign any values directly to these channels. As a reminder, the names of these channels are prefixed with an "r" indicating read-only.

Channel name

Meaning

HVDeform.rPtIPos.{X|Y|Z}

World position of particle

HVDeform.rPtIVel.{X|Y|Z}

World velocity of particle

HVDeform.rPtISpeed

World speed of particle

HVDeform.rPtIAge

Age of particle

HVDeform.rPtIID

Particle ID (unique per emitter)

HVDeform.rSurfID

Nearest surface ID

HVDeform.rCosIn

Cosine of particle angle of incidence with surface

HVDeform.rSurfDist

Signed distance of particle from nearest surface

HVDeform.rSurfNorm.{X|Y|Z}

Surface normal at surface point nearest the particle

HVDeform.rPoolDepth

Target pool depth

HVDeform.rStretchRadius

Target stretch radius

HVDeform.rAssumedHVSize

Assumed HV size

HVDeform.rHVSurfMult

HV surface multiplier

HVDeform.rPtIPos.{X|Y|Z}

Gives the position in world coordinates of the particle currently being evaluated by the HVDeform plugin. This can be used to apply position dependent deformations to a particle.

HVDeform.rPtIVel.{X|Y|Z}

Gives the velocity in world coordinates of the particle currently being evaluated by the HVDeform plugin. This can be used to apply speed or direction dependent deformations to a particle.

HVDeform.rPtISpeed

Gives the speed of the particle currently being evaluated by the HVDeform plugin. This is the length of the rPtIVel vector explained above.

HVDeform.rPtIAge

The age of the particle currently being evaluated. The age is given in seconds, not frames. This can be used to deform the particle as a function of how long it has lived so far.

HVDeform.rPtIID

Has the particle ID of the particle being evaluated by the plugin. Each particle from a given emitter has an ID which is unique over time. Note that it the ID is only unique with respect to the particles emitted by the same emitter. Particles from different emitters may have the same ID.

Note: channels can only store floating point numbers. The particle ID is an integer however. This means that if the ID number becomes larger than what can precisely be represented by a 32bit floating point number, the value in the rPtIID channel may no longer be exactly equal to the actual particle ID and uniqueness can no longer be guaranteed. In practice, for this to happen the emitter would have to emit well over 16 million particles.

This channel could be used as a random number seed for example.

HVDeform.rSurfID

Each HVDeform surface item has a unique ID. This ID number is shown in the HVDeform surface plugin description attached to the item. The rSurfID channel contains the ID number of the surface nearest the particle being currently evaluated. This could be used to apply surface-dependent rules.

If the surface ID is zero, there is no nearest surface. This can occur when there is no active HVDeform surface in the scene and Maximum surface distance is set to Ignore.





Trivia: the ID is the LW item ID plus 1, without the item type bits. In theory, like rPtIID, exactness could be lost if there are more than 16.7 million items in the scene.

HVDeform.rCosIn

The cosine of the angle of incidence of the particle being currently evaluated with the surface nearest the particle.

Mathematically, this is derived from the dot-product between the surface normal at the surface point nearest the particle, and the opposite of the particle's velocity vector. The sign of the value is set so that a positive value always means that the particle is approaching the surface, and negative means that the particle is moving away from the surface, no matter what side of the surface the particle is on.

Positive values indicate that the particle is moving towards the surface, and negative values mean that it is moving away from the nearest surface. A value of 1 means that the particle is heading straight for the surface, while -1 means that the particle is moving away from the surface as fast as possible. A value of 0 means that the particle is moving parallel to the surface.

Note: rCosIn only tells you the cosine of the angle between the opposite of the particle velocity and the surface normal at the surface point currently nearest the particle. A positive value does not mean that the particle is hitting the surface, or that it ever will. Nor does it say anything definitive about the angle of incidence if the particle does hit the surface at some future point in time.

HVDeform.rSurfDist

This channel gives the distance between the particle currently being evaluated and the nearest surface point. The value is signed. If it is negative, that means that the particle is below the surface as defined by the surface normal.

Note: if the particle is not on a surface, the nearest surface point is usually not the point where the particle is going to hit the surface.

HVDeform.rSurfNorm.{X|Y|Z}

This vector gives the normal at the surface point which is nearest the currently being evaluated particle.

Note: currently, the normal of the polygon in which the nearest point lies is used. The normal is not interpolated over the polygon.

HVDeform.rPoolDepth

This is the target pool depth set in the HVDeform general settings, evaluated for the current particle at the current time.

HVDeform.rStretchRadius

This is the target stretch radius set in the HVDeform general settings, evaluated for the current particle at the current time.

HVDeform.rAssumedHVSize

This is the assumed HV size set in the HVDeform general settings, evaluated for the current particle at the current time.

HVDeform.rHVSurfMult

This is the HV surface multiplier set in the HVDeform general settings, evaluated for the current particle at the current time.

## 8. Examples

### 8.1. The Waterball

In this scene water is poured over a sphere, which then flows down and ultimately drips off. The water is rendered using hypervoxels. The HVDeform plugin is used to flatten out the hypervoxel particles as it flows onto and over the surface of the sphere, and to stretch the hypervoxels as they fall off.

#### 8.1.1. Basic scene setup

Make a sphere of 10cm radius, and give it a copper surface. Place it in a scene and give it something to reflect, like a skytracer sky. Add a sticky collision effector to the sphere, with Type set to sphere and the Radius/Level equal to that of the ball.

Add a particle emitter approximately between 5 and 10 cm above the sphere. Make it a couple of centimetres in size. Have it create 500 particles per second. Set the particles to fall on the ball using the standard 1 gravity acceleration of -9.8 metres per second per second.

Activate hypervoxels on the emitter, and apply the Generic Water preset. Set the Particle Size to 15mm.

At this point, particles should be falling onto the ball, slide along it, and fall off near the bottom. The result looks something like the image below.

#### 8.1.2. HVDeform setup

The particles need to be deformed when they are on the ball. Add a HVDeform surface plugin as a displacement plugin for the ball. Set the HVDeform surface settings to use a smoothed mesh, with an Smoothing angle of 44.5 degrees (these are the defaults).

In order for HVDeform to properly deform the particles, some of the hypervoxel settings have to be set in a particular way:

- Stretch Direction set to Velocity
- Stretch Amount set to -100% (note the negative sign)
- Maintain Volume switched off
- Align To Path switched on

Add the HVDeform particle deformer plugin as a displacement plugin to the particle emitter.

#### 8.1.3. HVDeform general settings

Open the HVDeform particle deformer interface. Set the Assumed HV size equal to the Particle Size of the hypervoxels (15mm). As the standard Water hypervoxel preset is a surface hypervoxel, set the HV surface multiplier to 0.5. The Maximum surface distance can be set to infinite or ignore, as the number of polygons in the ball is probably relatively small. Alternatively, set it to about 4 times the particle size (60mm).

The Target pool depth is set to 3mm. Setting it too large will make the particles on the ball to be too lumpy. Setting it too small will cause the particles to be squashed so much that they become more like large disks stuck on the ball. A reasonable value for the Target stretch radius is typically twice the Assumed HV size, so set it to 30mm.

#### 8.1.4. HVDeform rules and groups setup

The behaviour of the particles in the scene can be classified into two cases:

- Particle on the surface
- Particle off the surface



When the particle is on the ball surface, it should be squashed to the Target pool depth. When the particle is not on the surface, two things can happen. First, the particle could be moving towards the ball from the emitter. In this case the particle should not be deformed. In the second case, the particle is falling away from the ball. In this case the particle is to be stretched as it falls off. The classification is refined to:

- Particle on the surface (squash)
- Particle off the surface:
- Particle moving towards surface (normal)
- Particle moving away from surface (stretch)

This is converted in HVDeform rules and groups in a straightforward manner as follows:

- on surface (rule)
- Off Surface (group)
- towards surface (rule)
- away surface (rule)

As the case where a particle is on the surface has only one rule, it does not need to be in a group. The towards surface and away surface rules are grouped. They could be put at the top level as rules without the use of any groups, but the contribution fractions will then become more difficult to set right.

#### 8.1.5. HVDeform rules and groups settings

The “on surface” rule contributes when, naturally, the particle is on the surface. So the Fraction is set to “On surface”. The Size is just the Assumed HV Size. When in contact with the ball surface, the particle needs to be squashed onto the ball, so Thickness is set to Squash and the Orientation to Surface normal.

If a particle is not on the surface, it must be off the surface. The Off Surface group contribution could therefore be set to be equal to 1 minus the contribution of the on surface rule. An easy way to do this is to set the Fraction to Remainder.

To set the fractions for the rules within the Off Surface group, assume that the particle is indeed off the surface. When the particle is moving towards the surface, nothing special needs to happen. The Fraction for the towards surface rule is simply Towards surface. The Size and Thickness are the same, equal to the Assumed HV size. In that way the particles will not be deformed.

If a particle which is not on the surface is also not moving towards it, it must be moving away from the ball surface. As with the Off Surface contribution, set the Fraction to Remainder. When falling away from the ball, the particles should be stretched out, so Thickness is set to Stretch. The stretching is done in the direction in which the particle is falling. The Orientation is therefore the Velocity vector.

#### 8.1.6. Finally...

That's it. Set rendering options, some anti-aliasing, maybe some motion blur, and render.

### 8.2. Bouncy Lava

The idea here is quite simple, if a bit bizarre. A blob of lava is going to bounce over a plane. The lava blob is made with a single hypervoxel particle, which is going to deform as it squashes against the plane and stretch as it moves off the plane.

#### 8.2.1. Basic scene setup

Add a plane or rectangle and add the HVDeform surface plugin to it. Turn off mesh smoothing in the HVDeform surface settings, so that the plane will be treated as perfectly flat.

Add a particle emitter a bit up in the air and off to the left side of the plane. Set it to emit one particle only which starts off moving to the right. Use gravity to have the particle fall towards the plane. Add a plane collision effector to have the particle bounce up when it hits the plane. Active the emitter in HyperVoxels, and use a lava preset to draw a blob of lava around the particle.

You should now have a blob of lava bouncing over the plane. What needs to be done now is to squash the particle as it comes down onto the plane, and stretch it as it goes up away from the plane.

#### 8.2.2. HVDeform setup

Add the HVDeform plugin to the emitter. As always, set the hypervoxels for the emitter with Stretch Direction to Velocity, -100% Stretch Amount, Maintain Volume off, and Align To Path on.

#### 8.2.3. HVDeform general settings

As the lava hypervoxel is a surface hypervoxel the HV surface multiplier in the HVDeform settings should be set to 0.5. Set the Target pool depth to half the hypervoxel size, and the Target stretch radius to 1.5 times the hypervoxel size.

#### 8.2.4. HVDeform rules and groups setup

We need two rules: one for when the particle is going down towards the plane, and one for when the particle is bouncing back up away from the surface. The two are mutually exclusive. If the particle is going down it is not going up, and vice versa.

#### 8.2.5. HVDeform rules and groups settings

The rule for going down should have its Fraction set to 1 if the particle is moving downwards, and 0 otherwise. This can be achieved with a simple expression. Set the Fraction to Custom, activate the envelope, and set it to the expression:

```
[Emitter.HVDeform.rPtVel.Y] < 0 ? 1 : 0
```

Replace Emitter with the name of the emitter if you called it something different.

When the particle is going down and hits the plane, it should be squashed, so set the Thickness to Squash. The Orientation is set to transition from Velocity to Normal as the particle hits the plane.

As the up rule is what should happen if the particle is not going down, and combined the up and down rule cover all the possible behaviours of the particle, the Fraction for the up rule is simply set to Remainder. As the particle goes up, it is stretched by setting Thickness to Stretch. The stretching is oriented the same as with the down rule. Note that Stretch and Squash behave the same when the particle is near the plane, so the down and up rules nicely join up when the particle changes direction from going down to going up.

#### 8.2.6. Finally...

You should now have a comically bouncing squashy stretchy blob of lava.

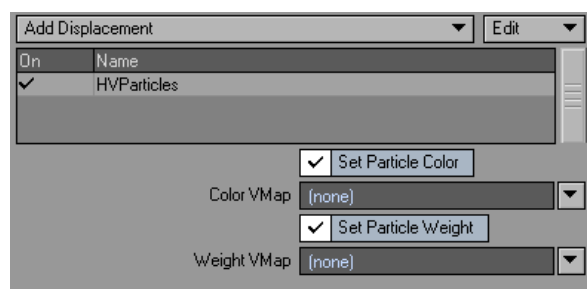
### 9. Known problems

- Particles aren't shown deformed in layout when the scene is loaded. Jiggle the frame slider to have them drawn properly.
- In some cases the mesh smoothing doesn't give the expected result. Particularly, a polygon is either smoothed completely or not at all. Subdividing and beveling edges can help.



## HyperVoxels Particles

The **HyperVoxels Particles Displacement** plugin lets you set the base color for a **HyperVoxels** particle to the color of the **Vertex Color Map**, if one exists. You can also set the particle weight using a **Weight Vmap**.



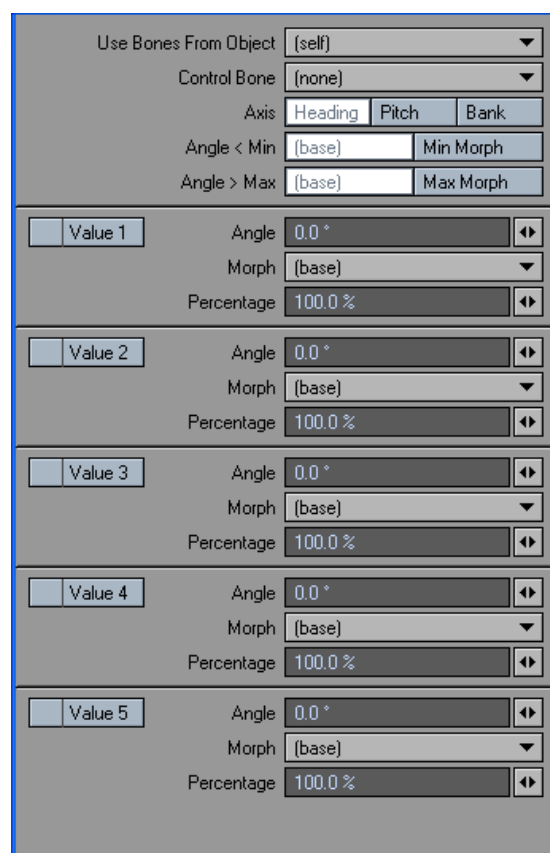
NOTE: These options are only available when HyperVoxels have been applied to an object.

## JointMorph

**JointMorph** is a legacy plugin replaced by **JointMorph Plus** detailed afterwards. Usage is pretty similar, so if you find yourself working on a scene using **JointMorph**, you should have no trouble adapting.

## Joint Morph Plus

**Joint Morph Plus** drives an **Endomorph** based on the angle of a bone. This displacement function replaces the common, but cumbersome, use of **Morph Mixer** in combination with an expression or plugin to control the morph channel based on a bone angle. This would allow you to, say, bend an arm using bones, but also control a **Morph Map** that might bulge muscles.



The **first three entries** define the **object** you are using the bones from, the **bone** and its **axis** that control the morphing. Usually the object will be **(self)** i.e. the object you are attaching to this plugin, but sometimes it makes sense to have the **Endomorph** controlled by another object's bones (e.g. if you have a high resolution object and a low resolution proxy).

The **next two entries** define what kind of morph is used if the control bone is below (**Angle < Min**) or above (**Angle > Max**) the defined range. If these options are set to **(base)**, the base model (i.e. no morphing) will be used, otherwise the lowest (**Min Morph**) value or the highest (**Max Morph**) value will be used.

The **next (up to five) values** define the **range** of the morphs. Every entry (make sure to activate the checkboxes at the left) defines a morph at a given angle. The values can be in any order, but make sure that you also define the angle for the base model (usually this angle will be at zero degrees). If you define multiple morphs for one angle, only the first entry will be considered.

The **percentage** entry scales the morph, usually this entry will be set to 100%.

Rotate the control bone now and you will see the effect of the morphing. If the angle of the control bone matches exactly a defined position, the corresponding morph will be used, otherwise the plugin will interpolate between two morphs.

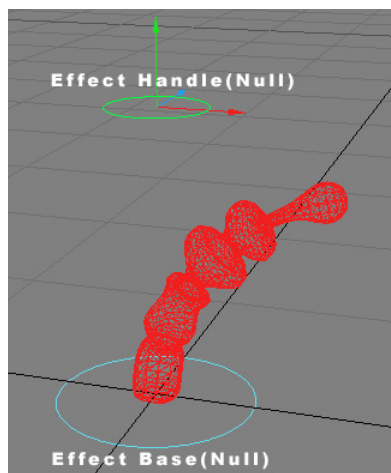
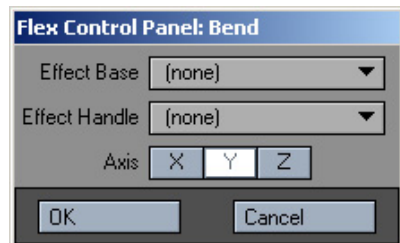


## Deform Displacement Plugins

The **Deform Displacement** plugins let you deform objects much like Modeler's **Flex** and **Deform** tools. Each needs two null objects to operate.

### Deform:Bend

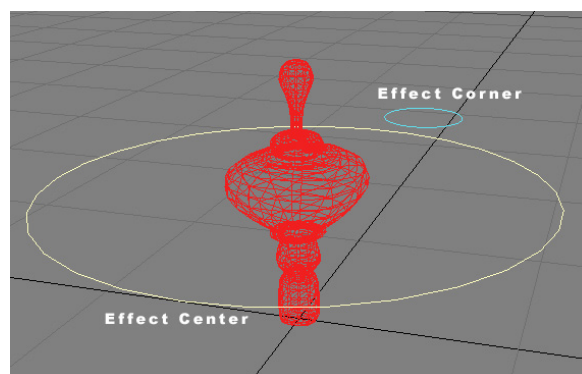
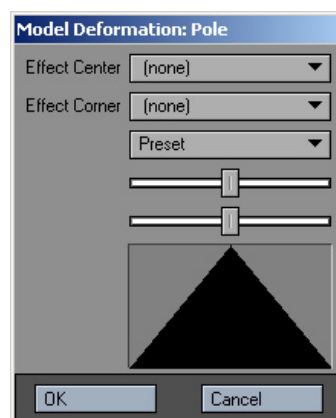
The **Effect Base** object defines the start of the bending point along the selected **Axis**. The **Effect Handle** object determines the direction of the bending.



NOTE: Nulls used in image are using Item Shape instead of the standard Null Shape.

### Deform:Pole

The **Effect Center** is the center of the effect and scaling it will distort the geometry. The **Effect Corner** defines the corner of the influenced area. Use the sliders to shape the influence area.

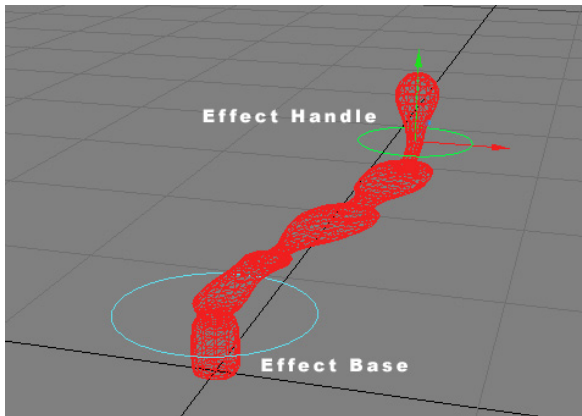
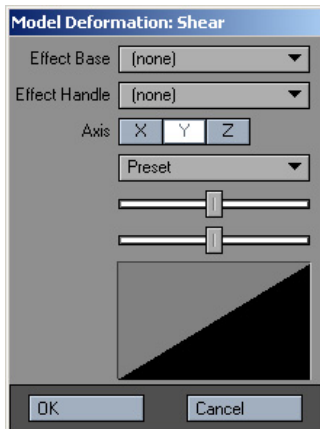


NOTE: Nulls used in image are using Item Shape instead of the standard Null Shape.



## Deform:Shear

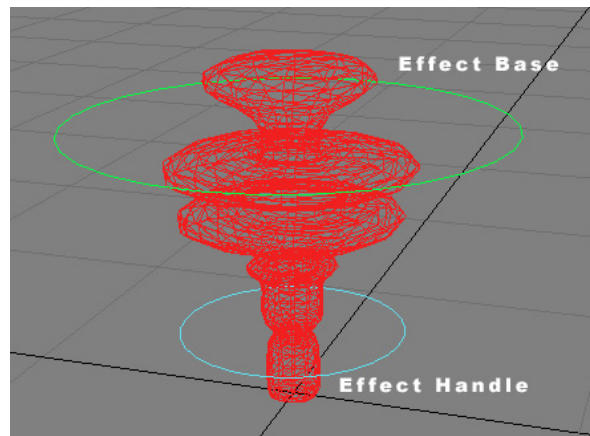
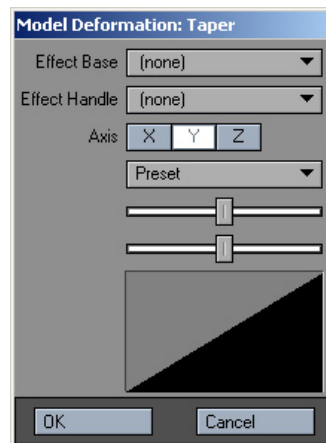
The setup for Deform:Shear is nearly the same as Deform:Bend. You can also control how the effect is applied from the base to the handle by using the two tension sliders. The top slider controls the beginning and the bottom one controls the end. Use the **Preset** pop-up to select from some common settings.



NOTE: Nulls used in image are using Item Shape instead of the standard Null Shape.

## Deform:Taper

Deform:Taper works just like Deform:Shear, except that you size the handle instead of moving it.



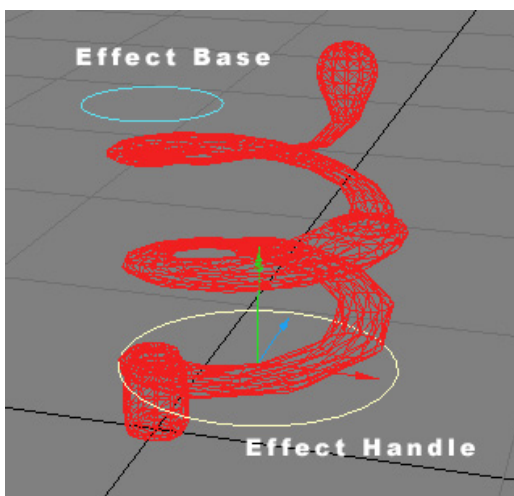
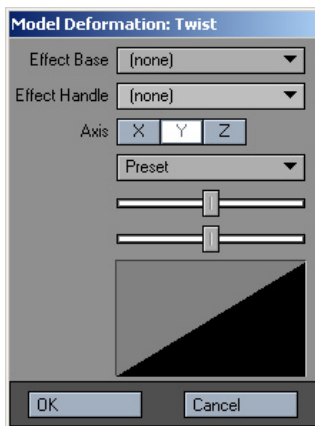
NOTE: Nulls used in image are using Item Shape instead of the standard Null Shape.





## Deform:Twist

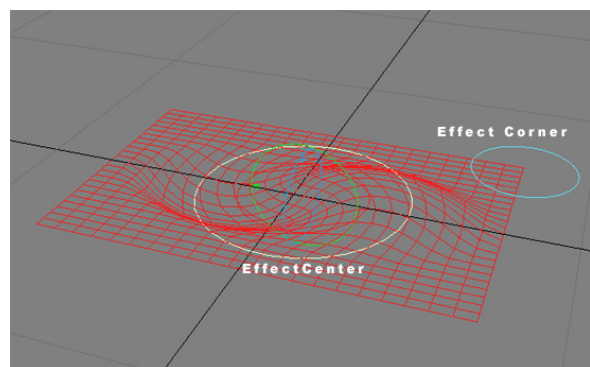
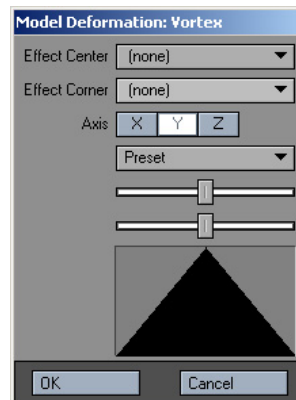
Deform:Twist works just like Deform:Shear, except that you rotate the handle instead of moving it. The center of the twisting is defined by the base object.



NOTE: Nulls used in image are using Item Shape instead of the standard Null Shape.

## Deform:Vortex

The **Effect Center** is the center of the effect and rotating it around the selected **Axis** will distort the geometry. As such, if you select the X axis, you rotate the object's pitch; for Y, you rotate heading; and for Z, you rotate bank. The **Effect Corner** defines the corner of the influenced area. Use the sliders to shape the influence area.



HINT: Add Deform:Vortex three times using a different Axis setting on each to have full rotational influence.



NOTE: Nulls used in image are using Item Shape instead of the standard Null Shape.

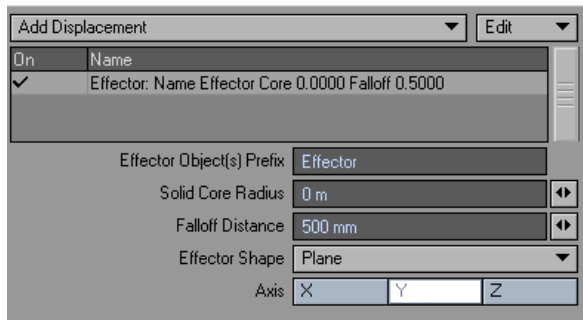


## Displacement Texture

Use **Displacement Texture** to use a texture as a **Displacement Map**. You're probably thinking that you can already do this with the **Displacement Map** function. However, using the plugin lets you order the deformation when you use multiple displacement plugins. For example, **Dynamics** totally overrides any effect the **Displacement Map** function has. With **Displacement Texture**, you could place it after the **Dynamics**.

## Effector

**Effector** causes *effector objects* to repel or attract the points of the affected object. The effector objects may be any objects you wish, but Null objects work best.

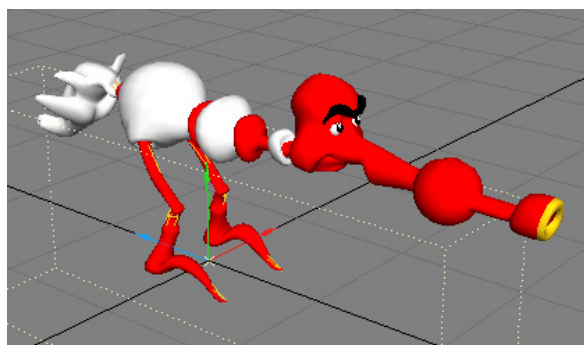


The **Effector Object(s) Prefix** is a name prefix with the default of Effector, and any object that begins with this name will be an effector. This lets you have more than one effector based simply on their name.

**Solid Core Radius** defines a spherical area, within which all points are equally affected. There is a gradual falloff of the effect between the **Solid Core Radius** and **Falloff Distance**. Points outside the **Falloff Distance** are not affected at all.

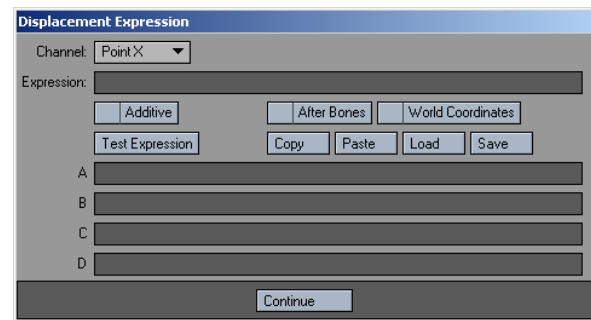
You can also choose the **Effector Shape**: it can be a **Point** or **Plane**. If it is a **Plane**, you need to specify the **Axis**. **Plane** can make an impenetrable plane that begins at the negative side of the axis based on the effector position (like keeping feet squashed against the floor). **Falloff Distance** and **Solid Core Radius** will have no effect.

The impact of the effector object is set and animated by keyframing its XYZ **Size** channels. Positive values repel and negative values attract.



## Expression

Expression is a sister plugin to the **Expression channel modifier**. You use it in the same way.



You can set independent expressions for the XYZ point displacements using the Channel pop-up menu.

Some global options are available that are not on the channel modifier version. The **After Bones** option will cause the displacement to occur after bone displacement. With **World Coordinates** checked, the XYZ variables in the expression are in world space, not local space.

## LW\_HVRealFlow\_import: Legacy Tool

Legacy plugin for using **RealFlow** data. New importers have been written for using **RealFlow** with LightWave that ship with **RealFlow**.



NOTE: Unless you have RealFlow, you will never really have a reason to use this plug-in.

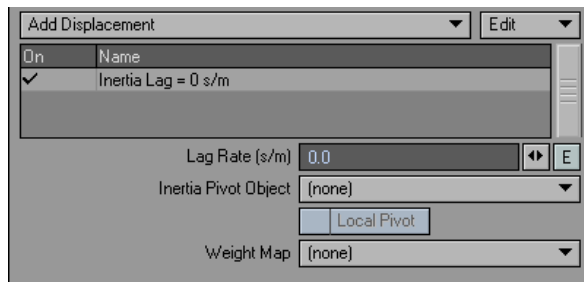
## LW\_HyperVoxel\_Particles: Legacy Tool

This is a Legacy plugin for older scenes that used **HyperVoxels** with tools like **RealFlow**.

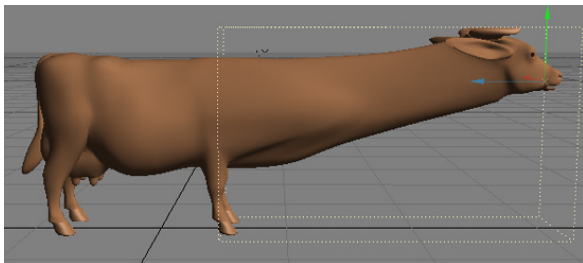


## Inertia

**Inertia** creates *lazy points*. It will delay the point positions in an object, and cause them to catch up over a period of time.



**Lag Rate (s/m)** is the amount of delay (in seconds) for points that are one meter away from the pivot point of the object — points closer or farther will be adjusted accordingly. If you had the pivot point for the cow object at the tip of her nose and moved her forward, her nose would move with the pivot point; however, the rest of her would be delayed. If you set **Lag Rate** to 1, then the portion of the cow that is one meter from her nose would be one second behind.



Normally, the object's movement triggers the effect; however, you can point at a different object instead by using **Inertia Pivot Object**. Then, the object doesn't even need to move to get the effect. **Inertia Pivot Object** replaces the item's pivot point, and becomes the center from which the inertia acts. Points farther away from this center have a larger delay in their animation — they are delayed by a time equal to the distance multiplied by the **Lag Rate**. Activating **Local Pivot** uses the pivot point's local coordinates rather than world coordinates.

You can also specify a **Weight Map**. Zero weighting will result in no delay for those points. Using 100% is the same as not using a **Weight Map** on those points.

## Lscript and Lscript/RT

These two items will allow you to apply an Lscript.

### Morph Mixer

Use the **Morph Mixer Displacement** plugin to animate using **Endomorphs**.



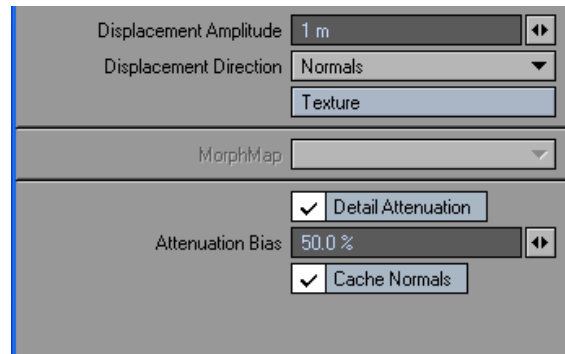
NOTE: See Morph Mixer Section, later in this chapter and starting on page 458, for more information.

## NormalDisplacement

**NormalDisplacement** can displace each vertex either along its normal or using **Morph Maps**. **NormalDisplacement** differs from regular displacement in that with regular displacement, the direction is fixed by you (or texture). With **NormalDisplacement**, the displacement is set by the geometry, an effect similar to using **Smooth Shift** in Modeler.



NOTE: A vertex's normal direction is the average of the polygon normals it is connected to.



An image of normal displacement would be very helpful in describing the tool's use.

The **Displacement Amplitude** sets the amount of displacement. **Displacement Direction** lets you choose whether the direction of the displacement is along the vertex **Normals** or using **Morph Maps**. The **Texture** button brings up the **Texture Editor**, where you can add a texture (like in a regular displacement texture).

If you are displacing along **Morph Maps**, set the **MorphMap** pop-up menu to the target **Morph Map**.

The **Detail Attenuation** option reduces the displacement in highly detailed areas, preventing the geometry from intersecting itself. This is accomplished by weighting each displacement by the neighboring polygons area, so vertices in small polygon areas are displaced less than in large polygon areas. You can adjust this weighting with the **Attenuation Bias** parameter. A bias of 0 percent will attenuate most polygons and 100 percent will affect only very small polygons.

The **Cache Normals** option causes the normals to be calculated only once. These results are used for later evaluations, making the evaluations much faster. This option is recommended for solid objects, but not for objects that are being deformed (e.g., with bones).



NOTE: If you are network rendering, "Cache Normals" will not work since the scene is unloaded and reloaded for each new frame. Instead of using "Cache Normals" for a solid object that does not change over time, freeze the deformed shape with the "Save Transformed" command, save it under a new name, and use the new object in place of the old.



NOTE: Play your scene while tweaking the settings to get real-time feedback.

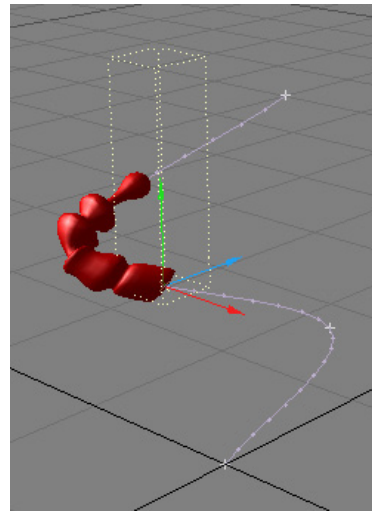


Remember that textures are three-dimensional and generally have different values for any point in 3D space (assuming your texture is something dynamic, like Turbulence). The value of the texture at each vertex acts as a multiplier against the **Displacement Amplitude** value. So if a texture was at 50 percent for a particular vertex and the **Displacement Amplitude** was 500mm, that vertex would be displaced 250mm (500mm x 50 percent). Each vertex would have its own evaluation.

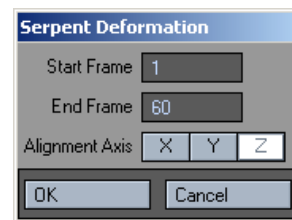
Because each vertex gets its own texture value, when using **Morph Maps**, the amount of morphing is usually different for each vertex. If the texture is animated, you will see the amount of morphing change over time.

A cool trick would be to use a gradient texture with Distance to Camera or Distance to Object set as the Input Parameter. The gradient could just be a ramp from 100 percent to 0 percent. Then, you can cause a morph to occur based on the distance to the object or camera.

## Serpent



Serpent is a displacement plugin that deforms an object so that it *hugs* its path. The object is expected to travel along some axis of alignment.



**Alignment Axis** sets the axis along which the object will be aligned. The **Start Frame** and **End Frame** parameters turn the deformation on and off at those frames, respectively.

Object deformation is usually not pretty when the object slides past the **Start Frame** and **End Frame**. To get around this glitch, give the object a nice straight lead-in, possibly with a linear keyframe. The length of this segment should be about the length the object extends past its center (0,0,0 in Modeler). Also, let the motion path and end frame extend beyond the end of your intended animation, or end with a nice straight segment along the **Alignment Axis**. Don't make the bends in the curve too sharp relative to the thickness and subdivision level of the object.

Serpent is computationally intensive. Using the following steps will maximise your efforts:

**Step 1:** Set up the motion path (i.e. keyframes) before applying the plugin to the object.

**Step 2:** Apply the plugin to a low-polygon-count stand-in object and refine your Scene.

**Step 3:** Use **Items > Replace > Replace With Object File** to replace the stand-in object with the finished version.



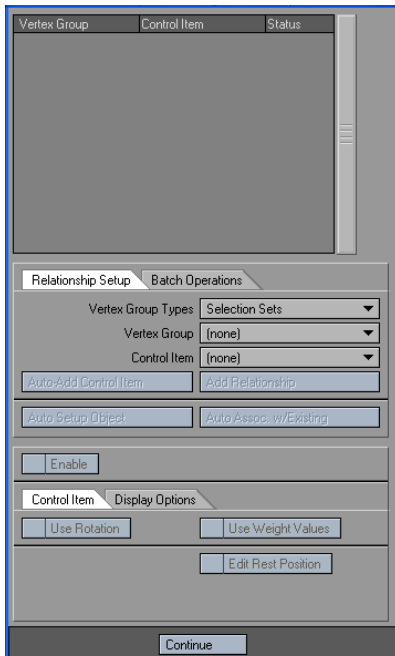
NOTE: Serpent does not work well with Align To Path controllers (Motion Options Panel) activated for the object.



## Sock Monkey

**Sock Monkey** is an animation plugin that uses scene items — typically null objects — to deform an object's mesh. You can deform an object like it is a puppet.

**Sock Monkey** parallels bones in many respects. You may want to use it if it matches your animation style. One big difference, however, from using bones is that the control objects are external to the object.



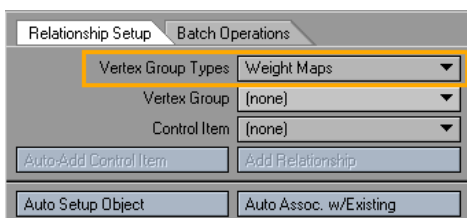
### Basic SockMonkey Setup:

**Step 1:** Create an object in Modeler using selection sets or **Weight Maps**. These will define the regions of the mesh to be deformed (e.g., chest, upper arm, lower arm, hand, etc.). Using **Weight Maps** lets you vary the influence based on the vertex weighting.

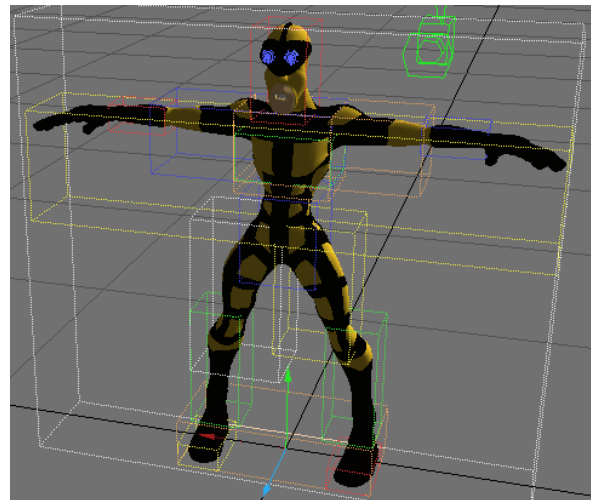
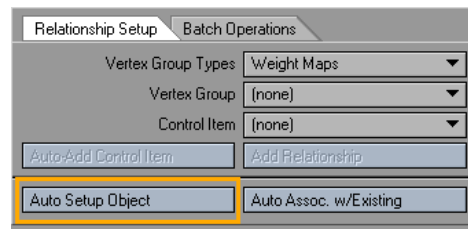
**Step 2:** Load the object into Layout and add the **SockMonkey** displacement plug.



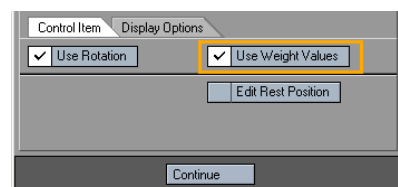
**Step 3:** Open the **SockMonkey Options Panel**. On the **Relationship Setup** **Tab**, set Vertex Group Types to the type you used in step 1.



**Step 4:** Click the **Auto Setup Object** button. Control items (objects) will be added to the scene and a *relationship* with each vertex group will be set up.



(If you are using **Weight Maps**, select each group and activate the **Use Weight Values** option (**Control Item Tab**.)



To manually add a control item, select the selection set or **Weight Map** from the **Vertex Group** pop-up menu. Then, click the **Auto-Add Control Item** button. If the item already exists in the scene, select it from the **Control Item** pop-up menu and click the **Add Relationship** button.

Instead of using **Auto-Add Control Item**, which creates the control items, **Auto Assoc. w/Existing** will add relationships to existing scene items. Just make sure the scene items have the same name as the groups before running. In other words, if you have a **Weight Map** called *Shoulder*, you must have, say, a null object called *Shoulder*. Note that this operation will not position the scene items like **Auto-Add Control Item** does.



NOTE: Control items don't have to be null objects, although this is usually the case. They can be lights or even cameras.

### Control Item Tab

The settings on the **Control Item Tab** affect the selected group in the list. **Use Rotation** will cause the control item's rotation to influence the group. When using **Weight Maps**, you'll probably want to activate **Use Weight Values** so the weighting of vertices will affect influence.

Like bones, control items have *rest positions* that provide the starting point of their influence.

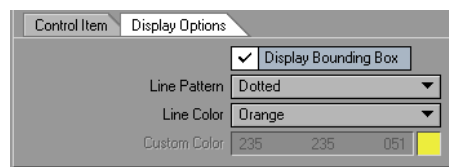
To change a control item's rest position, activate the **Edit Rest Position** option, move the item and then deactivate the option.



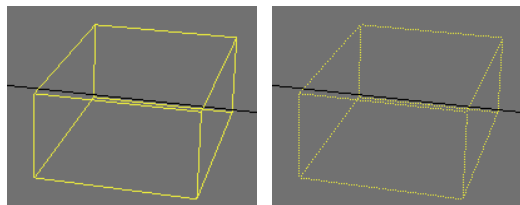


## Display Options

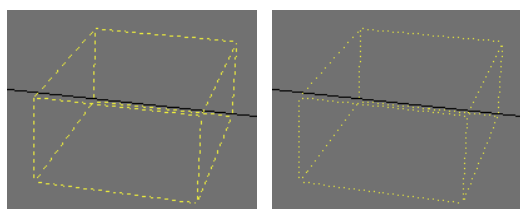
Use the settings on the **Display Options Tab** to change how (or if) the group bounding box is displayed in viewports.



Use the **Display Bounding Box** option to toggle the bounding box on and off. Choose a **Line Pattern** for the bounding box.



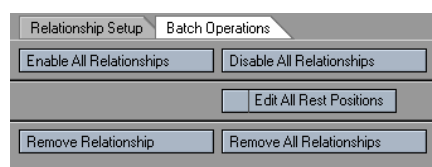
Left to Right: Solid, Dotted



Left to Right: Dash, Dash-Dot

Choose a preset color from the **Line Color** drop down menu or pick a **Custom Color** for the bounding box.

## Batch Operations



Use **Enable All Relationships** and **Disable All Relationships** to activate or deactivate all of the group/control item relationships.

The **Edit All Rest Positions** option allows you to edit the rest position for all control items.

The **Remove Relationship** button removes the group/control item relationship for the selected group. You can also remove all relationships in one step using the **Remove All Relationships**.

## Lazy Points

Lazy Points alters time for the points of an object it affects. When the Lazy Points Plugin is active for an object, the further an object's points are from its center, the longer it takes them to move, rotate, size and stretch.

## MD\_Reader



NOTE: These Plugins were designed to work with Motion Designer and have become Legacy tools.

**MD Reader (or MD\_Plug** when loaded) is the **Displacement Map** plugin that applies the motion data (MDD file) to the object in Layout.



NOTE: MD\_Reader is automatically added to the Target when you click the Activate button on the MD Property, Objects Tab. It is removed if Deactivate is clicked. As such, you will normally not need to add or adjust this plugin manually.

## Setting Options

**File Name:** Use this to load the file.

**Frame Offset:** The default load time is at frame 0, you can use this setting to offset the load time.

**Replay Speed:** If you want the motion data to replay at a different speed, adjust this setting. 100% would be the same, 50% would be replay at half speed.

**End Behavior:** This setting determines what happens with the motion data once it has finished.

**Stop:** The motion data will stop.

**Loop:** The motion data start at the beginning again once it has finished.

**Comp:** Plays back the MDD file successively.

**PingPong:** The motion data will reverse once it has reached the end, then go forward once it has reached the beginning again.

**Interpolation:** The motion data will be interpolated linearly or with splines.

**Key-Move:** Used to specify whether or not to displace the shape using the standard keyframe. Since the MDD already contains displacement information, setting **Key-Move** to **ON** can duplicate the displacement of shapes. As such, you will usually set this to OFF. However, this will also disable any movement in Layout.

**Match-By:** Matches the MDD by the specified setting.

**Coordinates:** Matches the MDD data with the object in Layout by the coordinates of each point in the object. The number of points in the MDD and the new object do not need to match with this option.

**Nodes:** Uses the point order of the vertexes contained in the object. Using this option, points must be in the same order on the object as the MDD file. The advantage for using this Match-By option is lighter processing is necessary.

**Load Limit(frame):** Limits to the MDD to the specified amount of frames, with 0 being no frames are limited.

**Vertex Cache Units:** Allows the MDD to scale the data brought in.



MD\_Reader.

NOTE: When using MD with other displacement plugins (which should be loaded after MD) be sure Key-Move is ON. Motion Designer ignores any plugins loaded above



## MD\_MetaPlug

MD\_MetaPlug is a **Displacement Map** plugin for extending MD\_Plug and applying a MDD file to the object in Layout without restricting the shape or the number of points. This lets you create the animation, and lets you completely separate the object for animation and the object for rendering. For example, you might use this to add buttons on a dress after the MD calculation.

### Setting Options

Specify the target MDD file in the **MDD Filename** field. The details include the number of MDD frames, recorded time, and the number of points on the plugin button. Verify the information.

Specify the object used for the calculation of the MDD file as **Cage Object**. This object should be made up of triangles and quads. The smooth reshaping is executed based on the lattice by polygons.



NOTE: You may run into problems if the Cage Object is too rough. Subdividing the Cage Object may help.

In the **ActionStart** field, specify the time (in seconds) at which motions based on the MDD should start.

Use the **EndBehavior** pop-up menu to specify what happens at the end of the motion based on the MDD. **Stop** maintains the last state, **Repeat** repeats the motion, and **Composite** plays back the MDD file successively.

**Key-Move** is used to specify whether or not to displace the shape using the standard keyframe. Since the MDD already contains displacement information, setting **Key-Move** to **ON** can duplicate the displacement of shapes. As such, you will usually set this to **OFF**. However, this will also disable any movement in Layout.

The **Smoothing** option attempts to smooth the reshaping. If unchecked, the reshaping will pass through the vertex of the Cage Object.

Activate **Disable** to turn the plugin off without losing your settings.

## MD\_MetaPlug\_Morph

MD\_MetaPlug\_Morph is a displacement plugin that enhances the functions of MD\_MetaPlug. By itself, MD\_MetaPlug cannot use normal morphing information because it ignores bones, **Morph Mapping**, and **Displacement Maps**. However, if you use MD\_Plug with MD\_MetaPlug\_Morph, you can use normal morphing data with Motion Designer.

The MD\_MetaPlug\_Morph plugin can be added before or after MD\_Plug. Make sure MD\_Plug's **Key-Move** is set to **OFF**.

MD\_MetaPlug\_Morph has one pop-up menu called **Morph Mode**. Set this to **One time morph** to execute morphing only one time. This mode is appropriate when the morphing is from **Morph Mapping**. Use **Every time morph** to execute morphing for each displacement process. This mode is appropriate when the morphing varies, like the **Displacement Map** of waves. The **Non morph** setting simply disables this plugin.



NOTE: Using MD\_Metaplug\_Morph is CPU intensive, so be sure to select the appropriate Morph Mode.

## MD\_Baker

MD\_Baker is a **Displacement Map** plugin that incorporates motion data from Layout into Motion Designer. MD\_Baker can handle motions, such as those of bones affected by **Displacement Map** plugins. The resulting data becomes an MDD file to be used with the main Motion Designer plugin.

**MDD Filename:** This is the name the file will be saved as.

**File Format:** Determines the motion file type.

**Bake Mode:** Sets what the motion data will record. Some file types only recognize data from specific mesh types.

**Cage Only:** The cage is the mesh before any subdivision is applied. Choosing this option will apply only motion data of the cage.

**Subdivision Only:** Subdivision occurs after the cage. Choosing this option will record only the subdivided mesh data.

**Cage and Subdivision:** This option will record both the cage and subdivided mesh data.

**First Frame:** Sets the first frame to start recording on.

**Last Frame:** Sets the last frame for recording.

**Frame Rate:** Sets the number of steps for recording. A setting of 1 will record the next frame, while a setting of 2 will record every other frame.

**Save OBJs:** Checked on will save an OBJ file with the same name as the object.

**Save LWOs:** Checked on will save an LWO file with the motion file.



NOTE: When you use MD\_Baker with other Displacement Map plugins, be sure to add MD\_Scan last.

### To use MD\_Baker:

**Step 1:** Add MD\_Baker to the **Deformations Tab** of the **Object Properties Panel** for the object whose motion you wish to incorporate.

**Step 2:** Double-click the added plugin to open its **Options Panel**.

**Step 3:** Define the desired **MDD filename** and set the parameters. Click **OK**.

**Step 4:** Click **OK** for the next dialog that appears.

**Step 5:** Make a preview animation.

**Step 6:** Open the **MD\_Baker Options Panel** again. This time a different version of the panel will appear. Click **OK** to save the MDD file data.

**Step 7:** Remove MD\_Baker or deactivate it.

**Step 8:** Load the file into MD\_Reader.

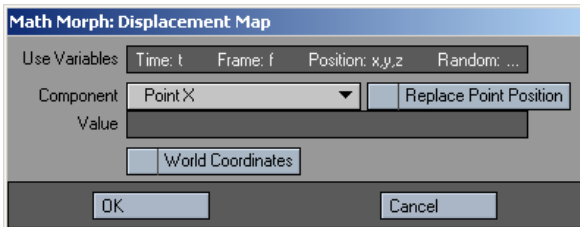


## MD Multi Baker

MD Multi Baker is similar to MD\_Baker and can export multiple objects MDDs at once, see the File\_Menu>Export section of Chapter 16 for more details.

## Math Morph

This tool allows you to affect an item's geometry by entering explicit equations for each of that item's position components, X,Y and Z. The equations may use some of the current values of the item's animation channels.



## Metamotion: Metaform Shaper and MetaformTarget — Legacy Tools

Metamotion was designed to make organic and natural-looking character animation easier. It lets you work with an object with a lower polygon count to set up your animation. This object can be manipulated with bones or other deformation tools and is smoothed by metaforming, just before rendering.



NOTE: These tools have been made obsolete since LightWave 6.0 and are included only to work with older scene files.

## Morph Gizmo Render: Legacy Tool

This tool allows you to render old LightWave 5.0 scenes that used **Morph Gizmo**. Morph Gizmo has been replaced with **Morph Mixer**.

## Saslite

**Sasquatch Lite** is LightWave's hair and fibre tool. For more information see the **Sasquatch Lite** section earlier in this chapter,

## Spline Control

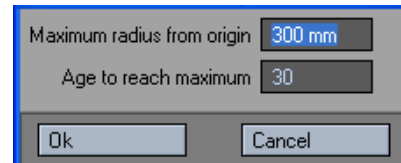
The **Spline Control** tool (**Modify>Spline Control**) is a simple way to animate hoses, tentacles, etc. using a spline with control handles. For more information see the **Spline Control** section earlier in this chapter, starting on page xxx.

## Steamy Particles: Legacy Tool

**Steamy Particles** is a legacy tool used on older scene files that used **Steamer**.

## Trail: Particle Trail

This script produces a simplistic particle effect, similar to a fireworks sparkler. If you apply it to an object made up of single-point polygons, as the object moves, the script will displace individual points in a repeating motion to simulate a "sparkler" effect.



## Object Displacement: Morph Mixer

Using LightWave's **Vertex Map** feature, all of your morph targets can be wrapped up neatly into a single object, with point-offset information. This type of object is called an **Endomorph** and the targets are referred to as **Morph Maps**.

Another advantage of **Endomorphs** is that a *morphed pose* can be a mixture of multiple targets. With normal object morphing, discussed previously, you are limited to the morphed states between the beginning and target objects only.

What's more, animating an **Endomorph** is simple because you just keyframe your poses using the **Morph Mixer Displacement** plugin.



## Morph Mixer: Morph List

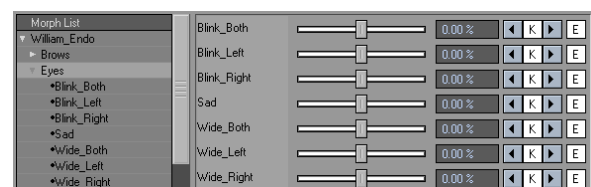
The **Morph List** will display all of the **Endomorphs** that have been created for the object. If no groups were created in Modeler they will all appear in one group. To have pre-defined groups use the group naming convention when creating your morphs.



NOTE: Placing a (.) after the first part of the name creates a new group. It's suggested that you name your morph using the group.pose format; like Eyes.Open or Mouth.

Smile.

**Morph Group Example:** To create a group called "Eyes" all of your eye morph targets should be named "Eyes.Name of-Morph".





NOTE: For more information about Endomorphs see the Endomorphs section at the end of the Map tab chapter in the Modeler part of the manual, starting on page 923.

## Managing Groups

You can move a morph from one group to another by simply dragging it to the desired group.



Create a new group using the **New Group** button. This allows you to re-organize your morphs without having to return to modeler. Use the **Rename Group** button to give any of the morph groups a new name. You also have the option to remove groups with the **Delete Group** button.



NOTE: Deleting a group that contains morphs will place those morphs in the previous group in the list. Also, deleting a group doesn't delete it from the object.

Clicking the **Graph Editor** button in the **Morph Mixer Panel** will launch

the **Graph Editor** with all the morphs channels in the channel list that appear in the Morph Slider area. The **Reset Group** button will set all the morph sliders in the Morph Slider Area back to 0%.

## Morph Mixer Options



**Morph Mixer** gives you the ability to save and load grouping information using the **Save/Load Grouping Data** commands. Use the **Save/Load Endomorph Mix** to reuse your morph animation data.

**Morph Mixer** has three **View Modes** to choose from. These modes will determine what morph sliders will appear in the **Morph Slider Area**.

**Normal** — Will display all the Morphs from the Current group.

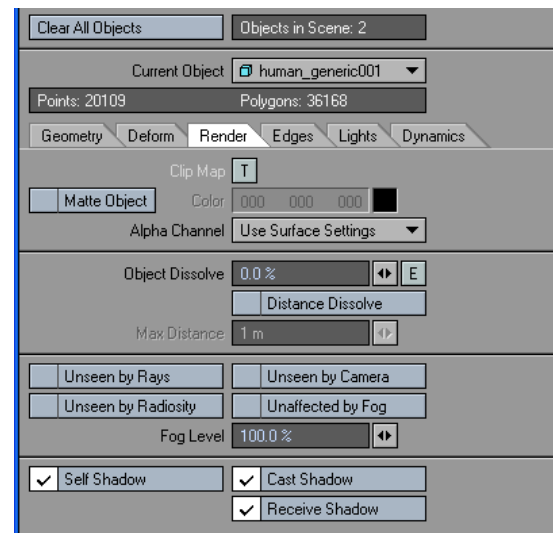
**Display all morphs as one group** — Every morph in the object will be displayed and treated as one group.

**Display Active Morphs Only** — Only morphs that have been animated will be displayed.

Use the **Set Slider Range** setting to set the Low/High (min/max) values for all sliders.



## Object Properties: Render Tab

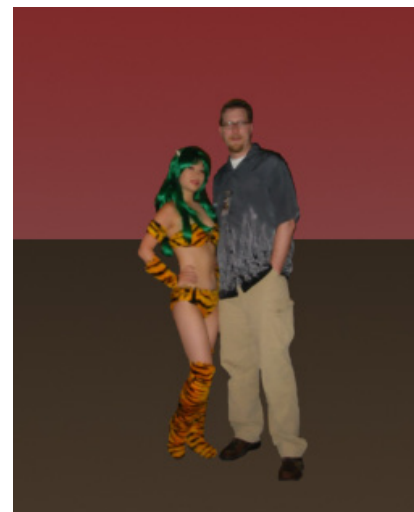


## Clip Mapping

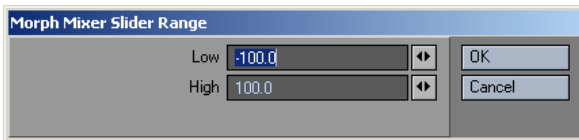
The **Clip Mapping** function, located on the **Rendering Tab** of the **Object Properties Panel**, offers a way to quickly alter an object. Basically, it allows you to cut away portions of an object using a texture. This is a great way of creating 2D pop-ups, as well as holes, tears, or grids in objects without having to model them.



Left: Flat Plane Object, Right: Clip Map Image

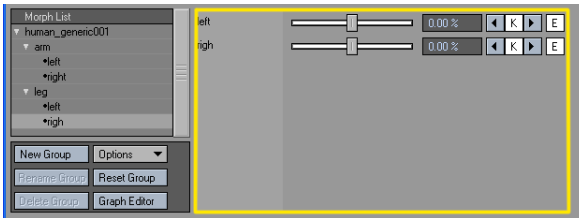


Rendered Image

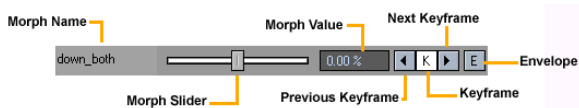


## Morph Slider Area

The **Morph Slider Area** is where you will do all your animating of the individual morphs.



## Individual Slider Banks



**Morph Name** — Name of morph for the Slider Bank.

**Morph Slider** — Slider used to interactively change the morph value.

**Morph Value** — Value of the current morph. You can manually enter in a value without using the slider.

**Previous Keyframe** — This button will skip to the previous keyframe in the animation.

**KeyFrame** — Displays Keyframe off and on.

Left click on "K" button creates key for the morph.

Left click on "K" button removes key if current frame has a keyframe for the morph.

Shift+ Left Click on "K" button keys all morphs in active group (respects view modes) at current value.

Shift+ Left Click on "K" button removes keys on all morphs in active group (respects view modes) if the current frame has a keyframe.

Ctrl+Left click on "K" button resets individual slider to zero and creates key.



**NOTE:** Keyframes are always automatically created when the slider is adjusted.

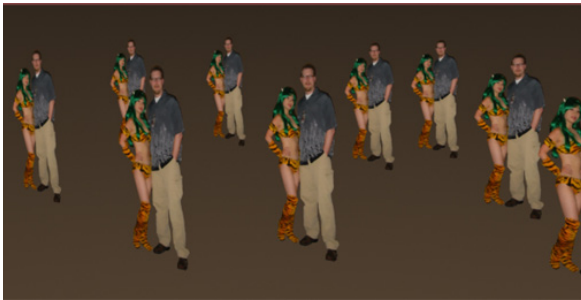
**Next Keyframe** — This button will skip to the next keyframe in the animation.

**Envelope** — This will launch the **Graph Editor** with the morph channel selected.





Object Cloned Several Times



Army of Dorks with Cute Anime Girls

There is one important distinction between a **Clip Map** and normal texturing options: there is no *partial clip*. The information in the **Clip Map** either cuts the relevant area away or leaves it intact.



HINT: If you need a *partial-clip* effect, try using a Surface Transparency Map instead.

If an image is used as a **Clip Map**, any value of 50% luminance (brightness) or higher will clip the corresponding part of the object, while a value below 50% will not. Using **Procedural Textures** works similarly, except LightWave calculates the *image* data instead of providing it in a picture. A two-color image will give you the most control over the results of a **Clip Map**.

Like **Displacement Maps**, **Clip Maps** are also saved as part of the scene file and not as part of the object. In order to load an object and its **Clip Map**, use **File > Load > Load Items From Scene**.



HINT: Place a clip-mapped flat polygon outside the camera view to fake shadows through grids, trees, or a window, using Shadow Maps. Use this where a shadow-mapped light source doesn't result in the correct shadow due to features like a transparent surface.

## Matte Object Option

A **Matte Object** option has been added to the **Render Tab**. It causes an object to be rendered in a single user-specified color (usually black or white) with no shading, overriding all of the object's surfaces.



NOTE: When an object is set as MATTE, it should take on the matte color where it is visible to the camera but it should not be seen by any other rays. In addition, the luminance channel should be 0 for the object's pixels. The matte color is still affected by fog.

## Alpha Channel Options

On the **Matte Object** section of the **Render Tab** panel there is also an **Alpha Channel** pop-up.

You can choose from the following:

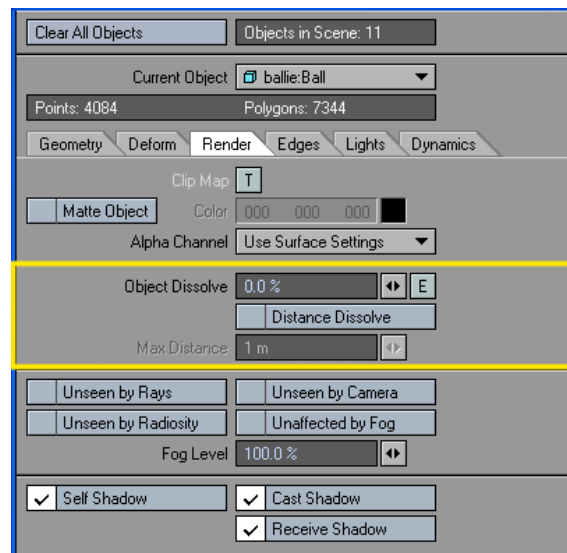
Use Surface Settings — uses the surface settings of the object to determine the object's effect on the **Alpha Channel**;

Unaffected by Object — to have the **Alpha Channel** be unaffected by the Object (essentially an Unseen by Alpha option);

Constant Black — the object is black as far as the Alpha Channel is concerned.

## Object Dissolve

Entering a value for **Object Dissolve** on the **Rendering Tab** of the **Object Properties Panel** will cause LightWave to render the object in a semi-dissolved state. If you enter 100 percent, LightWave will not render it at all. Since the value supports envelopes, you can change the dissolve amount over time. For example, you may wish to slowly dissolve clouds in on a rainy day, or replace one object with another by dissolving them in and out in the same frame.



NOTE: The appropriate Properties Panel will appear based on the current edit mode (Objects, Bones, Lights, or Cameras) when you press the **P** key or click the Item Properties button.

## Distance Dissolve

Activating **Distance Dissolve** lets you automatically dissolve the current object out after it has moved a certain distance from the camera. The **Max Dist** value determines the distance at which the object should be totally dissolved. The dissolve is gradual, thus the object will have some amount of dissolve any time it is between the camera and the **Max Dist**.

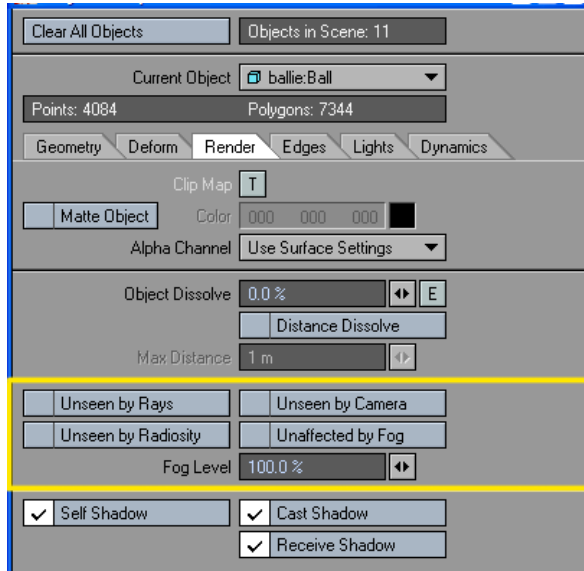


HINT: Underwater particles and moving stars often benefit from Distance Dissolve.



## Unseen By Rays

Selecting **Unseen by Rays** for an object tells LightWave to ignore the object in its ray-tracing calculations when reflection and refraction are involved. This means that the object will not appear in the reflections or refraction of another object. It will, however, render normally in the scene. This is especially handy for objects that are front projection-mapped; you probably do not wish them to show up in the reflections within other objects. **Unseen by Rays** will not affect the shadow options of a given object.



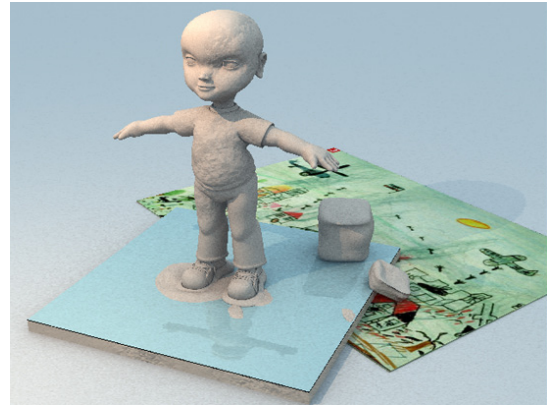
## Unaffected By Fog

Objects that have **Unaffected by Fog** selected will be excluded from the fog effect. Also, the fog **Level** setting, to the right of the **Unaffected by Fog** option, will let you apply an amount less than or greater than normal. Normal is 100%.

## Unseen By Camera

An alternative to making an object 100-percent dissolved is to activate **Unseen by Camera**. This makes the object invisible to the camera when you render; however, you will still be able to see it and work with it in the Layout window.

Although the camera won't see these objects, lights will. You can use this option to cast fake shadows into your scene from off-screen objects, like window pane frames.

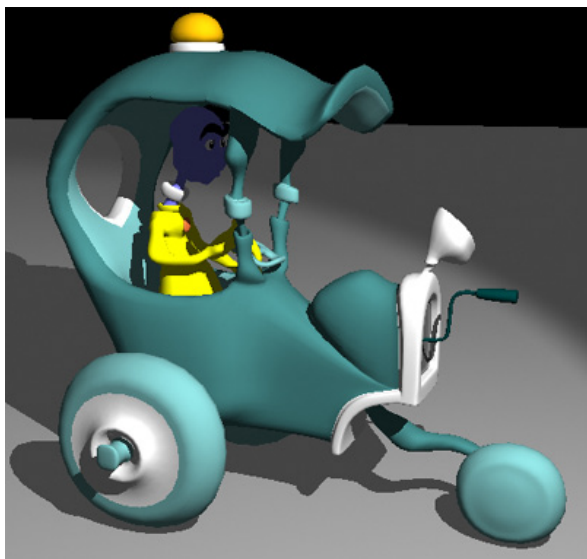


Left: Normal Render, Right: Unseen Clay Sculpture



## Object Shadow Options

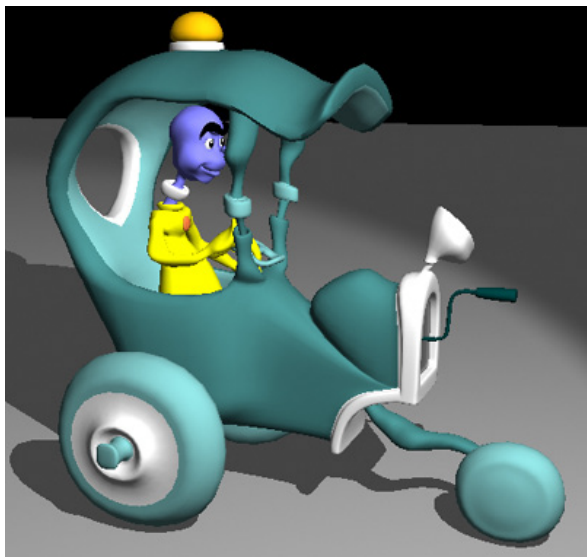
If a light has shadows enabled, any objects illuminated by it will generally cast shadows on themselves or onto other objects. However, you have complete control over this. LightWave lets you decide exactly which shadow options an object should employ. For example, you may wish for some objects not to cast shadows and others not to receive shadows. By default, all shadow options are on.



All Shadow Options Active

Judicious use of the **Self Shadow**, **Cast Shadow**, and **Receive Shadow** options (**Rendering Tab**) for the objects in your scene can greatly speed up rendering times for both ray-traced and shadow-mapped lights.

Deactivate **Self Shadow** if you do not want or need an object to cast shadows on itself. An egg is a good example of an object that cannot cast shadows on itself. A tree on the other hand is a prime candidate for **Self Shadow**.

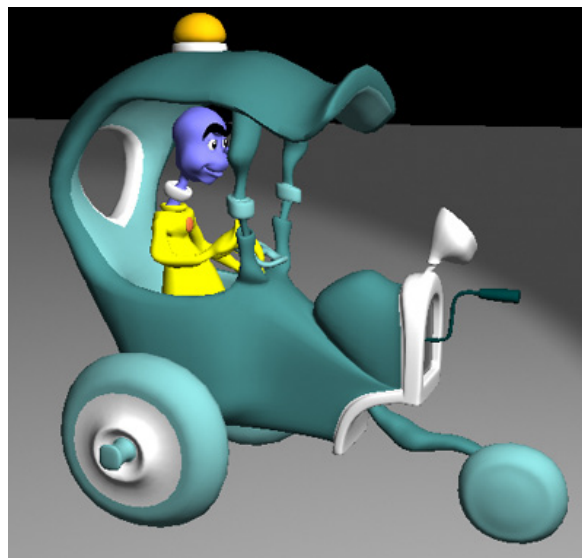


Self Shadow Inactive

Do not confuse **Self Shadow** with shading. An egg under a solitary light source may be shaded so that one side is darker, but this is shading and not casting a shadow on itself.

Deactivate **Cast Shadow** if you do not want an object to cast a shadow onto other objects.

A spaceship orbiting a planet would be a good example of an object that usually should not cast a shadow.



Cast Shadow and Self Shadow Options Inactive

Deactivate **Receive Shadow** if you do not wish an object to receive shadows from other objects. In the spaceship orbiting a planet scenario, perhaps a better option would be to turn off **Receive Shadow** for the planet as opposed to turning off **Cast Shadow** for the spaceship. You may want the spaceship to cast shadows onto nearby asteroids or other ships.

Shadow options apply whether LightWave is using ray-traced shadows or **Shadow Maps**. However, when you use **Shadow Maps** and you do not wish an object to receive a shadow, you need to deactivate both **Receive Shadow** and **Self Shadow**. Likewise, for an object that you do not wish to cast a shadow-mapped shadow, deactivate **Cast Shadow** and **Self Shadow**.

There is now a Shadow Offset text entry and mini-slider in the Render tab of the Object Properties panel. This defaults to 0. By entering a small amount (100mm worked in internal testing) the origin of the shadow rays cast by the object will be offset along the smooth surface normal by the Shadow Offset distance. This will cause nearby polygons on irregular objects to not cast shadows on their neighbors if they are roughly co-planar with them. More distant polygons on the same object or polygons that are not co-planar will still cast shadows as usual. This feature can be used to get rid of flickering shadows on the surface of irregular objects such as asteroids.

### Receive Shadows and HyperVoxels

Currently, plugins do not have access to all lighting information. As a result, **HyperVoxels**, for example, will continue to receive shadows, even if the **Receive Shadows** option is disabled.



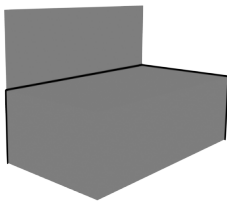
## Object Properties: Edges Tab

### Polygon Edges

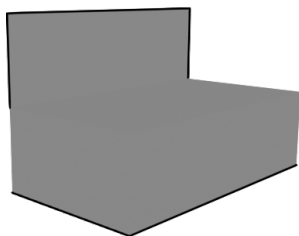
Normally, the actual edges of polygons do not appear in a rendered image. However, in some instances you may want the edges to show. The polygons themselves are still rendered with all surface characteristics, and outlines are added. Edge Color sets the color of visible polygon edges.

For example, you might want the finished animation to be done in wireframe with respect for hidden lines. (To see the full outline, use the Render Outlines on the Surface Editor's Advanced Tab. LightWave also lets you add a cel look to your rendered images, like the look you see in cartoons.

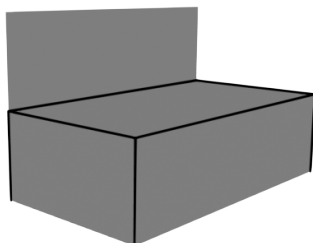
Polygon and surface edges are broken down into edge types that you can activate independently. You may also define a pixel width. You have great control over how cel-look objects are outlined.



Edges Selected

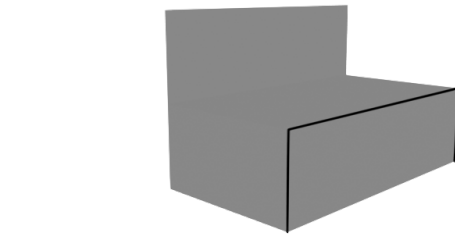


Selected

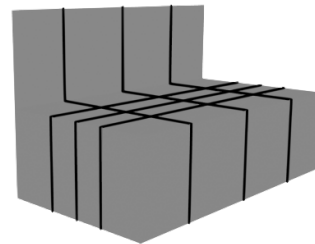


Sharp Creases Selected

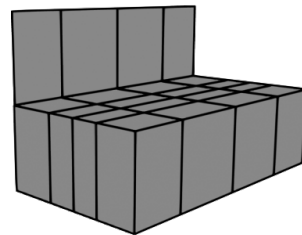
Silhouette



Surface Borders Selected



Other Edges Selected



All Edges Selected

Unshared Edges



name.

NOTE: For the images above, a box was made with the bottom cut out, then a new flap was put on at the top. The front side of the box was given a different surface

**Silhouette Edges:** When one polygon's normal faces towards the camera and shares an edge with another that is facing away from the camera, this is where the **Silhouette Edge** appears, along the line where the surface begins to move away from the camera.

**Silhouette Edges** appear on a shared edge between two polygons, where one polygon's normal faces towards the camera and the other polygon normal faces away from the camera. **Silhouette Edges** mark those edges that define where the surface turns away from the camera. Because **Silhouette Edges** depend on one surface normal facing away from the camera, they will never appear on double-sided surfaces, where all of the surface normals face the camera at all times.

**Silhouette Edges** share a real-world counterpart in life drawing, where a line defines where the surface turns away from the viewer.

**Unshared Edges** mark each edge that only belongs to one polygon. A single polygon floating in space will have all of its edges





marked as **Unshared Edges**, because each of those polygon's edges only belong to one polygon.

If you were to take a polygon and Knife it down the middle, splitting it into two, all of its outer edges would count as Unshared Edges. The Knife-edge down the middle belongs to both polygons (since the polygons still share points), and would not be considered as an Unshared Edge.

A solid volume with no overlapping points, such as a ball or box, will never have **Unshared Edges**, because all of its edges are shared. However, if you were to select a polygon on that volume, and cut-and-paste it to separate it from its brothers...you would create **Unshared Edges** around that selected polygon.

**Sharp Creases** serves three functions:

- 1) Any surface that does not have "Smoothing" checked will have all of its edges marked as **Sharp Creases**.
- 2) Any surface that has "Smoothing" checked will have only its "unSmoothed" edges count as Sharp Creases. This means that the Smoothing Angle of the surface controls the number of **Sharp Creases** on your surface.

**Sharp Creases** that fall under #1 and #2 are easily spotted in OpenGL. These edges are not smoothed over with the Phong shading in OpenGL, and are easily visible as "unSmoothed," or "sharp" edges.

Category #3 is different, but necessary for double-sided surfaces:

- 3) **Sharp Creases** act as "Silhouette Edges" on double-sided surfaces.

**Silhouette Edges** will never appear on a double-sided surface, where the normals of the surface face the camera at all times. **Sharp Creases** fill in the void, and mark the edges that define the point at which the surface turns away from the camera.

**Surface Borders** happen when two polygons with different surface names share the same edge. The edge marks the `_border_` between two polygons with two different surface names. And -- on a side note -- the surface properties can be identical, but if the surface names are different, then the shared edge that borders both polygons is marked as a "Surface Border."

**Other Edges** are for every other edge that does not fit into the other four categories: **Silhouette Edges**, **Unshared Edges**, **Sharp Creases** and **Surface Borders**.



HINT: To create a solid-wireframe animation, use the same Surface Color (Surface Editor) as the Backdrop Color (Effects Panel) and select all Edges with a contrasting Edge Color. Also, Edges are unaffected by Fog, so if you want a quick Alpha for the Edges, you can make the Edges white and fill the scene with a solid black Fog. That will give you an Alpha.

Edges can also get smaller as they move away from the Camera by activating the Shrink Edges with Distance option. The Nominal Distance setting determines the arbitrary distance from the Camera at which the edges are at their defined width. As the object gets farther away, the edges will get smaller.

## Edge Z Scale



NOTE: It is unlikely Edge Z Scale values outside the range of .95 to 1.0 would ever be necessary.

When pixels of a polygon edge are rendered, they will have depths that are approximately equal to those of the polygon that they belong to. This means that some of the pixels will be visible while others will be hidden, which could result in broken-looking edges.

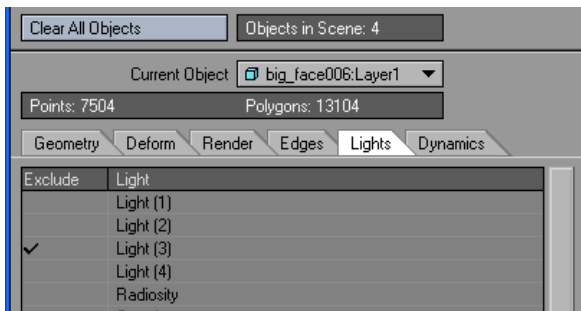
To prevent this, LightWave multiplies polygon-edge depth (i.e. Z buffer) values by a scaling factor, the Edge Z Scale, which is, by default, slightly less than one to ensure that the edges are always drawn on top of their polygons.

The default value, .998, should work in nearly all cases. However, in certain circumstances, like camera zoom factors that are extremely high, you may need to adjust this value. However, if the value is too high (i.e., too close to 1.0), the edges start to submerge into the polygon and break up. On the other hand, if the scale factor is too low, the edges will come in toward the camera too much and may even start to appear in front of other polygons that they should be behind. If either of these problems occurs, adjust the Z scale in the appropriate direction; otherwise, leave it alone.





## Object Properties: Lights Tab

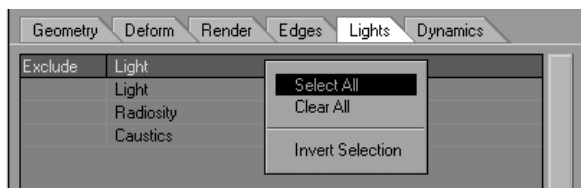


### Object Exclusions

You can exclude any set of lights, as well as radiosity and caustics, from the shading of any object. Simply click in the Exclude column to activate the exclusion.

If for some reason you need the excluded lights to continue to cast shadows, deactivate the Shadow Exclusion option. When this *global* setting is not activated, all lights will cast shadows even if they are excluded. You will need to add the Shadow Exclusion command (Lights command group) to a keyboard shortcut or menu to access it. Note that adding to a menu may be preferable so you can see its current state.

Use the drop down menu by right clicking to **Select All**, **Clear All**, and **Invert Selection**.



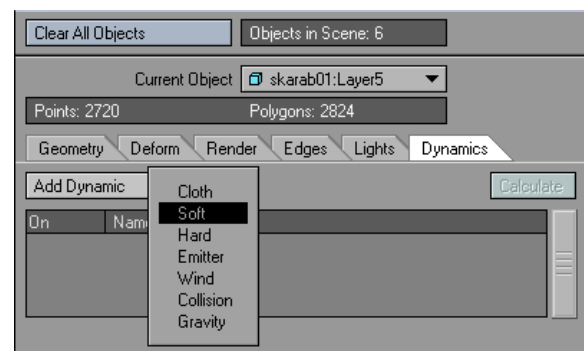
## Global Illuminations Tab

A localized set of Global Illumination values on a per-object basis has now been implemented. The controls are located on a new "Global Illumination" tab in the Object Properties panel that allows the new per-object GI values to be enabled and modified.

The per-Object GI controls are: (see Global Illumination section for details)

- \* Use Global check box: defaults to on for backwards compatibility - stored with the object info in the scene.
- \* Rays Per Evaluation
- \* Secondary Bounce Rays
- \* Angular Tolerance
- \* Minimum Pixel Spacing
- \* Maximum Pixel Spacing

## Object Properties: Dynamics Tab



The **Dynamics Tab** of the **Object Properties Panel** is where you can **Add Dynamics** and manipulate the settings.

### Add Dynamic choices:

Cloth — Cloth Dynamics.

Soft — Soft Dynamics.

Hard — Hard Body (Rigid Body) Dynamics

Emitter — Particle or Partigon Emmiter.

Wind — Wind controller.

Collision — Collision controller.

Gravity — Gravity controller.





---

## Chapter 18: Previzualization

---



## Previsualization

There will be times, more often than not, when you will want to plan out your animations or review your progress as you are working without doing a full render. This is the role of previsualization, or previz.

## Creating a Preview Animation

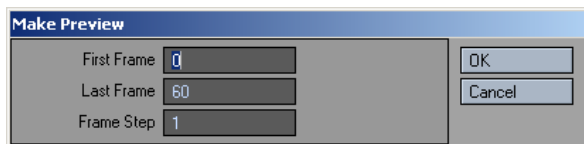
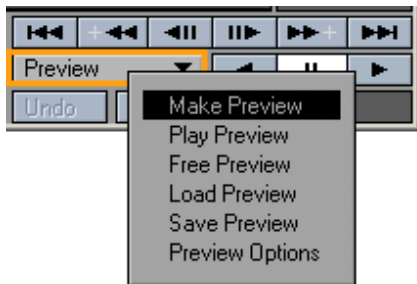
In addition to playing a scene using the transport controls, you can also create special preview animations. In most cases, these will give you a more accurate preview of your final animation.

### To Create a Preview Animation:

**Step 1:** If you are using multiple viewports, the top-left viewport is used for the preview, so make sure it shows the view you want.

**Step 2:** Make sure the viewport's **Maximum Render Level** is set as desired.

**Step 3:** Select **Make Preview** from the **Preview** pop-up menu. A dialog will appear asking you for first and last frame settings, as well as **Step**. This initially defaults to the scene settings, but may be set independently. Click **OK** and the preview animation will be created.

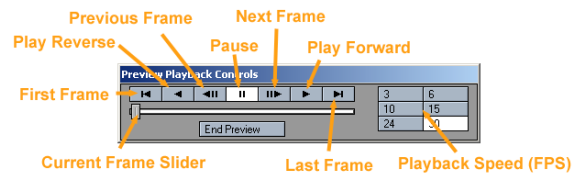


**Step 4:** When the preview is complete, the **Preview Playback Controls** Panel will appear.



NOTE: You can press the ESC key to stop creating the preview animation. The animation will still be playable up to the point of termination.

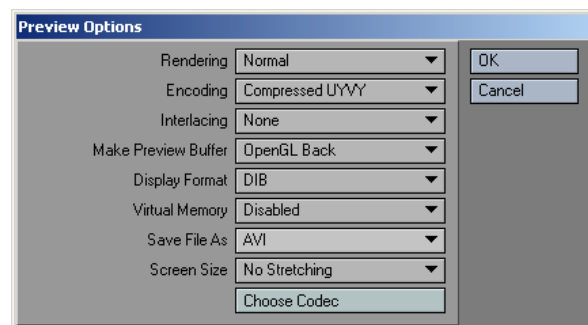
Here's a rundown of the **Preview Playback Controls**:



HINT: To save time creating a preview animation for a complex scene, do the *Texas Two Step*: use a Step of 2 and playback using a frame rate one-half the desired rate.

## Preview Options

Also from the **Preview** pop-up menu, you can replay the preview currently in memory, free it from memory, and load/save a preview animation file. Selecting **Preview Options** from the **Preview** pop-up menu will display a dialog.



Here, you can select a compression level for preview animations, as well as select the format and codec used when you save them. Compressed previews use much less memory, thus you can have a lot more frames before virtual memory starts being used, which bogs down playback. Compressed preview files are also smaller when saved to disk. You can also set whether or not to use virtual memory for preview animations.

Using the **Make Preview Buffer** option, you can choose to record the OpenGL back (offscreen) or front (on screen) display buffer. Other options may be available (e.g., Direct Draw). You may get better performance with different settings, depending on your video card.

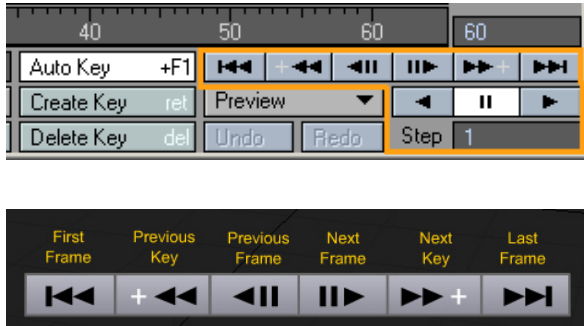
When **Virtual Memory** is set to **Disabled**, LightWave *attempts* to use only RAM for storage and playback. (Your operating system can still ultimately swap the data to virtual memory, however.)

Setting **Screen Size** to **Stretch to Fit** causes the image to fit the window, even if it is a different size than what was used when the preview was made.



## Layout Transport Controls

You can use the transport controls to navigate through your scene.



### Keyboard Shortcuts

There are also some keyboard shortcuts that can be used:

Previous frame (**Left Cursor key**)

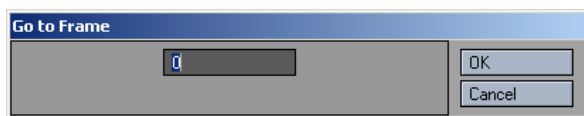
Next frame (**Right Cursor key**)

Previous keyframe (**Shift + Left Cursor key**)

Next keyframe (**Shift + Right Cursor key**)

### Going to a Specific Frame

If you press the **F** key, a dialog will appear. Enter the frame you want to go to.



### Playing the Scene

You can also play your scene backward (left arrow) or forward (right arrow) from the current frame using the playback controls. The **Step** field, in the bottom-right corner, determines the frame increment. A setting of 1 will play all frames; 2 will play every other frame, and so on.



The keyboard shortcuts for playing backwards, pause and play forwards are the PgDn, Ins and PgUp keys respectively.



**NOTE:** Your playback speed will vary depending on the complexity of your scene, object display mode, system capabilities and so on. Reducing the size of your Layout window can dramatically increase playback speed.



**HINT:** To take advantage of Play, you can modify an object in Modeler while it is animated in Layout (Note that the HUB must be activated).

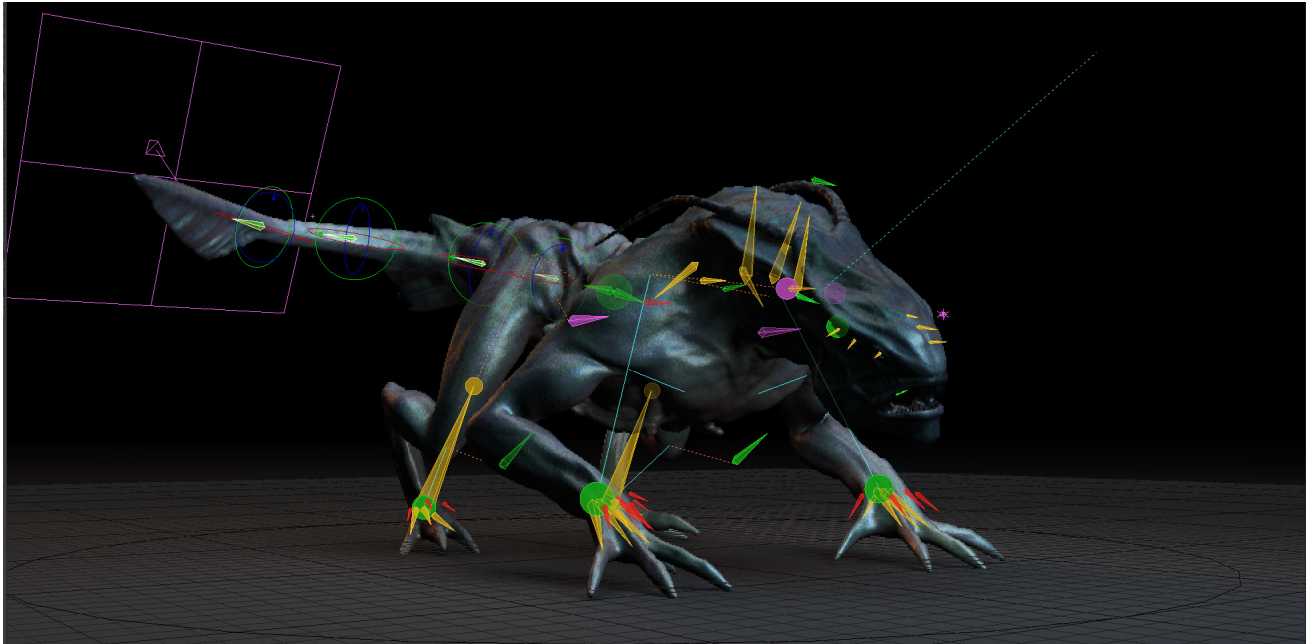
## Motion Blur and Depth of Field Preview

Using the **MB Preview** button (**Render > MB Preview**) or pressing **Shift+F9** will render an **OpenGL Motion Blur Preview** using the current **Camera** settings. The **Motion Blur Preview** can be used in any viewport type. Pressing **Shift+F9** will render an **OpenGL Depth of Field** preview using the current camera. This allows you to quickly visualise the results of the **Camera** settings without doing a full render.





## VPR (Viewport Preview Renderer)

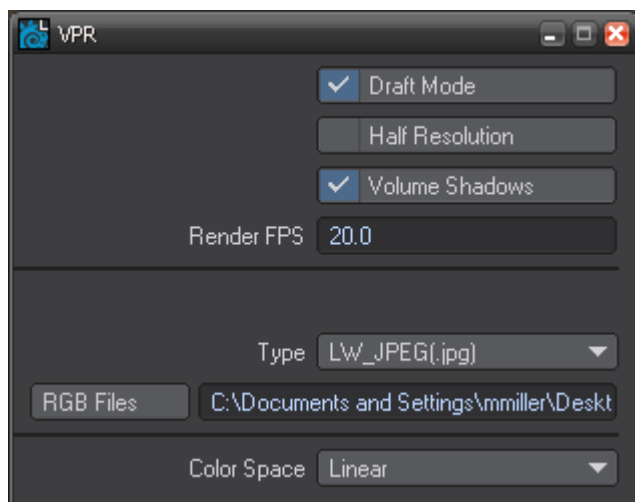


**VPR** is a native Interactive Render integrated with Layout. Essentially, **VPR** is the LightWave renderer in a next gen interactive form. So unlike **VIPER**, the new **VPR** Interactive Renderer handles Radiosity, Ray-Tracing, Reflections, Refractions, and Transparencies to provide stunning LightWave renders as quickly as possible. VPR also supports Hypervoxel rendering.

**VPR** is activated in the Viewport Options menu by selecting **VPR**. Once activated the viewport will begin rendering using the settings in the **Render Globals** panel. **VPR** will continually update when changes are made.

### VPR Render Controls

The render settings for **VPR** are located in the top-right section of each viewport. Look for this button:



**Draft Mode** is an optimized setting for VPR which does not use all of the settings but results in a faster image iterations. Using this mode does not affect any of your F9 renders.

**Half Resolution** will cause **VPR** to render only at half-resolution. This setting is also available in the Additional Viewport Options menu.

**Volume Shadows** will render shadows resulting from volumetrics in the scene. This includes fibers generated from Fiber FX.

**Render FPS** is the frames per second **VPR** will attempt to update the render. A **Render FPS** setting of 30 will attempt 30 redraws per second, while a **Render FPS** setting of 1 will attempt only one redraw per second.

NOTE: Some features will only work in the active viewport, including volumetric shading and plugin lights.

### VPR Save and View Controls

**Type** is the image file type.

**RGB Files** is the location where the files for the color information is stored.

**Color Space** sets the color space options for the saved image.

### Additional VPR Controls

A few more options are available in the Additional Drop Down menu next to the Viewport display menu:

**OpenGL Wireframe** will display the wireframe of the mesh over the **VPR** rendering.

**OpenGL Overlay** will display the OpenGL components such as the grid and transform widgets over the **VPR** render.

When you wish to save the render in the viewport, look for this button in the top-right portion of the viewport:



This will use the settings in the **Save Image** options described above.



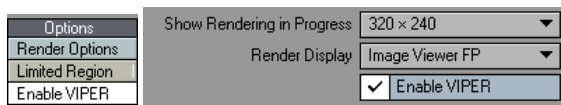
## VIPER

(default keyboard shortcut **F7**)

### Enable Viper

In the **Render Tab** menu, you can enable **VIPER** by clicking on the so-named button (**Render> Enable VIPER**). Be sure to turn it off when you are ready for your final render since it takes memory and processor time, slowing down your renders. **VIPER** stands for Versatile Interactive Preview Render and is a good way to speed up finalising your scene. It can preview volumetric lights, Sky Tracer, **HyperVoxels** and surfaces.

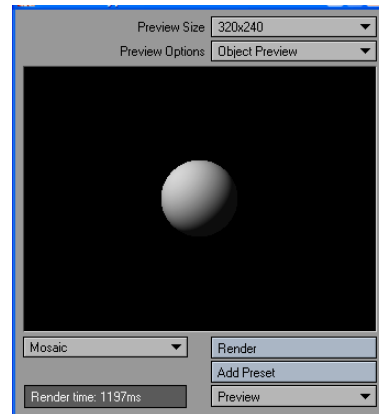
In the render options window, or the **Render Tab** menu, you can enable **VIPER** by clicking on the so-named button. Be sure to turn it off when you are ready for your final render since it takes memory and processor time, slowing down your renders.



## Surface Preview Mode

For surface previews **VIPER** needs to have a data buffer to work on so you will need to render an image for it to work with. Before you hit **F9**, make sure you have **Enable VIPER** turned on in **Render Options** or the button in the **Render Tab**!

Once your render has finished you can open the **VIPER** window by hitting **F7**. When you do, hit the **Render** button and you will see your image take shape in front of you. If you don't have a fast machine, a Draft Mode render (at half resolution) is often enough for you to get what you want and you can always check with a full-res render as and when needed. You can also change the size of the **VIPER** window to get more detail, but be aware that such a change requires a new **F9** render to take effect.



When LightWave renders an image, it generates much more information than the red, green, and blue components of the pixels you see in your images. The color components represent only a small fraction of the data that is generated. LightWave also generates, alpha (transparency), z-buffer (depth), luminosity, diffuse, specular, reflection, shading, shadow, geometry, object, diffuse shading, specular shading, and even custom surface buffers.

As a result, **VIPER** can determine not only what color a pixel is, but also how far back on the z-axis a pixel is, what surface it relates to, and so on. By manipulating this extra data, **VIPER** can change a surface color or **Specularity** setting and show the result amongst the rest of the scene without requiring another full frame render. This will even show the appropriate shading, scene lighting changes, as well as backdrop color changes.

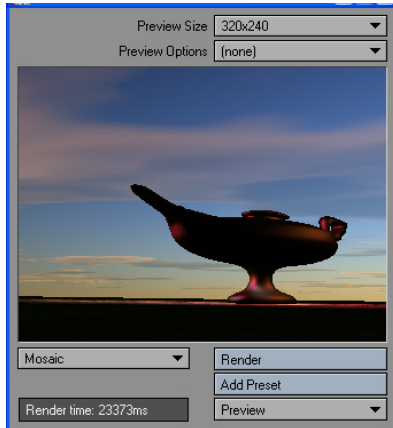
To select a surface in the **Viper** window, you can click on the part of the image containing the surface you wish to edit.

Since **VIPER** does not do a full-scene evaluation, there are some things that are not accounted for, like **Vertex Maps (Weight, UV, etc.)** on SubPatch objects, ray-traced effects (reflections, refractions, shadows), **Shadow Maps**, fog, double-sided polygons, radiosity, light falloff, and so on. As such, it is not a replacement for an **F9** render. Also, **VIPER** will not be affected by moving geometry.



## Other Preview Modes

**VIPER** can be used to preview **HyperVoxels**, **SkyTracer** skies and volumetric lights. In all of these cases full-scene evaluation is not required, so you don't need to perform a render before you can use **VIPER**.



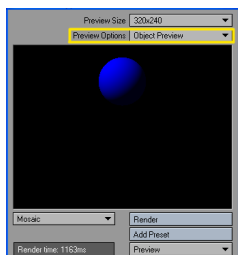
If you render your scene first you can preview **SkyTracer** in **VIPER** with the foreground scene elements.

There are two drop down menus that need to be explained at the top of the window: **Preview Size**, which dictates the size of your **VIPER** window and **Preview Options**. **Preview Size** allows you to dictate what sort of preview you want to work with. Be aware that in **Surface Preview Mode** you cannot change the **Preview Size** without doing another render.

## Preview Options

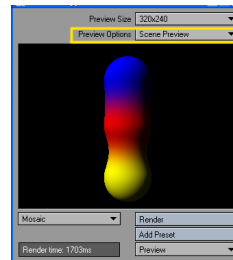
This drop down will only display the option (none) when you are using **VIPER** in **Surface**, **Volumetrics** or **SkyTracer Modes**. If you would like to see what this menu does, try adding some **HyperVoxels** to your scene and then use **VIPER**. You will then see that the **Preview Options** drop down menu has three options — **Object**, **Scene** and **Particle**. The difference is as follows:

### Object Preview



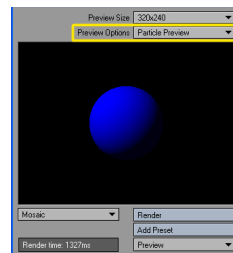
This will show the object you have **HyperVoxels** attached to. If, for instance, you have an emitter selected, the object preview will show the whole stream of particles coming from it. The active object is selected in the **HyperVoxels** window.

### Scene Preview



The scene preview will show all **HyperVoxel** objects that are visible from the camera view. If you have three emitters, this mode will show all three at once, rather than just the active one.

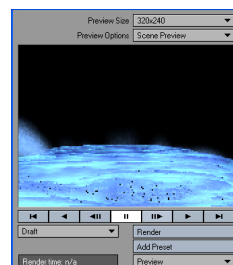
### Particle Preview



This mode shows a close-up of a single particle. This will allow you to see an emitted particle in detail.

### Animated Preview

When previewing **HyperVoxels**, skies or volumetric lights, a very powerful feature of **VIPER** is the ability to make a preview over time. If you click on the **Make Preview** button you can make an animation that can be played back within the **VIPER** window.



NOTE: For any of the **VIPER** modes, you can add a new preset to the current presets window (**F8**). This allows you to keep a library of useful settings for particles, skies or volumetric lighting types.



NOTE: If you have more than one object type in a scene — for instance, a fountain spraying water into an afternoon sky under the watchful eye of a streetlamp — and you want to be able to preview them all in **VIPER**, clicking on the interface for each object type — **Surface Editor** for the fountain, **HyperVoxels** for the water, **SkyTracer** for the sky and **Volumetric Options** for the street lamp — will switch **VIPER** to the appropriate mode.



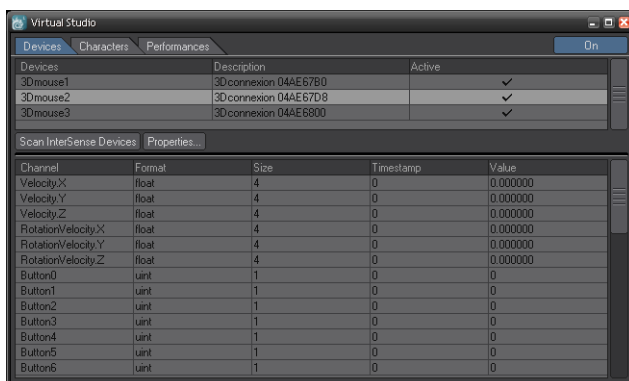
## Virtual Studio

**Virtual Studio** provides tools that let you control your virtual scene from your real-world studio. Using various input devices, real-world actions are captured and applied to a scene interactively. Supported devices currently include 3Dconnexion 3D mice products, such as **SpacePilot** and **SpaceNavigator**, as well as InterSense motion tracking systems including the Vcam virtual camera. With virtual studio, you can take control of the virtual scene in ways more comfortable to you than before possible. With the **Vcam**, your real-world cinematography becomes your scene camera's cinematography. A 3D mouse can aid you in a architectural walk-through. Your interactions can be recorded in one or more "TAKES" and played back. Performances allows you to easily switch among multiple TAKES for review by your colleagues. Like what you recorded, easily 'bake' it to your scene for even greater flexibility and blending with the rest of the scene.

The bulk of **Virtual Studio**'s capabilities is accessed via the "**Virtual Studio**" panel.

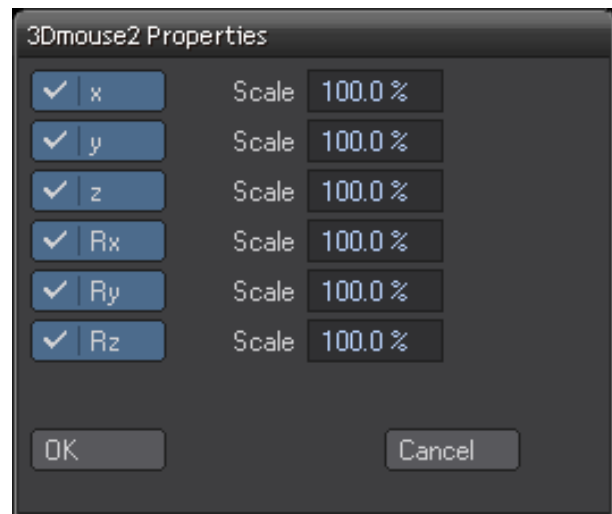


The **Virtual Studio** panel is organized into three 'tabbed' sections: **Devices**, **Characters**, and **Performances**. **Devices** are used to gather real-world input and place it into device channels. **Characters** represent controllable entities with traits that combine device channels. Traits can be recorded into multiple takes for later review. **Performances** are collections of performers, where each performer plays the role of a character and expresses the performance into the scene via shadows. **Shadows** are the way performers connect with the scene by mapping character traits to the scene, such as a camera, light, or object. In the upper-left corner of the panel is an 'On' button. When toggled on, **Virtual Studio** will accept device input and shadows will affect the scene.



The **Devices** tab defines user input devices supported, which currently include 3Dconnexion 3D mouse devices and InterSense motion tracking systems. 3D mouse devices are detected automatically, while InterSense devices can be found by pressing the **Scan InterSense**

**Devices** button. Each device has a name, description, an active state, a list of input channels, and device-specific properties. The properties button will bring up a panel specific to the selected device.

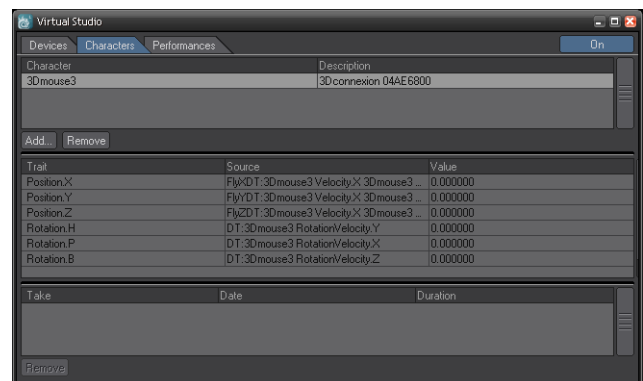


Each device input channel represents real-time data from the hardware device. These channels are fed into characters, which have traits that can be recorded and played back from. When the **Virtual Studio** is 'On', the panel will display this real-time channel data and can be used to confirm proper operation of a device.

Sensitivity defines how sensitive the subtle movements of the 3D mouse puck are. A value of 100% means that the movements close to center are just as sensitive as movements far from the center. Smaller values make movements close to center less effective. This allows users of various skill levels to make subtle adjustments without losing the ability to make extreme adjustments.

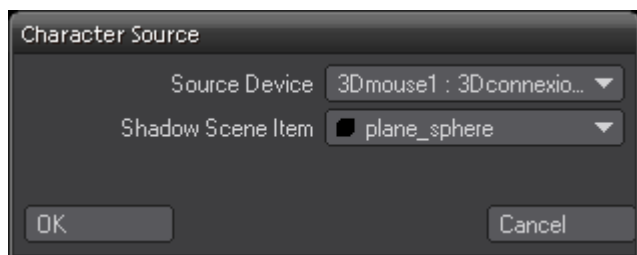
Scale defines what the extreme movement will be while moving the puck to its extreme in a particular direction. For translation, this refers to the number of grid units per second. For rotation, this refers to the number of half spins (180deg) per second.

The 3D mouse properties consist of enabled states for each degree of freedom (translation along and rotation about x,y,z axes) and scale factors for each axis. The **InterSense** device properties consist of a style and a reference item. The style helps identify whether a particular station device is a **Vcam**, hand wand, head tracker, or generic station; each has a unique set of support input channels.



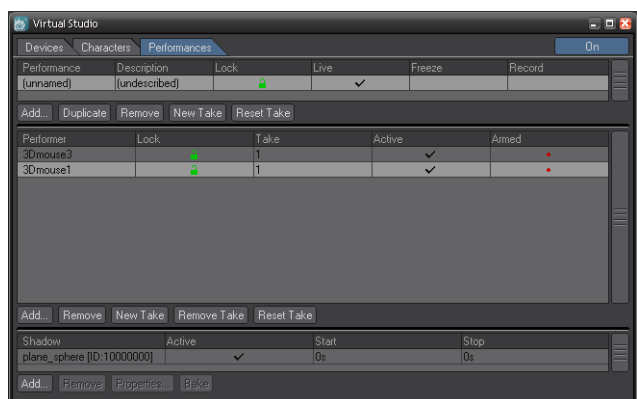


The **Characters** tab shows available characters, its traits, and takes for each trait. Characters have a name and description. Each trait has a name, a source, and a value. The source relates the trait to specific device input channels; therefore each character can technically link each trait to a different device and input channel. At present, however, these are predefined and all character traits are automatically linked to a single input device. The value is the current value of the trait which may be based on live input or a playing back a pre-recorded take. clicking on the value will reset it to zero; this can be useful if the value is getting out of hand. Pressing the **"Add..."** button displays a panel for setting up a new character.



The source device is a link to an input device and the shadow scene item is a link to the scene. In other words, the source is the real-world device to control the character and the shadow is the scene item being controlled by the character. Clicking 'OK' will create and set up the character as well as a performer in the active performance and a shadow for that performer.

Each character trait take has a numeric identifier, a modification date, and a duration. The take id is referenced by each performer. By maintaining multiple takes, a performer can choose among them allowing the operator to choose the best or provide a colleague with options to choose from. When a take is selected, it can be removed by pressing the **"Remove"** button.

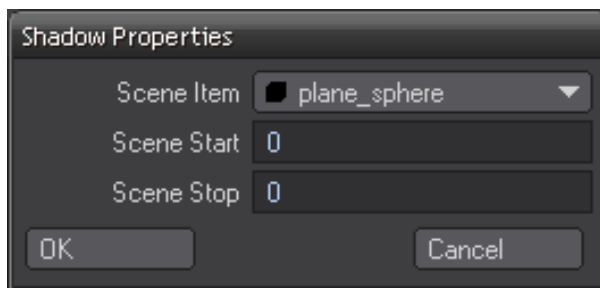


The **Performances** tab manages the recording and playback performers and how they affect the scene. Multiple performances are allowed, and switching among them is easy; simply select the performance name. Each performance has a locked state, live state, freeze state, record state, and a number of performers. When locked, changes to the performance are restricted. When live is enabled, device input is allowed to affect the performance. When frozen, performers will not modify their character traits regardless if they are from live input or a recorded take. This is useful to keep everything in place while performing some analysis or rendering a test frame. When recording is enabled, all performers that are 'armed' will be recorded. Only one performance can be active at a time, but each performance can be different. By default, there is a starting performance.

Each performer is linked to a single character and its traits. Each performer has a locked state, an active take id, an active state, an armed state, and a list of shadows. When locked, changes to performer

are restricted. The active take id indicates which take will be recorded to and/or played back from. When active, the performer will affect its character whether via live input or a recorded take. When not armed, the active take will be played back, while when armed, the active take will be recorded (but only when the performance recording is enabled).

Each shadow defines the mapping of character traits into a scene item. Without a shadow, none of the performance can be realized. Each shadow has a scene item, an active state, a start time, and a stop time. When active, the shadow will affect its referenced scene item but only for scene times between the start and stop times. When the stop time is the same as the start time, it is assuming no stop time is specified. A shadow can also be used to 'bake' to the scene item directly. Baking will convert a recorded take to key frames in the associated envelopes belonging to the scene item. Shadows work by assigning plugins to specific scene item envelope channels. For example, the "Position.X" will have a channel plugin that links it to the "Position.X" trait of a character. Once baked, the key frames can be manipulated like any other LightWave envelope. Baking is also available as an option via the **Graph Editor** panel; however, this approach to baking will disconnect the envelope channel from the shadow; additional setup work is required to reconnect the shadow if the bake was not desired. This setup is accomplished by reselecting the shadow scene item via the shadow **"Properties..."** button.



The panel allows adding and removing performers to/from the selected performance and allows creating new takes, removing the active take, and resetting (erasing) a take. Resetting a take is useful when the operator wants to record over an existing take. The panel also allows adding, removing, and duplicating performances as well as creating and resetting new takes for all performers in the performance. When creating a new take for a performance, all performers will have the same new take id; this consistency helps avoid confusion.

To record a take, a performer must be armed and its performance must be in record mode. The current scene time/frame specifies where to store trait values in a take. Device input occurs in real-time, while recording occurs in scene-time. This allows one to perform stop-frame animation or even reverse scene-time recording. Random access to the scene time can provide interesting results; but, more desirable is to use the **Layout** transport controls to play the scene time forward. When this approach is used, the recording will automatically stop once the last scene frame is reached. To play back what was just recorded, unarm the appropriate performer, adjust the scene time as desired and play forward.

**InterSense** devices specify position/orientation. The **Vcam** supplies a zoom rocker channel, which a shadow can apply to the scene camera zoom factor. The real-world position/orientation will be mapped into the virtual world (scene) using a **RVTransform** (Real-world to Virtual-world transformation) scene item. **Virtual Studio** automatically adds a **RVTransform** scene item when scanning for InterSense devices.





Scaling the RVTransform item will scale the real-world positions. By zeroing the scaling along a specific axis, one can create dolly moves or confined motion. By parenting the **RVTransform** scene item to another scene item, such as a moving car, it is possible to record special camera angles from a stationary real-world position.

**3Dconnexion** devices affect character traits by 'flying' them. The 3D mouse 'cap' controls relative movement rather than absolute position.

Device properties are remembered in Layout's config file, while all other **Virtual Studio** data is saved with a scene.

All recording/takes are measured in scene time units (which are seconds). The scene **FPS** (frames per second) value determines the conversion from seconds to frames and vice-versa. In order to record a virtual studio take faster or slower than scene time, we must control the scene time advancement. The current approaches are: 1) advance the scene time as fast as computer will allow, which is not consistent or predictable; 2) advance at the scene time such that it matches time in the real world (one second of scene time advances in one second of real world time); 3) have the user step through scene time using 2D GUI controls or keyboard shortcuts. None of these offer the ability to scale scene time relative to real world time smoothly and consistently, which is exactly what we need if we want to perform slow-motion or high-speed recording of the real-world input devices. We can not simply alter the scene FPS value. A scene-time to real-world-time scale factor would do the job; when 100% (1.0), one second of scene-time would pass in one second of real-world-time. When 200% (2.0) two seconds of scene-time would pass in one second of real-world-time. When 50% (0.5) 0.5 seconds of scene-time would pass in one second of real-world-time. The value would operate independent of scene FPS.

The **Layout** GUI currently has a preview "Step" field that is used when the play forward/backward buttons are pressed and only when the "Play at Exact Rate" feature is off. This field dictates how many frames to skip when advancing the scene time. Since it is a frames value, it is also dependent on the scene FPS value. The "**Play at Exact Rate**" feature is the only way to enforce a consistent passing of scene time relative to real world time. Since, the "**Step**" field is only valid when this feature is off, we can overload this screen location for a field that is only valid when this (exact rate) feature is on. So, when off, the field would be the frame step; and, when on, the field would be the frame rate factor (time scale).



By doing this, the virtual studio recording can be done at one rate and playback can occur at a different rate.



## Visor

Visor shows images and renders in a Layout viewport. The images and renders are organized in panes. A pane can show one or more images and/or renders, and there can be multiple panes in multiple viewports. Panes can be arbitrarily sized from taking up very little space, to a strip along the side of a viewport, to covering an entire viewport.

Visor can be used to keep reference photos conveniently available in Layout without having to open yet another window. Visor can also be used to show the last few F9 renders in a viewport without the need to have the render window open.

### Interacting with a pane

A pane can be activated by clicking in it or by selecting the Null to which the Visor custom object is associated. Note

that sometimes other items in the scene may interfere with clicking in the pane. Clicking near the corners usually works best.

If there are multiple panes, all the panes will at first become active. You can then click on the pane you want to work with by clicking in it again.

An active pane will show six handles. Five green ones which can be dragged around to resize and reposition the pane, and a red one which moves the slider. You'll need to have add some images with the Image Editor in order for the slider to be useful.

The panes can be deactivated by selecting some item in the scene. Initially you may have to do this by changing the Current Item. Afterwards Visor will keep track and return you to the last tool used and the items last selected.

## Images and renders

Panes can show both images present in the Image Editor, and any F9 renders. To have an image show up in a pane, simply load the image into the Image Editor.

To have renders appear in panes, the Render Display in the Render Options has to be set to the new entry called "Render2Image". This new option has been added by Visor and it adds renders to the Image Editor instead of having them appear in the render display. From there they are picked up by the Visor panes.



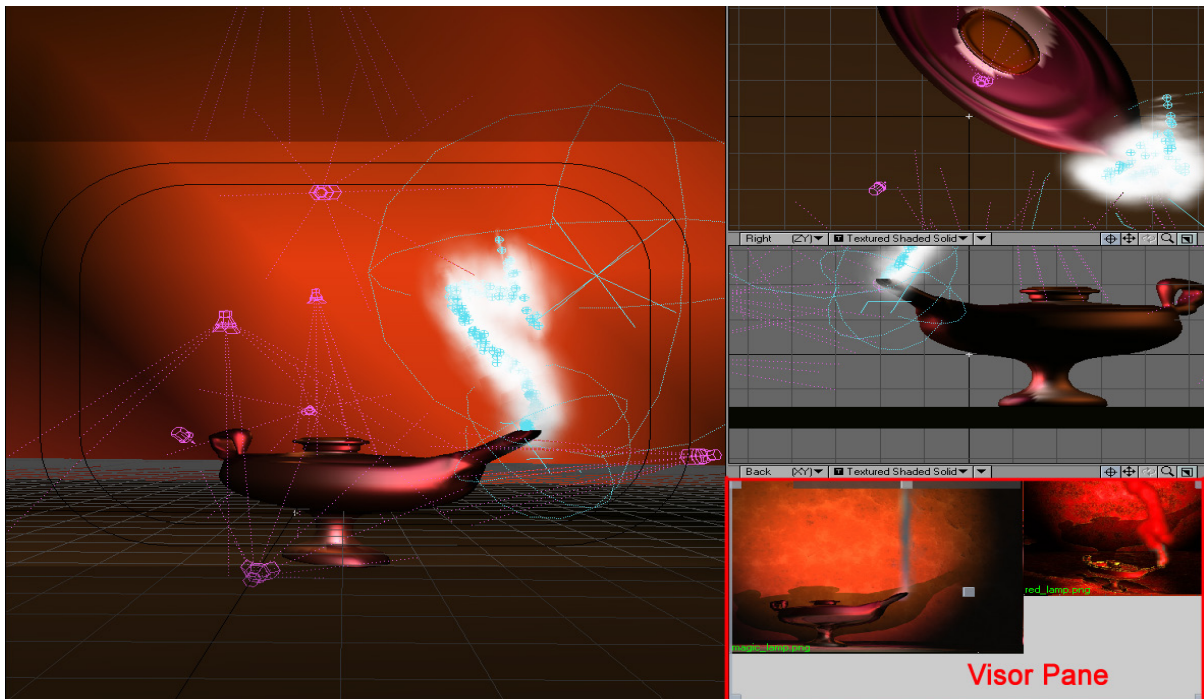
Note: Visor can be used without using Render2Image, but F9 renders won't appear in the panes.

References to images which are shown in a pane will be saved into the scene file, even if they aren't used as part of the scene. However, this does NOT apply to renders. Renders will be lost if the scene is cleared (you can always access a render through the Image Editor and save any you want to keep from there).

For renders, depending on how much space is available, Visor shows some helpful information about each render, for example:

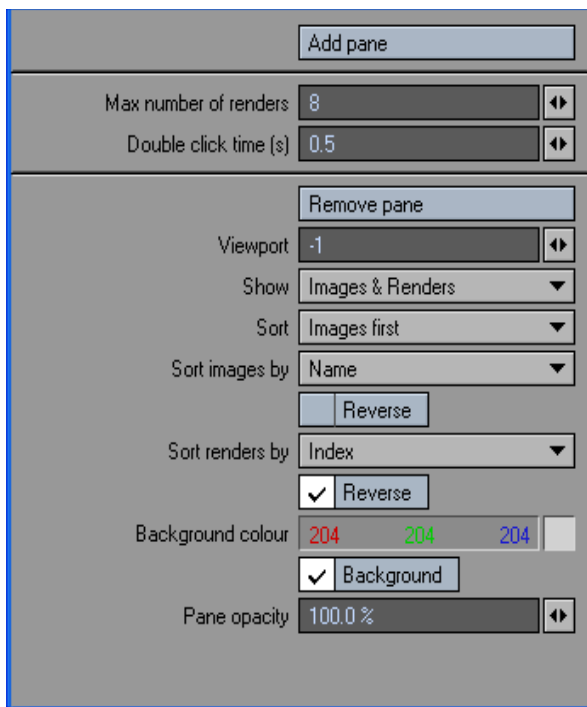
[8] 12/18/03 01:28:12, 23s, 12f, (640 x 480)

This reads as: this is the 8th render made, at the given date and time, took 23 seconds to render, it's a render of frame number 12, and was rendered at a size of 640 by 480 pixels. The date and time formats may vary depending on your locale.





## The interface



The interface can be opened either by double clicking on the Visor custom object entry, or by double clicking on an active pane.

As well as a button to add a new pane, there are also a set of global settings, and a bunch of settings for the currently active pane.

### Global settings

#### Max number of renders

Each rendering takes up memory to keep around. The number of renders Visor keeps around is limited to the given number of most recent renders. If the maximum number of renders is reached, the oldest render is deleted. Note that this does not affect mages, only renders.

Default: 8

#### Double click time

This gives the maximum amount of time (in seconds) between clicks on a pane in order for it to be registered as a double click. Double clicks on an active pane open up the Visor interface.

Default: 0.5

### Pane settings

#### Remove pane

Clicking this button removes the currently active pane.

### Viewport

Sets the index of the viewport in which the current active pane is placed. Viewports are numbered from zero up to a maximum of three. 0 means the first viewport, 1 the second, and so on. Which viewport is the first, second, and so on depends on the viewport layout.

Negative viewport numbers indicate a viewport index relative from the last viewport. So -1 means that the pane will always appear in the last viewport,

-2 means the second to last, and so on.

Default: -1

### Show

Selects if the currently active pane will show images only, renders only, or both types.

Default: Images & Renders

### Sort

Sets the sorting order between images and renders for the currently active pane.

Default: Images first

### Sort images by

Sets the sorting order of the images (if there are any) for the currently active pane. If Reverse is checked, the ordering of the images is reversed.

Default: Name

### Sort renders by

Sets the sorting order of the renders (if there are any) for the currently active pane. If Reverse is checked, the ordering of the renders is reversed.

Default: Index, Reverse

### Background color

Gives the color used to fill the background of the currently active pane. If Background is checked, the background is drawn, otherwise it isn't drawn.

Default: 204 204 204, Background

### Pane opacity

Sets the opacity with which the currently active pane is drawn. The pane opacity is only used if the pane is not active. When the pane is active, it will always be drawn completely opaque.

Default: 100.0%





---

## Chapter 19: Rigging

---





## Rigging: Introduction

Part of preparing a scene is the rigging process. Rigging a character, or any other object, involves setting up a system of interactions, in many cases using bones and applying the rules and limitations to those bones. There is no one way to prepare a rig, so your rig may have different rules from someone else and still achieve the same goal.

A number of tools can be applied to a rig, including all of the tools in the Setup Tab. Additional functions are available in the Motion Options Panel, where you can apply Inverse Kinematics and other rules for motion, and IK Booster, which can be used to augment normal IK, as a separate IK system, and even have dynamics applied to your rig. You can also Export and Import your rig data for use in other models.

## Setup Tab

### General

### Bone Edit Mode

**Bone Edit Mode (Setup > General > Bone Edit: Enter Bone Edit Mode/Exit Bone Edit Mode)** allows you to edit your character rigs quickly and easily within the scene you are creating.

### Enter Bone Edit Mode

(default keyboard shortcut **Shift E**)

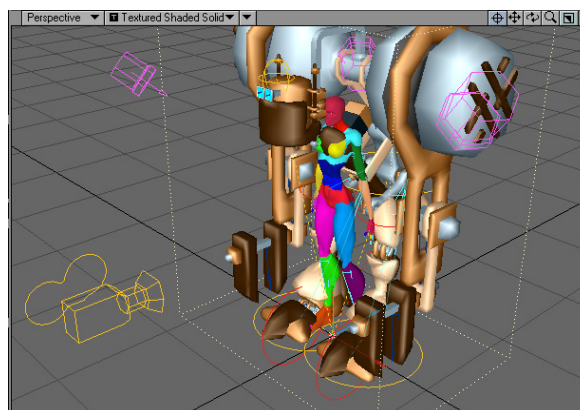
Entering bone edit mode will do a number of things to help you work on character rigs in your scene.

When it is invoked it will turn off the visibility of all items in the scene except the object that you have selected and will take you to frame 0. All of the items that have been hidden will also be locked so there can be no accidental selection of another object. Enable IK will also be toggled off but can be turned back on when needed.

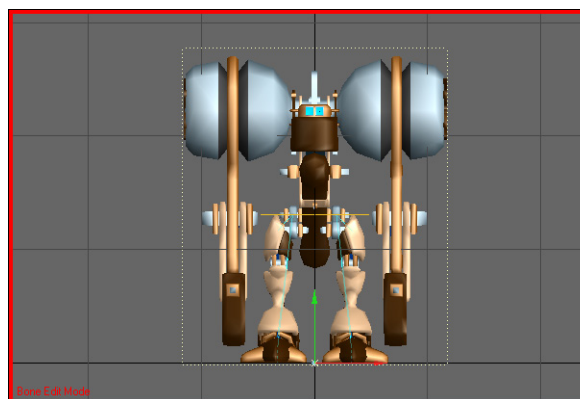


**WARNING:** If you use the **Scene Editor** while in **Bone Edit Mode** make sure you only edit the character that you had selected in **Bone Edit**

**Mode.**



Before entering Bone Edit Mode.



After entering Bone Edit Mode (note that a red border appears around the viewport).

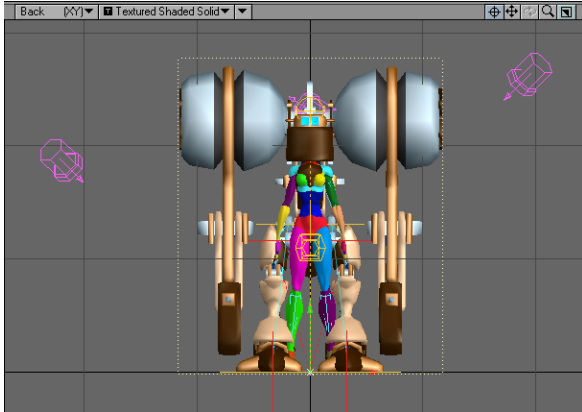
The other character that shared the same origin has been hidden and locked.



## Exit Bone Edit Mode

(default keyboard shortcut **[Shift] D**)

Exiting Bone Edit Mode will restore all visibility and locked items to their original state. You will have to move to the original frame manually.



After exiting **Bone Edit Mode**.



NOTE: Moving the time slider from frame 0 will cause LightWave to exit **Bone Edit Mode**.

The objects do not need to share an origin for **Bone Edit Mode** to work. Selecting any object in the scene and invoking **Bone Edit Mode** will work. But some of the **Bone** tools (Mirror Hierarchy) are easier to use if the character is worked on at X= 0, Z= 0.

### Bones On/Off

Turns all bones on or off.

**Enable IK** (default keyboard shortcut **[Shift] F8**)

When **Enable IK** (**Setup>General>Enable IK**) is active LightWave's IK will be applied to all items with IK set up on them.



NOTE: It is sometimes necessary to disable IK when editing a rig.

## Enable Deform

(**Setup>General>Enable Deform**) will toggle on or off LightWave's ability to deform a mesh with bones.

## Add

### Bone

**Setup > Add > Bone**. A bone will be added with its base at the selected object's local Origin, along its positive Z axis.

### Child Bone

(default keyboard shortcut =)

**Setup > Add > Child Bone**. A bone will be added with its base at the tip of the selected bone, along its positive Z axis. The new bone will be a child of the original selected bone.

### Draw Bones

Draw Bones will allow you to draw a bone in any orthogonal viewport (This will not work in Perspective). This bone will be parented to the object. Each successive bone added will also be parented to the object. If you need to add a child bone use **Add Child Bone** or **Draw Child Bone**.

#### To draw a bone in a viewport:

**Step 1:** In any Layout orthogonal-view viewport, select the object, or existing bone if you are drawing a child bone.

**Step 2:** Select **Setup > Add > Draw Bones**. (Since this is the initial bone, you can alternatively choose **Draw Child Bones**.)

**Step 3:** Place your mousepointer where you want to place the base of the bone.

**Step 4:** Drag your mouse to create the bone.

### Draw Child Bones

(**Setup>Add>Draw Child Bones**) allows you to draw child bones from a parent.

#### To use Draw Child Bones:

**Step 1:** Select a bone or object that you want to add a bone to.

**Step 2:** Run **Setup>Add>Draw Child Bones**.

**Step 3:** Use the **LMB** and draw out the bone to the length and in the direction you want.



NOTE: If you want to continue adding bones just keep drawing more bones. The next bone will be added as a child to the previous bone. If you want to add a child to a different bone than the one that was just created you will have to select the bone first.

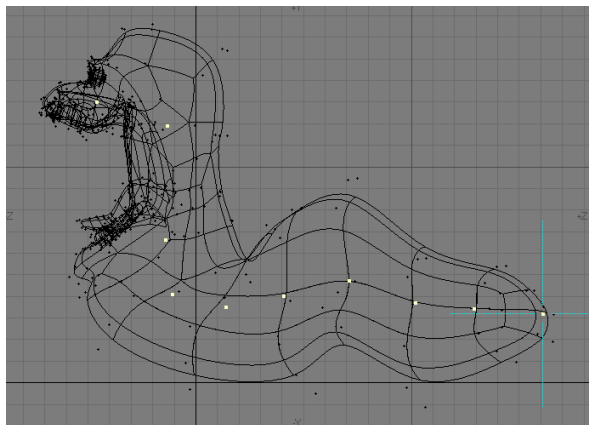


## Convert Skelegons

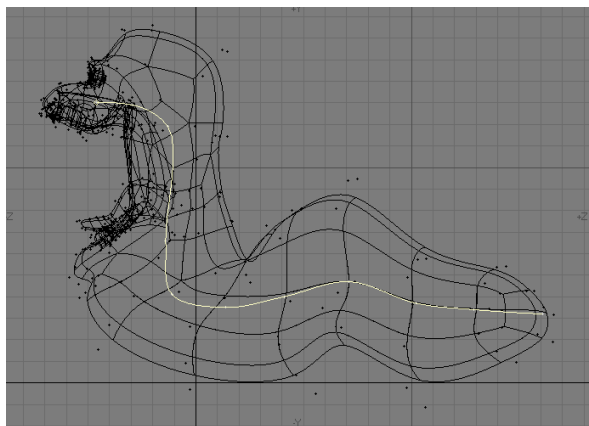
One way to create **Skelegons** is to create curves first. You can then convert the curves into **Skelegons**, which will be placed between the curve knots (points). With **Convert Skelegons**, (**Setup>Skelegons: Convert Skelegons**) you can even use points from the object to create the curves.

### To use Convert Skelegons:

**Step 1:** In an open Modeler layer, create points that will correspond to the bases and tips of the bones you want to eventually create. (You'll probably want the object to be boned in the background for reference.)

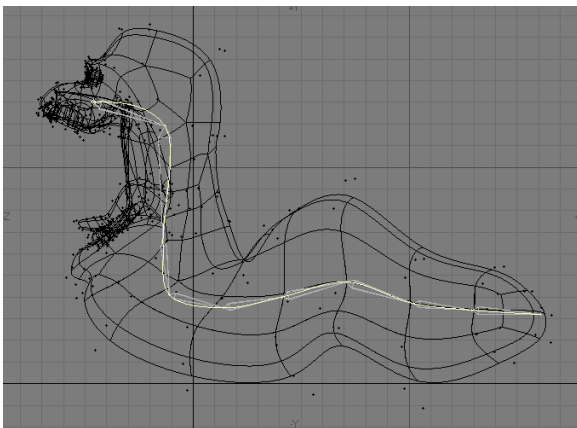


**Step 2:** Select points, working from the (theoretical) base of the first child bone to the tip of the last bone in the chain. Then choose **Create >Make Curve > Make Open Curve** to create an open-ended curve with your points.



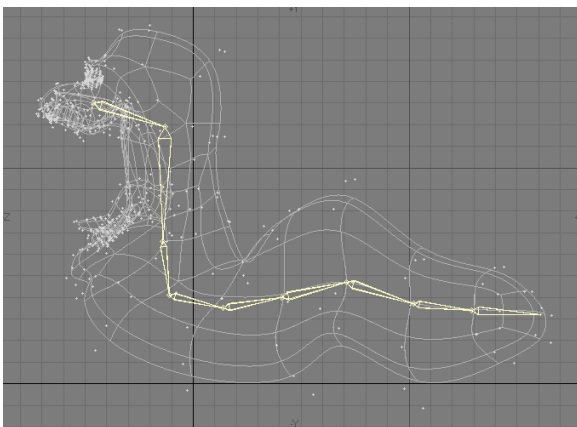
**Step 3:** Repeat the previous step for all chains, if you have more than one. You can use the same point in multiple chains.

**Step 4:** Still in Modeler, choose **Setup>Skelegons: Convert Skelegons** to create the **Skelegon** from the curves. You can edit the base/tip of the bones using standard modelling tools. The **Skelegon** bones will stay attached to each other.



If you desire, you can select and delete the curve afterward.

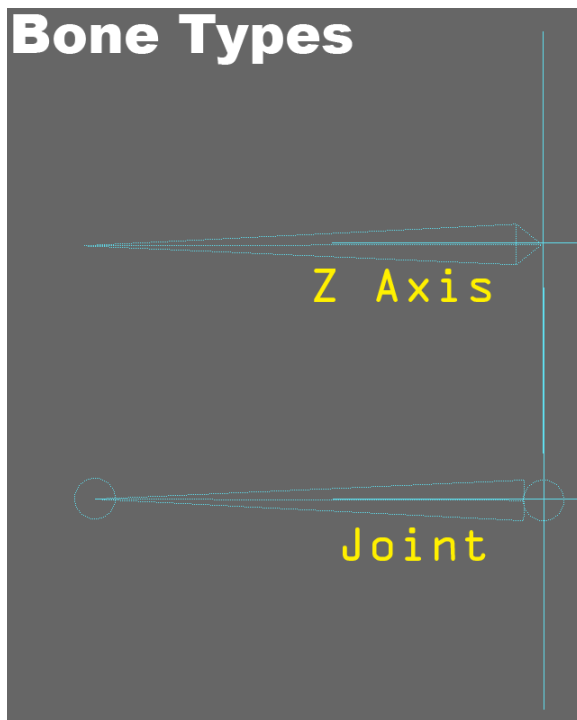
**Step 5:** If you want, you can optionally cut and paste the **Skelegons** into the layer with the normal object geometry.





## Joints

A new Bone type is now available, Joints.



The older Bone type has been labeled Z-axis. You can change the bone type in the Bone Type drop-down menu in the Bone Properties Panel.

The easiest way to visually tell a Joint from a Z axis bone is the small circle where the bone types are connected. The small circle is actually the Joint, two connected joints are represented by a triangle pointing towards the child joint. Keep in mind that a single joint will be represented by only the circle portion as it does not have a child.

Deformation is entirely defined by the change in position of the joint. Currently, joint rotation has no effect whatsoever on deformation. Joint-based animation and deformation control allows for a “stretchy bones” effect with considerable control for the user. A more organic and natural motion effect for shoulder and neck movements can be achieved, for example, or a more exaggerated effect for cartoon-type characters. Even zero-length bones (e.g. isolated joints) can cause deformation (activate them and turn off Multiply Strength by Rest Length).

**Add Joint:** A bone will be added with its base at the selected object’s local Origin.

**Add Child Joint:** A bone will be added with its base at the tip of the selected bone, The new bone will be a child of the original selected bone.

**Draw Joint:** Draw Bones will allow you to draw a Joint in any orthogonal viewport (This will not work in Perspective). This bone will be parented to the object. Each successive bone added will also be parented to the object. If you need to add a child bone use Add Child Bone or Draw Child Bone.



NOTE: As of this writing, Z-axis and Joints are not compatible, so it is recommended you do not mix the two types on a bone chain.

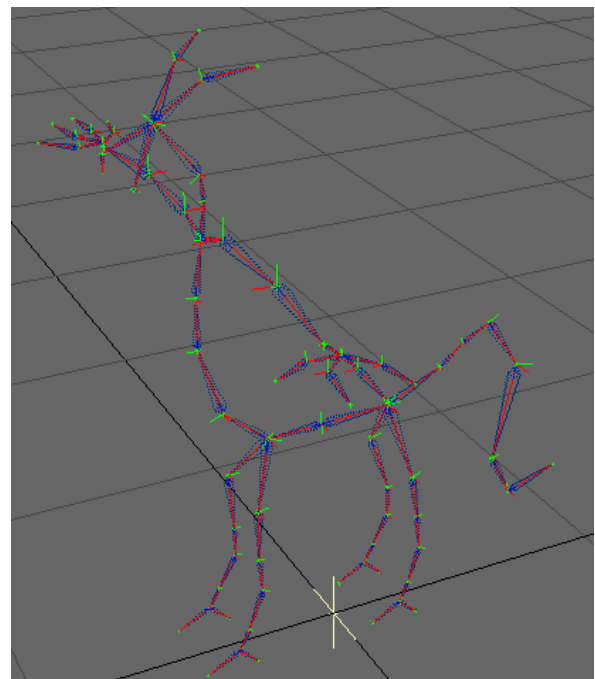
## Modify

**Joint Move** (default keyboard shortcut **Ctrl J**)

Joint Mover draws lines along each bone and puts a cross hair at the base of the child bone(s), which is usually coincident with the tip of the parent (but need not be).

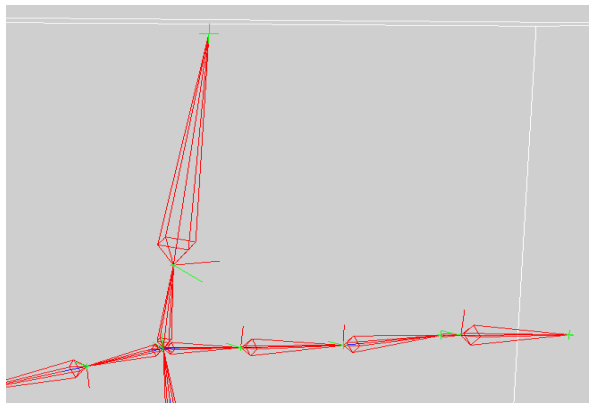
In any view, you can then click the **LMB** and drag the cross hairs to the new location where the tip of the parent and base of the child should be, in effect, moving the joint location between two bones. After the mouse button is lifted, hit the space bar to actually change the bones. Using the **RMB** will constrain the bone to move along its length.

Attributes edited in LightWave will be the parent’s rest rotation and rest length, and the children’s rest rotation, rest position, and rest length.

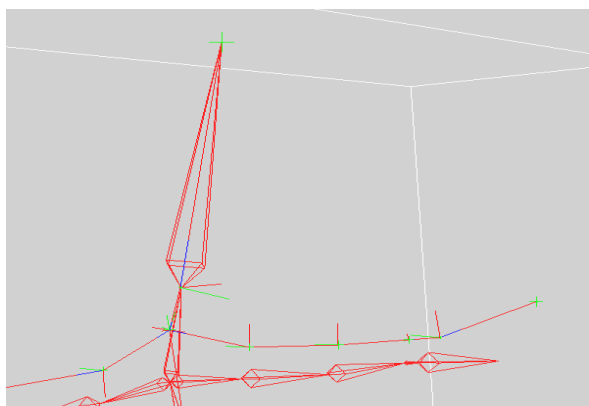




If you select the object and run **Joint Mover**, the tool will be drawn through the entire hierarchy.



Here is a hierarchy with **Joint Mover** active. You can see the green cross hairs that are drawn at the joints and at the tip of the single bone at the end.



Using the **LMB** each one of the joints and the tip of the bone at the end has been moved (one at a time) to a new location.

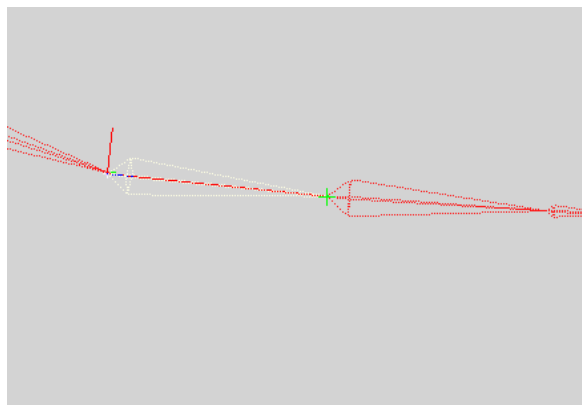
After all the editing is done, hitting the space bar will exit **Joint Mover** and move all the bones to the new locations.



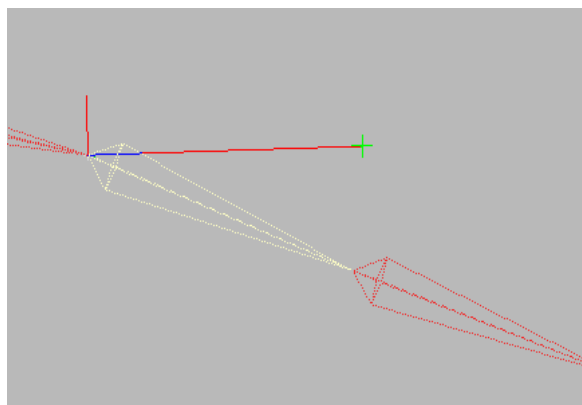
**NOTE:** If you want to execute your edits but keep **Joint Mover** active just invoke **Joint Mover** again instead of hitting the space bar. This will move the bones and keep **Joint Mover** on.



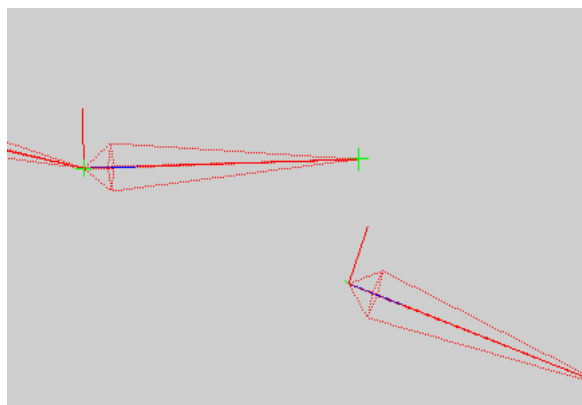
**WARNING:** These edits cannot be undone.



A bone's tip can be moved away from the base of its child by selecting the bone you want to edit and running **Joint Mover**. The tool will only be active in the selected bone which is not the case when the object is selected and the tool is drawn through the whole hierarchy.

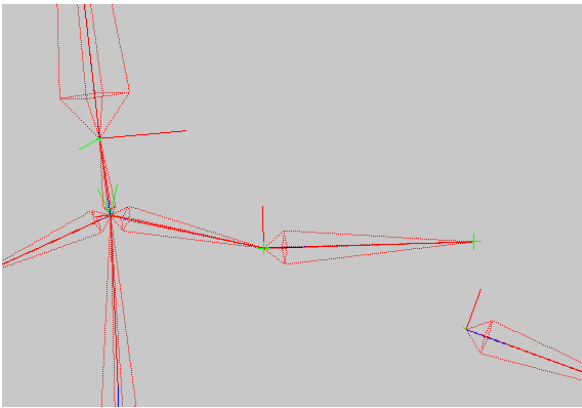


This will allow you to move the tip of the bone away from its child's base.

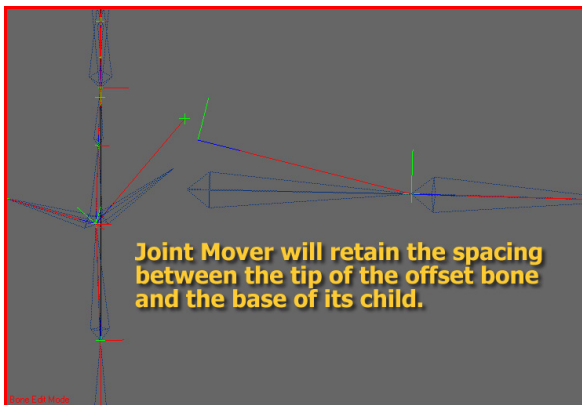


After the tool is dropped (hit space bar), the bone is moved to its new position.

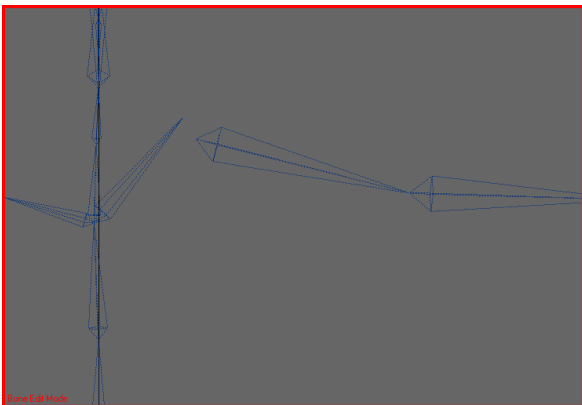




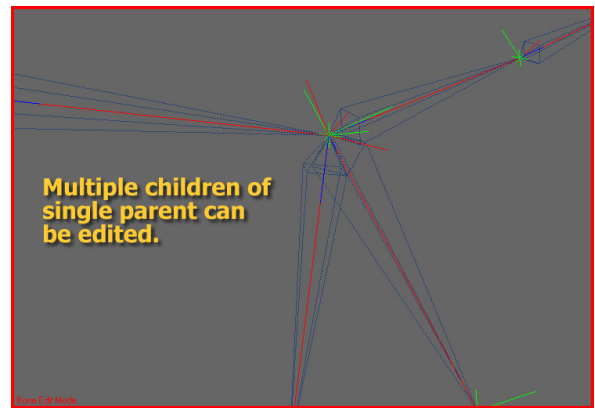
If you have bones whose tips have been offset, **Joint Mover** can still be used to edit the joints' position.



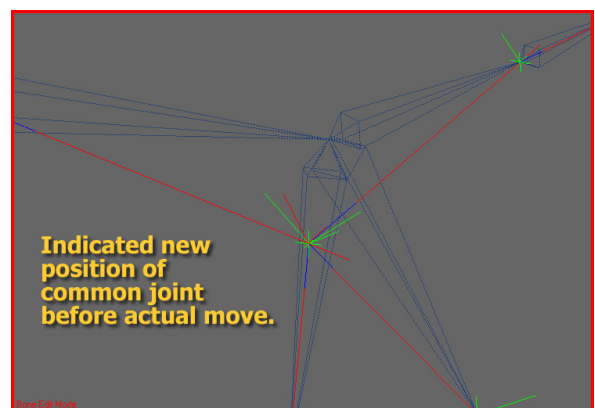
The tool will behave just as it does when the parent's tip and its child's base are touching.



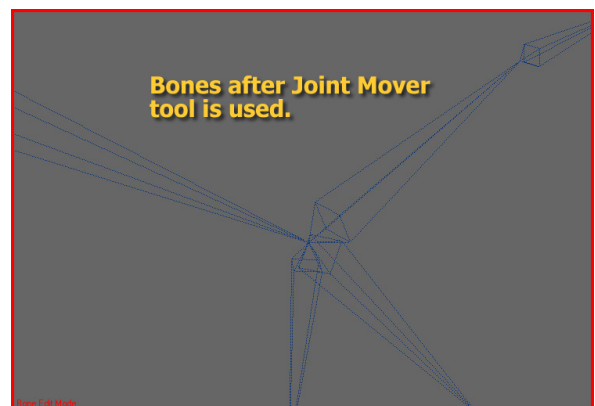
This is the result of that edit. The spacing is maintained even with the bones in their new position.



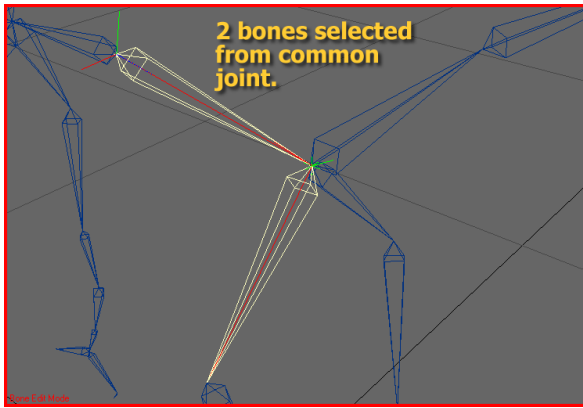
If multiple child bones share a joint with the same parent you can move them by selecting the object and drawing **Joint Mover** through the entire hierarchy. Or you can choose the bones individually.



Use the **LMB** and drag the common joint where you want it to be.



Once the edit has been executed, the bones will be drawn in the new location still connected.



You can choose what bones in a common joint you want to move. As long as the parent is selected, as many of its children as need be can be selected and moved.

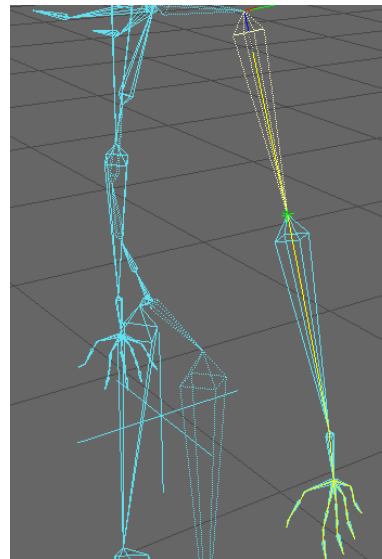
## Tip Move

(default keyboard shortcut **Ctrl** **T**)

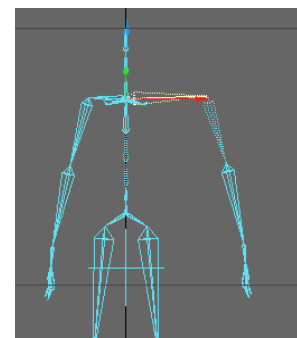
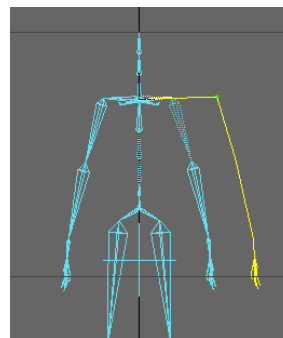
**Tip Move** will allow you to move an entire hierarchy by moving the tip position of the hierarchy's parent bone. Attributes edited in LightWave will be the bone's rest rotation, rest position, and rest length.

### Single Bone

If a single bone is selected, **Tip Move** draws a line along the bone and puts a cross hair at the tip of the bone. If there is a hierarchy with a child relationship to the selected bone a yellow line is drawn through the remaining bone but there will be no cross hair at the tips. Only the tip of the selected bone will be editable.



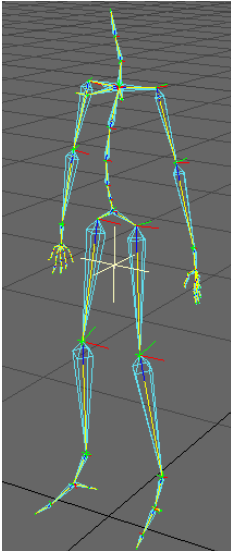
If you use the **LMB** to move the selected bone's tip, its hierarchy will move along with it. Even though the parent bone's angle and size change the hierarchy, it will maintain its original configuration. If the **RMB** is used on the selected bone's tip, it will expand along its own length.



Once you get the selected bone's tip where you want it you can hit the space bar and all the bones will be moved to the correct place. Or if you want to continue using the tool just select **Tip Move** again. All the bones will be moved and **Tip Move** will stay active.



## Entire Contiguous Hierarchy



If the object itself is selected and **Tip Move** is activated. The tool will be drawn through the entire contiguous hierarchy.

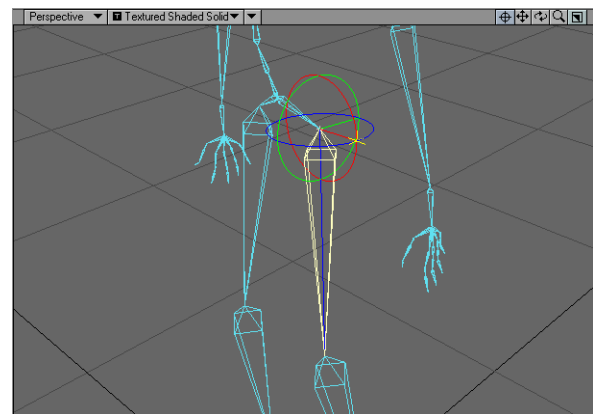
In this mode every joint will be editable instead of just a single bone. But the same rules apply to the child/hierarchy of the tip that is being moved. Even though the parent bones' angle and size change the hierarchy, it will maintain its original configuration. If the **RMB** is used on the selected bones' tip, it will expand along its own length.

Once you get the selected bones' tip where you want it, you can hit the space bar and all the bones will be moved to the correct place. If you want to continue using the tool just select **Tip Move** again. All the bones will be moved and **Tip Move** will stay active.

## Bone Twist

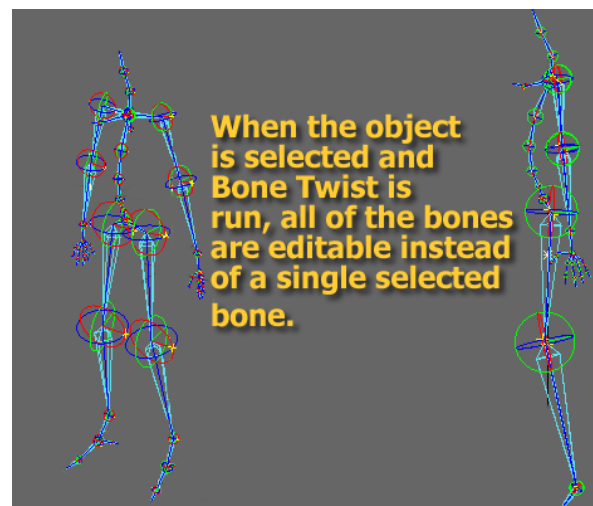
(default keyboard shortcut **Ctrl** **K**)

**Bone Twist** allows you to change the bank rotation of a bone while leaving children of that bone unaffected. If no bones are selected, the tool will allow you to go bone to bone and adjust each bone's bank. When invoked, this tool draws a circle around the base of the selected bone in blue, and draws representations of the three axes of the bone (in local coordinates). These drawn axes represent the right, up, and forward (x, y, and z) orientation vectors.



When the **LMB** is held down on the yellow cross and the mouse dragged right, the drawn axes rotate in a positive direction, when the mouse is dragged left, rotation is in the negative direction. Rotation is the similar to doing a Rotate operation in LightWave's standard interface, and changing the bank angle by dragging the yellow cross in the blue circle.

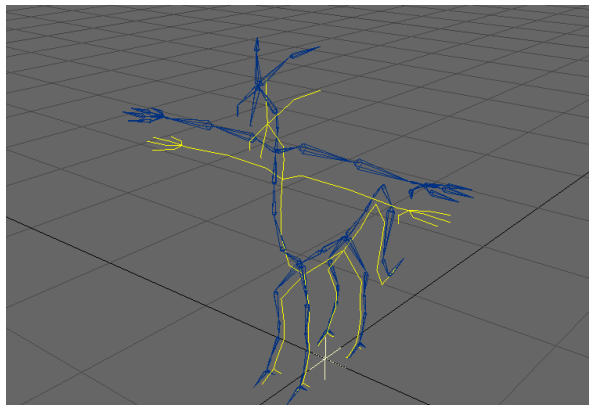
However, the children of the selected bone will remain in place with **Bone Twist**. After the mouse button is lifted, hit the space bar to actually change the selected bone's bank angle, or invoke the **Bone Twist** tool again to have the rotation of the bone occur with the tool remaining active.



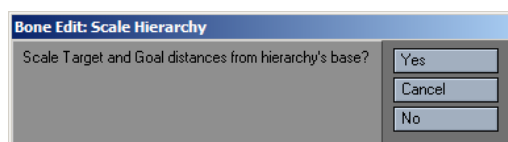
Selecting a bone and running **Bone Twist** will allow you to edit the bone selected and that bone only. Selecting the object containing the bones and running **Bone Twist** will allow you to edit all bones contained in the object. The reason for this is to give you the ability to isolate a bone that may be difficult to see when all bones have the tool active.



## Scale Hierarchy



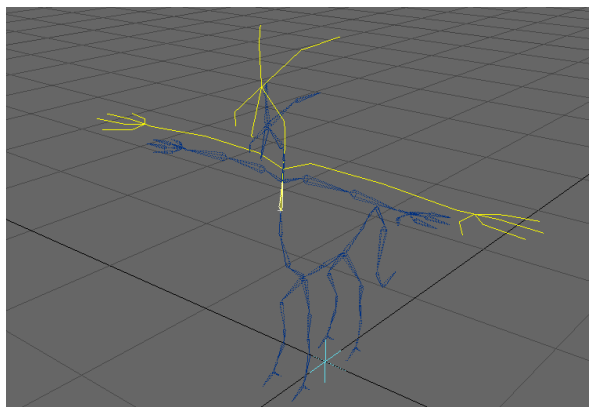
Scale Hierarchy will allow you to scale a hierarchy up or down. This can help you when you import an existing hierarchy into a new mesh or if you are reusing pieces of your rig in another part of your object.



Select the object to scale the entire bone structure. If you want to only scale a section, select the parent bone of that particular hierarchy. You can also choose to scale targets and goals along with the bones.

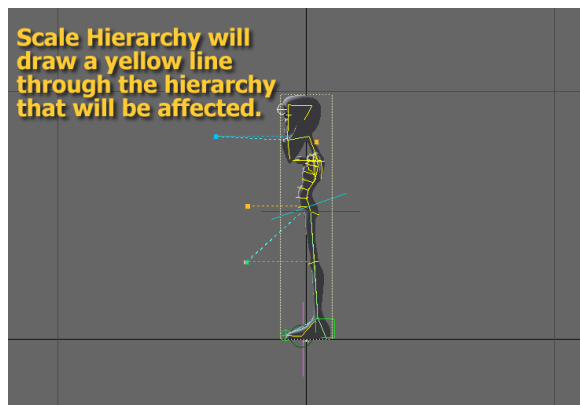


**NOTE:** Depending on the configuration of the hierarchy you may have to adjust the target and goal positions manually.

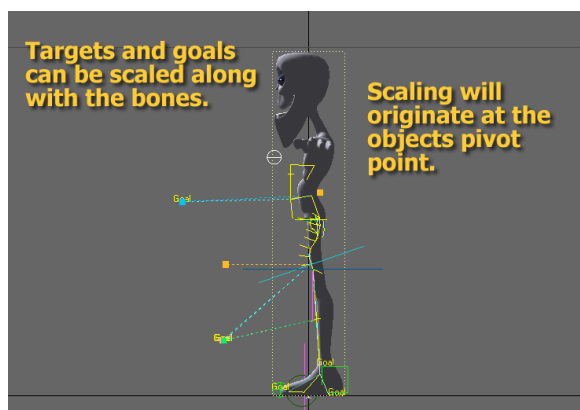


Selecting the parent bone of a hierarchy will allow you to scale just that part if needed.

Use the **LMB** to make coarse changes and the **RMB** to make fine changes.



In this example the object has been selected so Scale Hierarchy will affect all the bones.



If you choose to scale targets and goals. Scale Hierarchy will attempt to move them along with the bones.



## Orientation:

### Align Pitch

Twists the selected bone so that its pitch rotation axis is coplanar with that of its parent.

#### To use Align Pitch:

**Step 1:** Select the child bone whose pitch you want to align with its parent.

**Step 2:** Run Align Pitch located under **Setup** > **Modify** > orientation: **Align Pitch**.

**Align Pitch** will change the bone's rotation and automatically move to the next child bone. This makes it easy to just step down a chain you want to align.



**NOTE:** Ensure that IK is disabled and that **Bones** are turned off. An easy way to do this is to use **Bone Edit Mode**. **Setup**>**General**>Bone Edit:

**Enter Bone Edit Mode.**

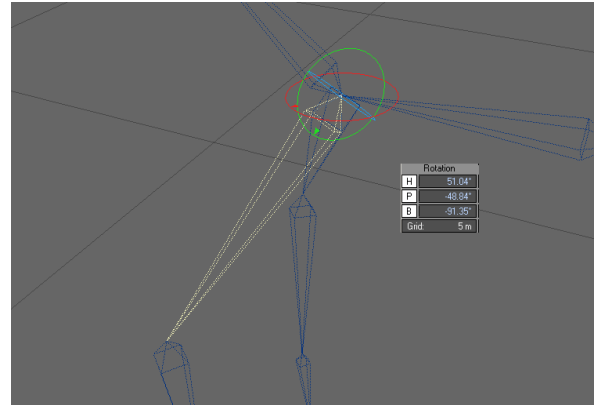


**WARNING:** This operation cannot be undone. Please save your work before executing the tool.

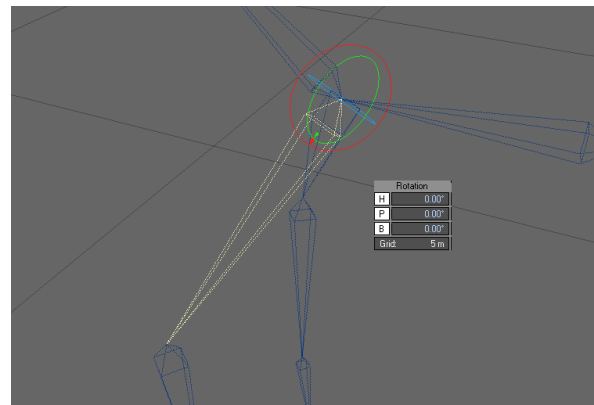
## Record Pivot Rotation

(default keyboard shortcut **Shift P**)

**(Setup>Modify>Orientation>Record Piv Rot) RPR** is used to zero out the current items' rotation at its current physical rotation. Basically it masks the real rotational values and allows LightWave to see H=0, P=0, B=0 after it is set. This can help avoid problems with Gimbal Lock.



Before RPR is applied in this example the bones heading (the red circle) is not aligned with its bone.



After RPR is applied the bones heading (the red circle) is now aligned with the length of its bone.



## Record Bone Rest Position

(default keyboard shortcut **R**)

Once a bone is placed and keyframed you must set its rest position and activate it. Usually, you will use the keyboard shortcut for the Record Bone Rest Position command, which is the **r** key. This does two things: records the rest position and activates the bone. From then on, any changes to the bone will deform the object.



NOTE: When adjusting a chain of bones individually you'll need to keyframe each bone (or use **Auto Key Create** in the **General Options**

**Tab** of the **Preferences Panel**) before moving on to the next bone.

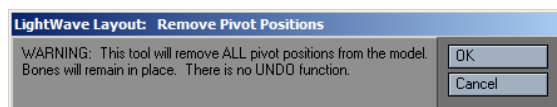
When you use the **Record Bone Rest Position** command, LightWave sets the rest position by copying the bone's current position and rotation values to the **Rest Position** and **Rest Direction** values (**Bones Panel**). This creates a point of reference, so that if you later move the bone (not the object), it can influence the object's shape.

If you need to reset the rest position, you may want to deactivate the bone before reapplying the Record Bone Rest Position command. Alternatively, you could manually edit the Rest Position and Rest Direction values.



NOTE: All of the **Bone Edit** tools (**Joint Mover**, **Tip Move**, **Bone Twist**) will record the bone's rest position automatically.

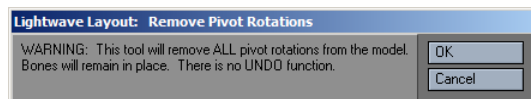
## Remove Pivot Position



RemovePivotPos will remove Record Pivot Position from the entire rig. When invoked, a warning will be displayed. Press OK and RPP will be removed.

## Remove RPR

(**Setup>Modify>Orientation>Remove RPR**)

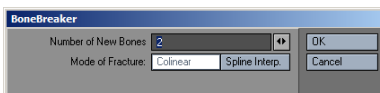


RemoveRPR will remove RPR from the entire rig. When invoked, a warning will be displayed. Press OK and RPR will be removed. The rotational values of the bones that had RPR applied will no longer read 0 degrees.

## Detail

### Bone Split

Allows you to break a single bone into multiple bones.

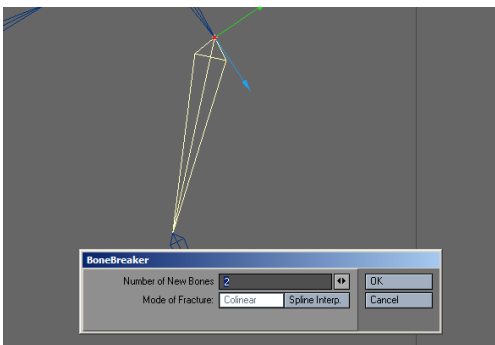


#### Number of New Bones

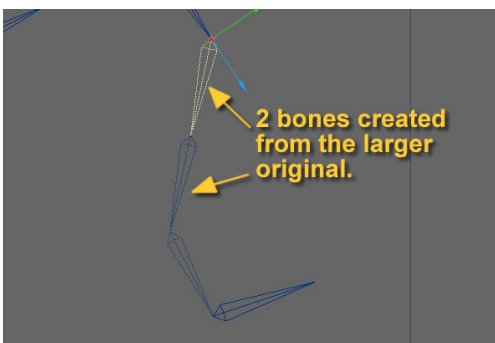
The number of bones that can be created range from 2 to 100.

#### Colinear

The Colinear option will break the original bone into 2 to 100 new bones whose total length will equal that of the original bone. The new bones will be created along the length of the original.



Before split.



After split.

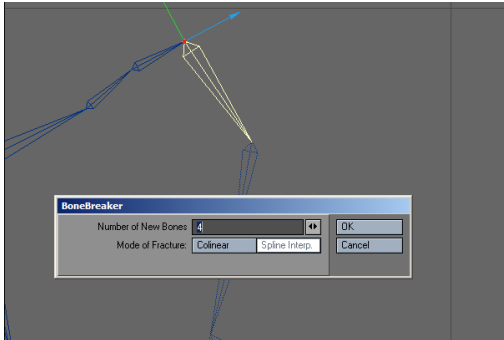
The new bones will retain the original bones' name but the string "\_1", "\_2", "\_3", etc... will be appended to them.



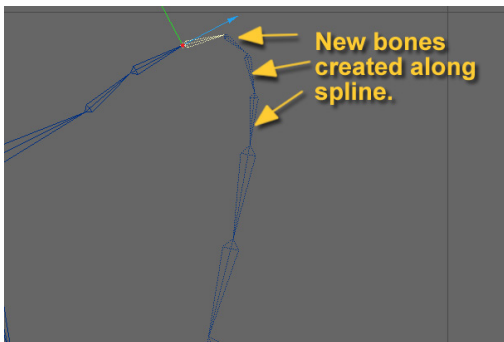


## Spline Interp

The spline interpolation option uses the direction of the parent of the selected bone and the direction of the selected bone, weighted by their size, and creates the new bones along a spline which gets its initial and final slope from the directions of the parent of the selected bone and the selected bone itself, respectively.



Before Spline Interp split.

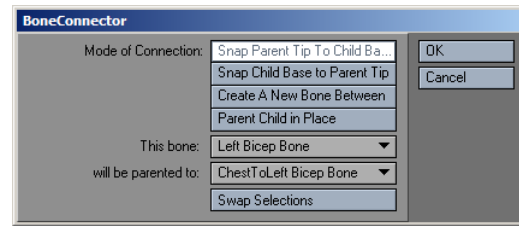


After Spline Interp split.

The new bones will retain the original bone's name but with the string “\_1”, “\_2”, “\_3”, etc... appended for each in series.

## Bone Connect

**Bone Connect** will connect any 2 selected bones, as long as the child to be of the pair is not already parented.



### Mode of Connection

**Snap Parent to Child** keeps the parent's rest position but swings its tip over to connect to the child's base end.

**Snap Child to Parent** keeps the child's tip position the same, but swings its base over to connect to the parent's tip.

**Parent in Place** creates a parent child relationship between the bones, but keeps their position and rotation in global coordinates the same.

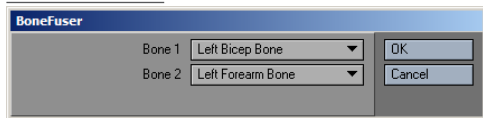
**Create a New Bone Between** connects the selected bones with a new bone between the parent's tip and the child's base.

**This Bone / Will be parented to** displays the selected bones and indicates which bone will be parent and which bone will be child.

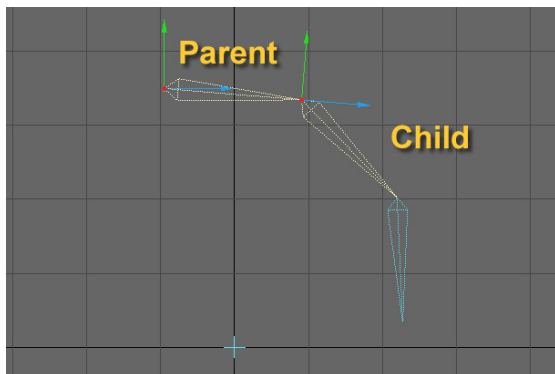
**Swap Selections** allows you to change the selection order without leaving the tool.



## Bone Fuse



Fuses two selected bones, if they have a parent/child relationship.



The newly created bone will have its base located where the parent's base was and its tip located where the child's tip was.



The newly created bone's name will be the original name of the parent.

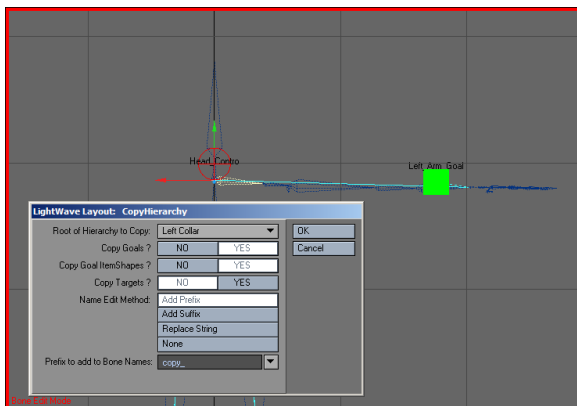
### UnParent Bone (default keyboard shortcut **Ctrl** U)

**(Setup>Detail>UnParent Bone)** The selected bone's relationship with its parent bone (if there is one) is severed, and it is parented to its object. After being "unparented", the selected bone's rest position is edited so that it remains in place, not moving, with respect to global coordinates.

## Edit

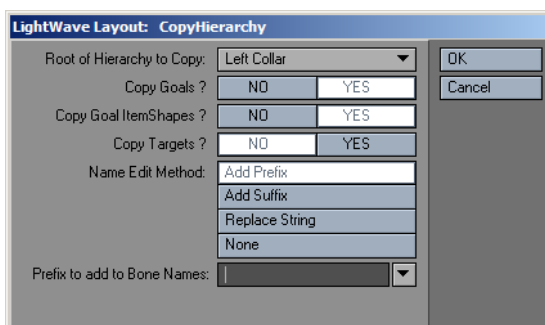
### Copy Hierarchy

When invoked, the selected bone and all of its children are copied along with goals and targets if you choose.



The copied hierarchy will appear next to the selected bone but slightly offset.

The parent of the newly created copy will be the same as the copied bone's parent.



Root of Hierarchy to Copy — Displays the selected bone that is the root of the hierarchy that you want to copy.

Copy Goals, Item Shapes, Targets — Select **YES** if you want any of these copied and **NO** if you only want the bones in the hierarchy copied.



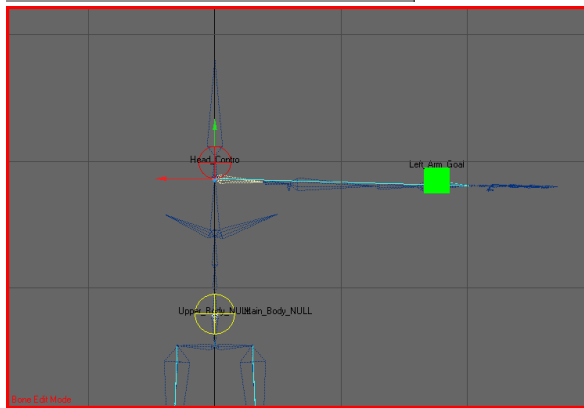
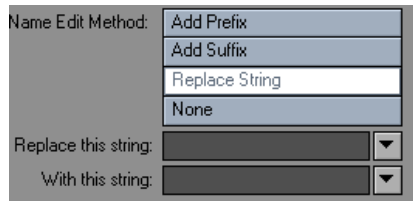
## Name Edit Method

### Add Prefix/Add Suffix

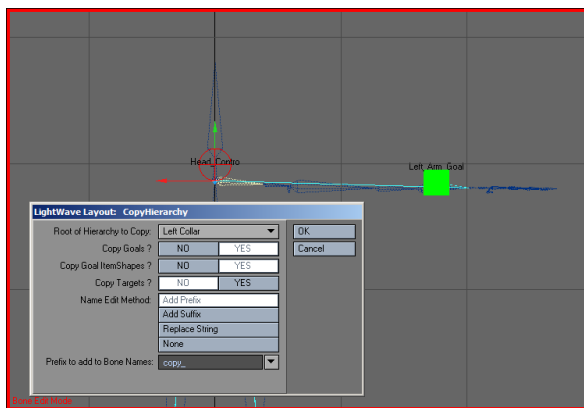
The user interface panel allows you to add a Prefix or Suffix. You can type in your own into the **Prefix/Suffix to add to Bone Names** text box. Or choose one from the **Prefix/Suffix to add to Bone Names** pop up menu.

### Replace String

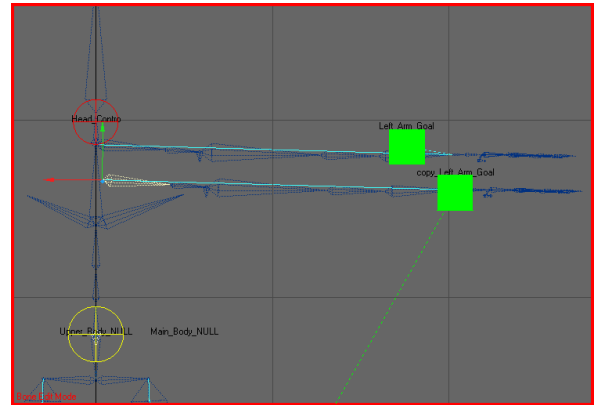
You also can choose to replace an existing text string in the original hierarchy with a new text string. Type the string you wish to replace into the **Replace this String** text box and the string you want to replace it with into the **With this String** text box.



Select the root bone of the hierarchy that you want to copy.



Run **Setup>Edit>Copy Hierarchy** and select copy your options and set the **Name Edit Method**.



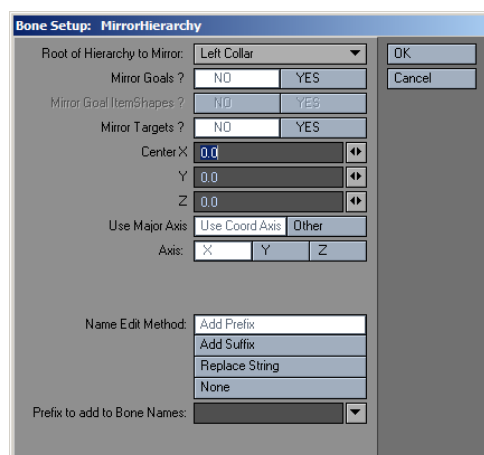
The copied hierarchy will be placed right below the original and parented to the same bone as the original. If you want to parent it to a different bone run UnParent Bone and then use Bone Connect to parent it to a new bone.



## Mirror Hierarchy

(default keyboard shortcut **Ctrl W**)

When invoked, the selected bone and all of its children are mirrored about a plane.



### Root of the Hierarchy

This displays the selected bone whose hierarchy will be mirrored.

### Mirror Goals, Goal Item Shapes, Mirror Targets

Select **YES** if you want any of these mirrored along with the hierarchy and **NO** if you only want the bones in the hierarchy copied.

### Center

Allows you to add an offset from the center.

### Use Major Axis

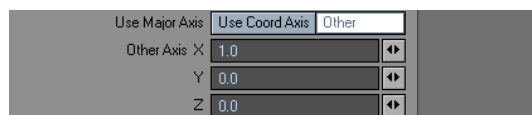
Use Coord Axis — You can choose this option to mirror across the X, Y or Z axis.



### Axis

Choose the Axis you wish to mirror across.

**Other** — This option will allow you to mirror using compound angles. Every unit you add to each axis adds to the angle of the resulting mirror.



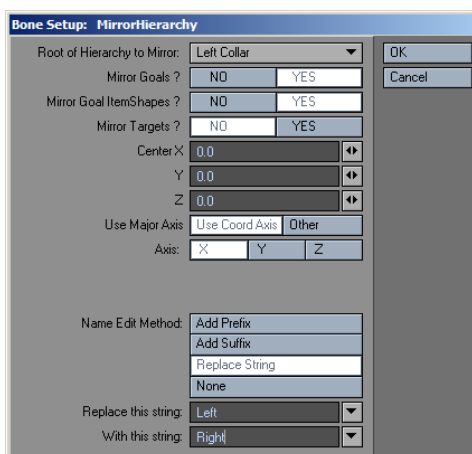
### Name Edit Method

#### Add Prefix/Add Suffix

The user interface panel allows you to add a Prefix or Suffix. You can type in your own into the **Prefix/Suffix to add to Bone Names** text box. Or choose one from the **Add Prefix/Suffix to add to Bone Names** pop up menu.

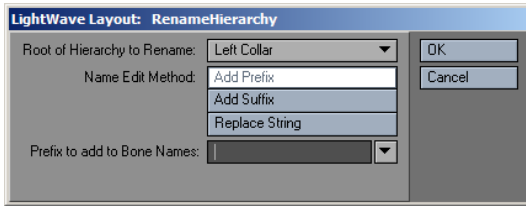
## Replace String

You can also choose to replace an existing text string in the original hierarchy with a new text string. Type the string you wish to replace into the **Replace this String** text box and the string you want to replace it with into the **With this String** text box.

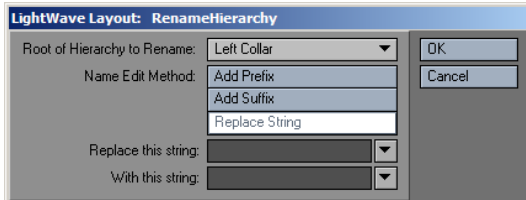




## Rename Hierarchy



When invoked, the selected bone's hierarchy can be renamed. The options are Add Prefix, Add Suffix, or Replace String.



### Name Edit Method

#### Add Prefix/Add Suffix

The user interface panel allows you to add a Prefix or Suffix. You can type in your own into the **Prefix/Suffix to add to Bone Names** text box. Or choose one from the **Prefix/Suffix to add to Bone Names** pop up menu.

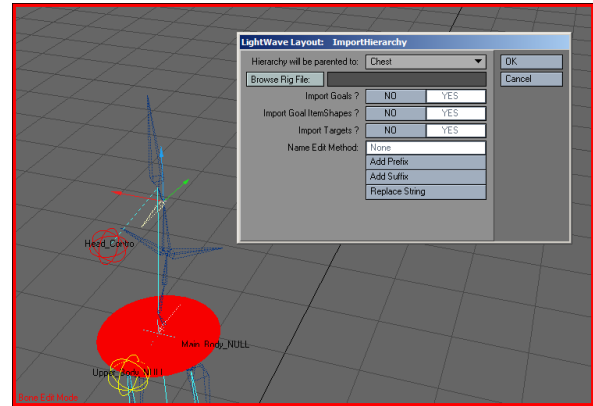
#### Replace String

You also can choose to replace an existing text string in the original hierarchy with a new text string. Type the string you wish to replace into the **Replace this String** text box and the string you want to replace it with into the **With this String** text box.

## Import RIG

(default keyboard shortcut **Shift I**)

(Setup>Edit>Import RIG) Import RIG will read files saved in the **RIG** file format (See the RIG SDK for more information) and load them into LightWave.



### Hierarchy will be parented to:

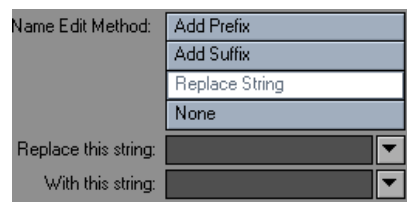
Displays the bone or item that the imported hierarchy will become a child of.

### Browse Rig File:

This will open a file requester so you can navigate to the location of the RIG file that you want to import.

### Import Goals, ItemShapes, Targets

Select **YES** if you want to import goals and targets along with the bones.



### Name Edit Method

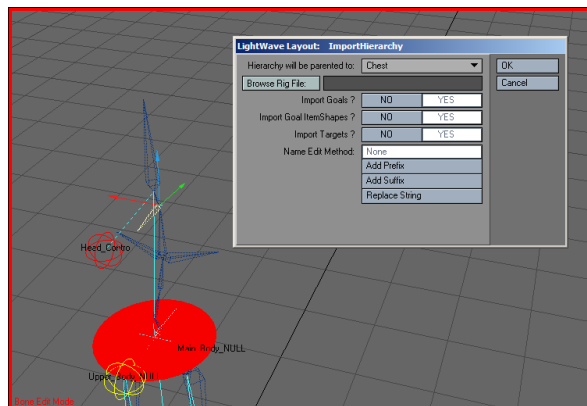
#### Add Prefix/AddSuffix

The user interface panel allows you to add a Prefix or Suffix. You can type in your own into the **Prefix/Suffix to add to Bone Names** text box. Or choose one from the **Prefix/Suffix to add to Bone Names** pop up menu.

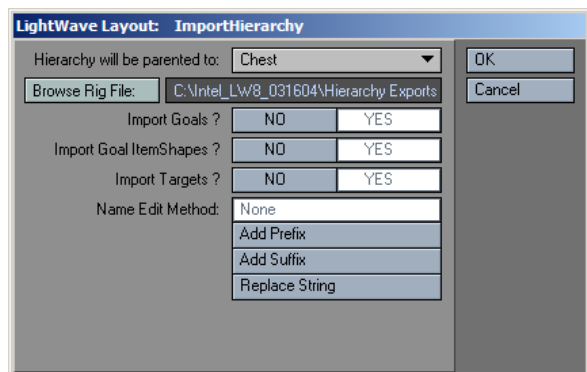


## Replace String

You can also choose to replace an existing text string in the original hierarchy with a new text string. Type the string you wish to replace into the Replace this String text box and the string you want to replace it with into the With this String text box. To use Import RIG to import a hierarchy:



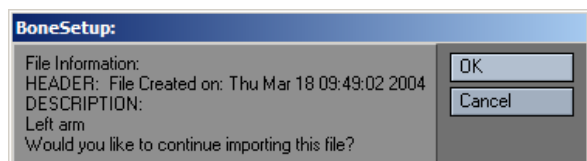
**Step 1:** Select the bone or object you wish to import to be parented to and run **Setup>Edit>Import RIG**.



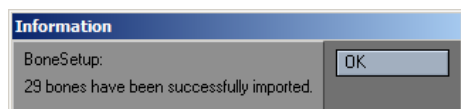
**Step 2:** Browse to the folder that contains the RIG file you want to use.

**Step 3:** Select the import options.

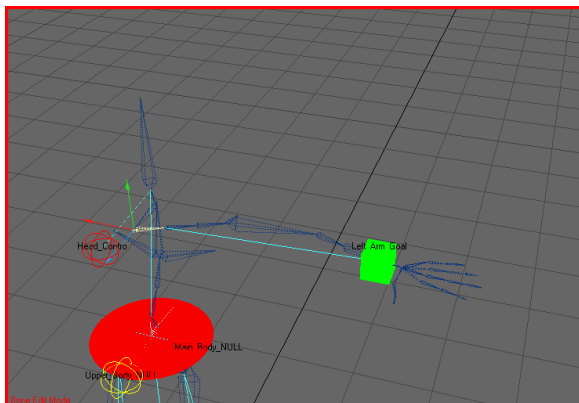
**Step 4:** Click **OK**.



**Step 5:** The RIG information dialog will pop up. Press **OK**.



**Step 6:** Another dialog will appear with bone count information. Click **OK**.



**Step 7:** The arm has been imported along with IK and custom object and parented to the selected bone.

Just enable IK and it will work.





## Export RIG

(default keyboard shortcut **Shift I**)

**(Setup>Edit>Export RIG)** With Export RIG (hierarchy) you can select a parent bone of a hierarchy or an object containing a hierarchy and export it for future use in another mesh. You can export goals, targets and keep the custom item' shapes that may have been added.

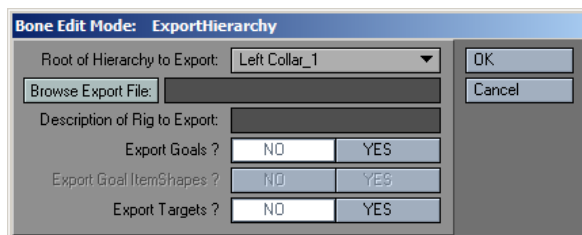
### What's Exported?

Most settings applied to the hierarchy will be exported. IK settings and other properties will go along with the RIG file. See the note below for limitations.



**NOTE:** With this first release there are limitations to what can be exported from the rig.

The motion modifiers that make up many of the rigs used right now are not supported, but will be very soon. A new scheme to allow plugins to communicate and share their settings data will be added. The first to have this done is the ItemShape plugin that allows you to customise the way Nulls and other objects are displayed. This allows the RIG file format to read the data directly from the plugin. As the other plugins add this feature, the RIG file format will be able to export and import ALL items and settings used in rigging a character.



### Root of Hierarchy to Export

Displays the parent or root of the hierarchy to be exported.

### Browse Export File:

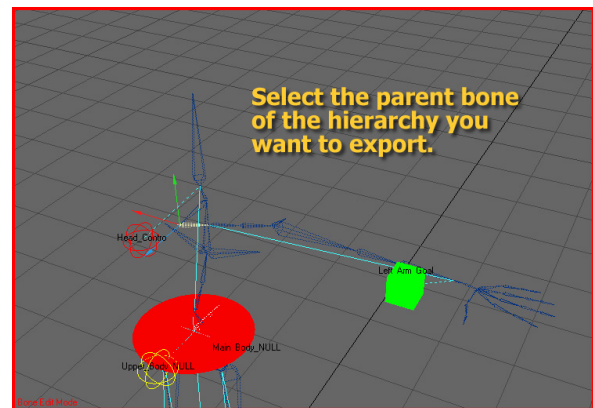
This will open a file requester so you can choose a place to save your export.

### Description of Rig to Export

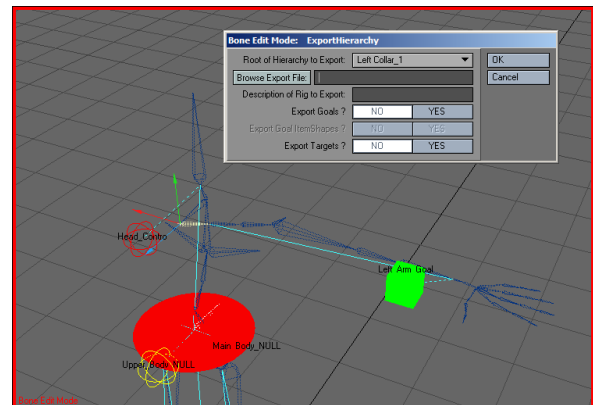
Enter a useful description about the export so you will know what you have when you import this file back into LightWave.

## Export Goals, ItemShapes, Targets.

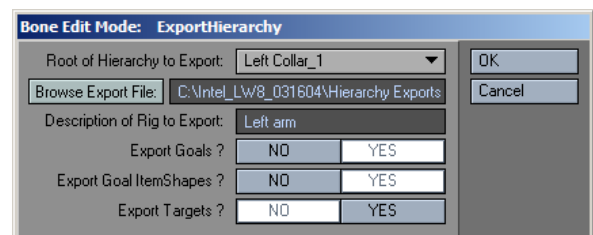
Select YES if you want to export goals and targets along with the bones. To export a hierarchy from a selected bone:



**Step 1:** Select the bone that is the parent of the hierarchy you wish to export.



**Step 2:** Run **Setup>Edit>Export RIG** and the dialog box will appear.



**Step 3:** Browse to the folder you wish to export to and choose the settings.

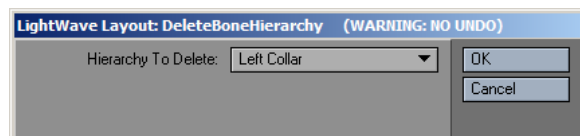
**Step 4:** Click **OK**.



## Delete

### Delete Hierarchy

Delete Hierarchy (**Setup>Edit>Delete>Delete Hierarchy**).



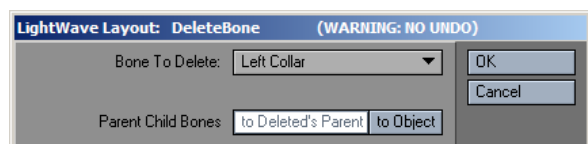
When invoked, the selected bone and all of its children and grandchildren, etc. are deleted, leaving all other bones in place.



NOTE: You may find the need to reparent the children using Bone Connector.

### Delete Bone

Delete Bone will delete the selected bone without causing the remaining bone to move.



#### Bone To Delete

Displays the name of the selected bone.

#### Parent Child Bones

to Deleted's Parent — Will parent the remaining bones (if there are any) to the deleted bone's parent.

to Object — Will parent the remaining bones (if there are any) to the object.

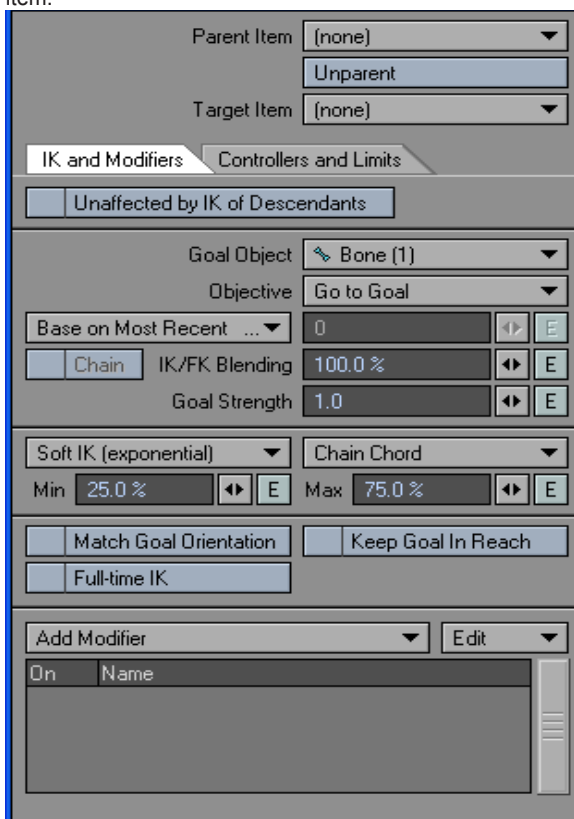


NOTE: This operation cannot be undone.

## Motions Options Panel

### Motion Options Panel *(default keyboard shortcut M)*

When an item in Layout is selected, clicking the Motion Options button opens a panel containing various controls for the selected item.



#### Parent Item

This brings up a pulldown list that allows you to parent the current item to any other item in the list excluding bones, which can only be parented to another bone belonging to the same object or to the object they belong to.

#### Unparent

Unparents the current Item from its current parent.

#### Target Item

This brings up a list of items that will allow the currently selected item to target or point at.

### IK and Modifier IK and Modifiers Tab

#### Unaffected by IK of Descendants

This option is used to define the root of an IK chain. It acts as an anchor for IK based motions and stops the motion from passing back beyond that item.

#### Goal Object

This is an item which is usually the "handle" at the end of an IK chain. This handle serves the same function as a control rod on the end of a muppet's arm or the strings on a puppet.



## Objective Menu

This menu selects how the item uses the goal. The default is “Go to Goal”, which is the familiar way of trying to get the item to be as close to the goal as possible. “Point at Goal” is similar to targeting in that it aims to make the item's Z-axis point at the goal. “YZ Plane through Goal” the IK solver will attempt to get the YZ plane of the effector to go through the goal position.

### Chain

Activates the IK/FK blending options for the entire chain.

### IK/FK Blending

Will blend the calculations for inverse kinematics and the keyframes for forward kinematics based on the percentage, with 100% being full FK.

### Soft IK

Soft IK can be used with the “Go To Goal” objective. Its purpose is to smooth out the transition from a bended chain to a straightened out chain. It does this by changing the position of the goal that the chain uses, moving it closer to the chain root.

The threshold at which it starts to do this is set by the Max value. If the distance between the chain root and the goal is larger than that indicated, the soft IK will kick in smoothly.

There are three ways to specify what “the distance between the chain root and the goal” means:

- \* Chain Chord: the Max value is given as a percentage of the straight-line distance between the chain root and the goal.
- \* Chain Length: the Max value is given as a percentage of the distance between the chain root and the goal, as measured along the chain.
- \* Given Lengths: the Max value is used as the distance.



Additionally, as well as a soft IK on stretching the chain, there is a soft IK on squashing the chain as well. If the distance between root and goal is less than a given minimum value, then it will start to push the goal away.

### Full-time IK

When this option is off, IK is not calculated full-time. This means that animating the goal object will have no effect on the items with IK active. The goal object will affect the IK items while posing. The items that are part of the IK chain will have to be keyframed manually. In this mode the Goal object acts as an aid to pose the IK chain. When this option is turned on, IK is calculated full-time. This allows the goal object to be used to animate with full-time and the items in the IK chain do not need to be keyframed.

## Goal Strength

This option is used to determine how much the end of the IK chain follows or is attracted to the goal object. This is particularly important when using multigoal IK chains. One goal for example could be used to control the placement of a leg/foot and another to determine the orientation of a knee joint. You would want the goal at the end of the leg to have a greater goal strength than the one at the knee. Sometimes this takes a little tuning to get right. With a character that is modelled to human scale, goal strengths from 30 -100 work well. Interestingly enough the smaller the character the larger the goal strength and the bigger the character the smaller the goal strength needed.

### Match Goal Orientation

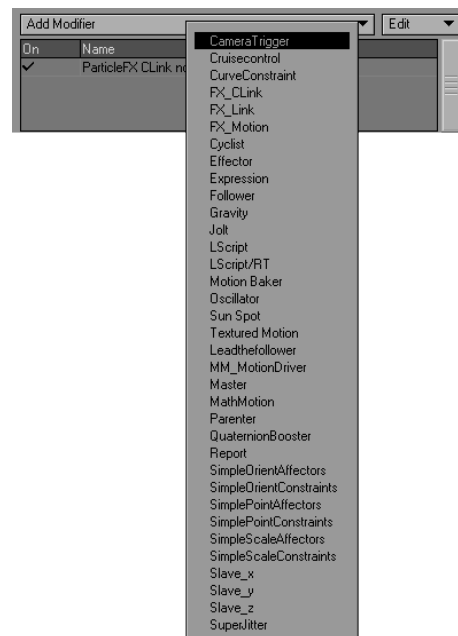
As stated before the Goal object acts as a sort of handle or the control rod on the end of a puppet's arm. By turning on match goal, the last bone in the IK chain will match its orientation to that of the goal objects. This allows things like a foot or hand to stay locked in place rather than pivot around the goal object.

### Keep Goal Within Reach

It is possible for the goal object to move away from its IK chain once it exceeds the length of the chain. This is normal but can be disconcerting to some people. If this option is turned on, the goal object will always stay at the end of the IK chain.

### Add Modifier

This panel allows additional plugins included with LightWave or available from third party sources to be activated. These plugins will allow motions to be controlled or modified in various ways. Next to each motion modifier in the list is a check box. This can be used to temporarily activate or deactivate the modifier next to it.



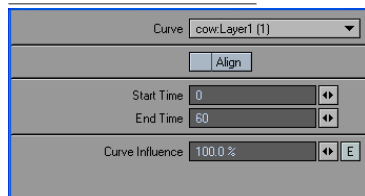


## Camera TriggerCruise Control

Will move item with constant speed. The direction of motion will be the direction at time 0. There must be at least two keys in the object motion, preferably linear.



## CurveConstraint



CurveConstraint moves an item along a curve object, similar to a motion path. The curve object is selected on the **Curve** pop-up menu. (If there is more than one curve in the object, the first curve is used.)

### Align

Will rotate the object so that it faces along the curve in the traditional “Z axis” manner.

### Start Time/ End Time

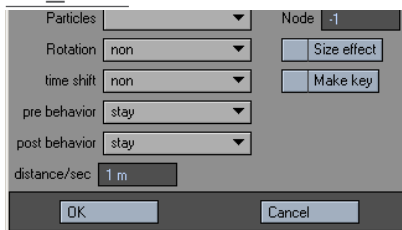
These values determine how long the item takes to traverse the curve.

### Curve Influence

The **Curve Influence** percentage blends the curve motion in or out.

CurveConstraint does not remain locked to the curve, but adds the curve's position to its own keyframed motion. To lock the item to the curve, simply parent the item to the curve, reset the item's position then delete all its keyframes. You might use CurveConstraint to move an object along the same curve used to create the geometry (or create the curve from the geometry), just like you would with a roller coaster.

## FX\_Link



### Particles

Choose particle group.

The Rotation drop-list determines which rotation method the particle will have when emitted. The default value, Non, has no rotation added.

The Random option gives the item a random starting rotation. Align to Path(h) and Align to Path(hp) will rotate the item according to the particle's path.

Time shift moves the start of the item's sizing/rotation motion according to the settings in the drop-list. The Non value doesn't alter the timing of the item's motion, it plays as it was originally keyframed. The Start Shift and End Shift options move the item's motion to the beginning or end of the particle's life. The Start Adjust (distance) and End Adjust (distance) settings will alter the frame rate of the motion according to the value in the Distance/Sec field. As the particle travels a certain distance, a percentage of the motion is animated.

The Pre and Post Behavior settings indicate what the item's rotation/scaling state will be when the motion isn't being animated. This represents the time before the particle is born and after it dies.

The Stay behavior will hold the first frame of the motion for the Pre Behavior and the last frame with the Post Behavior. The Original setting returns the object to its original state in either the beginning or ending of the animation.

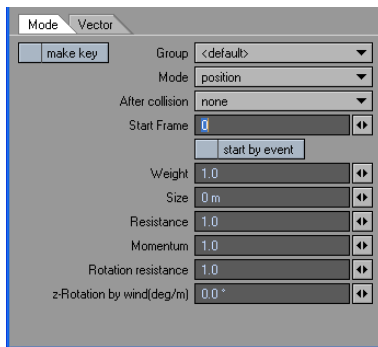
The Size Dissolve setting will dissolve the item in either the beginning or ending of the particles life. So in the Pre Behavior, while the particle is waiting to be generated, it's not visible. Then for the Post Behavior, after the particle dies, it becomes invisible.

Size effect applies the particle size to the object. The object's normal layout size becomes a factor where 0 = 0% and 1 = 100%. You can animate an overall size of particles using normal layout sizing functions.



## FX\_Motion

### Mode Tab



**make key** — Generates a key for every frame.

**Group** — Select an existing group or create a new one.

### Mode

**After collision** — Choose an action post collision: none, reverse, random, stop.

**Start Frame** — Choose what frame to start the effect.

**Start by event** — Use this option if you want to trigger the motion upon collision.

### Weight

This option sets an arbitrary weighting value that will influence how factors like gravity affect the item.

### Size

This option affects the outer boundary used for collision detection. It can also be used by volumetric plugins like **HyperVoxels**.

### Resistance

This option adds an *air resistance* effect. Items will move slower as this value is increased.

### Momentum

This option adds in a momentum or increased mass effect. Items will tend to resist being stopped or slowed down.

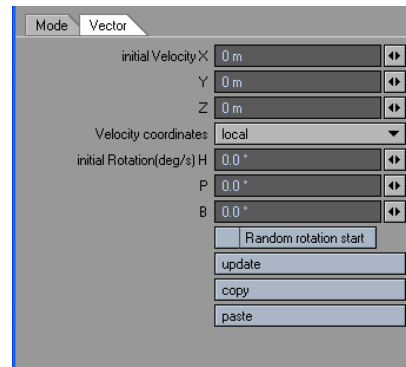
### Rotation Resistance

This option has the same effect as Resistance except that it affects the items' rotational channels.

### z-Rotation by wind(deg/m)

This option rotates the item in its bank channel as it moves or is affected by the wind.

### Vector Tab



### Initial Velocity

This option sets the initial direction and force.

### Velocity coordinates

This option sets whether these settings use the item's local axes or world coordinates.

### Initial Rotation(deg/s)

This option sets the items' initial HPB rotation.

### Random Rotation Start

This option randomises the initial HPB rotation.

### Update

This option updates/refreshes the current settings in the Layout viewport. This is good to do with a heavy scene.

### Copy

This option copies the current settings.

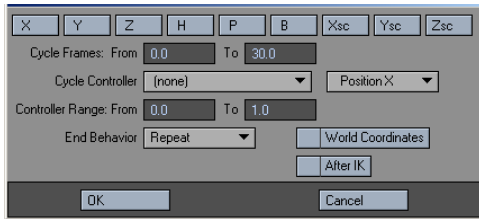
### Paste

This option pastes settings in the copy buffer over the current settings.



## Cyclist

Cyclist will associate the action of one item with a repeatable action (an animation cycle) of another item.



Channels are activated by clicking on the buttons that represent each channel at the top of the **Cyclist Panel**. All activated channels will have their values replaced by the animation-cycle values of that same channel, but at the time determined by the Cycle Controller. Channels that are not enabled will retain their normal value at the particular frame. For example, you can setup keyframes on wheels of a car so they revolve continually, while parented to a car. Then, apply cyclist to the individual wheels, setting the car (or master parent) as the cycle controller. When the car moves, the wheels will turn.

### Cycle Frames

Defines the frame range that contains the action that you want repeated (i.e., the animation cycle).

### Cycle Controller

Controls the item based on the selected control parameter, defined on the pop-up menu just to the right. The control parameters can be an item's position, rotation, scale, speed, or the distance it has traveled along its actual path length. The **Forward Progress**, **Side Slip**, and **Climb** parameters take into account the item's orientation and tell how far the item has been moving forward (Z axis), to the right (X axis), or up (Y axis), respectively. **Speed** looks at the item's velocity.

### Controller Range

Defines how much change is required to equal one full animation cycle. The unit of measure for this parameter depends on the selected control parameter. For **Position** and other **Distance** settings, the unit of measure is in the LightWave **Default Unit** defined on the **General Options Tab** of the **Preferences Panel**. Rotation uses degrees and scale is a factor where 1 equals 100 percent. Speed is in default units per second.

### End Behavior

Determines what happens after the first cycle is completed.

### World Coordinates

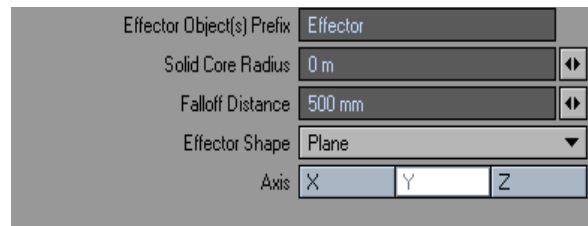
Select this button to use the **Cycle Controller's** actual (world) coordinates, rather than its relative coordinates. These could be different if, for example, the object is parented to another object.

### After IK

Although the motion of the item will always come from keyframes, the motion can be used to move an IK goal, if **After IK** is off.

## Effector

Effector causes *effector objects* to repel or attract the motion path of the affected object. The effector objects may be any objects you wish, but Null objects work best.



### Effector Object(s) Prefix

Use this to set the prefix name. Any object that begins with this name will be an effector, allowing you to have more than one based simply on their names.

### Solid Core Radius

Defines a spherical area, within which all objects are equally affected.

### Falloff Distance

There is a gradual falloff of the effect between the **Solid Core Radius** and **Falloff Distance**.

Objects outside the **Falloff Distance** are not affected at all.

### Effector Shape

There are two choices, **Point** or **Plane**. If you choose **Plane**, you will need to specify the **Axis**.

The **Axis** buttons will be greyed out if you choose **Point**.

The impact of the effector object is set and animated by keyframing its XYZ **Size** channels.

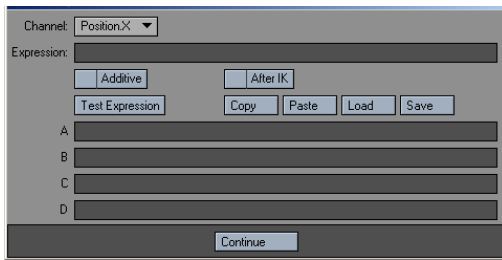
Positive values repel and negative values attract.





## Expression

This is a motion-modifier version of expressions, which can use the results of other motion modifiers, as well as IK. This is not possible with a channel modifier.

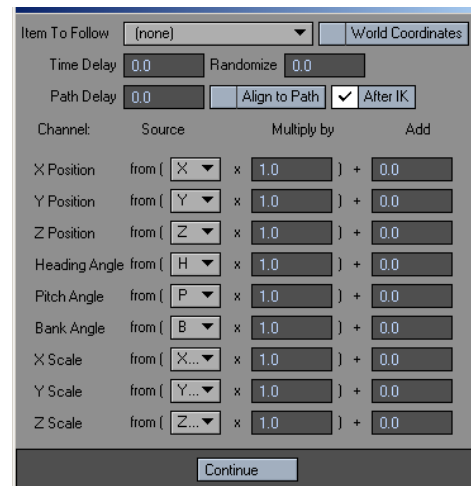


### Channel

You can select multiple channels from the pop-up menu and apply different settings for each (i.e., you don't just pick one). They can all share scratch variables (A through Z) and the expressions can be aware of other components of the item's motion. Thus, H can depend on Z, for example, without trying to reference the item's object.

## Follower

Using the Follower item motion modifier is similar to parenting an object to another, except that you have control over which motion channels you wish to inherit. You can also modify and delay the inherited value. Moreover, the motion can be inherited from the camera, a light, a bone, or any object in the scene. Follower looks at the pivot point of the item to follow, therefore, certain objects that follow another should match pivots.



### Item To Follow

Use the pop-up menu to define the item whose motion you want to use. This item would be considered the leader.

### World Coordinates

Activate this option to use the leader's actual (world) coordinates, rather than its relative coordinates. These could be different if, for example, the object is parented to another object.

### Time Delay

The amount of seconds entered is added to the current time. A negative number can be used if desired.

### Randomize

Is a maximum amount of time (in seconds) to be added to the delay. The actual amount will be between 0 and the **Randomize** value. This number may be negative or even larger than the **Time Delay**. (The latter would cause the item to sometimes anticipate the leader's moves and lag at other times.) The overall delay amount for each item is fixed over the course of the animation, so this is useful for basic flocking effects.



NOTE: The random number seed is taken from the object ID, which should be the same among ScreamerNet nodes and, thus, will cause items to have different delay amounts. The delay stays the same from frame to frame.



## Path Delay

This parameter specifies a fixed following distance along the path of motion of the leader, which is helpful for keeping cars in a train following correctly through accelerations.

## Align to Path

This option will align the follower object to its new path.

## After IK

Allows you to apply Follower taking (or not taking) into account changes from IK.

## Channel

Represents position, rotation and scale.

## Source

Select a **Source** for any of the leader object's motion channels you wish to use. Generally, the type of motion channels will match. That is, the follower object's, say, X Position will be determined from the leader's X Position. However, the settings can be different. For example, the follower object's **Bank Angle** might be derived from the leader's Z **Position**. Select **none** to disregard that channel.

## Multiply By

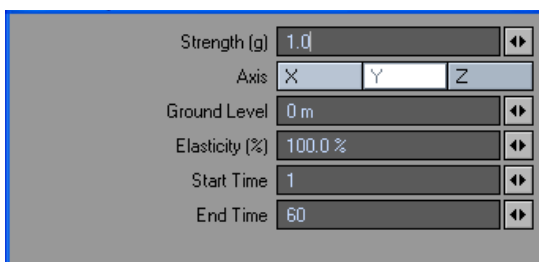
You can scale the value applied to a channel by inputting a factor other than 1 in the corresponding field.

## Add

The channel value may be offset by inputting a value other than 0 in the **Add** field.

## Gravity

Gravity simulates the effects of the Earth's gravity on an item's motion, making it fall naturally and even bounce with varying degrees of elasticity.



### Strength

This is the acceleration due to gravity in units of 'g', the Earth's standard gravity.

### Axis

The object's position will be altered along the specified **Axis**.

### Ground Level

Sets the height at which the object will meet the ground. This is measured in the **Default Unit (General Options Tab of the Preferences Panel)**.

### Elasticity

Describes the amount of energy lost on each bounce, which equates to the relative height of the bounces. Thus, 100% **Elasticity** is completely bouncy, and bounces forever, 0% doesn't bounce at all.

### Start Frame/End Frame

The **Start Frame** and **End Frame** parameters specify when the effect is applied.



NOTE: The most natural results will occur if all the keyframed motion within the range of frames is a single linear path. In particular, the trajectory is determined by an object's initial speed and direction of motion.



NOTE: Your object must start in a positive position along the selected **Axis**.



## LScript

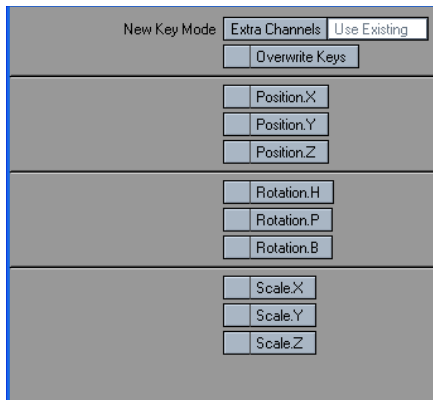
Allows you to add an LScript motion modifier to an item.

## Lscript/RT

Allows you to add a compiled LScript motion modifier to an item.

## Motion Baker

Motion Baker will freeze the motion of an item into individual keys. Motion Baker takes into account not only IK, but also motion/channel modifiers, align to path, and so on. Dragging your frame slider or playing the scene starts the computation.



### New Key Mode

#### Extra Channels

Places the keys in a special AfterIK channel group — you might use these with expressions.

#### Use Existing

Will create normal keys for the selected position or rotation channels.

#### Overwrite Keys

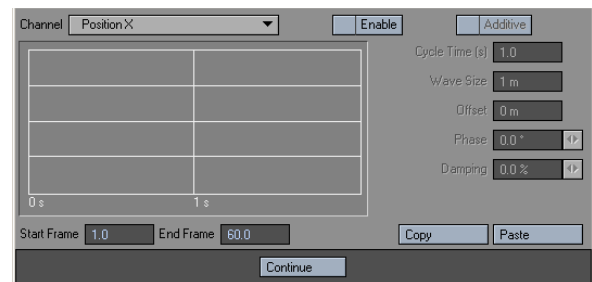
Any existing key data will be overwritten. Note that the existing key data will include the effect of Motion Baker recorded on a previous pass.



NOTE: Motion Baker's position in a list of item motion modifiers does not matter unless another modifier has an **AfterIK** option and it is active.

## Oscillator

Oscillator applies damped harmonic oscillator motions (i.e., decreasing waves) to selected animation channels of an object's motion. Examples of this effect are everywhere, from springs and guitar strings to a grandfather clock pendulum.



### Channel

Use the **Channel** pop-up menu to select an animation channel to be affected.

### Enable

Activate the **Enable** button to turn on the selected channel. The oscillator value will replace the normal channel value. You may independently enable multiple channels and each channel may have its own independent group of settings.

### Additive

**Additive** adds the oscillator value to the channel rather than replacing it. If this setting is active, a plus sign (+) will appear next to the channel name. If **Additive** is off, an asterisk (\*) will appear next to each enabled channel name.

### Cycle Time

**Cycle Time** is the period of the oscillation (i.e., wave), that is, the number of seconds between successive crests.

### Wave Size

Refers to amplitude; the oscillator adds/subtracts this amount at its positive/negative crests.

### Offset

The **Offset** value is also added to the oscillator value on each frame. Essentially, this sets the value of the horizontal axis that runs through the wave. By default it is zero.

### Phase

**Phase** sets where the wave crests with respect to the beginning of the cycle, it ranges from 0 to 360 degrees. Essentially, the wave is shifted horizontally.

### Damping

If **Damping** is applied, the crests will fall or grow over time, as is appropriate for harmonic oscillators. Damping units are a percent per cycle.



## Start Frame/End Frame

The **Start Frame** and **End Frame** parameters specify when the oscillator is applied.

## Copy/Paste

**Copy** and **Paste** use an internal clipboard to move **Oscillator** settings between channels or Layout items.



NOTE: The formula is: channel value = old value + size \* sin( phase + 2\*pi\*time/cycle time ) \* exp(-t\*damping factor). Where the damping factor is a special number computed from the **Damping** percentage.

## Sun Spot

Sun Spot is a motion modifier that will rotate an item, usually a **Distant** light, to match the sun's angle at the specified date and time.

Distance	1 km	↔
Second	0	↔
Minute	0	↔
Hour	8	↔
Day	2	↔
Month	February	▼
Year	2001	↔
Time Lapse	0.0	↔ E
Preset	San Francisco	▼
Longitude (E)	-122.43 °	↔ E
Latitude	37.77 °	↔ E
Time Zone	-8	↔

## Distance

The radial distance from the rotation center, at which the light is pointing.



NOTE: Remember that the position of a Distant light is not that important since the light will always come from the direction it is pointing, even behind the Distant light's position.

## Time Settings

Determine the starting angle. For example, the **Hour** field should be set from 1 to 24 and **Day** is the day of the selected **Month**.

## Time Lapse

When **Time Lapse** is set to 0, there is no sun rotation. A value of 1, will make the sun rotate in real-time, which is *very* slow (i.e., one second of animation equals one second of sun rotation in the real world). The rotation may be imperceptible in short animations. In such cases, you may want to leave it at 0 to minimise any impact on rendering time from moving lights.

You can accelerate the movement by increasing **Time Lapse**. For example, 86400 (60 seconds \* 60 minutes \* 24 hours) will cause one day's rotation to occur in one second.

## Longitude/Latitude

Set the **Longitude** and **Latitude** for the part of the world your sun is (theoretically) shining on.

## Time Zone

**Time Zone** is +/- Greenwich Mean Time (GMT). There are presets provided.

During night hours, the sun will stay at its lowest point at the end of its cycle and then pop to the starting point at the beginning of the next cycle. In other words, it will not revolve in a 360-degree circle.



## TextureMotion

TextureMotion lets you apply the *contour* of texture to a motion. Thus, if you used the same exact texture for a **Displacement Map** (on a subdivided plane), you could automatically have the item move over it following the contour without much effort!



### Offset

Lets you move the motion in the positive or negative direction along the selected **Axis**.

### Scale

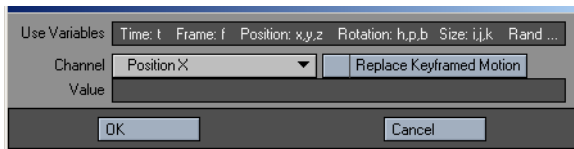
Acts as a multiplier for the motion. A value of 1 has no effect.

### Axis

Determines the perpendicular mapping axis, as would be the case with an **Image Map**.

## MathMotion

This tool allows you to apply simple mathematical formulas to various channels of items to control their motion.



### Use Variables

This field is where variables such as time, frame, rotation, position, etc are selected.

### Channel

This option allows you to choose which channel the variable will have an effect on.

### Replace Keyframed Motion

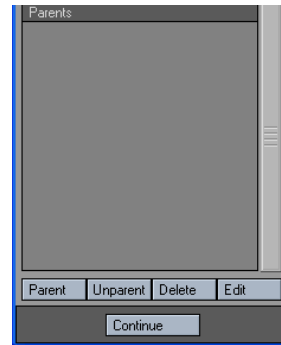
This option lets the formula contained in the Value field to modify or override the items keyframed motions.

### Value

This is the mathematical formula that will create the motions.

## Parenter

Parenter allows you to apply dynamic parenting.

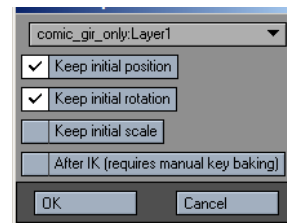


### Parent

Opens the Select new parent dialog.

### Select new parent

This option allows different parents to be selected over time.



### Keep initial position

This option allows the item to keep its initial starting position.

### Keep initial rotation

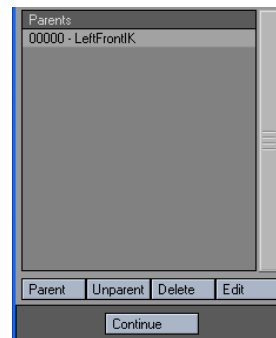
This option allows the item to keep its initial starting orientation.

### Keep initial scale

This option allows the item to keep its initial starting scale or size.

### After IK

This option will calculate the parenting after IK is calculated first. This also means that you will have to manually keyframe the item's position at that position when this option is used.



### Unparent

Clicking Unparent freezes or «bakes» the parented state at the current frame, but onlyuntil the next parenting entry, if any.

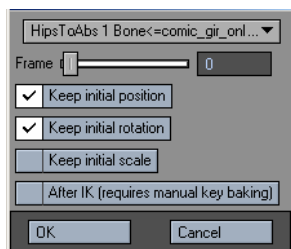


## Delete

Deletes the new parental relationship.

## Edit

Allows you to edit an existing parental relationship.



The only difference between the **Edit parent** and **Select new parent** panels is that a frame slider is added to the **Edit parent** Panel.

## LeadtheFollower

### To use:

**Step 1:** Create a motion path for an object, and then parent one or more other objects to it.

**Step 2:** Run LTF.LS for each of the child objects.

**Step 3:** Use the **Options** button to bring up a requester where you can set the number of frames each object should be in relation to the parent object. (If the parent's motion is set to **Repeat**, then the child object's motions will repeat too.)

## Simple Constraints

Use **Simple Constraints** (SimpleOrientConstraints, SimplePointConstraints, and SimpleScaleConstraints) to achieve “dynamic parenting” effects. You can independently inherit (world) position, rotation, and scale from other items in the scene and even use different items for different channels. These modifiers are “keyframe-aware” so you can have different settings at different times in your animation. You can also use **Simple Constraints** to set up your motion and then “bake” the motion into keys.

To use **SimpleConstraints**, simply open the **Motion Properties Panel** for the item you want to control and add one of the **Simple Constraints** motion modifiers. **SimplePointConstraints** is used for positional effects; **SimpleOrientConstraints** is used for rotational effects; and, **SimpleScaleConstraints** is used for scaling.

### Adding Constraints

The window lists each constraint entry in the order it becomes effective. The number on the left indicates the frame.

To set up a “constraint,” first make sure the Layout current frame is where you want the constraint to start. Click the **Add** button to add a constraint entry to the list. A panel will appear where you adjust the settings for the entry.

There are three main rows of settings. Each row controls the **XYZ position**, **XYZ scale**, or **HPB rotation**, depending on which modifier you are using. Each channel has its own targeted scene item.



**HINT:** The scene item selected on the pop-up menus will default to item currently selected in Layout just prior to clicking **Add**.

The **Weight** value determines how much the channel is “followed.” The default, 100%, will match the targeted channels exactly. You can use less than 100% or even more than 100%. Note that you can use the **Envelope (E)** button to animate the weighting.

SimpleConstraints uses the pivot points for the source and target for its computation, so keep this in mind if the effect isn't turning out like you expect. Also, world space position, rotation, and/or scale are used, overriding the source's parenting and keyframed data.

Select Null constraint if you want to disable and “bake” (see next section) the constraint at the current frame.

You may disable any channel by “clicking off” the **XYZ** or **HPB** button, making it unhighlighted. You can also set the selected item to none.

You can edit an existing entry by selecting it with your mouse and then clicking the **Edit** button.

### Baking

To disable all constraints within a range of frames, but retain the constrained states, you can “bake” the constraint effect into actual keys. However, note that keys are only created where the targeted object has a key, not at every frame. Also, the keys' Incoming Curve will be set to Stepped. This locks the state at each create key.





To bake the constraints, click the **Bake Range** button. In the panel that appears, set the sliders to define the range of frames you want to be baked. If you only want to bake the current frame, click Bake Current Frame to set the sliders to only the current frame. Click OK to execute the process.

After baking, the constraints falling within the baked range will become disabled. Obviously, subsequent adjustments to targeted items will not affect baked motions.

If you click UnBake Selected, the selected constraint will be re-activated. Any keys created from baking will remain; however, remember that constraints will override them.

### Deleting Constraints

To delete a constraint from the list, simply select it with your mouse and click the **Delete** button.

### Simple Constraints vs. Parenter

**Simple Constraints** are slightly faster than **Parenter** and allows you to control motion channels separately. In general, you should use Parenter, unless your item hierarchy is deep enough to cause performance issues.

## Simple Affectors

The Simple Affector modifiers (SimpleOrientAffector, SimplePointAffector, and SimpleScaleAffector) are set up exactly like Simple Constraints, described above. The difference is that the object retains its keyframe motion and is merely influenced by the target.

### Parenter

**Parenter** is also a dynamic-parenting modifier, but is easier to setup compared to **SimpleConstraints**. It also differs in that you can keep the (source) item's initial state and take only into account changes in the parent.

To use **Parenter**, simply open the **Motion Properties Panel** for the item you want to parent to something else and add the Parenter motion modifier.

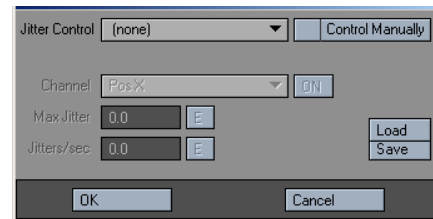
To add a parenting entry, change Layout's current frame to the desired time and click the **Parent** on **Parenter**'s settings panel. A dialog will appear. Select the parent item on the pop-up menu. You may independently select to keep initial position, rotation, scale, and/or parent after taking into consideration effects of IK. Click OK to close the dialog and add the entry to the list. Note that when you are using the After IK option, you will need to manually create a key (in Layout) for the parented item at that frame.

Clicking Unparent freezes or "bakes" the parented state at the current frame, but only until the next parenting entry, if any.

To delete an entry from the list, select it and click the **Delete** button.

You may edit a selected entry by clicking the **Edit** button.

## SuperJitter



### Jitter Control

Select an item to control jitter.

### Control Manually

Selecting this enables the **Channel** setting.

### Channel

Allows you to choose which channel (position, rotation, size) to apply jitter to. You have to click the **ON** button to activate the chosen channel.

### Max Jitter

Clips the amount of jitter that can be applied. The **E** button allows you to set an envelope.

### Jitters/Sec

Sets the frequency of the effect. The **E** button allows you to set an envelope.

### Edit

This allows you to check the properties of a motion modifier plugin or access its control panel.

You can also copy, paste and remove motion modifiers. This makes it easy to set up the attributes on one object and then copy/paste to other objects.



## Controllers and Limits Tab

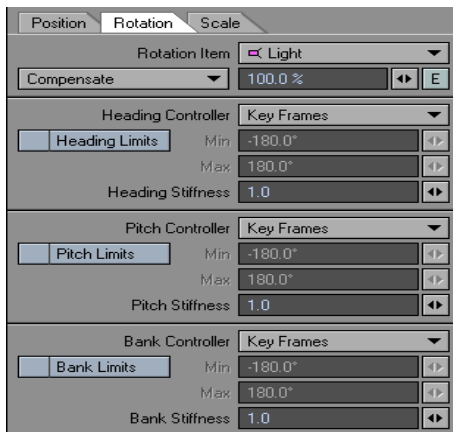
The Controllers and Limits tab has been updated to include Position and Scale controls. The limits are based on the parent space and not world coordinates.

### Heading, Pitch and Bank Limits

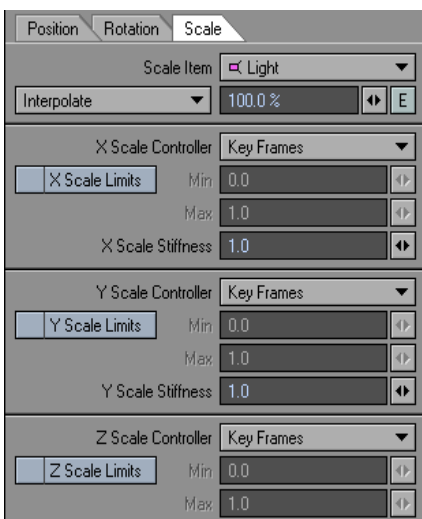
We can determine a range within which a rotational channel of an item can rotate. This is useful to keep joints from hyperextending or keeping IK controlled joints from “popping”. It is also useful to keep items that are very closely spaced from rotating in and through each other.

### Heading, Pitch and Bank Stiffness

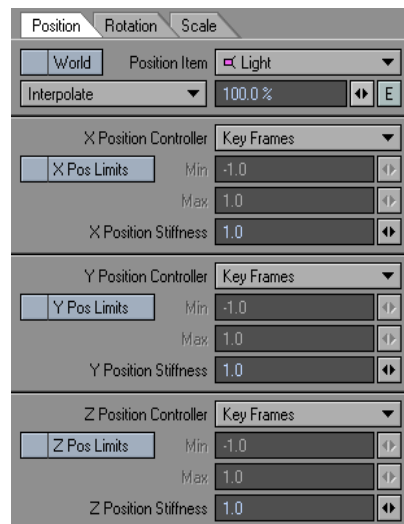
This setting allows you to set a stiffness or resistance on the rotational channels of an item. This works when the controller for a channel is set to IK. The **Stiffness** setting can help to determine which items in an IK chain will bend first or last. Higher values make a joint harder to bend and lower values easier to bend.



### Heading, Pitch and Bank Controllers



### Scale Controllers



### Controllers

These contain seven options for controlling the rotation, position and rotation channels. Not all options are available for all channels.

1. **Keyframe** — This simply means that you can manually orient an item in that channel and keyframe its position at anytime.
2. **Point at Target** — This allows a particular rotational channel to target an item. Target Item in the main motion panel must be set to some item.
3. **Align to Path** — This allows the rotational channel of an item to align to its motion path. With the Align to Path Look Ahead controller under the **IK and Modifiers Tab** we can make this channel look ahead or anticipate a turn.
4. **Inverse Kinematics** — This allows IK to control this channel. If other channels are set to FK we can selectively determine which rotational channels are controlled via IK or FK.
5. **Align to Velocity** — The channels controlled by this option will have the same values as the corresponding channels of the set item. The stiffness value can be used to interpolate. The one for position has the option to use the world coordinates of the position item (instead of local). The calculations currently assume that X, Y, and Z are controlled, so the result may not be optimal if not all three position channels are “Same as Item” controlled.
6. **Align to Pole** — This option adjusts the angle of the item so it points towards the pole. It is similar to the target item, and typically used in combination with it.
7. **Same as Item** — The channels controlled by this option will have the same values as the corresponding channels of the set item. The stiffness value can be used to interpolate. The one for position has the option to use the world coordinates of the position item (instead of local). The calculations currently assume that X, Y, and Z are controlled, so the result may not be optimal if not all three position channels are “Same as Item” controlled.

### Limits

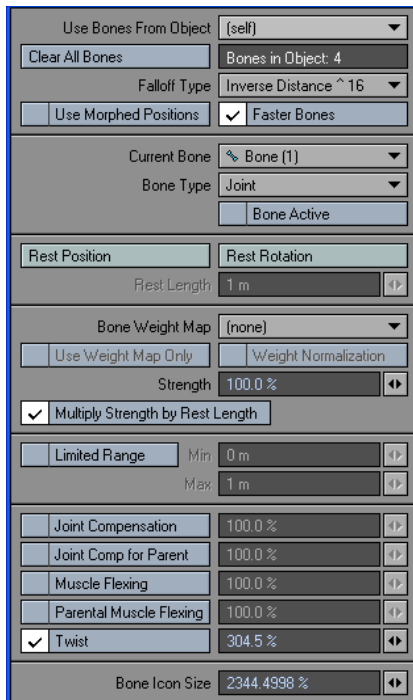
#### Record Minimum/Maximum Joint Angles

You can interactively set the rotational **Min** and **Max** values by rotating the item to the desired minimum direction and choosing **Setup > Motions > Limits > Record Minimum Joint Angles**. This will activate minimum limits for any rotational parameters (heading, pitch, or bank) that are currently active for the chosen item. Similarly, rotating the item to the desired maximum range limit and choosing **Setup > Motions > Limits > Record Maximum Joint Angles** will automatically input the values into the maximum limits for the active rotational parameters. You can deactivate **Heading**, **Pitch**, and/or **Bank** to avoid setting limits for the deactivated setting.



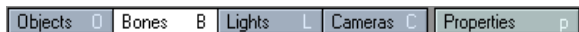
## Bone Properties Panel

The **Bone property panel** gives you access to all the settings available for bones in LightWave Layout.



Bone properties can be accessed in two ways:

1. Select the **Bones** button at the bottom left corner of Layout and then click the **Properties** button located in the same corner.



2. Press **Shift B** and then **P**.

## Use Bones From Object

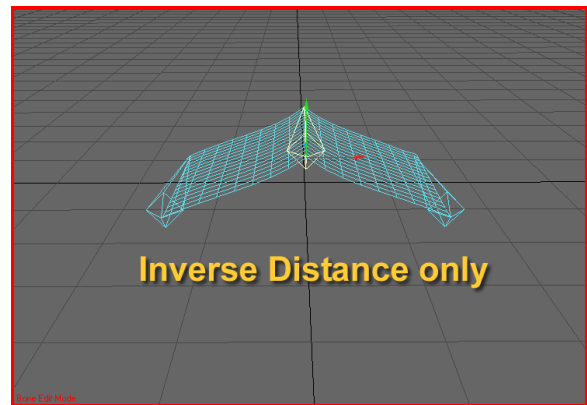
This pop up menu allows you to choose a different object whose bone properties you would like to edit. When you first open the **Bone Properties Panel**, you only have access to the bones of the object that you had originally selected. This allows you to change to a different object without having to leave the **Properties Panel**.

## Clear All bones

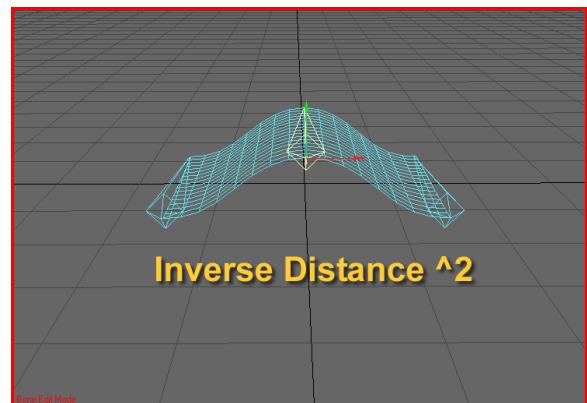
As the name suggests, this button will clear all bones from the current object.

## Falloff type

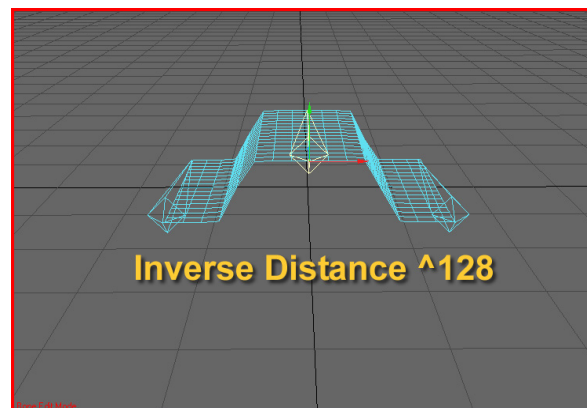
Falloff Type uses a mathematical formula to set the falloff strength of a bone's influence in the current object. An exponent is used ( $^2$ ,  $^4$ ,  $^8$ ,  $^{16}$ ) to figure out how quickly the falloff will happen. The larger the number the quicker the influence of the bones will falloff.



*Inverse Distance*



*Inverse Distance ^2*



*Inverse Distance ^128*

## Use Morphed Positions

With this enabled, bone deformations will be applied after morphs. This option can adversely affect performance; normally it should be turned off.

## Faster Bones

Each vertex will only be affected by the four closest bones in the object. This can increase performance in objects with more than four bones.

## Current bone

Shows the currently selected bone.



## Bone Active

When checked bone is on (active).

## Rest Position

This determines the position of a bone at the time it is rested. When a bone is rested in layout it has an effect on every vertex in the object. How much effect it has is called weighting (not to be confused with **Weight Maps**). How much a point is weighted towards a specific bone depends on several factors. The most important is distance from the bone and bone falloff.

## Rest Rotation

This determines the orientation of a bone at the time it was rested.

## Rest Length

This determines the length of a bone at the time it was rested.

## Bone Weight Map

Use this pop up menu to select a **Weight Map** to apply to the current bone. The influence a bone has on a point is now modified with the strength of the various **Weight Maps** on a given point.

## Use Weight Map Only

Only uses values attained from an applied **Weight Map** to set bone influence. Since each vertex can have multiple **Weight Maps**, various bones can have a distinct influence on a vertex based on the strength of each **Weight Map** it is assigned to.

Note that if a point has only one **Weight Map**, the bone assigned to that weight will have 100% influence on that point no matter what the value of the **Weight Map** is.

In order for a bone to have partial control over a point, that point must have at least two **Weight Maps** assigned to it. If the total value of the combined **Weight Maps** exceeds 100%, unpredictable results can occur. The same goes when using negative weights. To insure predictable results in this mode, it is best to also check **Weight Map Normalization**.

Another thing to know is that in this mode all other bone functions cease to apply.

## Weight Normalization

Use this option to obtain predictable results when using **Weight Map Only**.

This feature ensures that the total values of several **Weight Maps** applied to a vertex never exceed 100%. To calculate the amount of influence a bone will have on a particular point we have to know how many **Weight Maps** are assigned to bones and what their values are. For example, to calculate the amount of influence a bone will have on a vertex we have to know both the total number of **Weight Maps** on that point, and also their values. Assuming each **Weight Map** is assigned to a bone, we will conclude that the amount of influence a bone has on that point is derived as follows:

Point A has three weight maps assigned to it named Upp\_Arm, Low\_Arm and Elbow.

What is the amount of influence the Elbow bone will have on Point A ?

Point A = Elbow/(Upp\_Arm + Elbow + Low\_Arm)

## Strength

Whenever two or more bones are present, **Strength (Bones Panel)** will determine the influence one bone has over another when ranges of influence intersect. If the option to limit the bones range is inactive, all points within the object are considered completely within the influence range of each bone.

## Multiply Strength by Rest Length

The **Multiply Strength by Rest Length** option (**Bones Panel**) causes a bone to multiply its **Strength** by its **Rest Length** to determine the influence of the bone. Bones with larger rest lengths will exert greater influence over other bones with equal **Strength** values, but smaller rest lengths.

## Limited Range

When **Limited Range (Bones Panel)** is active, the **Min** value determines the sharp cutoff point of a bone's influence. Any points within this range are 100 percent affected by movement or sizing of the bone. Any points outside of this range are affected to a smaller extent or not at all depending on whether they fall within the **Max**. In an orthogonal view, the **Max** setting is visible in the viewport.

A short bone will exert a spherical influence, while longer bones exert a more oblong influence.



NOTE: The following options only operate in the pitch channel.

## Joint Compensation

Sometimes bones used to create *joints* can cause pinching or unwanted bunching, much like a folded garden hose. **Joint Compensation (Bones Panel)** will remedy this effect by making the affected points tend to maintain the original volume inside the joint. The result is a more realistic-looking joint. You can modify the compensation amount to increase or decrease the effect.

## Joint Comp for Parent

This option shears the points affected by the parent bone during the child bone's rotation. This compensation also alleviates the pinching effect often caused by bones. At the same time the point shear occurs, the bone will also try to maintain the volume inside the joint.

## Muscle Flexing

With this option the points affected by the bone will flare out as the bone is rotated. The effect simulates muscles bulging under the skin.

## Parental Muscle Flexing

This option is generally more commonly used than Muscle Flexing. For example, when your forearm rotates your biceps flex. The forearm is the child and the bicep is the parent. It would not look natural for the forearm to bulge as it is rotated, but it looks perfectly natural when the biceps flex.

## Twist

The rotation axis most aligned with the bone at rest is used. For zero-length bones the bank axis is used. The amount of twist is the change in rotation of that axis between rest and current (limitation: local rotation values are used, so a twist of a bone is not inherited by child bones). There is a Twist toggle in the bone panel (default on), and an input which controls how curly it is (0% is no curl, 100% is curl by the amount of twist over the length of the bone).

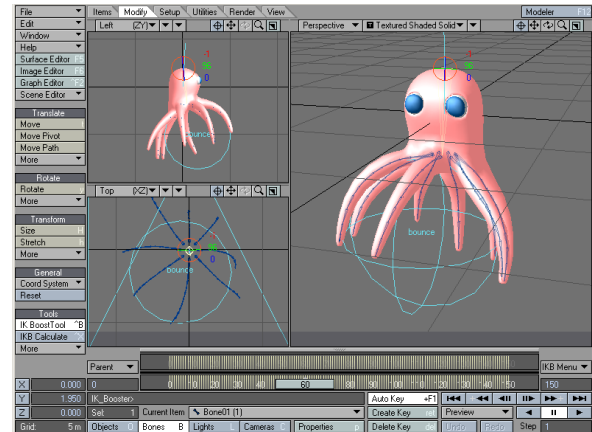
**Bone Icon Size** changes the size of the bone icon in the OpenGL display. It does not affect how the bone influences the mesh or other bones.



## IK Boost Tool

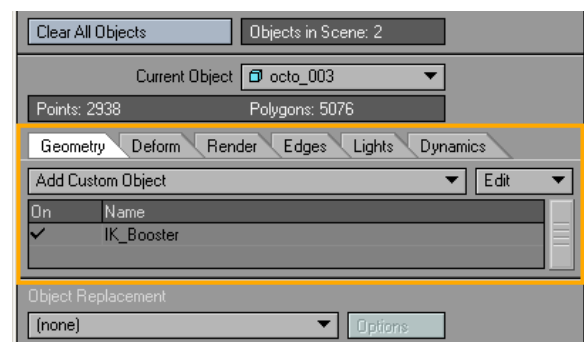
(default keyboard shortcut **Ctrl B**)

**IK Booster (Modify>IK Boost Tool)** is an animation system that can work independently from LightWave's base Inverse Kinematics or it can be used with LightWave's base IK. With **IK Booster** applied to an object you can gain access to **Bone Dynamics** and **IK Booster** character tools.



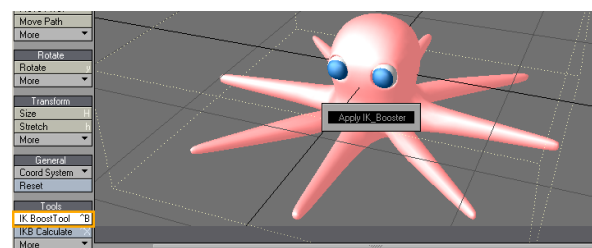
## Applying IK Booster

IK Booster can be applied to an object by adding it to the Custom Object list (**Add Custom Object**) in the **Object Properties Panel**.



You can also add IK Booster to an object using the **IK Boost tool (Modify> IK Boost Tool)**.

Select the **IK Boost** tool and **Right-Click** on the object pivot point to bring up the option to **Apply IK Booster**.



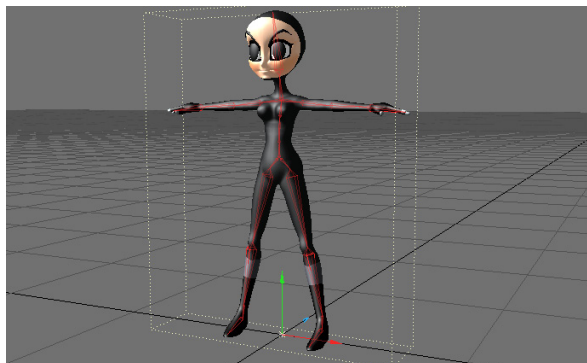
When IK Booster has been applied to an object all the bones in the object and all of the object's children will be set up with IK. Although IK Booster is ideal for character rigs, it can be a powerful tool for mechanical rigs as well.





## Steps for applying IK Booster:

**Step 1:** Load your object (s) into Layout and either create bones, or convert **Skelegons** to bones.  
In the case of a segmented object, make sure all parts are parented properly.

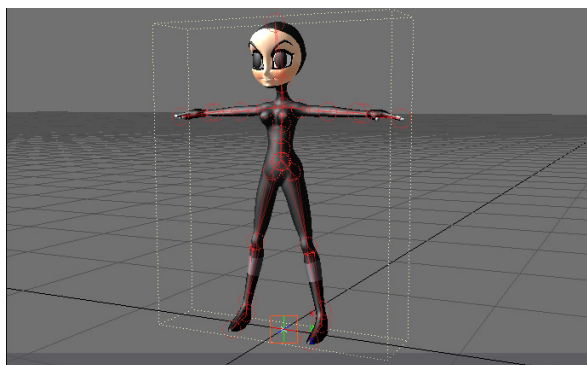


**Step 2:** Select the **IK Boost** tool (**Modify> IK Boost Tool**). Select the **IK Boost** tool and **Right-Click** on the object pivot point to bring up the option to **Apply IK Booster**. Select **Apply IK Booster** to activate **IK Booster**.



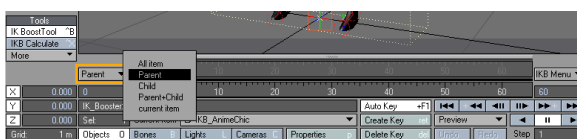
**NOTE:** IK Booster can also be applied in the **Object Properties Panel** as discussed above.

**Step 3:** Although there are some cases where Step 2 would be the last step before animating, in most cases you will want to apply further settings discussed below.

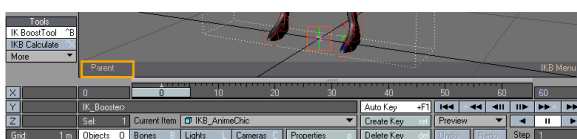


## KeyFrame Mode

IK Booster has several keyframe modes you can choose from the **Keyframe Mode** drop down menu. If the **DopeTrack** is active, the **Keyframe Mode** button will be located on the left side of the track.



If the **DopeTrack** is not active, the **Keyframe Mode** selection will be located on the left side of the IK Booster track. The **IK Booster Track** is located at the bottom of the viewport windows. The Keyframe Mode selector on the IK Booster track doesn't bring up a drop down menu, instead it toggles between the modes as you click on it with the **LMB**.



## KeyFrame Modes

**All Item** — When this mode is selected, if you create a key on an item, it creates a key for that item and the entire hierarchy.

**Parent** — With Parent mode, if you create a key on an item, it creates a key for that item and every parent up that item's chain.

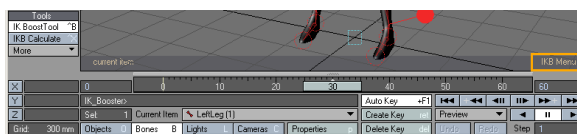
**Child** — In Child mode, if you create a key on an item, it creates a key on that item and all of its children.

**Parent + Child** — In this mode, if you create a key on an item, it creates a key for that item and every parent up that item's chain, and every child down the chain.

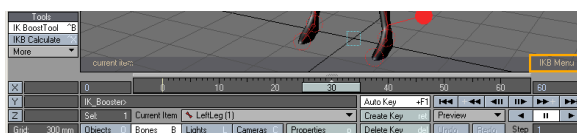
**Current Item** — With Current Item mode, if you create a key on an item, it only creates the key on that item.

## IKB Menu

The IK Booster menu has several options to work with. If the **DopeTrack** is active, the IKB Menu will be located to the right of the **DopeTrack**.



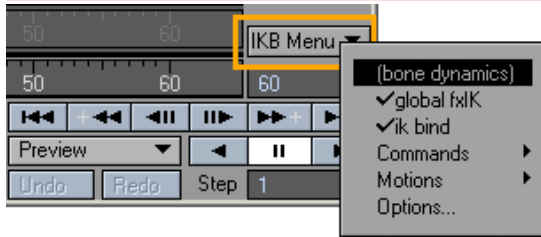
If the **DopeTrack** is not active, the IKB Menu will be located on the right side of the IK Booster track. The **IK Booster Track** is located at the bottom of the viewport windows.







## IKB Menu Options



### Bone Dynamics

This option enables and disables all bone dynamic functions for the object that IK Booster has been applied to.

### Global fxIK

IK Booster provides a global IK on/off switch. When this option is active all hierarchies in the object will be solved with IK. When this option is not active only the current hierarchy will be solved with IK.

### IK bind

IK Booster has the option to “fix” an item, which locks the position of the item’s controller. When IK Bind is active all fixed items will stay fixed. When it is not active, items that are fixed will ignore their “fix” setting and will no longer be locked to their position.

### Commands

Reset Rotation — This command will set the rotational values of the item(s) back to their original value.



NOTE: The KeyFrame Mode selected will determine the items affected by Reset Rotation.

re-BakeSpot-all — Any Bake Spot in the scene will be re-baked when this command is selected.



NOTE: Any Bake Spot that was not previously baked will not be affected.

### Motions

IK Booster has several motion options to help speed up animating and increase productivity.

Motion Copy — This function will copy the object’s entire motion to the clipboard.

Motion Rollback — This function will paste the last motion that was copied to the clipboard.



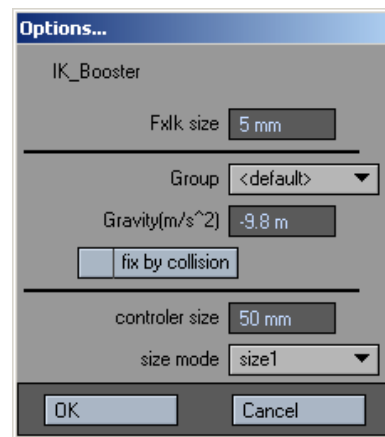
NOTE: This will remove any work that you have done since the time of copying the motion.

Motion Save — This function will save the entire motion data of the IK Booster object to a text file. This gives you the ability to re-use motions from your object on objects that share the same setup.

Motion Load — This function will load an IK Booster motion data file that was previously saved using the Motion Save command.

### Options

This brings up the **IK Booster Options Panel**.



FxIk size — This setting determines the accuracy of the IK solving. The higher the number the less accurate the IK solving will be.

Group — you can group Dynamic objects with a user-defined name to prevent unwanted interaction. This also works with Particle FX controllers. This becomes very handy when you are working on a complex scene and you want certain Wind emitters to only affect certain objects.

**<default>** Includes all groups.

**<new group>** Creates individual group. Can associate functions within a group.



NOTE: You can assign IKBoost settings to a group shared with Particle FX.

Gravity — Setting for downward (-y) gravity.



NOTE: This option only applies to objects that have Bone Dynamics active.



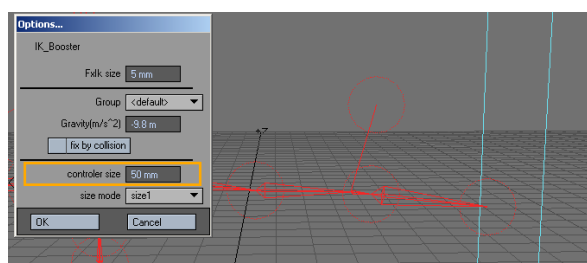
HINT:  $-9.8$  in the Y is considered the value of earth gravity. If you are simulating any terrestrial effect, make sure you use this value..

Fix by Collision — This function will fix items that pass through a collision object.

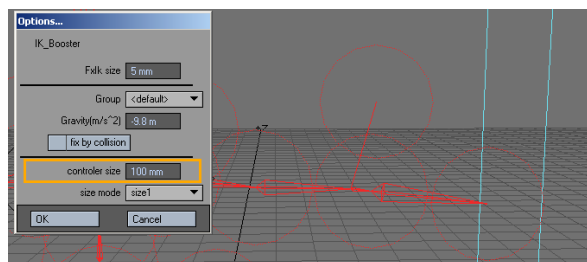


NOTE: This option only applies to objects that have Bone Dynamics active.

Controller Size — Use this setting to change the size of the controller when it is unselected.



Controller Size: 50mm



Controller Size: 100mm

Size Mode — This setting determines the size of the controller when it is selected.

Size 1 — Smallest selected size.

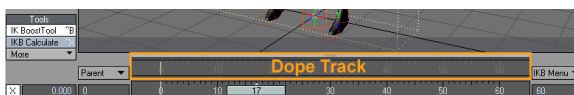
Size 2 — Medium selected size.

Size 3 — Largest selected size.

Not Change — The selected controller size will be based on **Controller Size**. The controller size when selected and unselected will be the same.

## IK Booster and the Dope Track

When the **DopeTrack** is active while in IK Booster mode, there are two different menus that can be used, the Right Click menu, and the Right Click and drag menu. You will also see the **Keyframe Mode** and **IKB Menu** buttons that are unique to **IK Booster** discussed earlier.



## Right Click Menu

Right clicking on the dope track will bring up a menu with several functions described below.



Copy key from current — This function will copy a keyframe from the current frame (the frame the time slider is at), and paste the keyframe on the frame you right clicked on.

Make Key — This function will create a keyframe at the frame you have right clicked on.



NOTE: If you would like to create keyframes for a range of frames, use the Bake Keys function in the Right Click and Drag Menu.

Delete Key — This function will delete the Keyframe at the frame you have right clicked on.



NOTE: If you would like to delete a range of frames, use the Delete Keys function in the Right Click and Drag Menu.

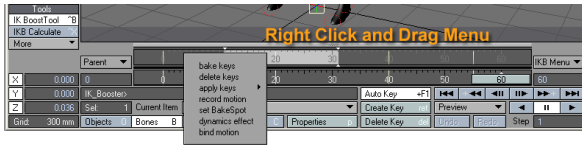
Save Pose — This function will save an external file that contains the keyframe data of the IK Booster object for the frame you right clicked on.

Load Pose — This function will load a previously saved Pose file on the frame you right clicked on.

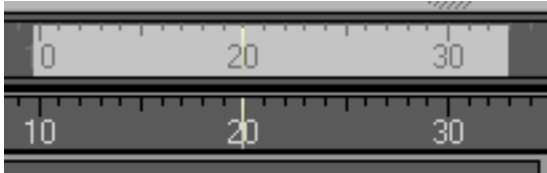


## Right Click and Drag Menu

Right clicking and dragging to select a desired time range on the Dope Track will bring up a menu with several functions described below.



**Bake Keys** — This function will create keyframes for the selected time range.

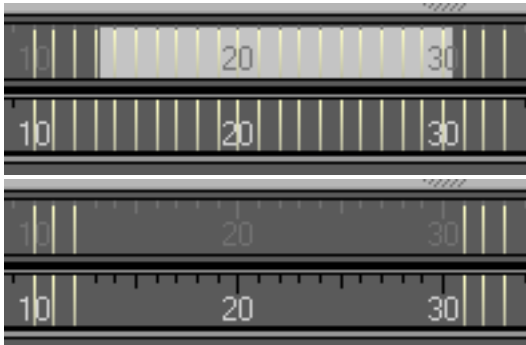


*Time Range Selected*



*After Bake Keys*

**Delete Keys** — This function will delete keyframes in the selected time range.

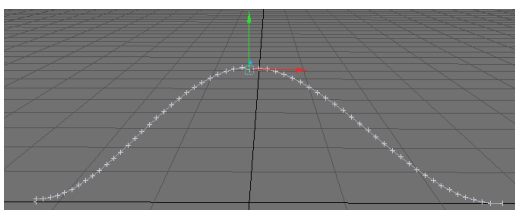


*Above: Time Range Selected, Below: After Delete Keys*

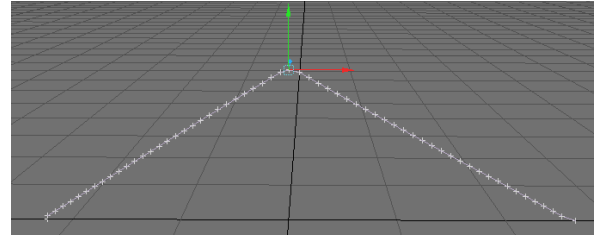
## Apply Keys

All three “applies” use the first and last keyframe in a range as boundaries, and the keyframe that the slider is currently sitting on (Current Frame) as a control point.

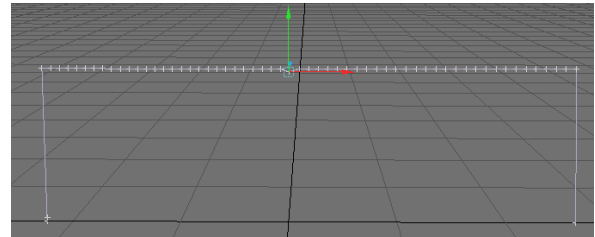
**Soft** — In the case of “Soft Apply”, a Hermite spline is generated using these keyframes as points to smooth out the intervening keyframes. Default Hermite coefficients are used.



**Linear** — “Linear Apply” uses the same keyframes, but simply “increments” the keyframe values between the control points.



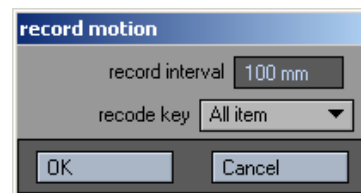
**Flat** — “Flat Apply” snaps all intervening keyframes to the value of the control keyframe (the one the slider resides on).



## Record Motion

This function will record the motion of the selected controller at fixed distance intervals.

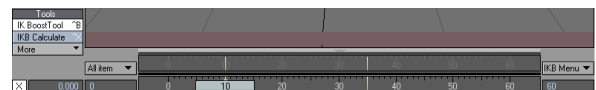
Right Click and drag to select the time range, select Record Motion, and set the options in the **Record Motion Panel**.



**Record Interval** — The set distance for each keyframe.

**Record Key** — Determines what items motion will be recorded using the various Keyframe Modes.

After you set the options for Record Motion, the IK Booster track will display red to show that you are recording. After you have moved the controller through the time range the track will return to the default mode.

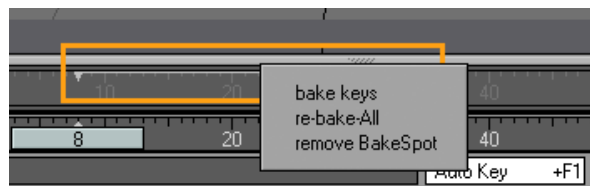


**Set Bakespot** — This function will create a Bakespot in the select time range. A Bakespot is a time range in which you can bake keys into the desired time range so that they will remain for future operations.





Bakespots have a right click drop down menu that gives you the following options:



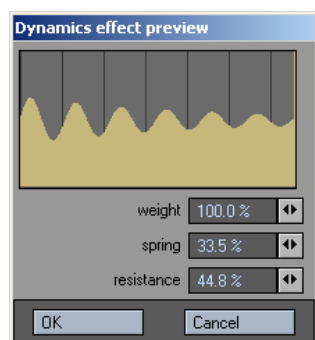
**Bake Keys** — This function will create keyframes for the selected bakespot.

**Re-bake-All** — Any Bake Spot that has been baked in the scene will be re-baked when this command is selected.

**Remove Bakespot** — Clears the bakespot from the track.

## Dynamics Effect

This function will apply dynamic motion to a controller over the selected time range. When this function is selected the **Dynamic Effect Preview Panel** will appear.



The graph in the preview, displays the motion that will be applied to the controller. The line in the background represents one second intervals. This graph gives you a visual representation of the motion that will be applied.

**Weight** — Defines the weight of the controller. You can achieve heavy motions by increasing the Weight value, and you can create light motions by decreasing Weight.

**Spring** — Controls the *springiness* or stiffness of the controller. Reducing the **Spring** coefficient creates soft motions, while raising the Spring coefficient produces motions with a stronger repelling force. Setting the coefficient to an extremely large value creates stiff motions.

**Resistance** — Controls the amount of air resistance.

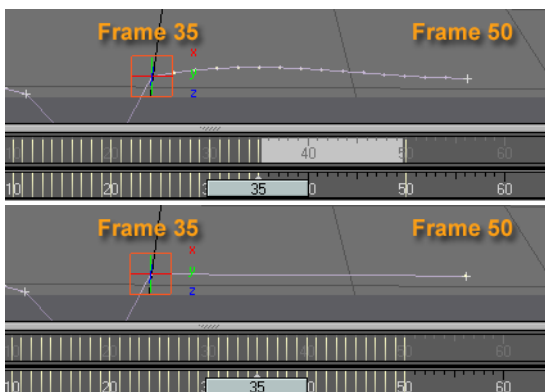
Once applied a keyframe will be placed on every frame in the selected time range.



**NOTE:** No calculations are needed when using this function.

## Bind Motion

This function will bake keyframes for the entire time range based on the keyframe of the first frame in the range.



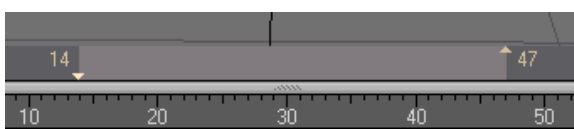
The example above shows bind motion in action. The Time range selected is from frame 36 to 49. When **Bind Motion** is selected every frame in the time range uses the data from Frame 35.

## IK Booster Track

If the **DopeTrack** is not active while in IK Booster mode, the IK Booster Track becomes functional. The **Keyframe Mode** and **IKB Menu** buttons appear on the track as discussed earlier and all operations for IK Booster that were available on the Dope Track are available for the IK Booster Track.



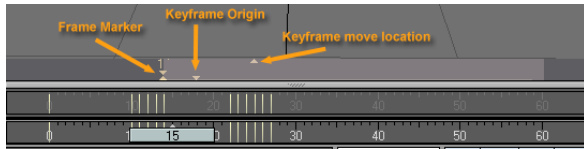
**IK Booster Track Time Ranges** — The IK Booster track displays time ranges differently than the DopeTrack. The time range will be shaded as it is in the Dope Track, but only the first and last frame numbers will be displayed unlike the DopeTrack.





## Time Range Keyframe Shifting

You can use the IK Booster Track to shift a selected area of keyframes. This function is unique to the IK Booster Track and can be very useful.



**IK Booster Frame Marker** — Place a frame marker by left clicking on the IK Booster Track. The Frame Marker is the frame on the track that marks the frame that the time range ends or begins depending on which side of the marker you click and drag in.



**NOTE:** To remove the frame marker, simply left click on it.

**Keyframe Origin** — This marker shows the original location of the keyframes before any movement has taken place.

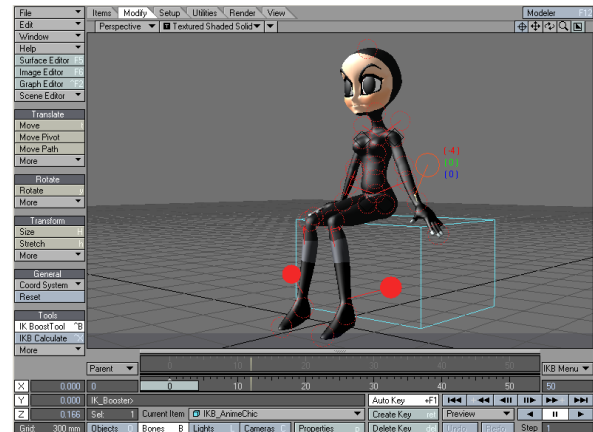
**Keyframe Move Location** — This marker shows where the Keyframe Origin is being moved to.

The **Time Range** will be shaded to display what area of time is being affected.

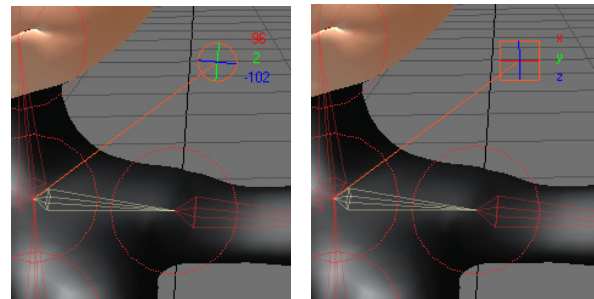


**NOTE:** The items that are affected by Time Range Keyframe Shifting are determined by the Keyframe mode.

## IK Booster Viewport Controllers



Once you have applied IK Booster to your object you have many options for displaying and working with the controllers that are created for each item. The controllers appear as circles while in **Rotate** mode and boxes while in **Move** mode.



Left clicking and dragging on the controller will allow you to move and rotate the controller freely. You can also left click and drag on individual channels to constrain the movement of the controller.

### Rotate Mode:

**Red** — Heading Channel

**Green** — Pitch Channel

**Blue** — Bank Channel

### Move Mode:

**Red** — X Channel

**Green** — Y Channel

**Blue** — Z Channel



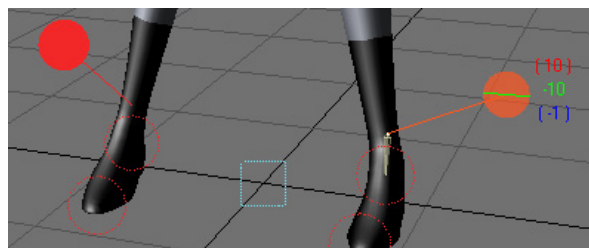


## Controller Right Click Menu

Right clicking on the controller will bring up the controller menu.



**Fix** — This function will lock a controller in place. The controller will be displayed as a solid circle to indicate that it is fixed.



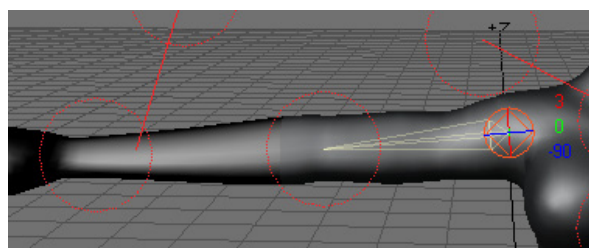
*Ankle Controllers are fixed*

**IK Stop** — This function will make the selected controller the new base of the IK chain, and only controllers farther down the chain are affected by any IK operation. The controller will display a diamond shape to indicate that IK Stop is active.

A good example of using **IK Stop** is wagging the tail of a dog. Imagine a string of bones inside a solid dog object, extending down its tail. You want the tail to follow the dog as it moves about, but you do not want the dog's body to wag as the tailbones are animated. You activate **IK Stop** for the bone at the base of the tail to accomplish this.

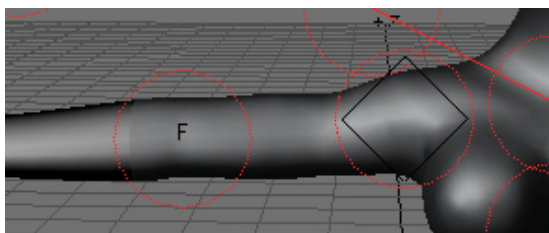


**NOTE:** This function is similar to Unaffected by IK of Descendants.

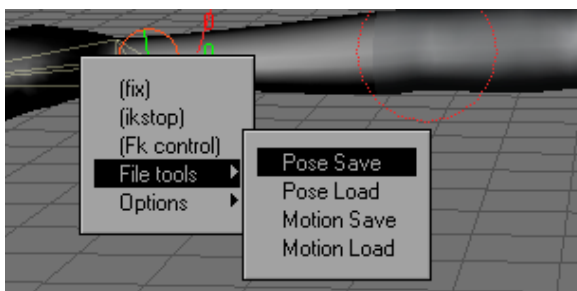


*The Shoulder Controller has IK Stop applied.*

**FK Control** — This function will change the controller from Inverse Kinematics to Forward Kinematics (FK). Forward kinematics is the standard type of motion generated in LightWave when you rotate/move any parents in a chain. The position of the last controller in the chain is determined indirectly by the combination of all positions of any parent controller up (i.e., forward) through the chain. (For instance when you move your upper arm, your forearm and hand also move with it.)



**File Tools** — The options listed below are file management tools for a controller.



**Pose Save** — This function will save an external file that contains the position of the selected controller for the frame you are currently on.

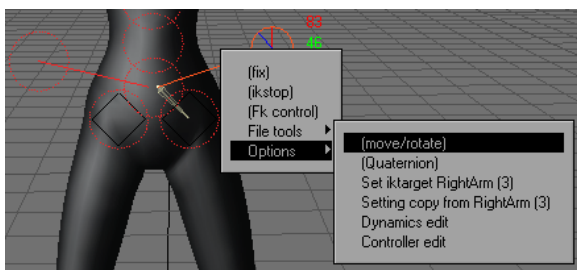
**Pose Load** — This function will load a previously saved Controller Pose file at the frame you right clicked on.

**Motion Save** — This function will save the entire motion data for the selected controller to a text file. This gives you the ability to re-use motions from the selected controller.

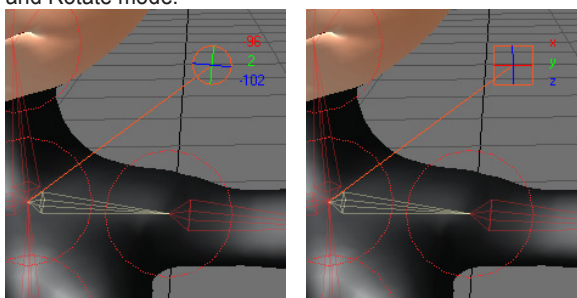
**Motion Load** — This function will load a controller's motion data file that was previously saved using the Motion Save command.

## Controller Options

The **Options** drop down menu has several functions to work with.



**Move/Rotate** — This function will toggle the control between Move and Rotate mode.

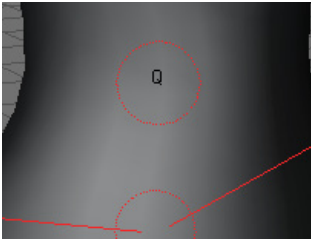


*Left: Rotate, Right: Move*



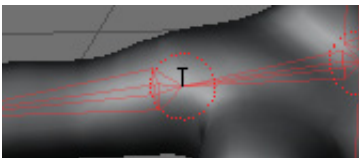


**Quaternion** — This function will activate quaternion inverse kinematics solving for the selected controller. Quaternion IK will help any controller that suffers from gimbal lock and the controller will be displayed with a “Q” to indicate that Quaternion is active.



**FUN FACT :** Interesting side note: The quaternions are members of a *noncommutative* division algebra first invented by William Rowan Hamilton.

**Set IK Target** — This function will allow one controller to drive another controller’s motion. In other words, the currently selected controller drives the previously selected controller. The controller will be labelled with a “T” to indicate that Set IK Target has been applied. Choose **Reset IK Target** to remove the target setup from the controller.



### Steps for setting an IK Target:

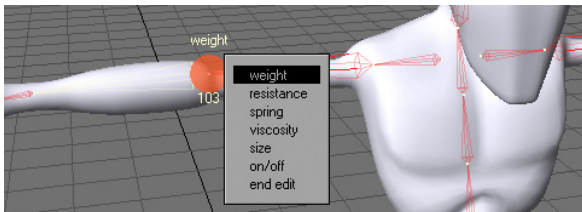
**Step 1:** Select the controller that you want to animate.

**Step 2:** Select the controller you want to drive the animation.

**Step 3:** Select Set IK Target. The target is setup and ready to go. Use the Target controller to drive the controller in step 1.

**Setting Copy** — This function will copy the settings from the previously selected controller. This can be a real time saver when setting up controllers on like items. One example would be setting up the left leg and copying the settings over to the right leg.

**Dynamics Edit** — This function will activate Dynamic Edit mode. Dynamic Edit mode allows you to set the dynamic properties for your controllers.



To change the values for **Weight**, **Resistance**, **Spring**, and **Viscosity**, simply left click and drag on the control. Dragging left will decrease the value while dragging right will increase the value.

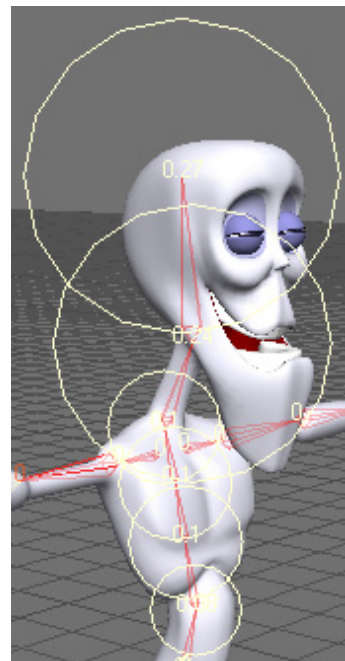
**Weight** — Defines the weight of the controller. You can achieve heavy motions by increasing the Weight value, and you can create light motions by decreasing Weight.

**Resistance** — Controls the amount of air resistance.

**Spring** — Controls the springiness or stiffness of the controller. Reducing the Spring coefficient creates soft motions, while raising Spring coefficient produces motions with a stronger repelling force. Setting the coefficient to an extremely large value creates stiff motions.

**Viscosity** — Controls the impact of a collision. A controller with a higher Viscosity value tends to keep its shape more. If a controller bounces, a higher Viscosity value will have less bounce motion because the Viscosity absorbs the bouncing force.

**Size** — This setting determines the area of influence the controllers have. The image below shows various sizes on the controllers that make up the head and torso area of this character.



**On/Off** — This function will determine whether a controller will be affected by dynamic calculation.

**End Edit** — This function will take you out of Dynamic Edit mode.

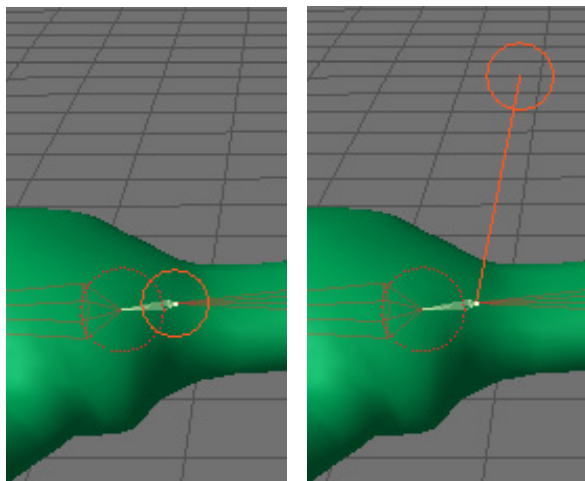


**NOTE:** Bone Dynamics must be active for these settings to be calculated.



## Controller Edit

This function will activate Controller Edit mode. While in Controller Edit Mode you can freely move the controllers by left clicking and dragging them to a location.



Left: Before Edit, Right: After Edit

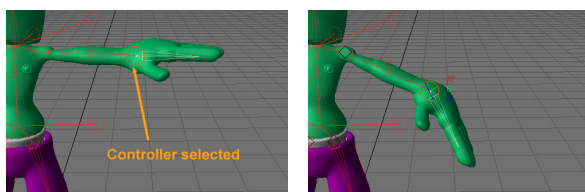
**Controller Edit Mode** has several options that can be used to customise your **Controller** settings. Right click to bring up the **Controller Options** menu.



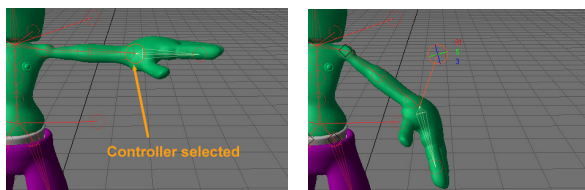
**Reset** — This function will reset the controllers back to its default setup.

**Float /Unfloat** — This function toggles the controller between Float and Unfloat.

**Float** — The controller moves along with the item.



**UnFloat** — The item moves but the controller is anchored.



**Size Mode** — This setting determines the size of the controller when it is selected.

**Size 1** — Smallest selected size.

**Size 2** — Medium selected size.

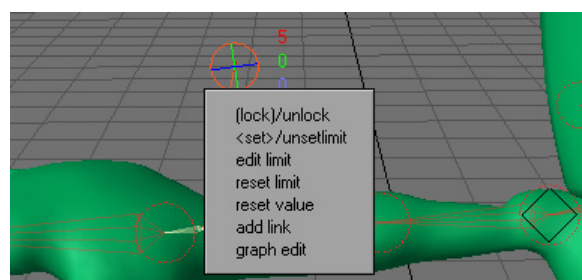
**Size 3** — Largest selected size.

**Not Change** — The selected controller size will be based on **Controller Size**. The selected and unselected controller size will be the same.

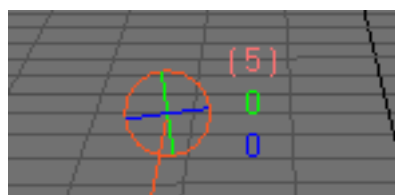
**End Edit** — This command will exit Controller Edit mode.

## Channel Menu

Right clicking on an individual channel will bring up the Channel menu. Each channel can be configured independently.

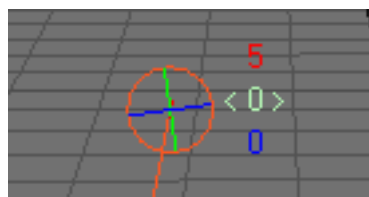


**Lock/ Unlock** — This function will lock and unlock a channel. When it is locked, the number is surrounded with ().



The Controllers Heading Channel is Locked

**Set/ Use Limit** — This function will limit the channels value range. When a channel has a limit, the number is surrounded with <>.



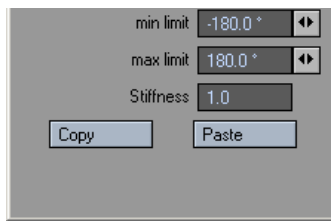
The Controller's Pitch channel has a limit.



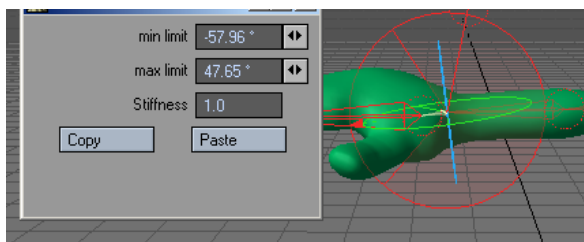
**NOTE:** By default the limits are -180 to 180. This is not ideal in most cases so you will want to edit the limits.



Edit Limit — This command brings up the **Edit Limits Panel**.



You can independently limit heading, pitch, and bank rotations between maximum and minimum values. You can prevent actions like bending a leg backwards at the knee (unless you're animating a sports injury). To use the limiting feature, set the **Min** and **Max** values on the **Edit Limits Panel**. Setting rotation limits can be particularly important when using inverse kinematics. When setting the **Min/Max Limits**, you will see a visual guide in the viewport that represents the limits.



NOTE: The maximum value must be greater than the minimum.

**Stiffness** — Controls the springiness or stiffness of the limit. Reducing the Stiffness coefficient creates soft motions, while raising the Stiffness coefficient produces motions with a stronger repelling force. Setting the coefficient to an extremely large value creates stiff motions.

**Copy and Paste** — Use these commands to copy and paste settings between channels.

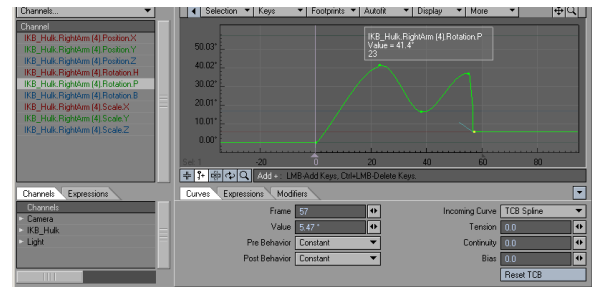
**Reset Limit** — This function sets the maximum and minimum settings of the limit to a present value. When **Reset Limit** is applied, you can edit the **Maximum** value by holding down the **Ctrl** key + left click and drag to the value that you want the limit to be set at.

**Reset Value** — This function will set the value of the channel to zero (0). This is very handy when you need to quickly set the channels value back to zero.

**Add Link** — This command will add a **Booster Link** to the selected channel. The reference controller will be the last controller selected.

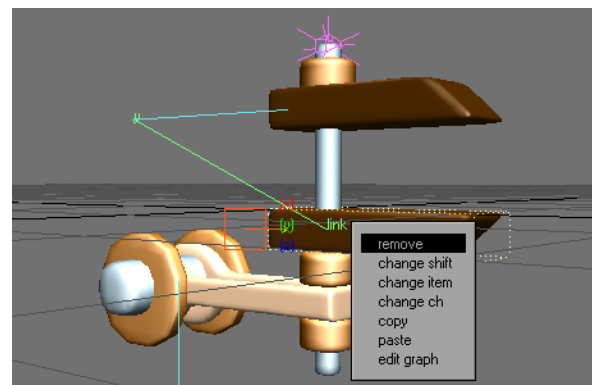
## Graph Edit

This command will launch the **Graph Editor** with the selected Channel active.



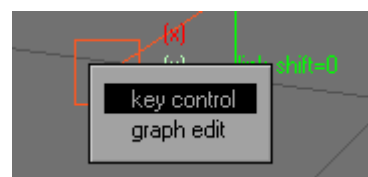
## IK Booster Link

IK Booster Link can be used to link a channel from one controller to a channel in another controller. The Linked controller can be driven by the referenced control. This is very similar to how expressions can be used without writing any expressions.



**Remove** — This command will remove the Booster Link Modifier from the channel.

**Change Shift** — This function will adjust the motion of the linked controller at different times during the referenced controllers motion. To remove Change Shift from a channel, simply right click on the channel and choose **Key Control**.



**Change Item** — This function will change the reference controller to the last controller selected.



NOTE: This option can be changed in the **IK Booster Modifier Panel** under the **Reference Item** drop down menu as well.

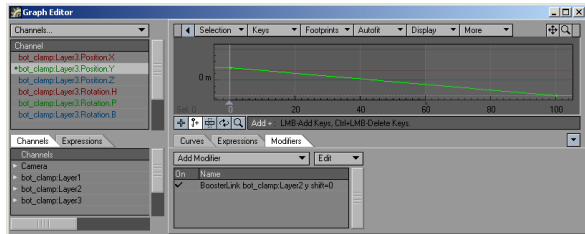
**Change ch** — This command will allow you to change the channel that drives the selected controller.



NOTE: This option can be changed in the **IK Booster Modifier Panel** under the **Reference Ch** drop down menu as well.

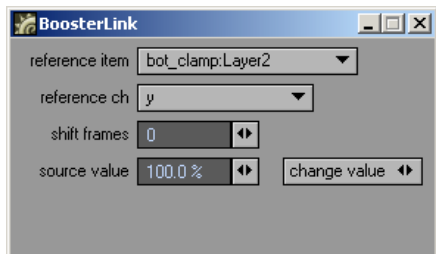
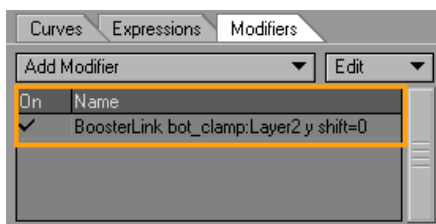
Copy and Paste — These commands will copy and paste **Booster Link** settings.

Edit Graph — This command will launch the **Graph Editor**.



NOTE: IK Booster Link will abide by any limits that have been set on the channel that is driving the current controller.

## IK Booster Link Modifier Options



Reference Item — This function will change the reference controller. The reference controller is what drives the linked controller.

Reference Ch — This command sets the channel of the Reference Item that drives the linked controller.

Shift Frames — Delays the linked motion by the number of frames specified.



NOTE: Entering “5” moves the frames -5, and -10 moves it +10.

Source Value — This function sets the percentage between the minimum and maximum value of the referenced channel that will be used.

Change Value — This function will change the value of the selected channel.

## IKB Calculate (default keyboard shortcut **Ctrl** X)

The IKB Calculate operation (**Modify> IKB Calculate**) is used to calculate IK Booster's Bone Dynamics and generate keyframes based off those calculations. A keyframe will be created on every frame much like motion-capture data.



NOTE: Bone Dynamics need to be properly set up before this will work. Otherwise, clicking it causes an error.

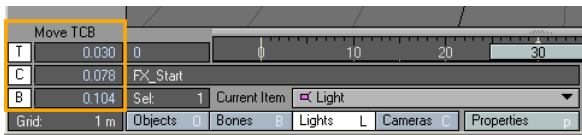


NOTE: The **IKB Calculate** button will also calculate dynamic objects as well. For more information about **IK Booster** and **Bone Dynamics** see the **IK Booster** section.



## Move TCB Tool (default keyboard shortcut **Ctrl** **G**)

The **Move TCB** tool (**Modify > Move TCB**) allows direct manipulation of Tension, Continuity and Bias in the main layout viewports. Use the input field in the lower left hand corner or interactively drag in the viewport to change the settings.



### Mouse Operations:

**LMB** - Tension

**Ctrl+LMB** - Continuity

**RMB** - Bias



NOTE: With AUTOKEY OFF — it will only allow you to adjust where a keyframe already exists.  
With AUTOKEY ON — it will add a new keyframe, if you adjust where there isn't a keyframe already.

### TCB defined:

**Tension** causes an object in motion to slow down, or move a little bit less in each frame, as it nears the keyframe, and to accelerate as it passes the keyframe (-1 = low tension, 0 = normal tension, 1 = high tension). Without **Tension** (i.e., value of 0), the object would pass through the keyframe position at a constant speed. Positive values slow an item through a keyframe (*ease-in*) while negative values speed it up (*ease-out*).

A high **Tension** value (1.0) is often used at the end of a flying logo move in order to make the logo come to a gradual stop. High **Tension** at the beginning of this move would have the logo start slowly, while a negative value would have the logo start quickly.

**Continuity** accentuates a break or change in an object's graph (-1 = sharp, 0 = normal, 1 = smooth). Negative **Continuity** gives a sharper transition in the spline path at a keyframe, while positive **Continuity** gives a broader transition (sometimes *over-continuous*) through a keyframe. Negative **Continuity** is usually used to replicate a sharp change in motion such as that of a falling ball striking a floor and quickly reversing direction.

You would rarely want to use a positive continuity — this will cause an object to overcompensate as it passes through the keyframe and appear to stutter or *roller coaster* at the frame.

**Bias** determines whether an item's spline path *leans* to one side of a keyframe or the other (-1 = more slack incoming, 0 = equal slack, 1 = more slack outgoing). You accomplish this effect by moving the *slack* in the spline path to one side or the other of a keyframe. This serves to accentuate motion — the incoming motion by undershooting the keyframe and creating a feeling of anticipation, or the outgoing motion by overshooting the keyframe. For example, a race car moving through a turn could use either a negative or a positive **Bias** setting to a) anticipate the turn with a

negative **Bias**, or b) overshoot the turn with a positive **Bias**.

Negative **Bias** values place the slack before the keyframe while positive **Bias** values place it after the keyframe.



NOTE: You can also adjust **TCB** settings in the **Graph Editor**, under the **Curves** tab at the bottom of the panel.







---

## Chapter 20: Animation

---



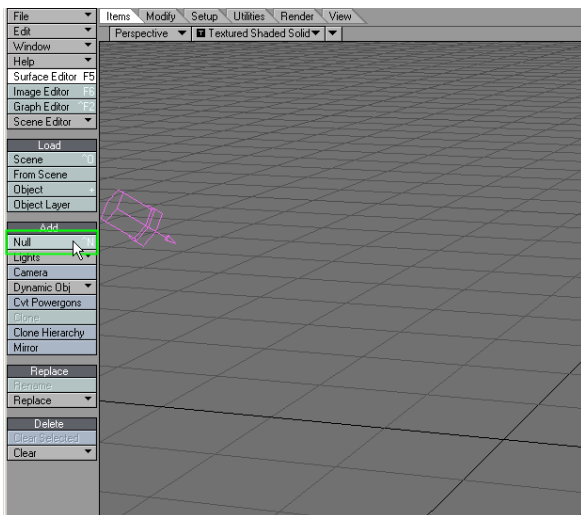
## Animation

Now that you have reviewed cameras, lights, objects, rigs, and almost everything else you need set up in your scene, it's time to discuss animation.

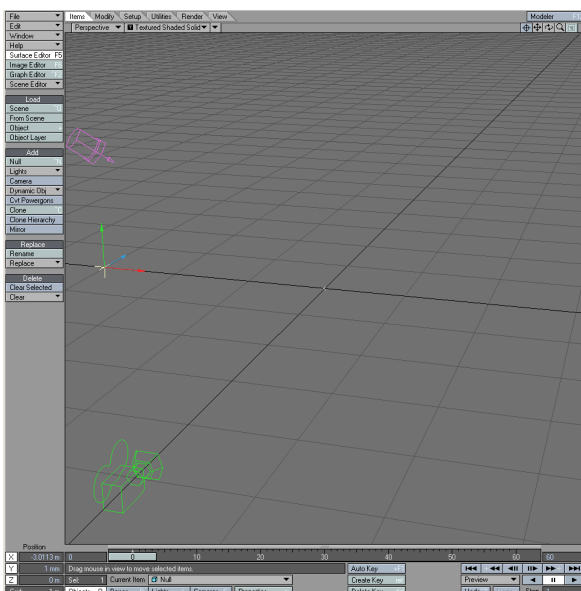
## Keyframing

The way to tell the computer where an object — be it camera, models or lights — is at any given point is to use keyframes. Keyframes fix an object's location and rotation for a given moment in time. Setting one at the start of your animation and one at the end means that LightWave will automatically interpolate between them for the duration of the animation. In fact it will create these in-between frames, called inbetweens, between every keyframe resulting in a fluid animation. To get you started, let's create a piece of the easiest animation possible.

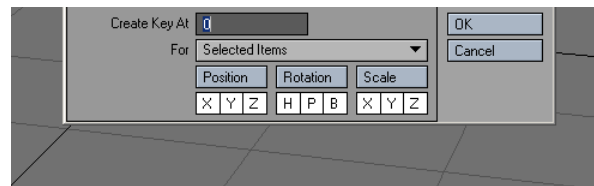
Start LightWave and check to see if Auto Key, at the bottom of the interface window, is highlighted. If so, click on the button to turn Auto Key off. Add a null object by clicking on the Add > Null button (you can also use the keyboard shortcut Ctrl N). It will appear at 0, 0, 0. A requester will also open asking you to name the Null — just accept the default "Null" and click OK.



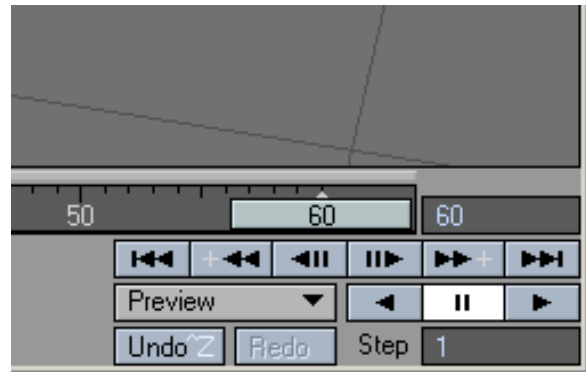
Move the null away a bit. We've moved it three meters left along the X-axis.



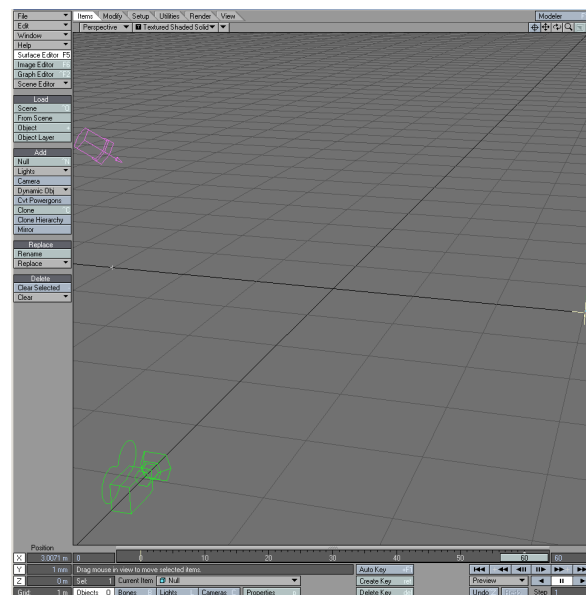
Hit the return key twice. You will see a window appear the first time you hit return and be dismissed the second time. You've now set a keyframe for the start of the animation of your null.



Move the slider at the bottom of the Layout viewport to the end of the slider range — it will be at frame 60 by default.



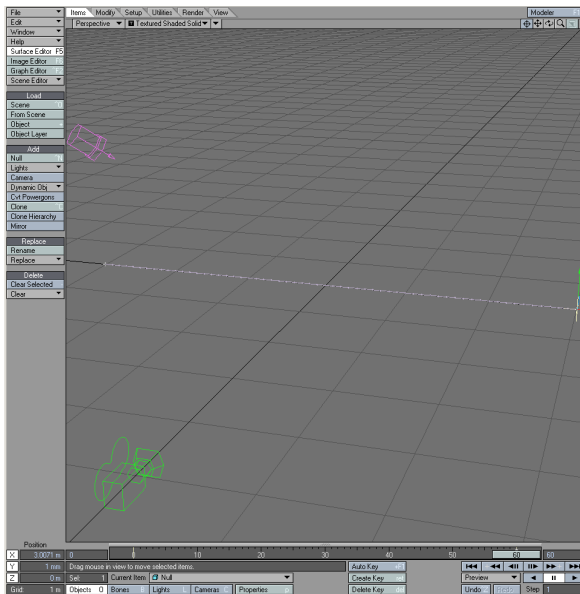
Move the null along the X-axis again, this time to the right six meters.



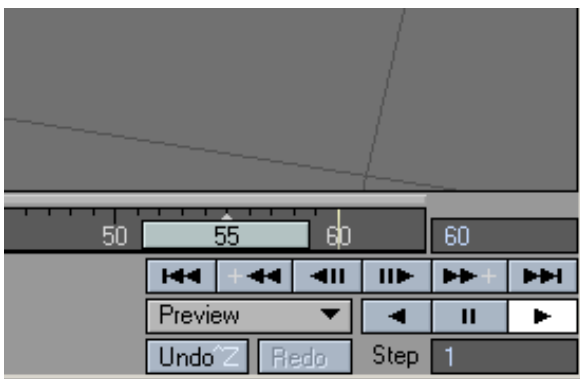
Hit Return twice again to set a second key frame. You will see two things. The first is that there's a line on frame 0 in the timeline under the Layout viewport (there's also one under the slider at frame 60).



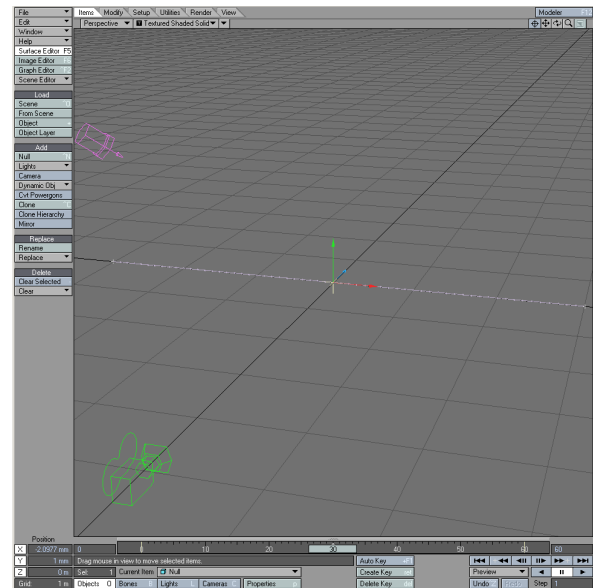
now, but it won't be as visible), the second is that there's a dotted line marking the path that the null object is taking. If you were to count the dots by hand, you would see that there were 60 dots, but just take our word for it. These mark frames along the way.



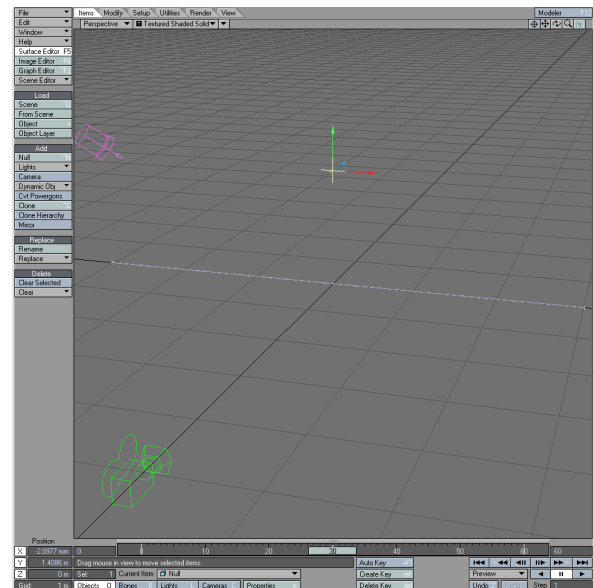
If you hit the play button just under the right-hand side of the Layout viewport you will now see your null moving across the screen and starting again when the slider reaches the end of the timeline. Congratulations! You've made an animation. It's not exactly Disney calibre, but it is an animation.



To make it slightly more interesting, stop the playback by hitting the pause button. Move the timeline slider to the middle of the timeline.

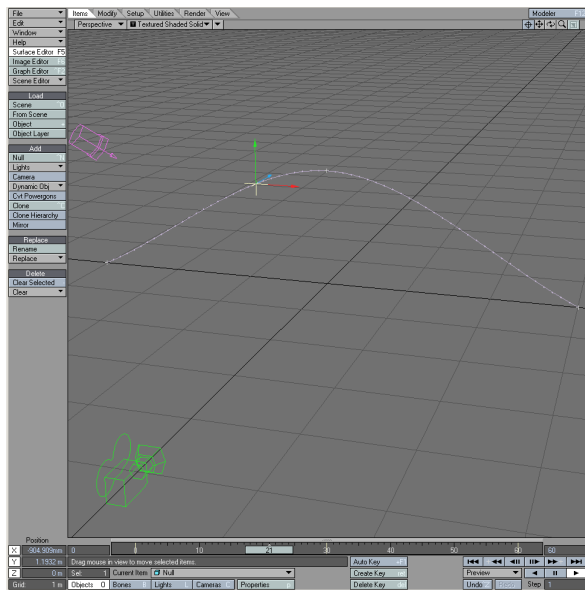


Move the null upwards about four meters. You can do this by grabbing the arrow pointing upwards from the null, or by using the right mouse button.

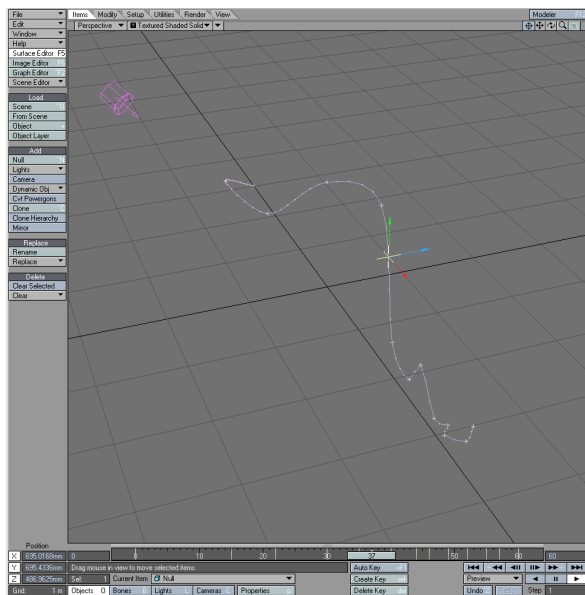




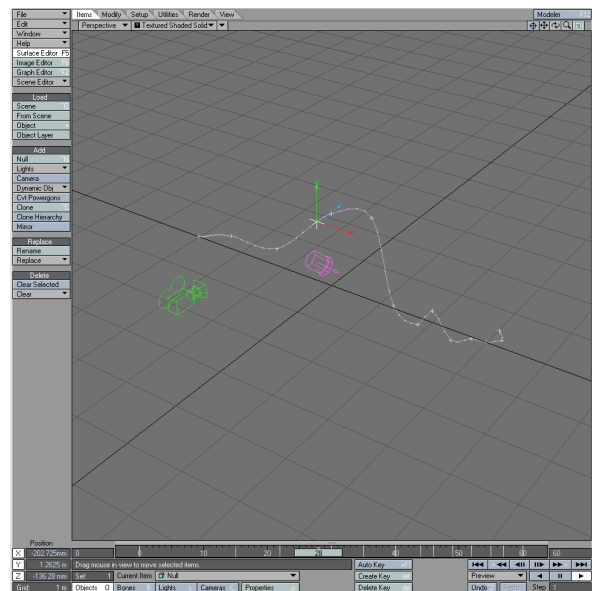
Hit the return key twice. Now when you press play again, you can see that LightWave has taken note of your changes and made the appropriate in-between frames for the motion to proceed smoothly.



Add some more keyframes (remember to press the Pause button). Create a real rollercoaster ride for the Null to take.



When you want to check the null's motion path, press play. When your viewport is in Perspective mode, if you use the navigation controls at the top right of the viewport, you can see how the animation looks from any angle. See something you don't like? Press pause, move the timeline slider to the frame you want to change and move the null. If you click on Auto Key to turn it back on, you will find that don't even have to press the Return key twice to set the keyframes — keyframes are created automatically when you move the object.



All items in a LightWave scene can have independent keyframes — while your null is getting sick on its rollercoaster ride, you can have the light in the scene changing direction and the camera following it around to make it even more disoriented. Even better, just because you've keyframed an object's heading, there's nothing to say that you can't keyframe its pitch and bank on different frames. Go back to our short tutorial and get everything — and I mean everything: null, camera, light — moving. Don't forget that in addition to the Move tool you used to reposition the null, you also have the Rotate tool (keyboard shortcut: Y) and the Size and Stretch tools (keyboard shortcuts: Shift+H and H). You'll find all these tools on the Modify Tab.

Normally speaking, LightWave doesn't work in seconds because motions tend to be too quick and seconds too coarse a measurement. It works in frames — there are 25 to a second in PAL and 30 frames to a second in the NTSC TV system. For film work, there are 24 frames to a second.

Keep working on your simple animation until you start to feel the null's pain at being forced to ride the rollercoaster one more time... This is easy and you'll be glad you've practised before things start getting harder.



## Navigating a Scene



Just under the Layout viewport (or viewports) is the frame slider that we've already used for our exciting null animation. Although its main action is obvious, you also have the ability to change the range shown and also to pan through this range easily. To scrub through the frames you can move the slider with the mouse or you can use the cursor keys. Left and right move you through the frames one at a time, and holding down the shift key at the same time moves you to the previous or next keyframe. If you hold down the alt key while you move the slider with the mouse you will pan the frame range – this means that you will keep the same number of frames on the slider, but the start and end points will change accordingly. If, for instance, you had a 500 frames long animation, the amount of action would cause the detail in the frame slider to be confused. But if you set the end frame to a 100, you can then pan through the other 400 frames of your animation, so this 100 frames could show the range between 0-100, or 357-457, as you like.



**NOTE:** You can also jump to a specific frame in your scene by hitting the F key on the keyboard, which will bring up a requester where you can enter a frame number.

## Playing a Scene



Under the transport controls for the scene, you will see two playback arrows and a Pause button, and under that a step counter. The playback arrows allow you to playback the scene in either direction while the step counter allows you to set how many of the scene's frames you wish to see played back. Setting this counter to 1 will play every frame, setting it to 2 will play every other frame, and so on.



**WARNING:** Make sure your scene is not playing back before you move anything. If you have AutoKey on, you will end up with keyframes all through the animation.



**NOTE:** Your playback speed will vary depending on the complexity of your scene, object display mode, system capabilities and so on. Reducing the size of your Layout window can dramatically increase playback speed.



**HINT:** If the Hub active, you can modify an object in Modeler even while your scene is being played back in Layout and the object will automatically update to reflect the changes you make.



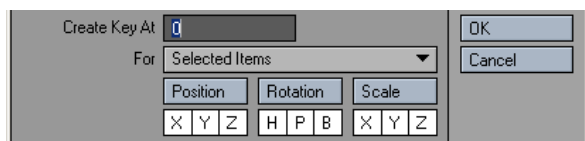
## Creating Keyframes

As we've already learned, objects are always loaded at the Origin. This is the center of Layout's grid: 0, 0, 0 and every new item will have a default keyframe for frame 0 here. If you never create another keyframe for an item, it will stay at this default location throughout the animation.



NOTE: Frame 0 is the default starting point, but you can create keys at frames at less than 0, if you need to.

As we saw in the tutorial at the start of this section, we pressed the return key twice. A window appeared but was dismissed by us hitting the return key the second time. Let's go in for a closer look at this window.



By default the Create Key At field will contain the current frame. You can change the frame number to create the key - using the animation channel values for the current frame - at a different frame by entering a different frame number here. This is a good way to copy a keyframe. You can also re-pose your item and create a new key over an old one.

The For pop-up menu has several options. You can create keys for:

**Selected Items** — All items that are selected, which will always include the current item.  
Generally, this will be the one you use the most.

**Current Item Only** — Only the current item, even if others are selected.

**Current Item and Descendants** — The current item and any child items.

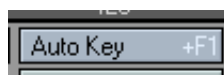
**All Items** — All items in the scene. Beware that this is not limited to just items of the same type. A key will be created using the current Animation Channel settings for all objects, bones, lights, and cameras.

The dialogs for creating motion keyframes for scene items have three rows of Toggle buttons. This lets you create animation channel keys independently. The Position, Rotation and Scale buttons allow a row to be turned on or off with one click. All channels are enabled by default and their state is remembered.



NOTE: Keys created using the Create Motion Key dialog use the Graph Editor's Default Incoming Curve type, discussed later. Generally, we suggest you use TCB Splines and not Bezier Splines. Bezier tangents are determined at the moment of creation and don't automatically adjust as new keys are created. This can cause awkward movement through keys.

## Creating and Modifying Keys Automatically



To automatically create or modify keys you must activate the Auto Key option on the main interface. This is the global on/off switch for automatically creating keyframes. It works in conjunction with the Auto Key Create setting (General Options Tab of the Preferences Panel).

Auto Key is a time-saving feature for advanced users and can be very useful for quickly roughing out a motion path; however, beginners may want to stick with creating keys manually. Most of the exercises in the manuals assume that Auto Key is not active.

Make sure you are always aware of the status of Auto Key and Auto Key Create. These options can result in an animation with extraneous keyframes.



HINT: A good way to get familiar with this feature is to set Auto Key Create to Off and leave Auto Key active in the General Options requester (keyboard shortcut: O).

This setting means that existing keys will update automatically if you adjust them, but new keyframes are not automatically created if you move, rotate, size or stretch an item. You can use the Undo function if you make a mistake when adjusting a keyframe.



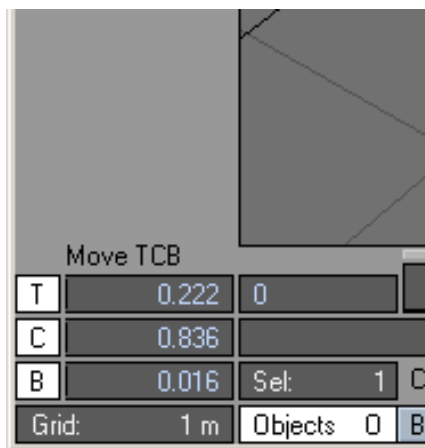


## Editing Motion Paths Directly in a Viewport

You can also move an item's motion path directly in a viewport using the Move Motion Path tool (Modify > Translate: Move Path).



The Move Path tool allows you to shift your completed motion path for an item, keeping all the keys in the same places relatively. Let's say you had a very complicated scene, it would be useful to be able to move to an empty portion, create the motion path for your object and then move it into place. With Move Path, that's exactly what you can do.



Another tool in the Modify menu will directly affect your motion path. At the bottom of the Modify list you will find Move TCB. This allows you to set the biases discussed in the Graph Editor section of the manual, but from Layout, rather than having to visit the Graph Editor itself. To use it, move to a keyframe, click on the Move TCB tool and you will notice that the lower left corner now displays T, C, B instead of the more familiar X, Y, Z or H, P, B.

You can enter figures in here for what you wish your TCB settings to be, or you can use the mouse:

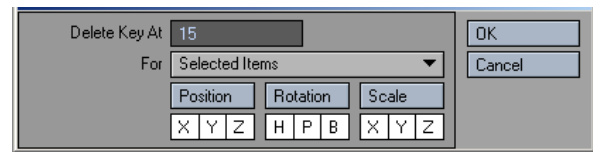
LMB: Moving left and right alters the Tension

Ctrl+LMB: Moving left and right controls the Continuity

RMB: Moving left and right controls the Bias

## Deleting Keyframes

You delete keyframes in the same way you create them.



*To delete a keyframe:*

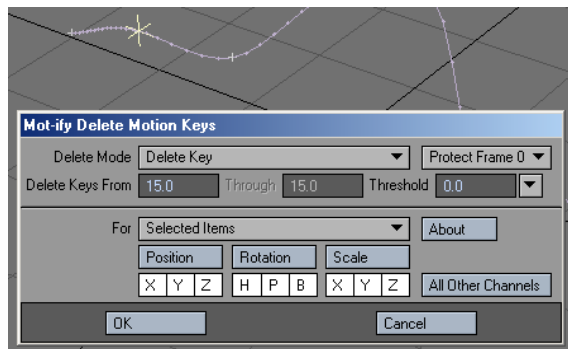
Step 1: Select the item. Normally, you will also go to the keyframe you wish to delete.

Step 2: Click the Delete Key button or press the Del key.

Step 3: The Delete Motion Key dialog will appear. It uses all of the same controls previously described for the Create Motion Key dialog.



## Delete Motion Key Plugin



Use the TMP Motify Delete Motion Key generic plugin (Utilities > Additional > Motify) to delete keyframes, clear motions, delete ranges of keys, delete keys within a threshold, and more. Motify can be used to completely replace the built-in Delete Key dialog.

### To use Motify:

Step 1: Select the item(s) you want to delete keyframes from. Normally, you will also go to the keyframe you wish to delete.

Step 2: Run Motify

Step 3: Choose a Delete mode.

Step 4: Enter a range of keys in the Delete Keys From and Through input fields.

Step 5: Enter a Threshold.

Step 6: Choose a For mode.

Step 7: Enable or disable channels

Step 8: Click OK.



NOTE: Many of the above steps are optional. For example, to simply delete a keyframe at the current time (using the previous For mode), run Motify and

click OK.

The For pop-up menu determines which objects will have their keys deleted and has several options. You can delete keys for:

**Selected Items** — All items that are selected, which will always include the current item.

Generally, this will be the option you use the most.

**Selected Items and Descendants** — The selected items and any child items.

**Current Item Only** — Only the current item, even if others are selected.

**Current Item and Descendants** — The current item and any child items.

**All Items** — All items in the scene. Beware that this is not limited to just items of the same type.

The Position, Rotation and Scale buttons allow a row to be turned on or off with one click. Only the selected channels will be processed. You can Shift click to invert a group's selection state.

The All Other Channels button is a powerful feature, but can cause a lot of damage to your scene if used incorrectly. All non-motion channels will also be processed (the Position, Rotation and Scale buttons still determine if the motion channels are processed). This includes all envelopes applied to those items (like Camera Zoom Factor and Light Intensity), envelopes for applied plugins (such as Morph Mixer channels), and surface envelopes applied in the Surface Editor. Use this feature carefully, especially when deleting keys on multiple items at once.

The About button will open an informational dialog, including a list of keyboard shortcuts.

## Delete Mode

The Delete Mode determines what is done to the item's motion. The default mode, Delete Key, deletes the key at the frame entered in the (Delete Keys) From field. Both integer and fractional keyframe values can be entered.

The Threshold field can be used to determine how close a keyframe has to be to the defined frame to be deleted. This is useful for deleting fractional keyframes when Allow Fractional Current Frame is disabled on the General Options Tab of the Preferences Panel.

**Delete Keys Within Range** will delete all keyframes between and including the From and Through frames. The Threshold setting, in this case, will extend the range lower and higher - all keys within the range will be deleted.

**Delete Keys Outside Range** will delete all keyframes outside, but not including, the From and Through frames. Threshold is not used in this mode, as it only applies when the end frames are also deleted.

**Delete Keys Before Range** deletes all keyframes before, but not including, the From frame. Similarly, **Delete Keys After Range** deletes all keys after, but not including, the Through frame.

With **Clear Motion**, all existing keyframes will be deleted. This will only clear the channels marked at the bottom of the dialog.

If all keyframes become deleted in any of these modes, a new default keyframe is created at frame 0 with a position and rotation of 0.0 on all axes, and a scale of 1.0.



## Threshold

Threshold determines how close a keyframe has to be to the From and Through frame numbers in order to be deleted. The default value of 0.0 means that the key must exactly match the frame numbers entered. A value of 0.1 means that any keyframe within 0.1 frames will be deleted. For example, if you are trying to delete keyframe 20 with a Threshold of 0.1, all keys between 19.9 and 20.1 will be deleted. A value of 0.5 can be used to ensure that any fractional keys between the current frame and the next or previous frames are deleted without going into the domain of the next keyframe. The small pop-up menu to the right of the Threshold input field contains a number of reasonable presets values.

## Protection

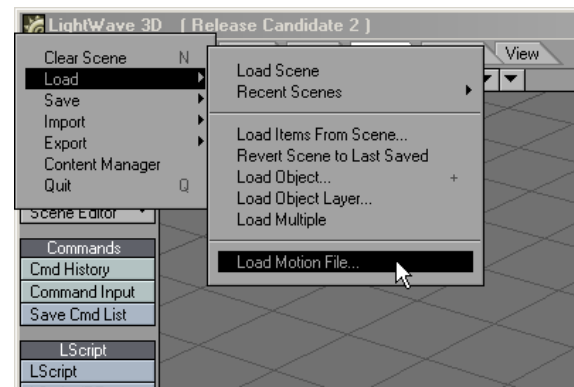
The Protection pop-up menu can be used to ensure that certain important keyframes are not deleted. This is especially useful when using the Delete Keys Outside Range or Clear Motion modes, where deleting all keyframes could ruin your scene or destroy your character's bone setup and IK poses.

No Protection means that no keyframes will be protected from deletion. This allows you to delete any keyframe.

Protect Frame 0 will ensure that no keys at frame 0 are deleted. Similarly, Protect Neg & 0 will protect frame 0 and all negative keyframes.

Protect First Key and Protect Last Key will keep you from deleting the first or last key in the channel, respectively.

## Saving and Loading Motion Files



To save the selected item's motion path to a file, choose File > Save > Save Motion File. Use the filename extension .mot when saving a motion file. Mac users should note that this extension is not automatically added — you must type it as part of the file name and then save the file. To load a saved motion file and apply it to the selected item, choose File > Load > Load Motion File.



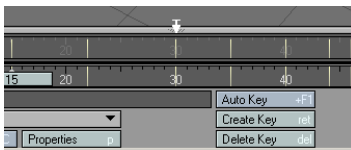
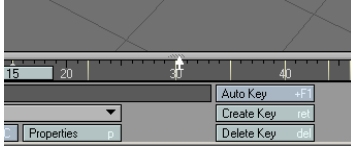
**NOTE:** Modeler's Path Extrude and Path Clone commands use these files to execute their operations.



## DopeTrack

If you've noticed, just above the time slider that shows all the frames in a LightWave scene there is a thick bar. Clicking on this seems to open a new time slider, it's actually a tool called the DopeTrack, to help the workflow of animators.

The DopeTrack allows you to play with keys. Not only their position in time and the scene, but also to alter their properties. You can set up markers here to tell you when things should happen in the scene and bake ranges of keys to set your animation from frame to frame.



You'll notice that when the DopeTrack is closed and your mouse pointer is over the thick bar, your mouse pointer will change from being a cross hair to an arrow pointing upwards. When DopeTrack is open and the mouse is over the thick bar, it will be a downwards-pointing arrow.

What's the point of DopeTrack? Let's say you have an animation and there's a keyframe at frame 13 that you really want at frame 12. You could open the Graph Editor and move the keys there. If you have a lot of items in your scene, it may take a while to find the object you want to edit.

Let's say you've moved this frame, but then your boss decides he wants an identical keyframe a bit later in the animation – say ten frames on. Then, being the indecisive management type he is, he wants to have those two keyframes moved along a bit in the animation. You could do this in the Graph Editor, but instead, you can just sit there nonchalantly, with a breezy “no problem, boss!” and just use the DopeTrack...

When you first open the DopeTrack it just looks like another timeline, only with darker numbers. The difference is that you can grab any keys on the DopeTrack and shift them about. Moreover you can grab a section of the timeline and any keyframes in that area will be selected. You will then be able to move all the selected keyframes at once. By default, the lines representing the keys are simple pale yellow lines, but if you click the right mouse button over the DopeTrack, a menu will pop-up where you can enter Channel Edit Mode, where the three channels that make up a keyframes X, Y and Z coordinates are represented by the red, green and blue lines. If you are in Rotate Mode the red, green and blue represent the Heading, Pitch and Bank channels. If you turn off some of the channels over on the left, you can then select just the channels you wish to move.

If this were all DopeTrack could do, it would be enough, but there's more. A right click on the DopeTrack itself will reveal a menu that gives even more options.



### Select All Keys

Selects all the keyframes visible on the timeline

### Clear Selection

Deselects all selected keyframes

### Add Marker

This adds a marker to your timeline that is visible no matter which item you have selected in the scene. You can also set a text that appears in the normal info window under the timeline. You can select this menu item, or you can hold down the shift key and double click to set a marker. You can set a marker anywhere on the DopeTrack, regardless of the location of the timeline cursor.

### Delete Marker

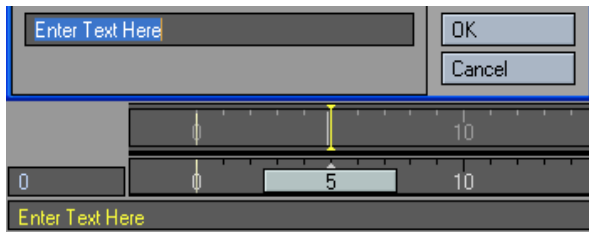
Position the timeline cursor on a marker and this option becomes available.

### Clear All Markers

Clears all the Markers you have placed on the DopeTrack.



### Set Marker Text...



Put your timeline cursor on a marker and it will become bright yellow and this menu item will become available. A marker is visible whichever item you select — it is based on the scene rather than an individual item. Whenever the time slider is on the frame where a marker is placed, and the DopeTrack is open, the marker will light up in bright yellow and the Marker text will appear in the status window. There is a limit of 62 characters visible in this status window, although there is no limit on the quantity of characters you can put in the Set Marker Text... window.

### Baking

If you right click and drag a selection on the DopeTrack this menu option will become available to you. Baking allows you to generate keyframes at every frame over the region you specify, to make sure of your animation or for export to another package.



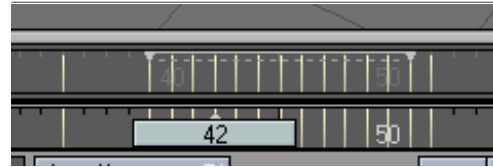
You can set a local or global bake zone. A local one bakes the current item and is shown in the same color as the keyframe markers on the DopeTrack, whereas a global bake zone will bake all items through those keyframes and is shown in blue. You can resize a bake zone by dragging the arrows at either end of it to make it shorter or longer, or move it in time by dragging on the line between them.

You can also hold down the Alt key and LMB drag on the time line and this will create a local bake zone immediately. There will be no need to even use the menu to set it.

Either way, when you have selected a zone to bake, the next thing is to know exactly what "baking" means. It means that you will create a keyframe at every frame in the zone you've selected. This sets in concrete your motion path, which can be useful if you have something precise to follow, or you need to export your motion to another application, like a game engine, for instance. Game engines don't usually like IK, so by baking the motion your character goes through; you can replicate the effects of using inverse kinematics, without actually using it.

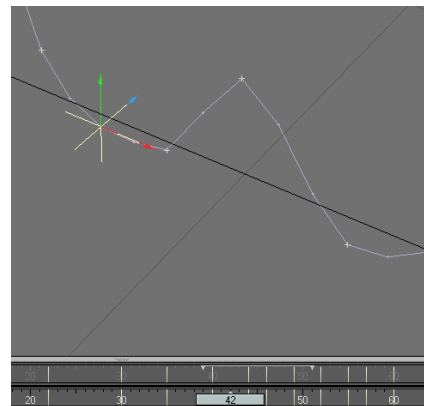


NOTE: If you have local bake zone(s) marked, but want to clear them, you can choose Clear Current Zone from the Baking sub-menu by right clicking on the DopeTrack again. Clear All Zones will remove both local and global zones.

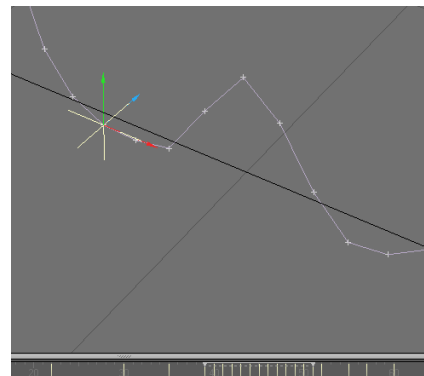


We now have our baked keys, let's move onto the Apply sub-menus. All three Apply sub-menus use the first and last keyframe in a range as boundaries, and the keyframe that the slider is currently sitting on as a control point. The slider should be sitting inside the baked range.

First, here's our scene before a bake of the motion:

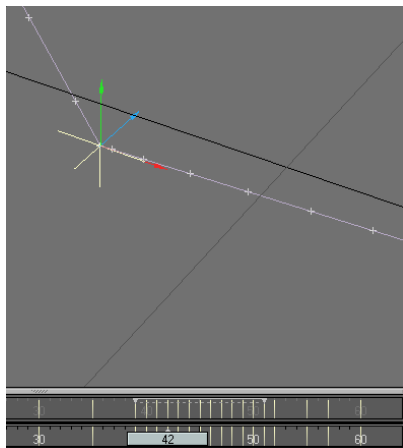


Here's our scene after a bake:



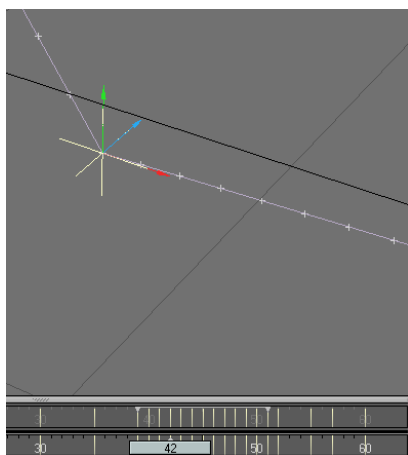


Now, here's the scene with the various "Applies" applied. Remember the shape that results depends on the position of the frame slider. For demonstration purposes we will continue to leave ours on 42, even if the results could be more dramatic with it elsewhere. Also to be noted is that if you don't like the results of a particular Apply, you can change it for one of the others with no penalty. However, you can't go back to the previous un-Apply'ed state, so make sure you save your scene before you use the Appls...



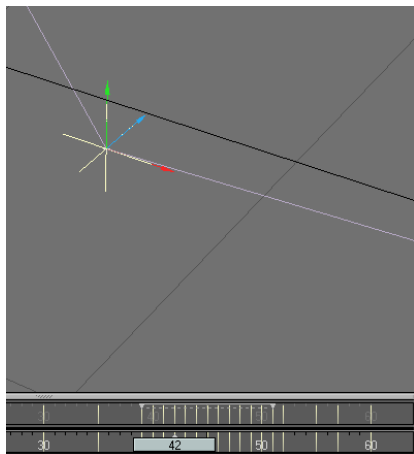
### Soft Apply

In the case of "Soft Apply", a Hermite spline using default Hermite co-efficients is generated using these three keyframes as points to smooth out the intervening keyframes.



### Linear Apply

"Linear Apply" uses the same keyframes, but simply "increments" the keyframe values between the control points.



### Flat Apply

Snap all intervening keyframes to the value of the control keyframe (the one the slider is currently on).



NOTE: To get rid of a set of baked keyframes, select them all by wiping over them using the left mouse button to select them all and then click the right mouse button to access the Delete Keys menu item — the Del key will bring up the Delete Keys window that will force you to delete the keys one by one.

### DopeTrack\_Apply Zones

The Apply Zones in Dope Track have now been enhanced with color for improved navigation. The new zones use a CYM (Cyan, Yellow, Magenta) color scheme.



Local Zones are represented with Cyan. Global Zones are represented with Magenta. Proxy Zones, such as IK Booster, are represented in Yellow.

### Snap Keys

If you have received a scene that uses fractional frames, but you don't want to use all or part of those keys, you can select the keys that are off-integer frame numbers that you wish to adjust, and use Snap Keys to quantise them to the nearest whole frame number.

### Cut, Copy, Paste Keys

As you might expect, these allow you to move keys around the scene by cutting or copying them from where they are, and pasting them where you want them. If a group of keys are selected and you copy them, you will paste them starting at the time slider position.

### Add and Delete Keys

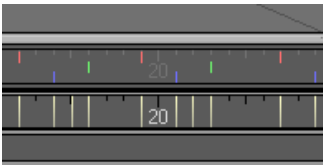
The Add function will create a keyframe in the DopeTrack where the mouse is at the time, so if you have an item at frame 10 in the scene and you Add key at frame 40, it will duplicate your object's position from frame 10 at frame 40.

### Channel-edit Mode

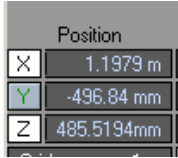


By default, the keyframe marks in DopeTrack are the same shade of yellow as the keyframes in the normal time slider. By switching to Channel-edit Mode you can choose individual channels by turning off the ones you do not wish to select in the bottom left-hand corner of the Layout window. When you switch back to Keyframe edit mode, any channels you have separated in time will become new keyframe lines in their own right.





Once you've shifted the channels separately, new keys appear in the normal timeline.



Turn off individual channels to move the remaining keys where you want them.



NOTE: You can combine the keyboard and mouse to make DopeTrack even easier to use:  
 Alt LMB copies a keyframe  
 LMB doubleclick creates a keyframe  
 Shift LMB doubleclick creates a marker  
 Alt LMB Drag creates a local zone for baking

## BioVision Motion Capture Support

LightWave provides a couple of plugins to support the BioVision (BVH) motion capture file format. The MoCap\_BVH\_Setup generic Layout plugin (File > Import > MoCap\_BVH\_Setup) reads a BioVision BVH file, creates bones, and applies the motion capture data to them. For any use of Motion Capture files, you will need to have AutoKey (Shift F1) enabled.

Set Start Frame Offset to the frame you want the motion to begin.

Bone Scale Factor allows you to set a scale for your object from the start and may need some repeated experiments to get close to the scale of your object. Often BVH files come in at an extremely large scale. As a start, try a scale of 0.1.

The Bone Name Postfix is simply a number appended to the end of all of the bone names (e.g., LeftKnee\_1).

After you run the plugin, replace the top null (in the created bone hierarchy) with the object to be animated. You could also use the Use Bones From Object feature on the Bones Properties Panel.

If you need to change the initial resting position of bones, make sure you reset their rest positions (use the r key). You'll probably need to adjust some of the individual bone properties after you run the plugin.

Applying motion capture to a LightWave object is an arduous process fraught with difficulties but yields good results.

MoCapSkelegons is a Modeler plugin that creates skelegons in Modeler that match the initial rest position of the Biovision BVH data. Use it to determine the correct scale, position, etc. for your object mesh. This object can then be used with the bones created using the MoCap\_BVH\_Setup generic plugin in Layout.

### Motion Preview

There is also a custom object plugin that can be used to preview BioVision motion data. The preview is fast and accurate. Use it to determine if there were any errors in the motion conversion.

Simply add the MotionCapturePreview custom object to a null object on the Object Properties Panel. (If you run the MoCap\_BVH\_Setup generic Layout plugin, a null called "MotionCapturePreviewNull" will automatically be added to the scene with the custom object plugin already applied.)



## Keyframer

Keyframer is essentially a collection of keyframing utilities. Its functions can be applied to currently selected items or defined by an external text file. You can even save and load frame and motion data.

To use Keyframer, choose Utilities > Plugins > Additional > Keyframer. The functions are divided into three different tabs on the panel: Standard, Transfer, and Other.

## What is affected

When you choose certain functions, the Select Range and Objects Panel will appear. Here, you set which range of frames, scene items, and channels to affect. All operations only modify items listed in the Affected Channels and Affected Objects lists.

The Start and End values define the range of frames that will be affected.

## Affected Channels

The Affected Channels list shows all of the channels that will be affected by operations. You may remove a channel by selecting it with your mouse and then clicking the Remove button.

If you click the Save button, you can save the list of Affected Channels into a text file. You should use the file extension “.lw\_channels”. The file contains just a straight list of the channels, like:

```
Position.X
Position.Y
Position.Z
Rotation.H
Rotation.P
Rotation.B
Scale.X
Scale.Y
Scale.Z
```

You may replace the list of Affected Channels with the channels listed in a text file using the Load button.

Clicking the Default button returns the list of Affected Channels to the default ones.

## Affected Objects

The Affected Objects list will default to the currently selected item(s) in Layout. You can list all scene items by clicking the All button.

If you click the Save button, you can save the list of Affected Objects into a text file. You should use the file extension “.lw\_items”. It will be just a straight list of the scene items, like:

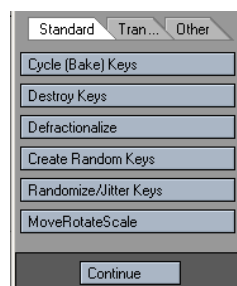
```
Null (1)
Null (2)
Light
Camera
```

You may replace the list of Affected Objects with the scene items listed in a text file using the Load button.

The Use selected objects only option, will cause the list to only show the item(s) currently selected in Layout.

The Include all subchildren option will cause all children of the listed items to be shown.

If Use custom loaded is enabled, all of the items in the Affected Objects list will be affected. Otherwise the list is always dynamically determined by what is selected in Layout, what the children are, and so on.



## Standard Tab

Choose Cycle (Bake) Keys — Keys within the range are copied and pasted after the End frame plus the Cycle Gap. The copy is repeated by the number times set with the Repeats slider.

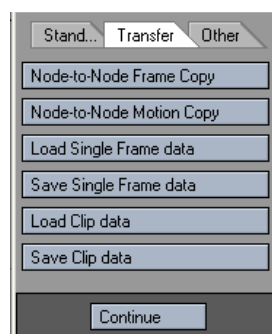
Destroy Keys — Keys within the range are deleted.

Defractionalize — Fractional keys within the range are rounded to the nearest integer value.

Create Random Keys — Creates new randomly spaced keyframes between the Start to End range. The animation curve “should” maintain its shape, if possible.

Random/Jitter Keys — This adds “noise” (i.e., jitter) to existing keyframes between the Start to End range.

MoveRotateScale — Modifies the position, rotation, and/or scale of items. If the Add method is used, the entered value is added to existing key values. If Multiply is used, the existing key values are multiplied by the entered value (be careful using 0!). The Overwrite method replaces existing key values with the entered values.





## Transfer Tab

**Node-to-Node Frame Copy** — Copies the Position, Rotation, and/or Scale of the Source to the Target at the current frame. If the Use World co-ordinate Copy option is enabled, the target will take on the “world space” values of the source (i.e., not the local values). Note that you will still need to create a key with the copied values.

**Node-to-Node Motion Copy** — Copies the entire motion of the Source to the Target. (This is the same as saving and loading a motion file.) If All descendants is enabled, all descendants of both the Source and Target will be used and modified. If you are using All descendants, the Source and Target hierarchies should be identical or results will be unpredictable.

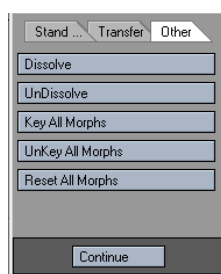
**Load Single Frame data** — Use this function to load the motion data saved with the Save Single Frame data function. Note that the position and rotation channels are loaded regardless of what appears in the Affected Channels list. Also, it will only affect the same item (when the data was saved)-that item must be in the Affected Objects list.

**Save Single Frame data** — This function will save all of the motion data for the Affected Objects at the current frame. Note that the position and rotation channels are saved regardless of what appears in the Affected Channels list. The file should be saved with the “.lw\_frame” filename extension.

**Load Clip data** — Use this function to load the motion data saved with the Save Clip data function. Use the Start and End sliders to trim data. Regardless of the Start and End, the range is always copied in starting at frame 0. Note that the item must have the same name as the item used to save the data.

**Save Clip data** — This function will save all of the motion data for the Affected Objects for a defined range of frames. Note that all channels are saved regardless of what appears in the Affected Channels list. The files are saved to a directory you specify.

Only the data at actual keys within the range is saved. If the start of your range is not a key, the data for the next key (in time) is used. Keys retain their position in time even if you don’t save starting from frame 0.



## Other Tab

**Dissolve** — This function adds keys to the Object Dissolve channel (Object Properties, Render Tab). The object will be 100% dissolved at the Start frame up until the End frame, where it becomes 0% dissolved.

**UnDissolve** — This function is the opposite of Dissolve, described above.

**Key All Morphs** — Sets keys in every MorphMixer channel at the current frame for items with MorphMixer.

**UnKey All Morphs** — Deletes keys in every MorphMixer channel at the current frame for items with MorphMixer.

**Reset All Morphs** — Sets key values to zero in every MorphMixer channel at the current frame for items with MorphMixer.



## Modify Tab

### Translate

#### Move

(default keyboard shortcut T)

When you move an (unparented and unrotated) item in Layout, generally your mouse movements have the following effects:

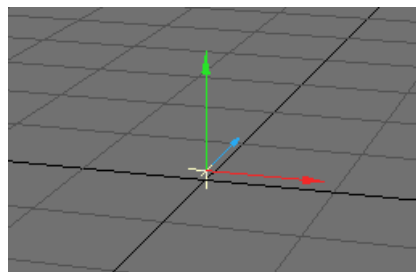
Movement	Move
Left/Right LMB	Left/Right
Up/Down LMB Fwd/Back(3D*)	Up/Dn(orthogonal)
Left/Right RMB	Up/Down

\*3D=Perspective, Light, and Camera views



NOTE: For parented and rotated items, see also the subsequent discussion on Coordinate System.

You generally use a viewport's arbitrary horizontal and vertical axes to adjust an item's position (Move tool). As such, no matter how much you have rotated a Perspective view, dragging your mouse left or right will always move the item left or right on your display. Dragging your LMB left/right in the Right view would move an object along the Z axis. The same mouse movement in the Back view would move it along the X axis.



#### Local Axis Adjustments

Sometimes you may want to move an item using its local axes instead. You can do this by simply holding the CTRL key as you drag. The movement will be along the object's local axis, no matter what view you use (or how much you rotate the item).

Movement (Ctrl)	Move
Left/Right LMB	X axis
Up/Down LMB	Z axis
Left/Right RMB	Y axis

### Move Pivot

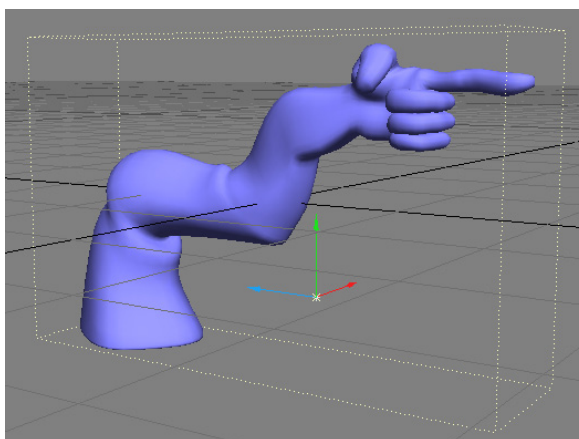


WARNING: Understanding pivot points is fundamental to understanding LightWave.

The *pivot point* is a point of reference used for all objects and does not correspond to any point used in an object's geometry. The pivot point is the center of position, rotation, and scaling. By default, it is located at the object's local Origin. The pivot point, a small yellow star, becomes visible when you select an object.



NOTE: You cannot animate the pivot point.



Pivot Point (Shown selected with handles)

#### Moving the Pivot Point

You can modify the pivot point from its default position in Layout or Modeler. Think of this as giving an offset amount from the object's local Origin. When you move the pivot point, the object will appear to stay in the same position. On the face of it, this might seem confusing to you; if the pivot point is the point used to reference position, the object should move if the pivot point is moved. The reason the object doesn't move is because LightWave automatically adjusts the object's position by the same amount the pivot point is moved.

#### To move the pivot point in Modeler:

Select View> Pivot to activate the Pivot Point tool and move the crosshairs to a new position in any viewport.



NOTE: The Modeler section of this manual has additional info about moving the pivot point in the section on Modeler's View Tab.



### To move the pivot point in Layout:

Step 1: Select the object.

Step 2: Select Modify > Translate: Move Pivot.

Step 3: Move the pivot point as you would any item in the Layout. (If you select the Move tool and look at the object's Position settings, you will see that they have been changed to compensate for the movement in the pivot point — the object will not visibly move.)



NOTE: We strongly suggest that you leave an object at its default rotation and scale before you move its pivot point in Layout or you may get unpredictable results.

### Layout or Modeler

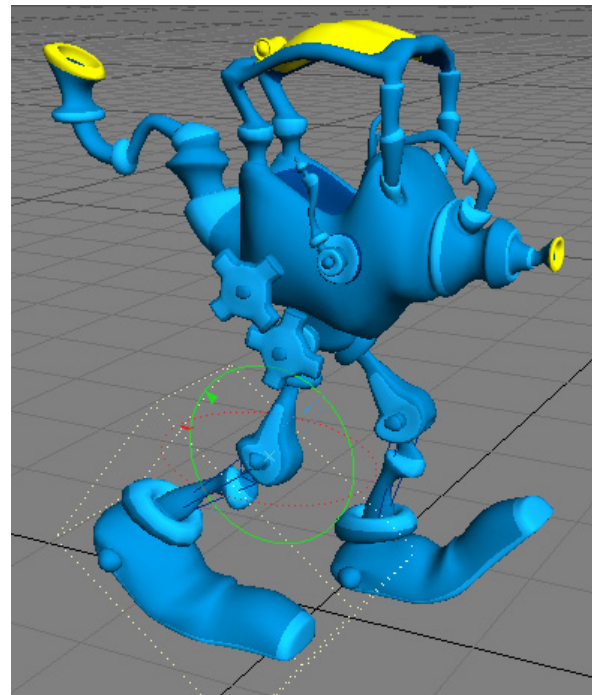
Setting the pivot point in Modeler saves its position in the object file. Setting it in Layout only saves the data in the scene file. As such, it is usually best to set your pivot point position in Modeler.

### Why Move the Pivot Point?

You might be wondering why you might want to move the pivot point. Why not just model all objects so that the Origin is the desired center of rotation? Well, you certainly could do this and should probably try to do this as much as possible. However, there are often circumstances when multiple objects are just parts of a larger element. You want the individual parts to be positioned in their respective locations by default. But you may want to rotate them individually, around their own center point.

Let's take puzzle pieces for example, where all of the pieces fly in to form the complete puzzle. If you modelled each piece centered around the Origin, it would be a real pain to move each piece perfectly into its final resting place. It would be much easier to load each piece by default at its resting position and then set arbitrary starting positions and rotations. To do the latter, you need to move the pivot points to the center of each piece.

A robot is another good example. The arms, legs, feet, torso, etc., are all separate objects, each modelled in their natural resting position in Modeler. However, all parts must be *rotatable* around a unique axis. This is possible only by moving the pivot points of each object.

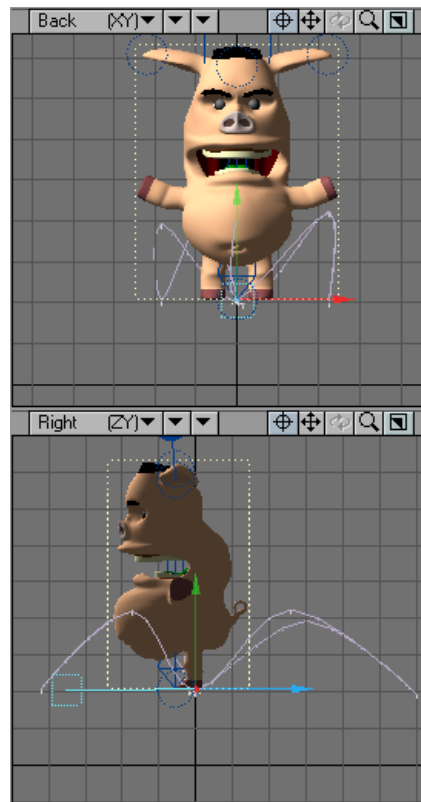
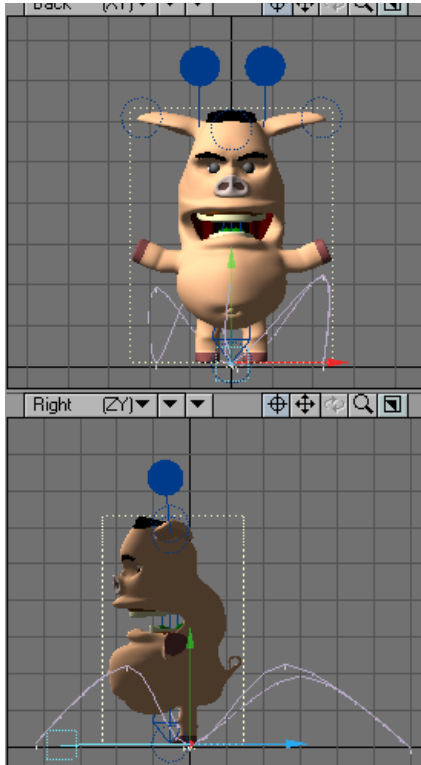


*Pivot Point moved so that rotation is at the Knee.*



## Move Path

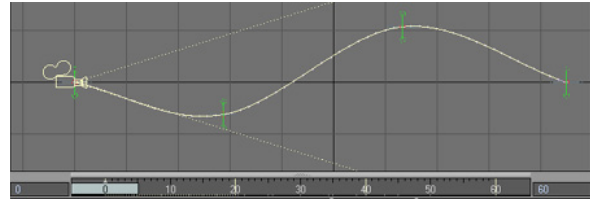
You can move an item's entire motion path directly in a viewport using the Move Motion Path tool (Modify > Translate: Move Path).



## Path Tool

(default keyboard shortcut **Ctrl Y**)

The Path Tool (Modify > Translate: Path Tool) will allow you to directly edit the current item's motion path, right in a Layout viewport. To use, simply activate the Path tool and drag any of the current item's keys. The perfect tool for tweaking an item's motion path.



Green Handles on *motion path* represent each Keyframe.



NOTE: The key will only move along the vertical and horizontal axes of the view, even in perspective.

### Add to Position

You can numerically add to Position values by using Add to Position (Modify > Add to Position). It works like editing the Position input fields in the lower-left corner of Layout, but adds to the existing values instead of replacing them.





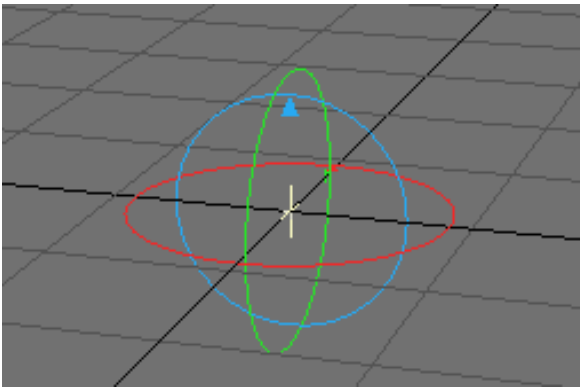
## Rotate

### Rotate

(default keyboard shortcut Y)

When you adjust rotation, in contrast to adjusting position, the action is relative to the global axes around the item's pivot point. By default, the pivot point is at the item's local Origin.

Rotation uses a similar three-coordinate system: Heading, Pitch, and Bank. (These are rotations around each of the axes: Y, X, and Z, respectively.) You can think of heading as the movement in shaking your head "no." Pitch is like the movement in shaking your head "yes." Bank is like the movement of tilting your head left and right.



Heading, Pitch, Bank

When you rotate an item, your mouse movements have the following effects:

Movement	Rotate
Left/Right LMB	Heading
Up/Down LMB	Pitch
Left/Right RMB	Bank

## Rotate Pivot

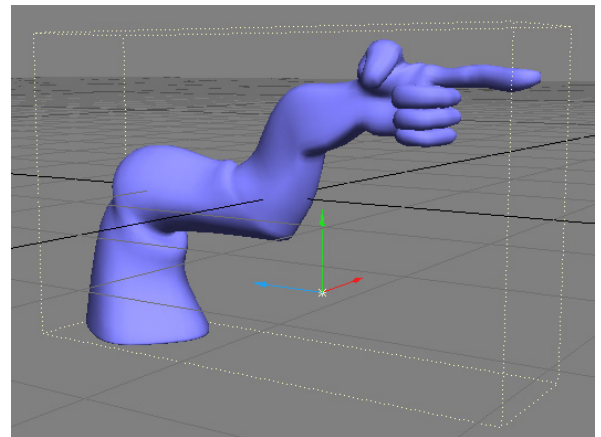


**WARNING:** Understanding pivot points is fundamental to understanding LightWave.

The *pivot point* is a point of reference used for all objects and does not correspond to any point used in an object's geometry. The pivot point is the center of position, rotation, and scaling. By default, it is located at the object's local Origin. The pivot point, a small yellow star, becomes visible when you select an object.



**NOTE:** You cannot animate the pivot point.



Pivot Point (Shown selected with handles)

### Rotating the Pivot Point

You can rotate the pivot point, which sets a starting point for rotation that is different from the default rotation, which is in alignment with world coordinates.



**HINT:** Pivot rotation is most useful when used with bones. You may want to stick with normal item rotation for other rotational needs.



**NOTE:** Unlike the Move Pivot tool, which changes the position values, when you use the Rotate Pivot tool, it does not compensate with changes to rotation values.

If you have rotated an object and wish to transfer its current rotational state to the pivot rotation, choose Setup > Orientation > Record Pivot Rotation, which runs the Record Pivot Rotation command. Then you can start all over as far as rotating the item is concerned. It will add the rotations to any existing values for pivot rotation.



## Add to Rotation

You can numerically add to Rotation values by using Add to Rotation (Modify> Rotate: Add to Rotation). It works like editing the input fields in the lower-left corner of Layout, but adds to the existing values instead of replacing them.

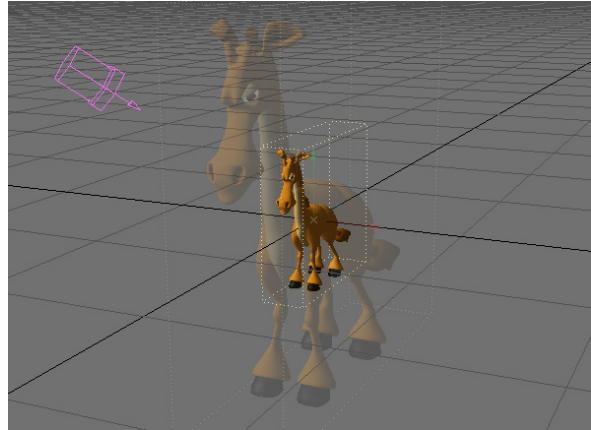
Heading	0.0	OK Cancel
Pitch	0.0	
Bank	0.0	

## Transform

### Size

(default keyboard shortcut **Shift H**)

You can scale an object (but not Distant lights, Point lights, Spotlights or cameras) using the Size tool (Modify> Transform: Size). Size scales your object proportionately along all axes around its pivot point.



NOTE: Linear and Area Lights can be Scaled.

When you Size an object, it is scaled equally along all axes. Dragging left will make it smaller and dragging right will make it bigger.

### Stretch

(default keyboard shortcut **H**)

You can Stretch an object (but not Distant lights, Point lights, Spotlights or cameras) using the Stretch function (Modify > Transform: Stretch). Stretch lets you scale each axis of an object independently around its pivot point.



NOTE: Linear and Area Lights can be stretched. Increased size for these types of lights also changes the softness of shadows.

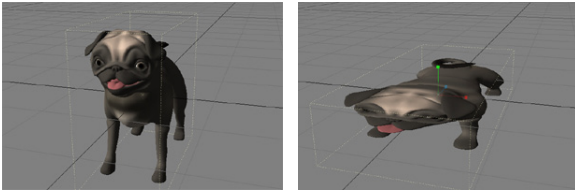
When you Stretch an object, your mouse movements affect the following axes:

Movement	Stretch
Left/Right LMB	X axis
Up/Down LMB	Z axis
Left/Right RMB	Y axis

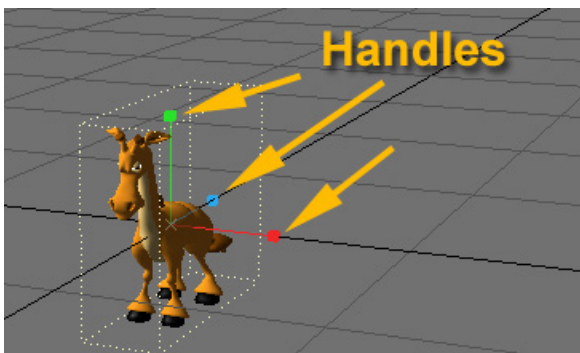


## Squash

The Squash tool (Modify > Transform: Squash) is similar to the Stretch tool; however, when one of the scale channels is modified, the other two channels are automatically adjusted to preserve the object's volume. Squash lets you scale each axis of an object independently around its pivot point.



NOTE: Linear and Area Lights can be squashed.



Use the handles shown to perform a squash operation.

## Add to Scale

You can numerically add to Scale values by using Add to Scale (Modify > Transform: Add to Scale). It works like editing the input fields in the lower-left corner of Layout, but adds to the existing values instead of replacing them.



## General

### Coordinate System

The Coordinate System setting (Modify > General: Coord System) affects the Move, Rotate, and Move Pivot Point tools.

**World** (default keyboard shortcut **Shift** F5)

World allows easy movement based on the world axes, even for items deep within a hierarchy that contain rotated parents.

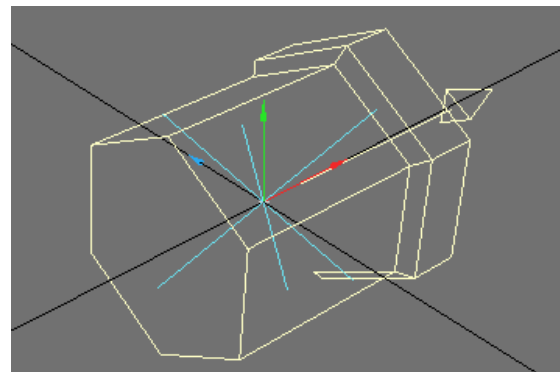
**Parent** (default keyboard shortcut **Shift** F6)

Parent is for movement based on the axes of an item's parent(s). If an item has no parent then this setting is equivalent to World.

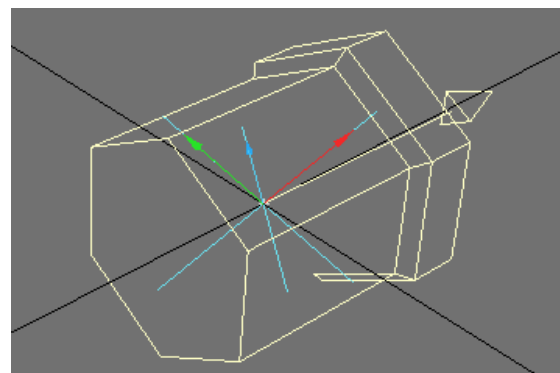
**Local** (default keyboard shortcut **Shift** F7)

Local is for movement based on an item's own axes (such as dollying a rotated camera along its view direction). For un-rotated items it is equivalent to Parent. Local can be temporarily activated by holding Ctrl.

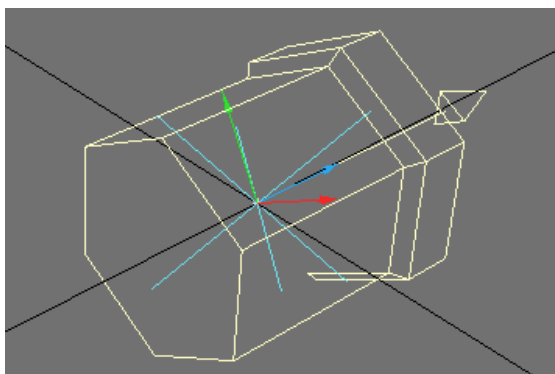
Below, we have rotated a null and then parented a distant light to it. The Show Handles option, discussed later, is also active for illustration purposes. The handles will line up with the movement axes that would be used if you dragged your mouse.



World



Parent



Local

Note how with World, the handles line up with the lines on the grid. With Parent, the handles line up with the rotated (parented) null. Finally, with Local, the handles line up with the distant light's rotation.



NOTE: The Position and Rotation Coordinate System settings are independent. To change the system, select either the Move or Rotate tool first.



NOTE: One thing to remember about Local and World rotation is that they are only for interactive manipulation. Internally, the Parent system is always used since it's the only one that can encode absolute rotation values. This will affect how an item's orientation is interpolated between two keyframes. As such, rotating the pivot point might still be useful in some situations.

## Avoiding Gimbal Lock

*Gimbal lock* normally occurs after you rotate an item 90 degrees on its pitch using the Parent coordinate system. Once this has occurred, rotating the object about either its heading or its bank axis gives you the same result—you have lost the ability to rotate about the object's local heading.

Gimbal lock is commonly a problem with bones in a hierarchy. By nature, bones have a starting position that is often rotated 90 degrees on their pitch, like bones in the arm of a human figure. You can avoid Gimbal lock by rotating using the Local coordinate system.

## Reset

You can quickly reset position and rotation to their initial states by selecting the item, activating the channel you wish to reset (i.e., Move, Rotate, Size, and other tools), and then clicking Modify > Reset. Each function must be reset individually.



NOTE: If you plan to move a pivot point, you should reset the item's position and rotation first.

Reset is not an undo feature, although it can sometimes work similarly. Resetting restores the state for the selected channels to what it was when the item was first loaded or created.



NOTE: If you have Position or Rotation axes deactivated, Reset will have no effect on those settings.

## Tools

### Light Intensity

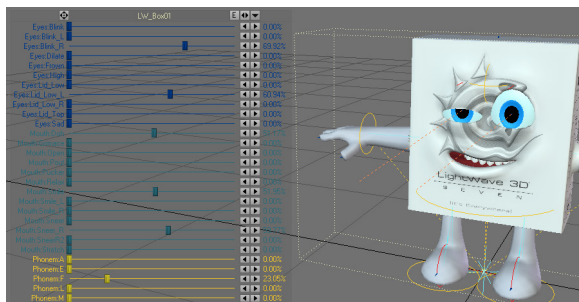
(default keyboard shortcut **Ctrl** **H**)

Selecting a light, or several for that matter, and using this tool allows you to adjust light intensity interactively by holding down the LMB and moving the mouse left and right. Feedback is given down in the bottom left corner as usual. If you adjust more than one light at once, the feedback area reports how one is being adjusted, but all that are selected are adjusted equally. Make sure you multi-select with the shift key rather than making a banding box.

### Sliders

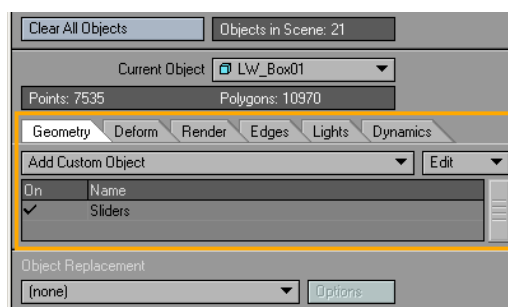
(default keyboard shortcut **Ctrl** **D**)

Sliders (Modify > Tools: Sliders) are slider gadgets that are displayed over viewports. An individual slider is tied to a specific animation channel. A slider will indicate the current value of a channel and also let you interactively adjust that channel value. Sliders are useful for all kinds of animations, especially character animations. A good example can be an animated hand, with multiple bone movements for multiple fingers.



To use sliders in a scene:

Add the Sliders custom object (Object Properties Panel) to any object.



To display and interact with the sliders, choose Modify > Sliders. The sliders for the last current object (with the Sliders custom object) will be editable when the Sliders tool is active. Keep this in mind if you have more than one object with the Sliders custom object.

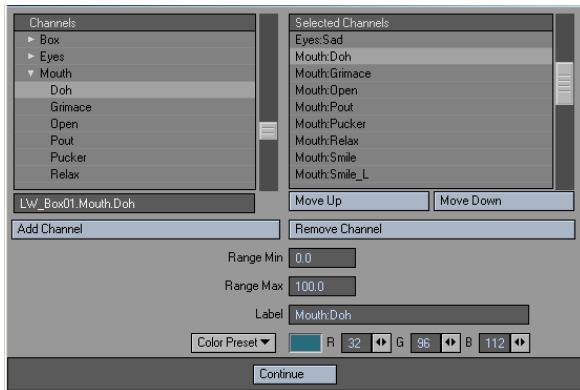


NOTE: Remember, as with all Layout tools, you'll need to select another tool (Move, Rotate, etc.) to deactivate the Slider tool.



To configure your sliders:

Open the Sliders Custom Object settings dialog from the Object Properties Panel.



The left window (Channels List) will list all of the channels in the scene. To attach a slider to a channel, simply select the channel in the left window and click Add Channel. Selected channels in the right window can be removed with the Remove Channel button.

The Range Min and Range Max settings define the interactive range of the slider. The underlying channel can go beyond these values, but the slider's range of control and feedback will be limited to this range. If the underlying channel goes outside the range, the slider value will turn red. Clicking on the slider handle will immediately change the channel to the slider's corresponding value.

The description Label will default to the channel name, but you may edit that if you desire. You can also set the color used for the slider with the Color preset pop-up menu or specific RGB values.

Working with OpenGL Sliders:



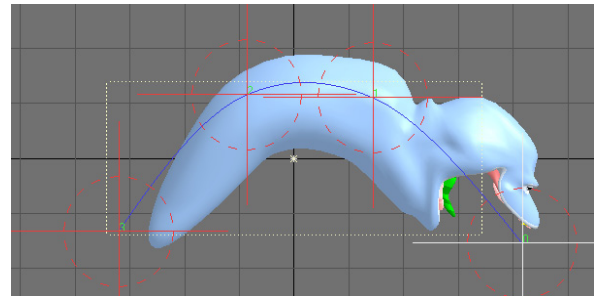
There are three controls along the top. Drag the far-left one to move the slider group. The Envelope (E) button will display the Graph Editor with the associated channels in the curve bin. The right/left arrowhead (<>) button can be dragged to scale the size of the slider group. The arrowhead button on the right will minimise/maximise the slider display.



NOTE: To simplify using sliders, use the Add Slider Bank generic plugin command (Not assigned to the interface by default). It will add a null object, apply the sliders custom object to it, and open the sliders interface, all in one step.

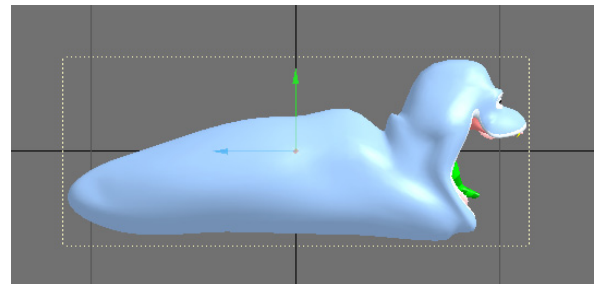
## Spline Control

The Spline Control Tool (Modify>Tools: Spline Control) is a simple way to animate hoses, tentacles, etc. using a spline with control handles. This tool will create a spline the length of the object with a user specified amount of control points that can be used to deform an object.

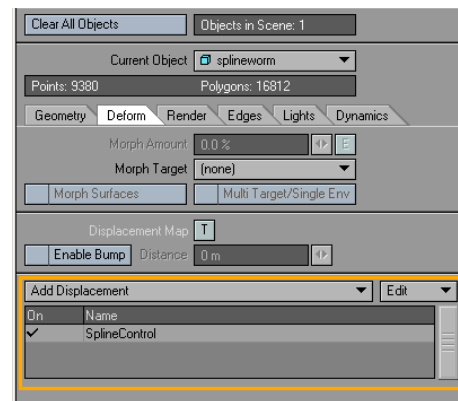


To use Spline Control:

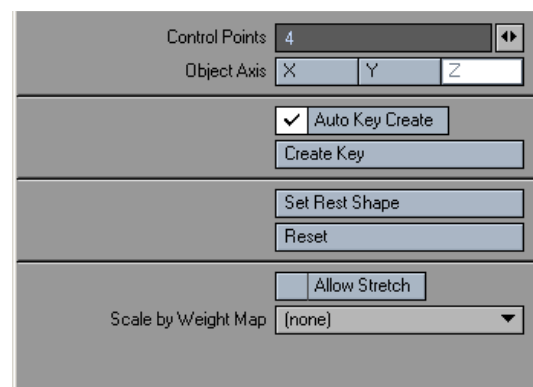
Step 1: Select the object that you would like to animate with Spline Control.



Step 2: Activate under deformations, SplineControl.



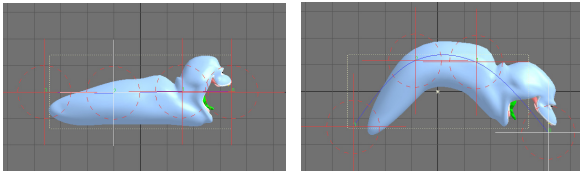
Step 3: Double Click one Spline Control to open the Options Panel.



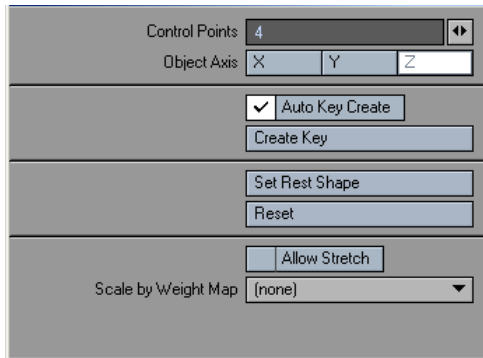


Step 4: Choose how many 'nodes' you want (you can change this later) and other options.

Step 5: Use the Spline Control button (Modify>Spline Control) to activate and move the nodes around.



#### Spline Control Options:



**Control Points** — This field will determine the number of control handles that will be generated.



NOTE: Each control point will be numbered starting from 0. This number will appear in OpenGL.

**Objects Axis** — Defines the axis the control spline will be drawn on.

**Auto Key Create** — Similar to Layout's standard Auto Key in that it will automatically create a keyframe as you move a control point. The only difference is that it will always create when on frame zero regardless of whether or not this option is active.

**Create Key** — When Auto Key Create is not active use this to create a keyframe.

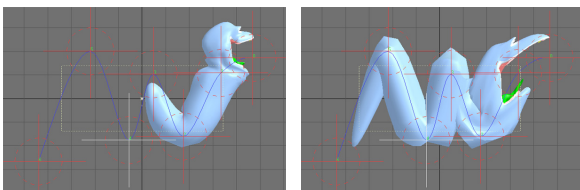


NOTE: Hitting Enter on the keyboard will not set a keyframe for a control point.

**Set Rest Shape** — Use this to change the default rest position of the control points in relationship to the object.

**Reset** — This function works like undo, however instead of just undoing the last move it will reset all control points to their rest position.

**Allow Stretch** — When this option is activated the object is able to be stretched past its original scale.



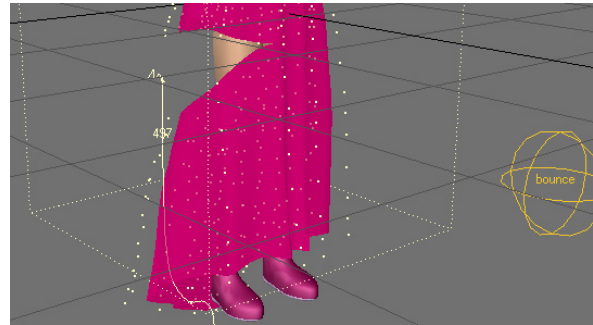
Left: Allow Stretch OFF, Right: Allow Stretch ON

**Scale by Weight Map** — This option will use the Weight Map selected and only deform that area of the mesh.

## Edit Tool

(default keyboard shortcut **Ctrl** **E**)

The Edit tool (Modify> Tools: Edit Tool) activates the ability to edit points on dynamic objects and particles. You will be able to visually see that you're in Edit Mode in the viewport. All the points that make up the object will become highlighted and all the Edit tool functions will become active.



NOTE: For more information about the Edit tools and some of its options, see the Dynamics and Particles sections.





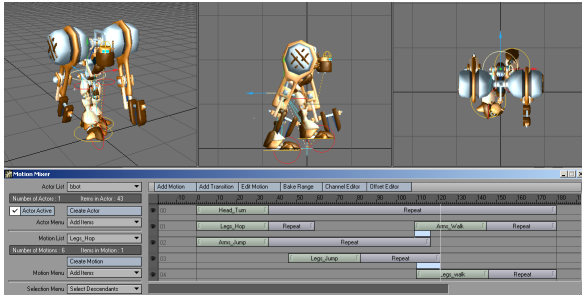
## Motions

Motion Options Panel *(default keyboard shortcut M)*

See chapter 19, page 224 for details for the Motion Options Panel.

## Motion Mixer

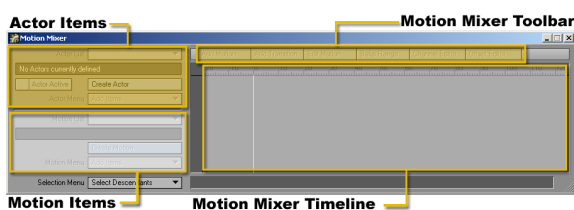
Motion Mixer was designed to bring the same concepts found in non-linear video editing to computer animation. You can store, re-use, and adjust motions for entire hierarchies. You can even mix motions together.



Motion Mixer's "actors" define objects or groups of objects, while "motions" define their animations. The Timeline gives you the ability to move, trim and scale motions together. You can add "transitions" to control exactly how motions blend together. Creating incredibly complex animations from a library of relatively simple moves is fast and easy, bringing exciting possibilities to your LightWave animations. The best way to get comfortable with Motion Mixer is to set something up yourself.

### Steps for using Motion Mixer:

Step 1: Choose Motion Mixer from the Windows drop down menu to run Motion Mixer. The Motion Mixer Panel is broken down into four areas, Actors and Motions on the left, and the Timeline and Toolbar buttons on the right. Beneath the toolbar is the familiar frame display.



Beneath the frame display are the Motion Mixer "tracks" where motions and transitions are placed. There can be any number of tracks, each of which can hold any number of motions or transitions.

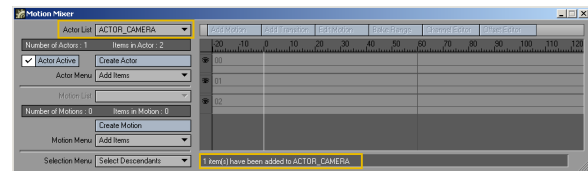
Step 2: In Layout, select the camera. Next, on the Motion Mixer Panel click Create Actor. Enter ACTOR\_CAMERA as its name and click OK.



When you create an actor, you are telling Motion Mixer which items in the scene you want it to control as a group. Scene items may be added or removed at any time and an actor can contain items of different types (i.e., objects, bones, cameras, and lights). However, an item can

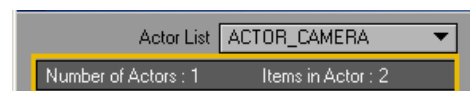
only belong to one actor. An actor may represent a character and its bone structure, a mechanical apparatus such as an aircraft's landing gear, or as in this case, an individual item such as the scene's camera.

Step 3: The panel will begin to come to life. The Actor List pop-up menu will display the "current actor", which we just created. If you had other actors, you could use this pop-up menu to change the current actor.



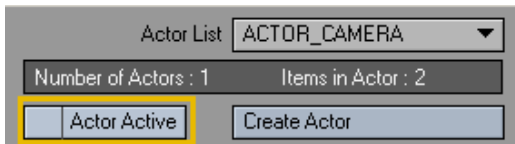
If there were multiple actors in a scene, you would choose the "current actor" from the Actor List pop-up menu. The current actor is the one you want to work with and the one whose tracks and motions are displayed on the Timeline.

The actor information display shows the number of actors in the scene and the number of items defined in the current actor.

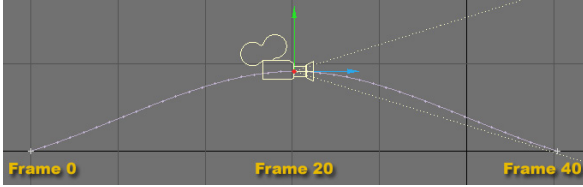


**NOTE:** Anything in Layout with an Envelope (E) button can be used by Motion Mixer. When you create an Actor, you will see that a new item has been added to the Actor, called "ExternalXChannels." This is a "virtual" item to which any channels that are not part of an items channel group are added. There is a new entry on the Actor menu, "Add XChannels," which will bring up a panel that will allow you to choose channels to add to the Actor. Double-click on a channel in the left-hand tree view to add it to the selection list.

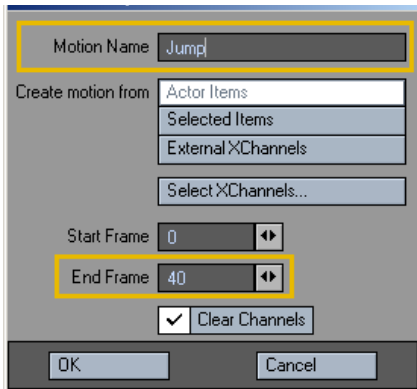
Step 4: Now, let's create a motion. Motions are segments of animation that can represent anything from a character's walk cycle to a jet fighter performing a barrel roll. When an actor is active, Motion Mixer controls it. So to create a motion in Layout, you must first deactivate the actor. Disable Actor Active.



Step 5: In Layout, keyframe the camera so that it moves straight along the Z axis from frame 0 through 40. Make a key at frame 20 that makes it jump up along the Y axis.



Step 6: Click Create Motion on the Motion Mixer Panel. Enter JUMP for the Motion Name and set the End Frame to 40, since that is the end of the motion we set up. Leave other settings at their default and click OK.



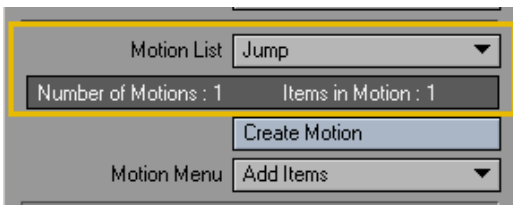
If we had used Selected Items instead of Actor Items, the motion would come from items selected in the scene.

"XChannels" are non-transformation channels (i.e., any channel other than Position, Rotation, and Scale). Select XChannels opens a new panel that allows you to specify which XChannels will be present in the motion. However, you can always toggle them on and off later in the Channel Editor.

Clear Channels removes the animation from the items in Layout — after capturing into Motion Mixer of course. Since Motion Mixer normally overrides and controls the motion, the animation is not usually needed. However, you may wish to disable Clear Channels, so you can create further motions from different parts of the same animation.

Step 7: The actual keyframed motion of the camera has now been "sucked" into Motion Mixer.

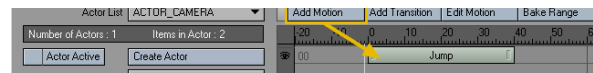
If you drag the Layout frame slider, you will notice that the camera doesn't move.



The motion information display will show the number of motions that are defined in the current actor and the number of items contained in the current motion. The Motion List pop-up menu will display the "current motion", which we just created. If the current actor had other

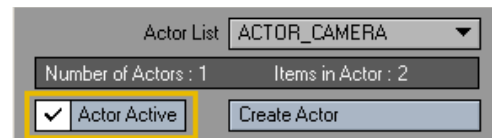
motions, you could use this pop-up menu to change the current motion — this is the motion you are working with.

Step 8: We can now add a motion to the timeline. Click Add Motion on the toolbar and then click on the 00 Timeline slot at frame 0. This will add the motion to the timeline.



If you want to cancel the operation, prior to clicking in the Timeline, click in an empty area of the panel, outside of the Timeline display. To remove a motion from the Timeline select it with your LMB and press the DELETE key.

Step 9: Enable Actor Active.



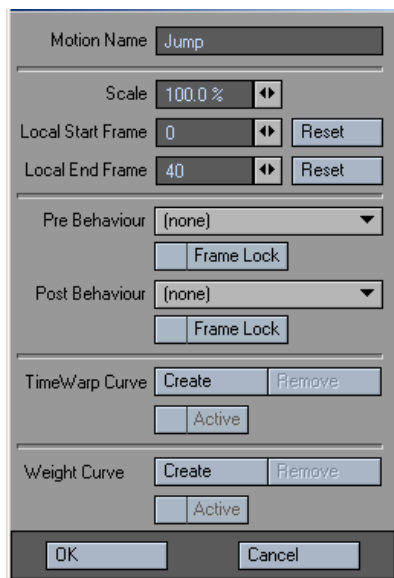
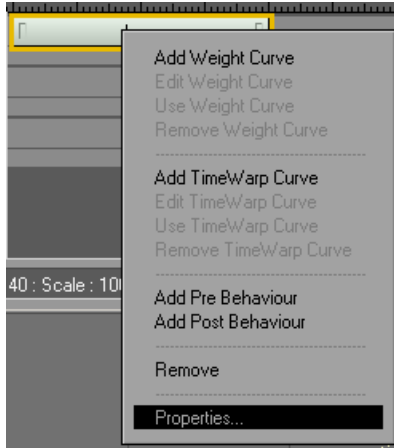
Drag the Layout frame slider. Motion Mixer is now in control and moves the camera. You can reposition the motion by dragging the center of the motion bar. You can scale the motion by dragging either end of the bar to resize it. The range and scale of the motion will be displayed at the bottom of the panel.

Step 10: Save this scene for future use!

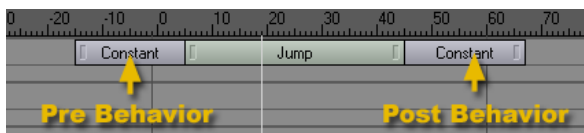


## Motion Properties

A motion's properties can be edited by right-clicking a motion in the Timeline. The motion can be renamed by simply editing the Motion Name field. The Scale field allows the motion's scale to be modified numerically. Use Local Start Frame and Local End Frame to trim the ends of a motion without changing the original data.



A motion uses "behaviours" in the same way that LightWave uses them in the Graph Editor. After you click OK, you will see any behaviours added to the appropriate ends of the motion on the Timeline. You can change the length of a behaviour by dragging its end.



## Pre and Post Behaviour Options

**Reset** — the motion value is reset to zero.

**Constant** — the values beyond the ends are equal to the first or last keyframe value.

**Repeat** — the motion repeats from the first to last keyframe.

**Oscillate** — the motion is mirrored over and over.

**Offset Repeat** — the motion repeats, but it is offset by the difference between the first and last keyframe values.

**Lock** — Normally, when you move a motion that has a behaviour, the behaviour will move along with the motion and retains its length. However, the Lock option "pins" the start or end frame of a behaviour to the Timeline. Dragging the motion will shorten or lengthen the behaviour. Eventually, it will push the behaviour once it has reached its minimum length.

**Remove** — the behaviour will be deleted/removed from the timeline.

## Weight Curve

Motions can now be assigned a Weight Curve which allow you to specify the influence a Motion has over time. These curves have the added benefit of allowing you to weight to any keyframed animation underlying the Motion in Layout. The weighting system works on a value of 1.0 indicating full influence and a weight value of 0.0 indicating no influence on the final animation. The difference with MotionMixer's weights is that as the weight approaches 0.0, and if there's underlying animation in Layout, then more of that underlying animation will be mixed into the final value:

Weight Value	Motion Influence	Layout Influence
1.0	100%	0%
0.75	75%	25%
0.5	50%	50%
0.25	25%	75%
0.0	0%	100%

If a Motion has no Weight Curve attached then it's treated as if it has a weight of 1.0. Weights can also be increased above 1.0. This is very useful for overlaying other Motions, the higher the weight value you apply the more influence that Motion will have over any others at that time (i.e., if you give a Motion a weight of 10.0, it will have ten times more influence over another Motion with a value of 1.0). When a Weight Curve has been applied, a small green bar is drawn under the Motion on the Timeline. Weight Curves can be added, disabled temporarily and opened in the Graph Editor either through the Motion Properties Panel or directly from the Timeline's context menus (see below).

## Curve Translation

Curves for Transitions, TimeWarpes & Weights are now moved and scaled to match their positions on the Timeline. This makes working with the curves far more intuitive; the frame indicator in the Graph Editor now shows the position on the curve at the current time.



## Motion Instancing

Motion Instancing is now implemented. Any number of instances of a Motion can be placed on a Track and each can have their own independent attributes (i.e., Item/Channel states, in/out points etc.)

To create an instance of a Motion, select the source Motion from the Motion Menu (this can also be a previously created instance) and click Add Motion.

Freeing the "source" Motion also removes all instances.

## TimeWarp Curves

These can be attached to any Motion (or instance) and allow you to vary the timing of the animation, you can speed up or slow down sections, go backwards or freeze time and continue. A small red bar is drawn under the Motion to signify that a TimeWarp curve is attached. These can also be temporarily disabled.

Controls to add/remove/disable TimeWarp curves are in the Motion Properties Panel and on the context menus.

## Relative XZ Custom Offset Type

This is a hybrid of the normal Relative Offset and the Absolute Offset types. All channels are evaluated as relative offsets apart from the Y axis. This allows motions to be blended relatively but retain their Y heights (i.e., you don't end up with a follow-on Motion under the floor or floating above the ground).

## Character Custom Offset Type

As the name implies this offset type is specially designed for characters and makes it very simple to align Motions automatically. Say you have a Motion where a character turns a corner and ends up facing 90 degrees, you append a Motion of a walk in a straight line. MotionMixer can now rotate that Motion so that it continues in the direction that the last Motion ended on.

The Adjust Alignment control can be used to modify the calculated rotation value allowing you to fine tune the direction your character travels.

The Compensate for Start Angle checkbox allows you to compensate for a Motion that doesn't travel along the positive Z axis. This is very handy for motion capture where motions are often not axis aligned (i.e., in situations where the performer runs diagonally to make the best use of the capture area, or probably more commonly, where people don't create their forward motion along the positive Z axis).

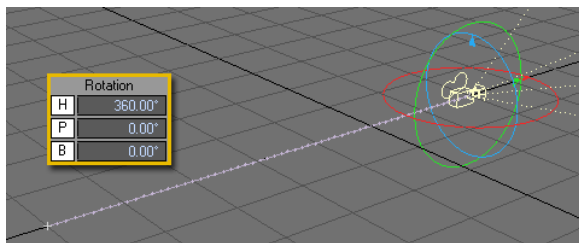
To use the Character offset type set any items that control the characters movement (e.g., pelvis bone, IK Goals etc.) to CHAR in the Offset Editor (Custom Offset type).

## Blending Motions with Transitions

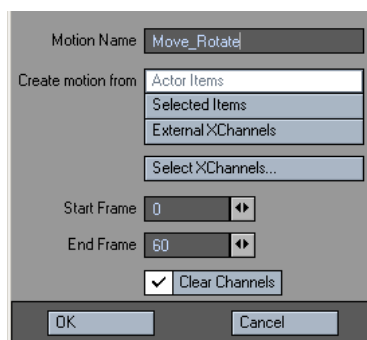
Transitions are one of the most powerful features of Motion Mixer. They allow you to blend motions together.

Step 1: Start off where the previous exercise left off. We want to create another motion so disable Actor Active.

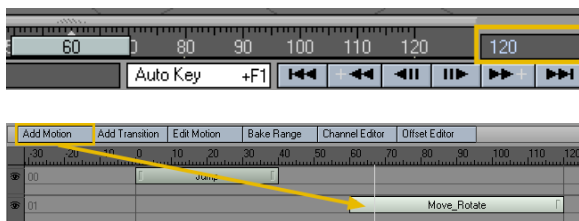
Step 2: Create a keyframe at frame 60 so that the camera moves along the Z axis, but rotates 360 degrees on its heading.



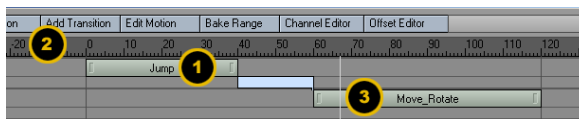
Step 3: Click Create Motion. Use MOVE\_ROTATE for the motion name. Other settings can be left at their defaults. Click OK.



Step 4: Set the scene's last frame to 120. Add the motion to slot 01 on the Timeline.



Step 5: Click on the Jump motion bar (1) and then click Add Transition on the toolbar (2). Next, click the Move\_Rotate Motion Bar (3). This adds a transition between the two motions.

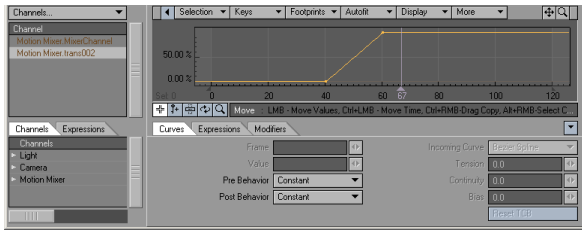


Press Layout's Play button and watch the movement of the camera. During the frames controlled by the transition (41 through 59), the camera's position is interpolated to line up with the start of the second motion. You should notice that the movement between the motions is quite abrupt. This is because it is a linear interpolation curve by default. For a smoother blend, read on.

Step 6: Right-click on the transition bar and choose Edit Transition.



This brings up the Graph Editor and the transition should be listed as an animation channel!



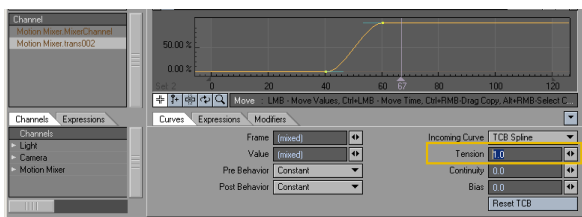
The vertical range of the curve represents the percentage of transition. For example: a value of 0% means 100% of the start motion and 0% of the end motion, 60% means 40% of the start and 60% of the end, and so on.

The horizontal axis, which normally is frames, represents how far through the transition you are—not the range of frames in the transition. For example: 0 means the start of the transition (i.e., 0%) and 100 means the end (i.e., 100%).



NOTE: Ignore the time slider in the curve window, if you're scrubbing the animation in Layout. It does not reflect the position within a transition.

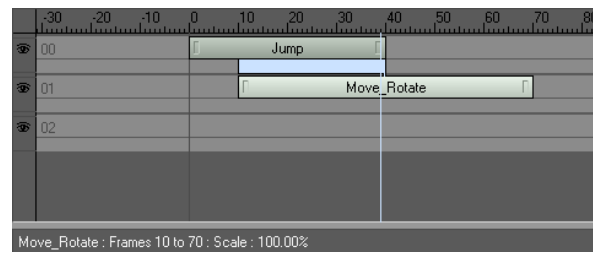
To smooth out the transition, you simply set the Tension to 1 for both Incoming Curves (they are both TCBSplines). You can do this quickly by selecting both keys and right-clicking on of them and selecting Ease In/Out from the pop-up menu.



Of course, since this is like any other curve, you can add more keys, use different Incoming Curve options, and so on. However, the first and last keys should always be left at 0 and 100. If you add new keys, only add them between 0 and 100 or you will likely get unpredictable results.

Step 7: Close the Graph Editor and watch the animation playing back. You'll see a smoother transition, but the difference in the end of the first motion and the start of the second is too great to achieve a totally smooth blend. The smoothest blends are most often achieved by overlapping the motions, so that the transition starts before the first Motion finishes and ends after the second has started.

Step 8: Drag the Move\_Rotate motion back so it starts at around frame 10. As you adjust the amount of overlap you'll see the blend become much smoother. Remember that you can drag motions while Layout is playing, so you can modify the transition overlap interactively.



NOTE: Motions can be dragged between Tracks.

## Renaming or Replacing Items in Layout

Renaming or replacing an item in Layout that belongs to an Actor will update the Actor and any motions. However, the channel names (as seen in the Graph Editor) will not be updated until the scene has been saved and reloaded. This does not affect the operation of Motion Mixer, but you should save and reload as soon as it is convenient.

## The Track Area

Clicking on the eye icon with your LMB can deactivate each track. When a track has been deactivated, all motions and transitions on the track are also disabled. This is useful for isolating a motion on another track, so that it can be worked on without the effects of any transitions or blending.



The display can be scrolled horizontally by holding down the ALT key while dragging your LMB on an empty area of the display. Scrolling the display vertically can be achieved by using your UP and DOWN cursor keys.

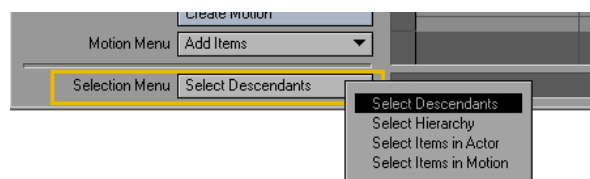
You can add or remove Tracks by using the RMB. Insert Track adds a new track to the Timeline at the end of the current list of tracks. Remove Track removes the currently selected track along with any motions on it. You can select a track by clicking on it.





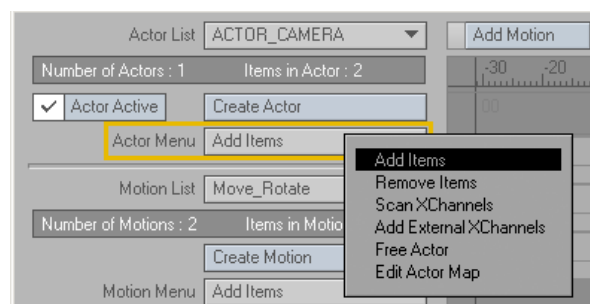
## Selection Menu

The Selection Menu can be used as shortcuts to select items in Layout.



## Actor Menu

The Actor Menu pop-up menu provides methods for working with actors and the items defined within them.



Add Items adds the currently selected items in Layout to the current actor.

Removes Items removes the items selected in Layout, from the current actor. Motion Mixer will no longer control these items. This does not clear the item from Layout, however. Clearing an item from your scene will also remove it from an Actor.

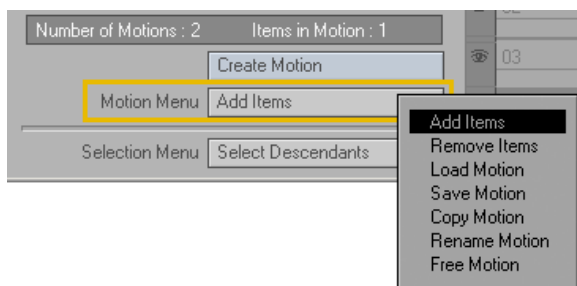
You can use Scan X Channels to add Endomorph channels, if Morph Mixer is added to an item *after* that item has been added to an actor. Motion Mixer allows you to mix all of the animation channels belonging to an item. The non-transformation channels (i.e., any channel other than Position, Rotation, and Scale) are called "X Channels." Other examples of X Channels are: a light's RGB color and Intensity channels, a camera's Zoom Factor, an object's Dissolve channel, and so on.

Free Actor removes the current actor from memory. All items contained in the actor are removed and all motions are freed.

Edit Actor Map opens the Actor Map Panel, discussed below.

## Motion Menu

The Motion Menu provides methods for working with motions and the items defined within them.



Add Items adds the currently selected items in Layout to the current motion. Use this when you need items of different types in the same motion.

Remove Items removes the currently selected items in Layout from the current motion. Any animation data for these items will not be restored.

Load Motions loads a motion into the current actor. Motion files have an .HMOT file extension and contain data for all the items in an animation. If the motion that's being loaded contains animation data for items that are different from those in the current actor, the Motion Mapping Panel will be displayed.

Save Motion saves the current motion to disk.

Copy Motion creates a copy of the current motion. You will be prompted for a name, after which the motion will be added to the Motion List.

Rename Motion allows you to change the name of the current motion. This can also be done in the Motion Properties Panel.

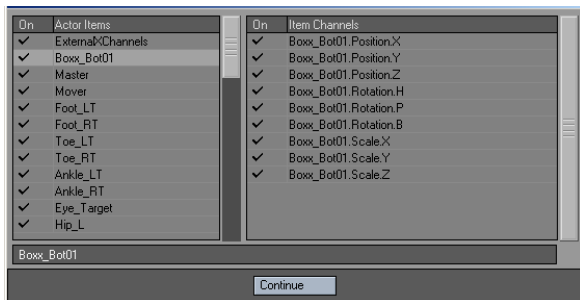
Free Motion removes the current motion from memory, all channels are cleared and the motion is removed from the actor.





## Actor Maps

The Actor Map informs Motion Mixer which channels in an item should be evaluated on playback and while “baking.” By default, all channels and all items are active.



Usually, an animation will not use all channels in all items. Having fewer channels active will improve playback performance in complex scenes and reduce the amount of keyframe data generated by the baking process. For example, in a character bone hierarchy, scaling is unlikely to be used, so these three channels can be deactivated for each item. Moreover, often only one item has any positional animation (e.g., the pelvis), so all the other items can have their position channels deactivated.

Finally, some bones, like an elbow joint, only rotate around one axis, so the other two rotation axes can be deactivated.

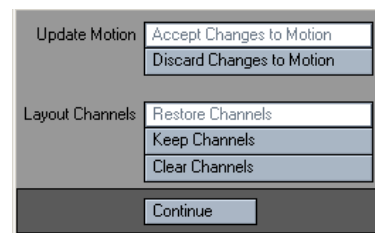
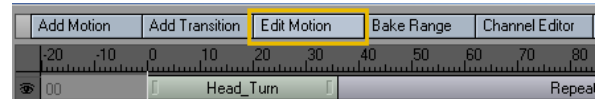
To toggle the state of an item or channel, click in the column marked On. A check mark in this column indicates that the item or channel is active. An [L] indicates that this item or channel is locked (i.e., inactive). This is a global setting and overrides any states set in the Channel Editor.



NOTE: Actor Maps are saved and loaded in the LightWave scene file.

## Editing Motions

The Edit Motion button on the toolbar places the currently selected motion back into Layout for editing. All other tracks and motions are disabled while this mode is active. Once the motion is in Layout, you can edit the animation as you would normally. When you’ve finished making changes, toggle the Edit Motion button off. This will bring up the Edit Motion dialog.

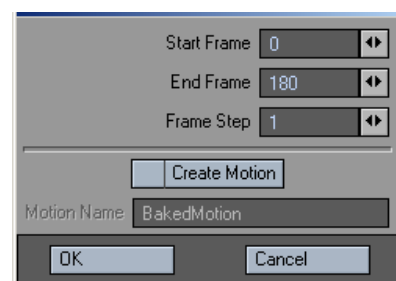
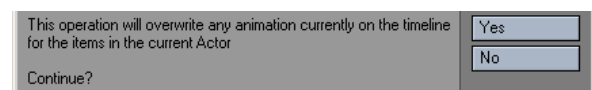


Accept Changes to Motion updates the motion with the changes you made. Discard Changes to Motion does not update the motion.

Restore Channels restores any animation that was present in Layout before editing the motion. Keep Channels leaves the motion in Layout. This is handy for creating new animations based off an existing motion. Clear Channels clears all the channels in Layout associated with the motion’s items. Motions are placed back at the frame/time from where they were created and are reset to their original (100%) scale.

## Baking Motion

The Bake Range button on the toolbar allows a range of frames to be recorded by evaluating the animation at a specified interval. This enables any combination of motions, transitions, and behaviours to be collapsed into a single motion that can be loaded back into Motion Mixer or just used in Layout.



The Start Frame, End Frame and Frame Step fields specify the range of frames that will be baked and the frequency of the evaluation. Note that Motion Mixer will always create a key on the first and last frames of the range.



NOTE: Often a Frame Step of 2 or 3 is accurate enough and much quicker to bake..



By enabling Create Motion, Motion Mixer will automatically create a new motion from the baked sequence and place it in the Motion List. The Motion Name field enables you to specify a name for this new motion. If this is disabled, the baked keyframes will be left in Layout, allowing you to perform further editing before creating a motion as described earlier.

The current Actor Map determines the items and channels that are baked. If an item or channel is locked, then no new keys will be made for it. This is an effective way of reducing the amount of new data created and optimising the resulting motion.



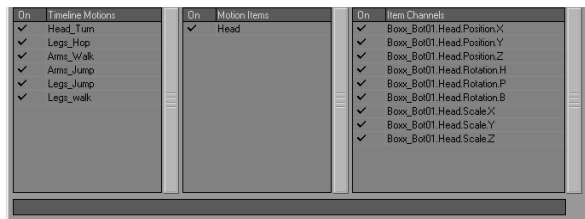
**NOTE:** Currently, you will need to bake the entire animation sequence, if you intend to render the scene with ScreamerNet. Once this is done, remove the Motion Mixer plugin from the scene and save the scene using a different name.



**NOTE:** Baking does not currently support X Channels.

## The Channel Editor

The Channel Editor gives you complete control over which motions, items and channels contribute to your animation. Open it by clicking Channel Editor on the toolbar.



The panel contains three lists. The list on the left controls the states of the motions that are currently placed on the Timeline. Clicking in the left hand On column, toggles the state of each motion. In the center of the panel is the Items list. This shows all the items that are controlled by the motion selected in the Motion List. Any or all of these items can be enabled or disabled. The list on the right shows the state of the channels for the item currently selected in the Motion Items list. If a channel is marked with the [L] symbol, that channel has been locked by an Actor Map and cannot be altered here.

The Channel Editor can be used to combine two or more motions together. For example, say you have a character walk cycle and a waving animation and you'd like to combine them to create a walking waving animation. First, you'd position each motion on the Timeline so that they play at the same time.

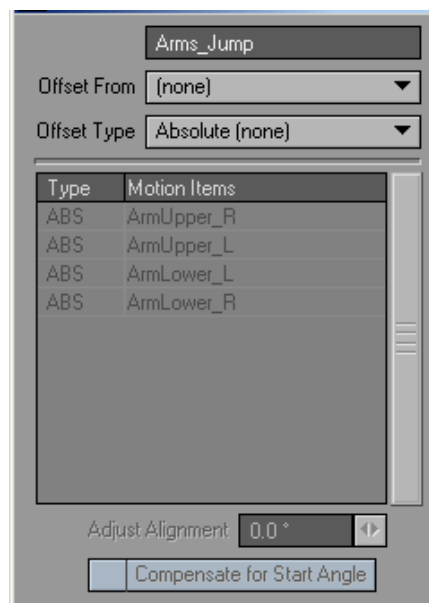
Then, open the Channel Editor to edit the state of the items in each motion. In the walk cycle animation, you'd disable all the items for the

upper body, leaving just the legs animating.

For the waving animation, you'd disable all the items relating to the character's lower body, leaving just the upper body animating. Now, when you play back the animation, you'll see the lower body animation from the walk cycle playing along with the upper body animation from the waving animation. The resulting animation can then be baked and saved for future use. Other animations could be layered on top of these, providing an easy way to add secondary motion to your work.

## Offset Editor

By default, Motion Mixer uses an absolute offset when evaluating motions. This means that each motion is evaluated in isolation from any other motion on the Timeline. Alternatively, a motion may be offset from another motion. This is called a "relative offset." If you want a motion to continue on from the point at which another motion has finished, you can use a relative offset.



*To set an offset:*

Step 1: Select the motion on the Timeline that you'd like to set the offset for and click on the Offset Editor button on the toolbar.

Step 2: Select the motion you want to create an offset from the Offset From pop-up menu and the type of offset from the Offset Type pop-up menu. Note that you cannot offset a motion from itself.

*To remove an offset:*

You can remove an offset from a motion either by setting Offset From to (none) or setting the Offset Type to Absolute.



## More on X Channels

When you create an actor, or add items to an actor, Motion Mixer looks for any non-transform channels and assigns them automatically. If channels are added to an item *after* it has been added to an actor, Motion Mixer will detect the change and add them to the actor. The exception to this is Endomorphs. Endomorph channels will be added to an actor automatically only when Morph Mixer has been added *before* the item was added to the actor. If Morph Mixer is added after that item has been added to an Actor, you must use Scan X Channels in the Actor Menu, which was mentioned earlier

Removing X Channels from an actor is achieved by either using the Remove Items entry in the Actor Menu, clearing the item from the scene, or manually removing the MM\_ChannelDriver modifier on the Graph Editor.

Custom Offset — when you add this to a Motion, you can choose, on a per item basis, whether to use Absolute or Relative offsets. This is very useful for animations containing translation. Just set the translation item(s) to Relative and the rest to Absolute. Changing the offset type is done by clicking on the column next to the item name in the Offset Editor list box, this will toggle between ABS and REL.

## Key Commands

Shift+LMB Click: Add/remove Motions to/from the selection.

Shift+LMB Drag: Moves selected Motions on the Timeline. Drag anywhere other than on a Motion.

Ctrl+Drag: Moves the entire contents of active tracks. Disable a track if you don't want to move its contents.

Ctrl+RMB Drag: Moves the contents of active tracks that have their start frames after the frame at which the drag operation is started.

Ctrl+MMB Drag: Same as Ctrl+RMB Drag, but affects only those Motions that have their end frames before the initial click frame.

Timeline can be zoomed by Alt+Ctrl+LMB dragging.

The Timeline will pan vertically during a normal Alt+LMB drag if the mouse is moved up or down more than the height of a track.

## Additional Notes

Displacement/HyperVoxel Texture Layer Position, Rotation and Scale envelopes etc. can be used. General XChannel support has been improved.

MotionMixer creates a config file (in the same directory as the LightWave configs) to store settings between sessions. Colors for Motion & Behaviour bars and the Toolbar are specified here. Other colors will be affected by any changes made to the LightWave color scheme.

Scenes containing the same Actor can be imported using Load from Scene, this creates a duplicate Actor.

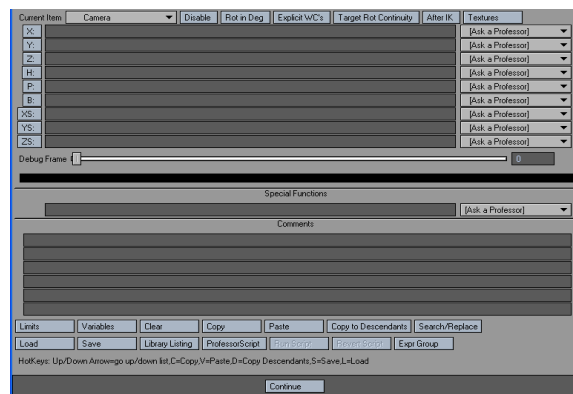
Motions on the Timeline that have been made inactive through the Channel Editor cannot be selected or moved.



## Relativity Expression Module

### What is Relativity?

Relativity is a motion and displacement plugin (applied via the Motion Options panel and/or the Object Properties panel) that allows the motions and/or deformations of one item (camera, light, object, or bone) to be related to the motions of another item via mathematical expressions. Below is a picture of the main Relativity motion plugin panel (accessed via the “Options” motion plugin button):



### Current Item

This menu allows you to bring up and edit the expressions of any item in LightWave that is currently using Relativity.

NOTE: You cannot access Relativity morph expressions from the motion panel or vice versa.

### Disable

Turning this button on will disable this instance of Relativity for this object, allowing the original keyframed motion to play through untouched. This setting is saved in the scene file for this instance of the plugin as well.

### Rot in Deg (Rotation in Degrees)

Allows rotation values to be calculated in degrees, as opposed to the default radians that LightWave uses internally. For most, it is much easier to think in degrees.

### Explicit WC's

This button forces Relativity to calculate the global position of a child object based on just the rotation, scale, and motion of each of its parents. LightWave normally supplies this information, but in the case of Full-Time IK and in cases where a motion plugin controls the motion of a parent or ancestor of the particular child object, LightWave supplies bogus world-coordinate information to a plugin. So, Relativity can go in and reconstruct the position explicitly using position, scaling and rotation data. Using this option can slightly increase the amount of time needed to perform an expression in Relativity.

### Motion Channels

It is ultimately the contents of these nine little fields that comprises all that Relativity really is. At its heart, you enter expressions defined in a simple expression language that end up relating the value of each motion channel to something else happening in LightWave. For instance, you could enter in the H slot the expression “X(light,t+2)”. What would happen then is that the expression interpreter would look at the light’s X position at 2 seconds before the time for the current frame and then pass that into LightWave as the heading rotation value for the camera. I could also make the expression 3\*X(light,t+2) and then have the X position of “light” multiplied by 3 before being passed as the heading value for the camera.

### Special Functions

This slot is used for several special functions that Relativity can perform, like targeting and dynamic parenting.

### Clear

Clicking this button will clear everything out of the current panel, allowing you to start over. If you happen to do this accidentally, click Cancel to lose the changes and then come back in.

### Copy/Paste

Copies/Pastes the expression into/from an internal buffer, all settings and variables will be copied and pasted as well.

### Copy to Descendants

Copies the current set of expressions to the descendants of an object. An example would be in setting up a motion cycle for the thigh of a character. You could then take those same expressions and copy them to the descendants of the current object. NOTE: The descendants will need to already have Relativity applied to them via their own Motion Graph panels. Relativity is unable to do this automatically due to limitations in the LightWave plugin architecture.

### Load/Save

Loads and Saves the expression settings To/From a file. The file format is just text with all the parameters listed sequentially in it.

### Search/Replace

Search for instances of a text string, and replace it with another. The search can either be case-sensitive or not.

### Comments

Put in anything you darn-well please in these slots...they are there to help you remember why you set up an expression the way you did.



## Professors

For those who feel rather math-shy, there are a number of “Professors” that allow you to set up common types of expressions. Note that next to the expression slot for each motion channel, there is an “Ask a Professor” menu choice. For most professors, it is context sensitive (i.e. for the Gears professor, the rotation used in the expression will match that of the slot selected...so the bank slot would be filled with a “B(object,t)”, the heading slot with H(object,t) and so on.). The “Dr. Follow” professor will actually place data in any slot specified and is thus not context sensitive. To use a professor, select it from the choice menu and fill in the blanks. If you’ve filled in everything correctly, your expression will be automatically set up for you. You can then take the expressions created and tweak them further.

A detailed staff listing of the professors and what they do can be found later in this chapter.

## Variables Panel

Above is a picture of the variable panel, accessible via the “Variables” button on the main panel. Relativity has a number of variables available that you can use in your expressions. Basically, each variable can be a sub-expression which then gets dropped into the grand expression for each motion channel. For instance, I could define the sub-expression  $X(\text{object1},t)*\sin(t*5)$  as a sub-expression for variable “A”. This can then be substituted into the final motion channel expressions. Say we put the expressions:

$3*\#a$

and

$2*\#a$

in the X and Y slots of Relativity...if you expanded things out, you’d end up with:

$3*(X(\text{object},t)*\sin(t*5))$

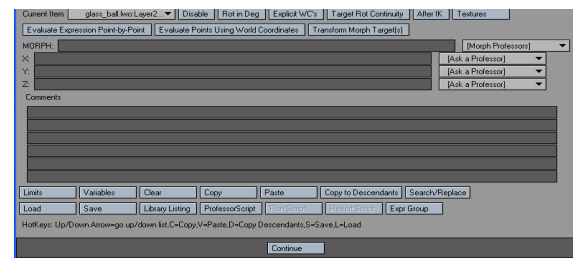
and

$2*(X(\text{object},t)*\sin(t*5))$

which are certainly more unwieldy to manage than putting the sub-expression in A and substituting in “#a” where needed.

In addition, note the two buttons, “Shift Variables Up” and “Shift Variables Down” on the panel. What these buttons do is move the variables up or down in the slots, replacing the variable names in all expressions accordingly. The reason for these buttons is if you as the expression designer decide you need some extra slots at the beginning or end of a group of expressions...finding and replacing all references to variables within the expressions as necessary.

## The Relativity Morph Plugin



Using the Relativity Morph plugin, found in the Object Properties panel Deform Tab, you can easily morph from one object to many potential targets, using complex expressions. You can even do additive, subtractive, multiplicative, and division-based morphs. There is a single expression slot “MORPH” into which you can enter any number of morph expressions (any other expressions in this slot will be ignored). Each morph expression has the following syntax:

<command character>MORPH(target,value)

where “command character” can be any of the following:

- & - replacement morph - this interpolates from the presently calculated morph into the target.
- + - additive morph - adds the values of the target points to the currently calculated morph
- subtractive morph - subtracts the values of the target points from the currently calculated morph.
- \* - multiplicative morph - multiplies the currently calculated morph by the values of the target.
- / - division morph - divides the calculated morph points by the point values of the current target

So, a typical morph expression might look like this:

&MORPH(morph1,#a)+MORPH(morph2,#b)-  
MORPH(morph3,#a+#b)

where #a, #b, etc. were derived from expressions in the variables panel.

You can enter any of the standard Relativity expressions in the variable slots, A through R, and then use these in the MORPH expressions. You can also place expressions directly in the morph value of the MORPH expression as well (e.g. “MORPH(targ1,sin(t))”)

You will also notice the options buttons at the top of the morph panel, labeled “Evaluate Expression Point-by-point”, “Evaluate Point using World Coordinates”, and “Transform Morph Target(s)”.

**Evaluate Expression Point-by-Point:** If this option is on, then the variables x, y, and z are replaced with the value of each point in the original object, as they are moved. This is useful for creating partial morphs, like effector-morphs, gradient morphs and ripple morphs. If this option is off, then x, y, and z use the keyframed motion values from the object itself for these variables.

**Evaluate Points using World Coordinates:** If this option is on, then the world x, y, and z values of each point is used in evaluating the morph. This will also cause the morph to be evaluated after bones have been applied. If this is off, then individual points are evaluated using their before-bones local coordinate.



Transform Morph Target(s): If this option is on (and it must go hand-in-hand with "Evaluate Points using World Coordinates"), the morph plugin will transform the target points by the motion of the morph target object, allowing you to have the morphed points move with their target.

It is highly recommended that you avoid using functions like DIST and ODIST in the expressions if you have any of the point-by-point options on...you will be waiting a long time for those expressions to finish with objects that have any significant point count.

There are several example scenes included in the latest example scene archive that show the new Rel\_Morph plugin in action.

In addition to the object morphing, the X, Y, and Z slots allow for expression based deformation on the X, Y, and Z coordinates of each point in your object.

## THE RELATIVITY EXPRESSION LANGUAGE

### Syntax

The Relativity language syntax is very similar to that of the NewTek "Math Motion" and "Math Morph" plugins, which ends up being a sort of LISP-ish type programming language. Basically, you enter a mathematical expression in a slot, and the plugin will evaluate it, applying the final value of that expression to that channel. In the slots on the variable panel, the expression is applied to that variable, whose value can then be placed in other expressions. Some simple expressions would be:

`sin(t)` - the sine of the current time value

`cos(x)` - the cosine of the x value of the object's own keyframed motion

`X(blah1,t)+Y(blah2,t)` - adds the X value of blah1 and the Y value of blah2

`#a*sin(t)+XS(SELF,t)` - multiply whatever's in the "A" slot by the sine of t and add the X scale keyframe value of its own keyframed motion at the current time.

`X(something,t-.5)` - gets the X motion value of object "something" where it was a half second ago (time is always in seconds)

The goal is to create an expression for each motion channel of interest. For motion channel slots left blank, the keyframed value for that channel will default through to the motion of the object.

### Object Names

In order for relativity to grab object names, full names (case insensitive) need to be supplied for all functions, object and bone names also need to be unique in order for Relativity to find the correct object. Object names that end with the ".lwo" extension need to have their names specified in functions in Relativity with their extension.

Null objects should be specified with their exact names (typically, they do not have the ".lwo" extension on their names). There are also 3 predefined object names:

camera - grabs motion info from the camera

SELF - grabs information from the item's own keyframe data

PARENT - grabs information from the object's parent.

## Variable Substitutions

If you click on the "Variables" button on the main Relativity options panel, you will notice a panel with a number of additional slots: A through R. These are there to allow the build-up of complex multi-expressions that would become quite unmanageable very quickly if all strung into one string. Relativity will scan each string, looking for a '#' character followed by a variable name and substitute that value in. For example, if I had:

X: 3\*#i

Y: 4\*#i

I: sin(X(something,t))

the value of I would be evaluated and X would be essentially set to  $3*\sin(X(\text{something},t))$  and Y to  $4*\sin(X(\text{something},t))$ . Note how using the variables also allows you to remove repetitive sub-expressions from your main expressions. In addition to that, each variable slot can take variables from the previous slots...so, we could have:

I: X(blah,t)

J: Y(more,t) + #i

K: Z(amore,t) + #i + #j

etc.

Relativity also has several special variables,

#ex

takes the numeric extension of the object's name, either the "(1)", "(2)", etc.) of a cloned object's name, or any numeric extension on an object's filename (i.e. myobject001.lwo, myobject002.lwo). This can be useful in setting up a set of expressions that automatically offset clones and numeric duplicates of objects into their appropriate position when cloned (this was used in the "newtank.lws" example scene to make all the tread links fall into their correct position. A simple example would be:

X: #ex

Now, anytime this object is cloned, the clones will each offset themselves by one meter in the X direction.

#frm

gets the number of the current frame.

#fps

gets the current frames-per-second value

#def

this passes through the default value for whichever motion channel it's on.. So, for example, if you had an object keyframed at X=2, Y=-3, Z=5, and then had the following expressions:

X: #def+2

Y: #def-6

Z: #def\*2

would change the coordinates of our object to 4, -9, 10.





In addition to the user-defined “slot” variables, Relativity has several predefined variables (these SHOULD NOT BE PRECEDED BY THE ‘#’ SYMBOL):

- x - the x position of the current item (in local coordinates)
- y - the y position of the current item (in local coordinates)
- z - the Z position of the current item (in local coordinates)
- t - the current time

## Comments

You can put whatever you want into the comment fields, which will be saved with the expressions in the scene file. In addition to that, the user can end any expression line with a comment preceded by either a double-forward-slash “//”, or a forward-slash followed by an asterisk “/\*”. Relativity will ignore anything after the comment character and will not send that part of the expression on to the parser/compiler/evaluator. An example would be the following, entered in the X field of the main panel:

```
X(object1,t) //here we're getting the X value of an object
```

or

```
X(object1,t) /*here we're getting the X value of an object
```

## Functions

Relativity supports all of the standard math functions supported by LightWave. These can be found listed in the appendix in the Reference manual. In addition, Relativity supports the following functions (NOTE: the function names are case sensitive, X is not the same as x). It should also be understood that object names containing these functions in their proper case may confuse the Relativity expression parser...so, for example, don't include an object's name as “myXobject”, instead, change the name to “myxobject” and all will be well with the world. IT IS HIGHLY RECOMMENDED THAT ALL OBJECT NAMES USED IN RELATIVITY EXPRESSIONS BE DONE IN ALL-LOWERCASE:

- X(object, time expression)
- Y(object, time expression)
- Z(object, time expression)
- H(object, time expression)
- P(object, time expression)
- B(object, time expression)
- XS(object, time expression)
- YS(object, time expression)
- ZS(object, time expression)

gets the x, y, z, heading, pitch, bank, x scale, y scale, and z scale of “object”, respectively.

- XW(object, time expression)
- YW(object, time expression)
- ZW(object, time expression)

gets the world coordinate position of “object”.

- XL(object, time expression)
- YL(object, time expression)
- ZL(object, time expression)

converts the global coordinate of “object” into the local coordinate space of the object to which we're applying Relativity.

Examples:

```
X(object,t)
```

This would get the X position of “object” at the current time.

- MOTX(object, time expression)
- MOTY(object, time expression)
- MOTZ(object, time expression)

These functions return a normalized vector representing the motion of the object. This can be useful, when combined with the vector parameters below, to gauge how much of the object's motion is forward and how much is side-to-side.

Example:

- X: MOTX(object,t)
- Y: MOTY(object,t)
- Z: MOTZ(object,t)

would cause our Relativity instance to “point” in the direction of motion of “object”.

- UPX(object, time expression)
- UPY(object, time expression)
- UPZ(object, time expression)
- FORX(object, time expression)
- FORY(object, time expression)
- FORZ(object, time expression)
- RITX(object, time expression)
- RITY(object, time expression)
- RITZ(object, time expression)

These are the Relativity vector parameter functions, and return a normalized vector representing the orientation of “object”'s local X, Y, and Z axes in global space (UP = Y, FOR=Z, RIT=X). See the included example scene “mo\_parms.lws” to see these in action.

- OBJDIZ(object, time expression)

evaluates the dissolve factor of “object”, which can then be used in your expressions.

Example:

```
OBJDIZ(myobject,t)
```

will evaluate the dissolve amount for “myobject”.



BONERX(bone)  
 BONERY(bone)  
 BONERZ(bone)  
 BONERH(bone)  
 BONERP(bone)  
 BONERB(bone)  
 BONERL(bone)

Get the rest X, Y, Z, heading, pitch, bank, or rest length of "bone".

Example:

BONERX(bone)+BONERY(bone)

adds the X and Y rest positions of "bone" together.

CAMZOOM(time expression)  
 CAMHORZ(time expression)  
 CAMVERT(time expression)  
 CAMFL(time expression)  
 CAMFD(time expression)  
 CAMFSTP(time expression)  
 CAMBLR(time expression)

Gets the zoom factor, horizontal view angle, vertical view angle, focal length (in millimeters), focal distance, f-stop, and motion blur factors, respectively, for the camera at time "time expression".

Example:

Z: Z(CAMERA,t)+CAMFD(t)

would position a null right at the focal point of the camera, provided it's looking down the Z axis and unrotated.

LIGHTCON(light,time expression)  
 LIGHTEDG(light,time expression)  
 LIGHTRED(light,time expression)  
 LIGHTGRN(light,time expression)  
 LIGHTBLU(light,time expression)

Gets the cone angle, soft edge angle, red color, green color, blue color, respectively of "light" at time "time expression".

Example:

LIGHTCON(light,t)

would get the cone angle of "light" at time "t".

XCYCLE(object, driving expression, start time in seconds, end time in seconds)

YCYCLE(object, driving expression, start time in seconds, end time in seconds)

ZCYCLE(object, driving expression, start time in seconds, end time in seconds)

HCYCLE(object, driving expression, start time in seconds, end time in seconds)

PCYCLE(object, driving expression, start time in seconds, end time in seconds)

BCYCLE(object, driving expression, start time in seconds, end time in seconds)

XSCYCLE(object, driving expression, start time in seconds, end time in seconds)

YSCYCLE(object, driving expression, start time in seconds, end time in seconds)

ZSCYCLE(object, driving expression, start time in seconds, end time in seconds)

- cycles the object's motion from the start time to the end time... whenever the decimal portion of the driving expression is 0, the position of the object at the start time is copied...if it is .99999, the position at the end time is copied, if it is 0.5, the position halfway between the start and end times is returned, etc.

Example:

XCYCLE(SELF, t, 0, 0.333)

This would cycle the X values between 0 and 10 frames (frame 10 = 0.333 seconds), repeating the cycle each time the time value ticks off a second.

XMINPATH(object, following distance, time expression)

YMINPATH(object, following distance, time expression)

ZMINPATH(object, following distance, time expression)

HMINPATH(object, following distance, time expression)

PMINPATH(object, following distance, time expression)

BMINPATH(object, following distance, time expression)

XSMINPATH(object, following distance, time expression)

YSMINPATH(object, following distance, time expression)

ZSMINPATH(object, following distance, time expression)

- gets the motion value at a minimum path distance of "following distance" behind the object's position at time t.

Example:

XMINPATH(leader, 1, t)

This would get the X coordinate of object "leader" 1 meter in path-length behind its current position.

TARGET(object, time expression)

entered in the special expression field for the object, target will target the heading and pitch of the item toward the target object, pointing the local Z axis at the targeted object.



Example:

```
TARGET(object, t)
```

would target the position of “object” at time t.

```
MATCH(object, time expression)
```

this function will match an unparented object to correspond with the object listed in the MATCH function, no matter how deeply “object” is buried in a hierarchy (including bones and child bones). Useful for such things as attaching objects to bones properly. MATCH is a special function, and as such, should go in the Special Functions slot.

Example:

```
MATCH(footbone, t)
```

This would match the position of our object to that of “footbone” at time t.

```
GAP(object1, time expression1, object2, time expression2)
```

this function measures the gap between the global position of object1 at time “time expression 1” and object2 at “time expression 2”

```
DIST(object, time expression)
```

gets the total path distance traveled (in global coordinates) from time 0 to the time specified.

Example:

```
DIST(whatever,t-0.5)
```

Gets the total path distance traveled by “whatever” from time 0 up to 15 frames ago.

```
ODIST(object,time expression)
```

this returns the oriented distance traveled by the object, with the orientation being determined by the direction of travel and where the object’s own local Z axis is pointing. Using this distance function, it is possible for an object to “walk” forward and backward, stepping properly as it does so. The original DIST function did not take object orientation into account and will return the total path distance traveled by an object regardless of the direction of the path motion with respect to its orientation.

Example:

```
ODIST(PARENT,t)
```

Gets the oriented distance traveled by the parent of our object.

```
MOVEVEC(velocity,time expression)
```

Entered in the Special Functions slot, this will move the object along the Z direction (wherever this happens to be pointing) at the specified

velocity (in meters/second). Using this function, you can simply rotate your object and have it smoothly fly wherever you need it to go. This function is perhaps most useful for situations where you have the rotation of an object controlled by some sort of “virtual steering mechanism” (like a virtual joystick or steering wheel) and want the object to respond to the rotation changes by moving in the new direction.

Example:

```
MOVEVEC(X(controller,t), t)
```

this moves the object by the velocity indicated by X(controller,t), incorporating the rotations of our object up to the current time to determine its current position.

```
SUMX(object,time expression)
```

```
SUMY(object,time expression)
```

```
SUMZ(object,time expression)
```

```
SUMH(object,time expression)
```

```
SUMP(object,time expression)
```

```
SUMB(object,time expression)
```

```
SUMXS(object,time expression)
```

```
SUMYS(object,time expression)
```

```
SUMZS(object,time expression)
```

this function adds the value of the object’s respective motion channel incrementally, once per frame. So, for example, if you have an object keyframed to X=0.5, at frame 1, SUMX would equal 0.5, at frame 2, SUMX would equal 1.0, etc. To use the SUM functions as a pseudo-time function (for throttle type controls), you will probably have to multiply the SUM expression by 1/(frames per second rate), i.e. 1/30 or 1/24, etc.

Example:

```
SUMX(blah, t)
```

this would sum the X component of object “blah” from time 0 to the present time.

```
IF(expression1 <comparison character(s)> expression2)
```



the IF function will evaluate expression1 and expression2, compare them using the <comparison> character(s) and evaluate to 1 if the comparison is true, 0 if it is false. The following comparisons are supported by IF:

- < - less than
- > - greater than
- <> - not equal to
- <= - less than or equal to
- >= - greater than or equal to

Example:

```
IF(X(blah,t) < 0)
```

this will evaluate to 1 if the x coordinate of "blah" is less than 0, and will evaluate to 0 if the x coordinate of "blah" is greater than or equal to 0.

```
AND(expression1,expression2)
```

Performs a logical AND between two expressions, if expression 1 and expression 2 are BOTH not 0, then the AND function evaluates to 1, if either or both are 0, then the AND returns 0.

Example:

```
AND(IF(X(blah,t)<0),IF(Y(blah,t)>0))
```

would return 1 if the X value of "blah" is less than 0 and the Y value of "blah" is greater than 0, otherwise, the AND would return 0.

```
OR(expression1,expression2)
```

Performs a logical OR between two expressions, if either or both expression 1 and expression 2 are not 0, then the OR will return 1. If they are both 0, the OR will return 0.

Example:

```
OR(IF(X(blah,t)<0),IF(Y(blah,t)>0))
```

would return 1 if either the X value of "blah" is less than 0, or the Y value of "blah" is greater than 0, or both. If neither is true, then the OR would evaluate to 0.

```
NOT(expression)
```

NOT will return 0 if the expression's value is not 0 and 1 if it is zero, i.e. a logical NOT operation. This is most useful for negating IF() expressions.

Example:

```
NOT(IF(x>0))
```

will return 1 if the x value of our item is not greater than zero, and 0 if it is greater than 0.

```
COND(expression <comparison character(s)> expression,trueresult,falseresult)
```

this function implements a conditional expression, basically giving the user an IF...ELSE kind of logical decision-making. The first part is identical to the format of the IF statement, this is followed by an expression to evaluate to if the condition is true (i.e. "trueresult"). If the condition is false, then "falseresult" is returned instead.

Example:

```
Y: COND(X(self,t)<=0,1,Z(object,t))
```

in this example, if the X position of the object itself is less than or equal to zero, COND will evaluate to 1. If it is greater than zero, the COND expression will evaluate to the Z value of "object".

```
FT(frameenum)
```

converts a frame into a time value in seconds (as Relativity equations use seconds to calculate the time). NOTE: frameenum must be a constant value...it cannot be an expression.

Example:

```
FT(30)
```

would convert to 1.0000, or 1 second, in the typical 30 fps scenario.

```
EVTIME(expression)
```

this function is a very powerful one, using it, you can set up an animation to play back when a specific trigger condition occurs. The way this function works is it will return a time value, indicating the last time that "expression" changed from 0 or less than zero, to greater than zero, most typically done by some form of IF statement. As long as "expression" remains greater than 0, EVTIME will keep ticking off time. Whenever the expression is 0, or less than zero, EVTIME will return 0. EVTIME will also return 0 if the expression starts out as true (i.e. greater than zero) without having switched from a false condition to a true condition. NOTE: EVTIME will only evaluate "expression" at frame boundaries, not at interframe time values in order to save calculation time. If "expression" changes from false to true at an interframe time, the time at the next frame will actually start the count.

Example:

```
Y(SELF,EVTIME(IF(Y(botfoot,t) < 0.1)))
```

This would start playing back the animation of the object itself once the Y value of botfoot gets reasonably close to zero. Something like this could be used to simulate camera shake whenever a bot's foot hit the ground.

another Example:

```
A: EVTIME(IF(GAP(object,t,sensor,t) < 1))
B: COND(#a <= FT(5), #a, FT(5))
C: EVTIME(IF(GAP(object,t,sensor,t) > 1))
D: COND(#c > 0,FT(5)+#c,#b)
X: X(SELF,#d)
Y: Y(SELF,#d)
Z: Z(SELF,#d)
```

In this example, say we have a door keyframed to open from frames



0 to 5 and keyframed to close from frames 5 to 10. In this example, an event trigger is thrown by A when object and sensor come within one meter of each other, causing the open animation to play. However, we don't want the close animation to occur until our object gets far enough away. So, the B expression limits the playback of A to frames 0 through 5 by returning the time value for 5 frames if A gets bigger than that. The C expression is set up so that it's triggered when the object moves further than a meter away. Finally, the D expression is set up to return the C value if it has a value, and the B expression otherwise. The net result of all this gibberish is that when "object" gets close to "sensor" the doors fly open. When it moves away from "sensor", they fly shut again.

EVONE(expression)

This function is the same as EVTIME, except it doesn't reset when the condition becomes false again. EVONE remains triggered no matter what happens to the tracking event.

VX(object,t)

VY(object,t)

VZ(object,t)

VH(object,t)

VP(object,t)

VB(object,t)

VXS(object,t)

VYS(object,t)

VZS(object,t)

computes the directed velocity of an object along the appropriate motion channel. NOTE: This value is sensitive to direction, so an object moving in the positive direction at the same speed as one moving in the negative direction would have opposite X velocities, one positive and the other negative, respectively. NOTE: All velocities are in meters per second (except rotational velocities which are in degrees per second if the "Use Degrees" button is on, in radians per second if "Use Degrees" is off), if you need a different velocity measurement, such as miles per hour, you will need to multiply the velocity value by the appropriate constants.

Example:

VX(object,t)

gets the X velocity of our object at time t (in meters per second)

SPEED(object,t)

SPEEDW(object,t)

gets the speed in local coordinates or world coordinates, respectively, of the object at time t. Speed is irrespective of direction, and takes all axial motions into account.

Example (a simple speedometer):

B: 180\*SPEED(car,t)

AX(object,t)

AY(object,t)

AZ(object,t)

AH(object,t)

AP(object,t)

AB(object,t)

AXS(object,t)

AYS(object,t)

AZS(object,t)

computes the acceleration of an object along the motion channel. In this case, acceleration is direction insensitive, returning the change in the absolute value of the velocity on the channel in question.

Example:

AH(object,t-0.1)

this would get the heading velocity of "object" at .1 seconds behind the current time.

SPDCHNG(object,t)

SPDCHNGW(object,t)

gets the change in speed over time of an object at time t, in local or world coordinates, respectively. This can be considered akin to acceleration, but in all directions, rather than just one axis.

Example:

Y: 30\*SPDCHNG(myobject, t-X(timecontrol,t))

This would measure the rate of speed change for object "myobject" at the current time minus the X value of "control" (in seconds).

ABSDEG(expression)

converts a rotation value in "expression" into an absolute degree value between 0 and 2\*PI radians, or between 0 and 359 degrees if the "Use Degrees" button is checked on. This can be useful when you have something continually rotating, and yet you want to create a behavior that works between a set of rotation values, like bones placed in a tire to do tire-bulge.

Example:

ABSDEG(P(obj,t))

would have the following values with corresponding values for the pitch of "obj".



Value of Pitch of obj    Value of ABSDEG

30 degrees	30 degrees
360 degrees	0 degrees
740 degrees	20 degrees

FRAC(controller object, # of frequencies)

gets a fractal noise value (a single value between 0 and 1.0) at the global position of an object, using the controller object to determine the position, rotation, and scale of the fractal texture. The way this works is as follows:

Relativity will grab the global position of an object and transform that coordinate into the coordinate space of the controller null. - Using this information, and the # of frequencies value, Relativity will apply a Perlin fractal turbulence function to the coordinate and wrangle out a single value between 0 and 1 for that point.

Example:

`X: X(SELF,t) + 0.5*FRAC(fracnull,3)`

this will add a bit of fractal noise to the X position of our object

Tips for using Fractal noise:

1. Since the FRAC function will only return a single value for the coordinate of the item involved, you may want to use several fractal controllers to apply different fractal values to different motion channels of an item to shake things up a bit more.
2. The fractal controller null determines the size, position, and rotation of the fractal texture permeating throughout space. So, for example, you have an object sitting still, but you want it to wiggle a bit, no problem, move or rotate the fractal controller object(s) and the fractal texture at the object's location will change. Another example: Say you want an object to bump around very quickly, changing values dramatically at almost every frame, just scale the fractal controller null to a very small size, and you'll reduce the size of the fractal "lumps" causing the texture to vary much more quickly from point to point. Conversely, larger scale values for the fractal controller will cause the variation to be much more smooth between points.

Remembering these tips can give you a high degree of control of the fractal randomness.

`NOISE(expr)`

returns a noise value based on the value of "expr". Basically, you feed it a value, and out pops a fractal noise value.

Example:

`NOISE(t)`

will return a noise value based on the current time.

`NOISE(3*t)`

will return a more "jagged" noise value.

`BLEND(value 1, at point 1, value 2, at point 2, current position)`

`TBLEND(value 1, at point 1, value 2, at point 2, current position)`

Blend can be used to linearly blend between values 1 and 2, with the result of blend equaling "value 1" when the "current position" expression is at "at point 1", and will return "value 2" when the "current position" expression is at "at point 2". BLEND basically sets up a linear interpolation relationship between the values, and will over-interpolate if "current position" is less than "at point 1" or is greater than "at point 2"...



NOTE: over-interpolation is not necessarily a bad thing, as many mechanical relationships can be set up by knowing relationships between various values at two distinct points, which can be used to define behavior outside of the range of those two points. If you desire the blend to stop when the current position goes outside the range specified by "value 1" and "value 2", use TBLEND (short for Truncated-BLEND) instead.

Examples:

`BLEND(0,3,5,6,X(blender,t))`

this would blend between the values 0 and 5, with 0 being returned by BLEND when X(blender,t) is 3, and 5 being returned when X(blender,t) is 6. Note: if X(blender,t) were 9, then the blend function would return 10 for its value, as it overinterpolates based on the value of X(blender,t). If this were a TBLEND instead, then the maximum value that the BLEND function would ever have would be 5, no matter how high X(blender,t) got.

`BLEND(X(obj1,t),0,X(obj2,t),1,X(blender,t))`

this would blend from the X motion channel of obj1 when X(blender,t) is 0 to the X motion channel of obj2 when X(blender,t) is 1. So, basically, X(blender,t) can be used as a toggle to switch the motion of our item from obj1 to obj2.





## The Relativity Staff Directory

The following information will give you an overview of the professors in Relativity and what they do.

- Dr. Item Picker
- Dr. Equation Maker
- Dr. Wheel Rotater
- Dr. Follower
- Dr. Oscillator
- Dr. Delayer
- Dr. Gear Grinder
- Dr. Cycler
- Dr. Motion Blender
- Dr. Blend Machinist
- Dr. Dist Maintainer
- Dr. Event Maker
- Dr. Camera Shaker
- Dr. Morph-o
- Dr. Target
- Dr. Matcher
- Dr. MoveVec
- Dr. Snake Maker

Most of the professors in Relativity are accessed from the menu next to an expression slot on the Relativity panel. What they do, is allow the point-and-click automated set-up of many commonly used expression set-ups. It is quite likely that this list will expand in future releases of Relativity. One thing to note about the professors: some professors are context sensitive, they will only set-up the expression for one motion slot at a time, and may change how they set up the expression based on which motion slot they're being run from. Other professors work on a more global level, and can set up multiple motion expression slots all at once. A few will even commandeer some variable slots. This will be noted as we go through the documentation, professors that set up multiple slots will have the statement "Multi-slot professor" before their description.

### Dr. Item Picker

Dr. Item Picker is a simple professor panel that will blast an object name onto the end of whatever you are typing. So, for instance, if you were typing:

2+X(

and then forgot the name of the object you wanted, you could use Dr. Item picker to pick the name you're interested in...so you would end up with:

2+X(mynull

and then you could add the ",t)" to finish off the expression. If you need to insert an object name into an existing expression, like:

2\*sin(XW(blah,t))

and realize that you forgot your object name and need to replace "blah" with the correct name, use Dr. Equation Maker below.

## Dr. Equation Maker

This professor is the most powerful, complex, and yet easy-to-use in your arsenal. It turns the task of hand-building an expression into an exercise in point-and-click. Now, whether or not your expression does what you expect, that's up to you, but this professor will do all it can to help steer you in the right direction. On this panel, you have a total of eight menus that allow you to pick eight different possible elements that make up your expressions: LightWave's math constants and functions, Relativity's arsenal of expression functions, Rel's special functions (for use only in the "Special Function" slot), Object names, bone names, light names, and special reserved Relativity objects (PARENT, SELF, camera, etc.).

To use this professor, simply select what you want from a menu and click it into place. As you select items, the text control at the top will give you a description of the choice you've made, to make sure you have what you're looking for...in this sample image, X(obj,t) has been selected, and the description tells you what this Relativity function does. If you click "Add Choice to Expression", the professor will append your choice to the current expression in the edit field.

You can also tell this professor that you want to insert a choice into the middle of your current expression, rather than just at the end. By placing an '@' character wherever you want an insertion point, you can make the professor insert your next choice where the '@' is, rather than at the end of the expression. So, we could start by selecting 'X(obj,t)' and clicking that into place. Then we could edit out the 'obj' and replace it with an '@'. Finally, we could select the actual object name we were interested in from the object menu, say it was called 'heavynull'. Once we click the "Add Choice to Expression" button, 'heavynull' will replace the '@' character, turning our expression into 'X(heavynull,t)'.

### Dr. Wheel Rotater

This professor has a simple task, making your object rotate as it tracks the motion of some other object. To use it, pick the object that will drive this rotation (this will most often be the direct parent, or an otherwise ancestor of your object). However, there is nothing that can stop you from letting the object's own motion drive it's own rotation. Here is a field-by-field break-down of the panel:

- Object Driving the Rotation: This is the object whose motion will cause the rotation of our current object to occur.
- Pick Vehicle: Brings up the object picker panel to get a point-and-click selection of the object you want.
- Diameter: Here you enter the diameter (i.e. the total distance across) of the wheel.



## Dr. Follower

### Multi-slot professor

This is the professor for setting up following motions of all kinds. You can set up many kinds of relationships, like a delayed following of another object, or have an object follow at a specified distance. You can even set up an offset-following motion where an object or group of objects will follow a leader, yet maintain a respectable distance from it. Here is a break-down of the fields:

- Object to follow: This is the object we want to follow...this is hopefully not a hard concept to grasp
- Pick follow object: allows point-and-click selection of object
- Channels to follow: allows you to determine which channels we're going to follow on...this will set up multiple expression slots at one time.
- Offset Motion by Difference at Frame 0: If this button is clicked on, then this object will follow its leader in a "flocky" sort of way...so the follower object will move with the leader, but maintain its relative position with respect to the leader.
- Follow at a specified distance: This slot, if it contains a value, specifies that we want to follow at a specific path distance behind the lead object. This is useful for things like roller coasters, trains, etc. etc. where you need successive cars to maintain an exact distance behind the leader or behind the car in front of it. The distance will be maintained no matter how fast the lead object moves.
- Follow with a constant time delay: This slot, if filled, will cause the follower object to either lag behind or anticipate the motion of the lead object by a specified time delay.
- Optional Time Delay Control Object: This slot will allow you to specify an object (often a null) as a delay controller.
- Channels to Control: Select the channel(s) for the delay object to control. If you only select the X channel, the X channel of the object will control the delay on ALL THE MOTION CHANNELS of our follower object. Otherwise, the X channel of the delay control will determine the X delay, the Y channel will determine the Y delay. Unclicked channels will not have the delay controller specified in them.
- Use Object Extension for Follow Delay: This will multiply the object's numeric extension or clone number (i.e. the (1) in 'null (1)' or the 3 in obj003.lwo) by the delay factor, either the time delay or the distance delay. This way, you could set up an expression for one object and copy it verbatim to other objects and have them lock into the right place behind the leader.
- Extension Offset: This is the offset for the extension...for instance, if you want object001.lwo to not have any delay, but object002.lwo to delay by one delay factor, you could make the offset 1, this would zero out any delay for object001 but turn it on for subsequent objects.

## Dr. Oscillator

### Multi-slot professor

This professor is the one for setting up any and all oscillating motions (i.e. objects swinging back and forth, bobbing up and down, wiggling around, weaving side-to-side, etc.). The setup of this panel is quite simple:

- Channels to wiggle: Select any and all channels that you want to oscillate.
- Wiggles per second: This determines the frequency of the oscillation.
- Frequency Controller: You can optionally have a frequency controller instead, it's set up so the X channel of the controller object controls the frequency (this can always be edited after the fact to a different channel).
- Wiggle Intensity: Determines the intensity of the oscillation.
- Intensity Controller: You can optionally have an intensity controller whose X channel controls the oscillation intensity.

## Dr. Delayer

### Multi-slot professor

Dr. Delayer allows you to delay an object's own motion by an offset in time. This is useful for moving the dreaded "locked 0 keyframe" to another time, so cyclic motions can start off at frames other than 0. It's also possible, using a delay controller, to soften acceleration and deceleration much more than when using just tension on your spline values. Here's a breakdown of the fields on this professor:

- Fixed Time Delay: Enter a value (in seconds, or add a 'F' at the end to specify frames) and your motion will be delayed by that amount.
- Delay Controller: Choose an object to control the amount of the delay, which makes the delay factor animatable.
- Channels of the controller object: This specifies which single channel of the delay controller will control the delay. You can also choose "all channels" to get a channel-by-channel delay control.
- Delay on which channels: Pick the channels to set up a delay on.
- Tie Delay to Object Extension: Use the object number extension in the expression as a multiplier by the delay factor. This is useful for setting up a sequence of numbered objects to delay by specified amounts.
- Extension Offset: This value is subtracted from the object extension before applying it to the expression.



## Dr. Gear Grinder

Dr. Gear Grinder is great for relating the rotation of one object to the rotation of another. You pick the controlling gear and then tell it the radius or diameter of each gear (note: you must use either radius or diameter values for both slots). If you are actually setting up gears, you could specify the number of teeth for each gear instead of radius.

## Dr. Cycler

Dr. Cycler is the ultimate professor for setting up all sorts of nifty cyclic relationships in LightWave. Before you run the professor, though, you need to have a cyclic motion keyframed for your object. Dr. Cycler will take that cyclic motion and figure out a way to apply that motion over-and-over-again (i.e. cyclically) over the course of your animation. Here's what all the fields mean:

- Motion Channel to Cycle on: The motion channel the cycle will be applied to.
- Object to Pull the Cycle from: You can select any object as the source of the cycle. In many cases, you would select "SELF" to choose the object's own keyframed motion as the basis for the cycle.
- First and Last frame of the Cycle: Again, this should be self-explanatory.
- Object driving the Cycle: This is the object whose motion is driving the cycle, unless you select "time" as the driving function, which makes the object irrelevant.
- Driving function: Choose what aspect of the object's motion is going to drive the cycle. Most of the options should be clear, path distance means the absolute path distance traveled by the object with no sense of orientation. Oriented distance will grow larger as the object moves "forward" and grow smaller as the object moves "backward". Oriented distance is probably what will be most often used. NOTE: if you happen to have constructed your object so that its "forward" end faces down the negative Z axis, you will need to go in by hand and edit the expression to have a negative ODIST value.
- Stride Length: Quite often a cycle will need to repeat over a specific distance (like 2 times the step distance of a character, or the distance around the path of a tank tread unit), the distance over which the cycle repeats is known as the "stride length".
- Optional Length Delay: This field can be used to push the cycle back over its stride length, used to offset opposing legs in a walk cycle, or pushing one element of many back down its path a little bit.
- Tie length delay to object extension: Use the clone number (i.e. the 3 in "null (3)", or the 10 in "obj10.lwo") as a multiplier for the length delay. You can use this field to create an expression that should automatically tailor itself to clones of that same object.
- Extension offset: This value is subtracted from the object's extension and can be used to turn the clone number of "null (1)" to 0, so the first object doesn't have an offset along its path.

## Dr. Motion Blender

### Multi-slot professor

Dr. Motion Blender can be used to blend between the motion paths of two objects interactively during an animation. What applications that might have in your animations is entirely up to you. Here is a breakdown of the fields:

- Object 1 and Object 2: Pick the objects whose motions you want to blend between. NOTE: if you put SELF in for one of the objects, it would blend this object's own keyframed motion with something else.
- What Channels will we be blending between: Pick one or more motion channels for the motion blend.
- Constant Blend Percentage: Pick a constant blend percentage between the motions.
- Blend Control Object: If this field is filled, it negates the blend percentage field. Use an object to control the blend.
- Channels of the controlling object: You can choose a single channel of the controlling object to control the blend on all channels of our blend object (for instance, the Y value of our object would control the blend as it moved from 0 to 1). If you pick "All Channels", then the X value of the control object will determine the X blend percent, the Y value will control the Y blend, etc.
- Truncated Blend: If the control object's values get above 0 or 1, cut them off at 0 or 1, otherwise, you'll get an "overblend".



## Dr. Blend Machinist

Dr. Blend Machinist is one of the most powerful professors. Using this panel, you can set up lots of interesting relationships between objects, where one channel of one object control the action on one or more channels of another object. For instance, imagine a foot where you need the toes to curl as it lifts off the ground. Or imagine a mechanical system where you need a piston to expand and contract with a rotation. Or imagine a key-state animation, where you can control the opening and closing of multiple bones/segments using a single control null (or group of nulls)...basically any kind of "reaction" to some slider value...Blend Machinist can do these and much more. Here's a breakdown of the fields:

- Truncated Blend: This will cut off the blend as it reaches its min and max values, a non-truncated blend will continue the blend beyond the min and max values.
- Type of Blend: A blend between values will do just that, blend between two values. A time blend will push the time value back and forth and play back a keyframed motion. Quite often, a time-based blend is what's required to properly blend-animate a motion that has more than two distinct keyframed values.
- First and Second Value: These can either be straight numerical values or expressions unto themselves...only used if we're blending between two values.
- First and Second Frame: Used if we're blending time, at the lower blend value, the current motion channel will go back to the start frame. As the blend value increases to the upper blend value, the current motion channel will play along to end at the second frame value when the blend value reaches the upper value.
- Track what value for the blend: You can either click on a motion channel to track, or enter an explicit expression by clicking on the "Custom Expr" button and designing an expression using Dr. Equation Maker from the "Expression" button.
- Expression: Used when the "Custom Expr" button is clicked.
- Lower/Upper Blend Trigger Values: Lower and upper values for the blend to track.



NOTE: if you need to set up a blend for more than one motion slot, you will need to re-select the professor next to each slot that you want set up.

## Dr. Dist Maintainer

### Multi-slot professor

In order to understand how to use this professor, you need to understand the two modes it works in. One is a linear distance maintainer, the other is a spherical distance maintainer. For the first mode, it's important that you choose wisely which motion slot you select the Dist Maintainer professor from, this will become the "sliding axis". For instance, say you want an object to slide up and down the Y axis to try to keep its distance constant to some other object, you would then select Dr. Dist Maintainer next to the Y slot. Note that Dr. Dist Maintainer doesn't make much sense in the rotation or scaling slots. In spherical distance mode, the expressions generated will draw a line from the current object to the tracking object and measure the length of that line. It will then move the current object along that line until the distance is the value desired. Here's a breakdown of the fields:

- Other object: This the object that you want to maintain a constant distance to.
- Distance to maintain: Distance to the maintainer.
- Spherical distance: Throw the professor into spherical distance mode, as described in the preceding paragraph. If this button is off, the distance formula works in sliding-axis, linear mode.



## Dr. Event Maker

### Multi-slot professor

Dr. Event Maker is one of the most powerful professors available in your arsenal. Using this professor, you can set up an animated event to play back when some particular action occurs: the object will play back its own keyframed motion in response. You can also set up a “closing event” for when the particular action undoes itself. A good example (indeed, an example scene), is a set of sliding doors, like on the turbo lift in Star Trek. As an object approaches the doors, they slide open, and then as the object gets far enough away, they slide shut again. But there’s lots more events than that...play around, experiment...see what automated responses you can come up with. Here’s what the fields on the panel do:

- Type of Event: If set to Resettable, the time of the event will reset every time the trigger event becomes false again, and then start ticking again while the trigger event is true. A one-time-only event will keep ticking even if the condition for the trigger event becomes false.
- Channels for Event: Pick the channels that you want this event to play back on. Note, you are selecting the object’s own keyframed motion for this event. If you want to play another object’s motion back instead, you will need to hand-edit the expressions generated to point to the object in question.
- Object to control the event: This is the object that we’re tracking.
- Look for Event: Here, you choose what event to look for. For now, it simply tracks whether a motion channel becomes greater than “>” or less than “<” a specific value. Notice if you choose “Dist >” or “Dist <”, the “From what object” field becomes active.
- Value: this is the value that our object must hit to trigger the event. Once it falls above or below this value (depending on whether you chose a greater-than or less-than button), the event will be triggered.
- From what object: If you are tracking distance, there needs to be an object to track the distance from.
- Play Event From/To Frame: Pick the first and last frame that our object will play back in response to the event trigger.
- Closing Event: You can set up a closing event that happens when something else occurs. NOTE: if you are choosing to undo some action, like having a door close back shut, it’s important that you set up both events so that they are mutually exclusive...i.e. if you have a door open when an object gets closer than 1 meter, you should not have it close when the object gets out to 0.5 meters... otherwise, there will be a conflict between the two conditions.
- All the other fields are functionally equivalent to the original event fields.

## Dr. Camera Shaker

### Multi-slot professor

This professor is great for not only setting up a camera shake when an event occurs, but also for making stuff jiggle with impact, and other sorts of impact-related events.

- Channels to shake up: Pick what channels you want to set the impact on. Next to each channel selector is a power field...increasing this value will increase the intensity of the impact. Decreasing will do the opposite.
- Object that should shake the camera: This is the object that will make the camera shake occur.
- Look for Event: Look for either a motion channel or the distance of this object from another to either become less or greater than some value.
- Length of shaking: Determines the length of the shaking.
- # Shakes/second: This is the frequency of the shaking, higher values for this will make faster shakes, lower values will make slower shakes.
- Would you like shake to fall off with distance?: You can have the shake fall off with distance.
- Distance when shake is optional: The impact expression will be at full force at this distance; closer, and it will become a super-impact; further, and it will become a soft impact.
- Quadratic Falloff: Make the impact distance expression fall off more rapidly as the object gets further away. A non-quadratic falloff doesn’t decay down as rapidly as the object moves further from the camera.
- Would you like the shake to fall off with impact velocity?: Turning this on will cause the shake to change depending on how hard the object hits the ground.
- Velocity strength factor: Changing this to a higher value will make the shake more sensitive. Changing it lower will make it less sensitive.



## Dr. Morph-o

Dr. Morph-o is the professor to use for setting up Relativity morphs. It should be noted that Dr. Morph-o currently only sets up morph expressions with either a morph controller or an effector. If you need some other expression to control a morph other than just the X value of a control null, you will need to hand-edit the expression to make it happen.

- Target: This is the target for the morph. NOTE: it doesn't have to have the same number of points as the base object, but it would be quite advisable.
- Sign of morph: Determines how the morph is applied; replace is like LW's traditional morph; add will add the values of the target points to the base object; subtract will subtract the points; difference is a cool morph, it morphs only points that are different between the base and target objects.
- Type of morph: Normal is like LW's regular morph. Effector will apply a morph in an effector like fashion, in a spherical gradient around an effector object.
- Object to control morph: Choose an object to control the morph.
- Morph effector: This is the effector that controls an effector morph
- Min radius: This is the area around the morph effector when everything is fully morphed.
- Max radius: In the area between the max and min radius, points close to the max will be barely morphed, while points close to the min will be almost fully morphed. Points outside the max radius are not morphed at all.
- I want a min and max null: Choose a min and max null to interactively animate the min and max values for the effector.
- min null: The minimum value for the null
- max null: The maximum value for the null.



NOTE: These last two don't have to be nulls, but usually are.

## Dr. Target

Dr. Target is used to set up a targeting relationship between the current object and a target.

## Dr. Matcher

Dr. Matcher is used to match an unparented, free-floating object to an object or bone deep in a hierarchy. If you want to dynamically parent/unparent an object to another, here's how to set it up:

1. create a null and use Dr. Matcher to match to whatever child object/bone/whatever you're interested in.
2. set up a motion blend using Dr. Motion Blender between your unparented object and the MATCH'd null.
3. when you want the object to move from unparented to parented, quickly keyframe the blend null over one frame. If you need to have motion blur and not have your object do some sort of odd sub-frame rotation, you might want to change the expressions for the Motion Blend from:

```
X=BLEND(X(SELF,t),0,X(matcher,t),1,X(blah,t))
```

to

```
X=BLEND(X(SELF,t),0,X(matcher,t),1,IF(X(blah,t)>0))
```

That will cause your motion blend to be instantaneous, even on a sub-frame level.

## Dr. MoveVec

Dr. MoveVec sets up a really odd form of motion...basically, whichever way a MOVEVEC'd object is turned, it will move in that direction. This was originally done as a means of making an object move as if it were under the influence of a joystick...just set up an expression to make it rotate accordingly and the object will act like it's in a video game.

## Dr. Snake Maker

Dr. Snake Maker is a generic level plugin that can set up a scene for you with a bone-snake (for path-level deformations) stretching through an object. Here's a breakdown of the fields:

- Set up bones for object: Picks your object.
- How many bones do you want: Tell it how many bones in the snake. Keep in mind, you want to balance between having enough bones for a smooth path deformation but not having so many that the scene becomes unwieldy.
- How long is your object: Measures your object and tell the nice professor how long it is. As an aside, the "head" of your object should rest at the origin (i.e. XYZ=0,0,0) and the "tail" stretch out into positive Z.
- Bone name prefix: Each bone will be named with whatever prefix you choose, followed by a number...i.e. SnakBon01, SnakBon02, etc.
- Target name prefix: The way a bone snake is set up in Relativity, a train of nulls follows your object around, and the bones target those nulls. This lets you choose the prefix for each target null.





---

## Chapter 21: Dynamics

---



## Introduction

LightWave provides the user with tools for simulation of real-world physics and dynamics. With these tools you can make your objects behave with the characteristics of cloth or of elastic or rigid bodies, and set characteristics for the environment in which your objects are interacting. You can also apply dynamics to the skeleton of your characters or articulated mechanisms, so that they too respond in a realistic simulation of an environment with gravity or wind, other objects to collide with, etc.

We'll begin our discussion of LightWave's dynamics with the modules for simulation of cloth, soft bodies, rigid bodies and particle effects. This will include a look at the integrated functions for wind, gravity, and more.

## Adding Dynamics to your Scene and Objects

### Items: Add: Dynamic Object

One way to quickly add dynamic objects to your scene is to use the **Add Dynamic Object** drop down list (**Items > Add Dynamic Obj**). Choose between **Collision**, **Wind**, **Gravity**, and **Particle** from the drop down menu. You can add one or more of these controllers to your scene. Then, each can be set with its own Parameter settings to create just the effect you are looking for. A dialog prompts you for a name when you add a dynamic object. Click OK to accept the default or type in the name you wish to use and click OK.

**Collision**-The Collision controller lets you add an element for the particles and/or dynamic objects in the scene to bump into.

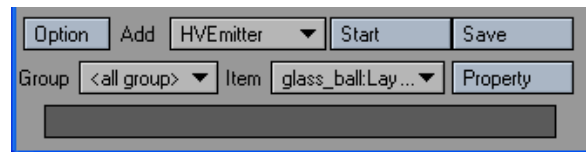
**Wind**- The Wind controller lets you add wind to blow your particles and/or dynamic objects around.

**Gravity**- Add a Gravity controller to add gravity-like effects to your particles and/or dynamic objects motions.

**Particle**- The Particle Emitter controller is the controller and the source for particles. There are two emitter selections: **HV Emitter**, which adds a HyperVoxels Emitter to the scene, and **Partigon Emitter**, which adds an emitter of single point polygons ("partigons") to the scene

## FX Browser

The **FX Browser Panel** is where you can add various controllers. From here, you can also open the property panels for the controllers that you add. To open this panel, choose **Utilities > Additional > FX\_Browser**.



### To add an FX controller:

From the **Add** pop-up menu, choose the controller you wish to add, load, or apply.



NOTE: FX controllers can also be applied to an item in the **Dynamics Tab** of the **Object Properties Panel**.



NOTE: When you add a controller from the FX Browser that is not one of the "Apply" options, a null object is added to the scene. Then, an FX custom object plugin is added (e.g., FX\_Emitter, FX\_Wind, etc.). Just check out the **Dynamics Tab** of the **Object Properties Panel**.

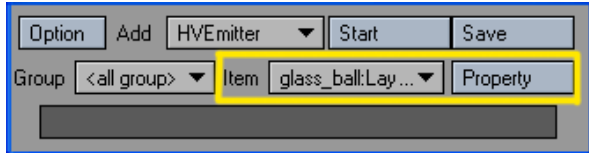
### To delete a controller:

Select and clear, just as you would any other Layout item (**Items > Clear Selected Item or - on the numeric keypad**).



### To open a controller's property panel:

Choose the controller from the **Item** pop-up menu and click the **Property** button to open the **Properties Panel** for the selected controller.



**NOTE:** You can also display the **Property Panel** for the currently selected controller by choosing **Utilities > FX\_Property** or by accessing the **Dynamics Tab** in the **Object Properties Panel**.



**HINT:** You can open multiple property windows, one per dynamic type, by holding CTRL then changing the selection in the Item Select Menu.

### The Start Button

Some effects require a “pre-calculation” before they can be previewed in your animations. These are instances where you have particles interacting or have post-deformed geometry. By activating the **Start** button, the FX will begin solving any simulations that are currently setup in the scene. The progress of the simulation is displayed in the output pane located at the bottom of the **Particle FX Browser**. A shortcut to this is on Layout's toolbar at **Modify > IKB Calculate**.



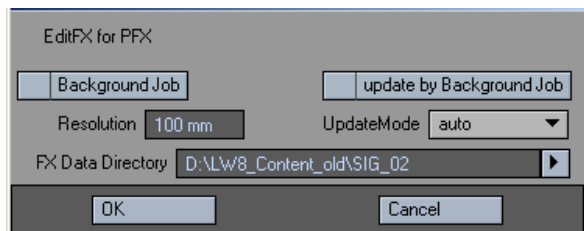
**NOTE:** Many scenes don't need this step, but clicking **Start** won't do any harm if you are unsure.

### The Save Button

Clicking the **Save** button saves all controller motions, using individual files, to your FX directory (defined on the **Option** dialog). These can be loaded using the **Load Motion** option on the **Controllers Properties Panel**.

### Options Dialog

The **Options** dialog has some options that affect the manner in which LightWave computes controllers.



The **Background Job** option activates multithreading. The **update by Background Job** option will update Layout when background tasks are complete. In most cases, you should leave these options in their default state of off.

The **Resolution** setting adjusts the parameter used when performing the physics simulations. The smaller the **Resolution** setting is, the more accurate the

simulation will be. However, this also has a direct effect on the rendering times.

In this dialog you can also set the **FX Directory**. This is the default directory used when saving **Controller** settings.

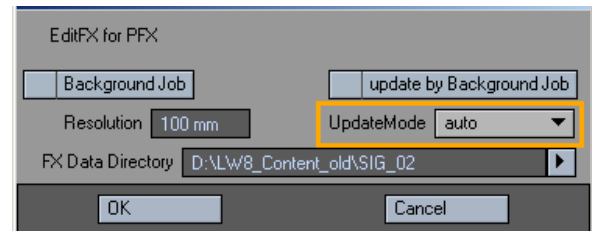
### Real-time Display

Just click the **Play** button in Layout and as you tweak the various controller parameters, you get real-time updates of how your changes affect the particles.



**NOTE:** Scenes with particles interacting or with post-deformed geometry require you to click the **Start** button first to pre-compute some motions. Clicking **Start** won't do any harm if you are unsure.

You can control how FX Controllers are updated using the **UpdateMode** pop-up menu.



**Auto** updates all emitters as parameters are modified. Use this for fast systems or in low-particle count simulations.

If you have multiple emitters and are using the **Select** update, then only the emitter showing in the **Select** pop-up menu, if any, will be updated.

**Adaptive** dynamically scales the number of visible particles based on your CPU performance. This setting attempts to keep interactive performance at a useful level.

User turns off particle updating completely. You may force an update by either clicking the **Start** button on the Particle FX Browser, or the **Update** button on the **File Tab** of the **Emitter Controller Panel**. This setting is useful in situations with heavy particle counts.

### Groups

You can group Dynamic objects with a user-defined name to prevent unwanted interaction. This also works with Particle FX controllers. This becomes very handy when you are working on a complex scene and you want certain Wind emitters to only affect certain objects.

**<All Group>** Includes all groups.

**<default>** Includes all items in scene.

**<new group>** Create individual group. Can associate functions within a group.



**NOTE:** You can associate with particle FX's functions using group function. If it is in the same group, you can associate settings. Also, if you do not want to associate, just by removing from the group, you can disregard the association.



## Object Properties: Dynamics Tab

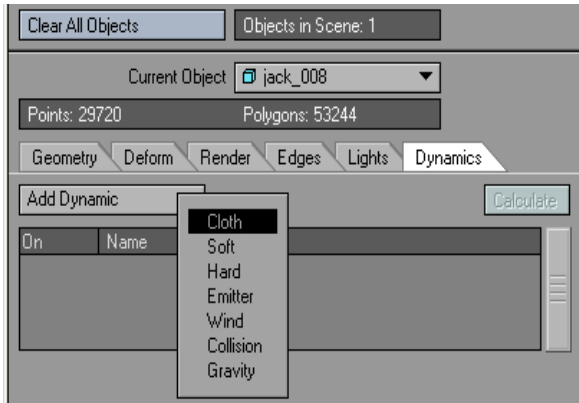


Figure 1: Dynamics Tab, Add Dynamic drop-down menu.

The **Dynamics Tab** of the **Object Properties Panel** is where you can **Add Dynamics** and manipulate their settings.

### Add Dynamic

The **Add Dynamic** Drop Down menu lets you select Dynamics to apply to an object.

Add Dynamic choices:

**Cloth-** ClothFX, applies Cloth Dynamics to a target object.

**Soft-** SoftFX, applies Soft Body Dynamics to a target object.

**Hard-** HardFX, applies Hard Body (Rigid Body) Dynamics to a target object.

**Emitter-** adds a ParticleFX Particle or Partigon Emitter to the scene. Note that if you wish to add a HyperVoxel emitter rather than a particle emitter, you should use the Item: Add: Dynamic Object: Particles menu option to add the HV Emitter.

**Wind-** adds a Wind controller to the scene, which can be placed, sized and oriented to affect a specific portion of the virtual stage.

**Collision-** adds a Collision controller to the scene, which can be placed, sized and oriented to affect a specific portion of the virtual stage.

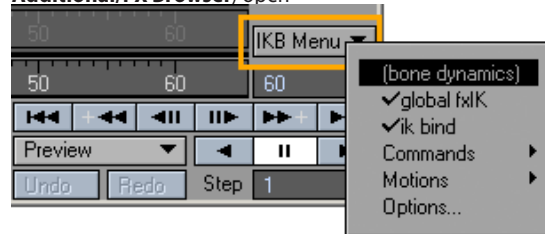
**Gravity-** adds a Gravity controller to the scene, which can be placed, sized and oriented to affect a specific portion of the virtual stage.

## Dynamics List Window

Add some text here describing the operation of the list window and how it is used.

## Calculate

Clicking on the **Calculate** button will perform the dynamics calculations for the scene. If the **FX Browser** is not open, please note that if you change focus to another application or attempt to move the Object properties panel, the viewports will stop tracking the dynamics progress and stop refreshing. Layout will appear to be locked up, but the calculations are continuing, and the interface will refresh again when they are complete. Having **FX Browser (Utilities/Plug-ins/Additional/FX Browser)** open



### Bone Dynamics

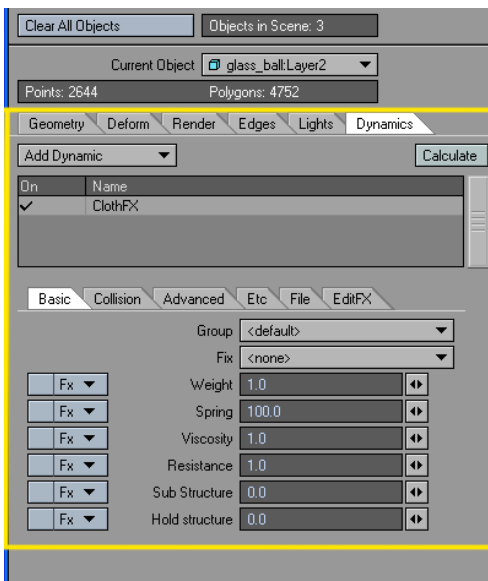
This option enables and disables all bone dynamic functions for the object that IK Booster has been applied to.



## Cloth Dynamics

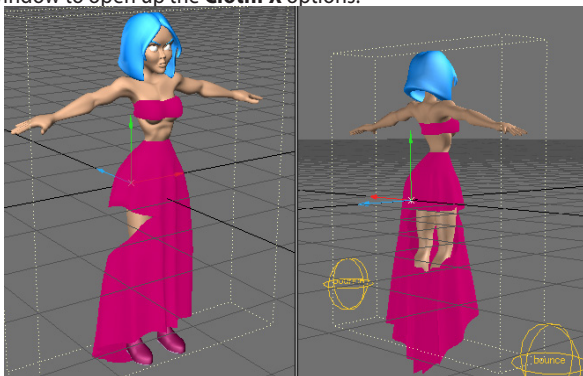
Cloth Dynamics (**ClothFX**) have a variety of uses for supplying physical characteristics and natural movement simulation to thin, flexible materials. Examples:

- Clothing on a character such as a dress or shirt
- Flag blowing in the wind
- Curtains
- Tablecloth or anything that needs to be draped over another object
- Mud flaps on a truck
- Hair



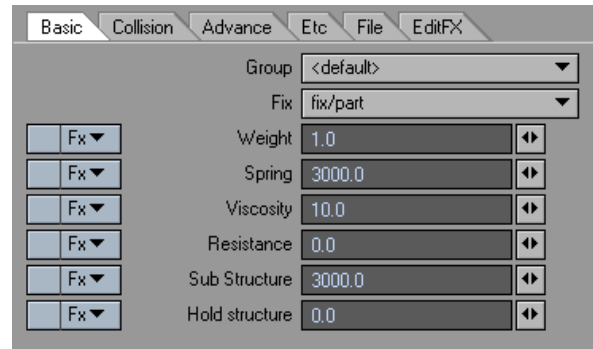
ClothFX Menu Tabs, Basic Tab menu

**Cloth Dynamics** are located in the **Dynamics Tab** of the **Object Properties Panel**. Choose **Cloth** from the **Add Dynamic** drop down list and then follow that by clicking on **ClothFX** in the Dynamics list window to open up the **ClothFX** options.



"Sew" function in use

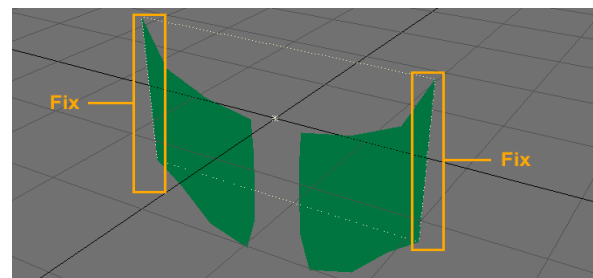
## Basic Tab



Basic Tab menu

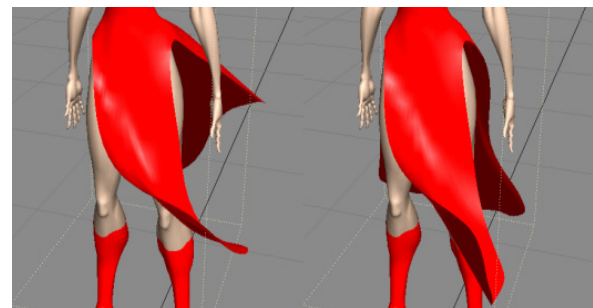
**Group**- you can group Dynamic objects with a user-defined name to prevent unwanted interaction. This also works with ParticleFX controllers. This becomes very handy when you are working on a complex scene and you want certain Wind emitters to affect only certain objects.

**Fix**- The shape of polygons with "Fix" on remains intact and is not affected by the dynamics. In the image below, the ends of the cloth have been *Fixed* so that they stay in place while the rest of the geometry is pulled down by gravity.



"Fix" applied to polygons indicated inside orange outlines – these stay in place, while gravity affects the rest of the cloth object.

**Weight**- defines the weight of the material. You can simulate the motions of a heavy material by increasing the Weight value, and you can simulate the motions of a light material by decreasing Weight.



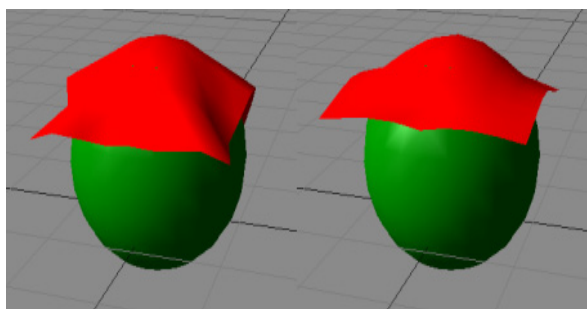


The dress on the right hangs down more due to its weight and also will not flap as much.

**Spring-** controls the *springiness* or stiffness of the material. Reducing the Spring coefficient creates soft motions, while raising Spring produces motions with a stronger repelling force. Setting the coefficient to an extremely large value creates stiff motions.

**Viscosity-** controls the level of impact that a collision will have on the shape of an object. An object with higher **Viscosity** value tends to keep its shape more. If an object bounces, a higher Viscosity value will have less bounce motion because the Viscosity absorbs the bouncing force.

The pieces of cloth below are the same except for the **Viscosity** parameter. Here, a ball is moving up, pushing the piece of a cloth. The cloth with the higher **Viscosity** value tends to keep its shape more.



Effect of viscosity parameter on cloth. Left = lower viscosity, Right = higher viscosity

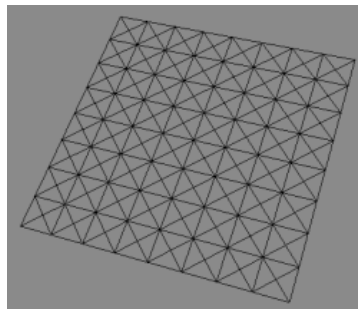
**Resistance-** This parameter is not a characteristic of the object itself, even though it can be set per object. **Resistance** controls the amount of resistance of the environmental medium in which the simulation is taking place. For example, this affects how quickly a cloth object moves through the virtual air in the scene by determining how “thick” the virtual air is. The default setting is 1.0; raising this number increases the resistance and lowering to a number between 1.0 and 0.0 decreases the resistance. This allows the simulation of a wide range of gaseous or even liquid media as the environment for the physical simulation. A negative number amplifies a collision of the cloth object. To simulate behavior within a normal atmosphere at sea level, a value of .14 would be appropriate. A value of 1.0 gives the medium a character close to that of molasses.



NOTE: If you are coming to LightWave from Maya, the function that is most similar from that application is “Damping.”

**Sub-Structure-** The surface of a pure, elastic body model has a high degree of freedom, which causes the object to easily distort. To prevent distortion, you can apply an auxiliary form, called a **Sub-structure**, to restrict the instability of the surface. This can improve your results with a two-dimensional object (i.e., one with no thickness) making it act as if it has some thickness. However, this reinforcement can require a high **Sub-Structure** setting that can take longer to calculate.

When **Sub-Structure** is used on a square mesh, it creates something that looks like two triangular polygons placed on top of each other in opposite directions. The **Sub-structure** mitigates the tendencies to bend and to resist bending in certain directions.



Substructure applied to the quadrangles of a mesh.

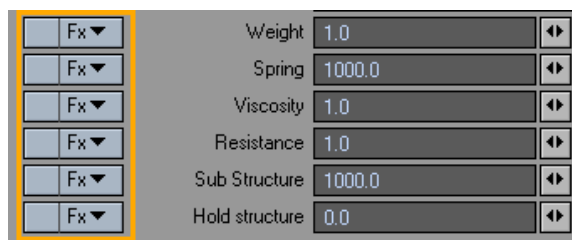
When the **Sub-structure** value is non-zero, the auxiliary form is applied. The higher the value, the more the form will have a tendency to keep its shape.

**Hold-Structure-** because a pure elastic body model simulates only the surface structure, it is limited to two-dimensional motions, such as a bed sheet. In order to simulate elastic three-dimensional motions, such as a lump of gelatin, a **Hold-structure** needs to be used. This parameter causes a surface to tend to maintain its original shape, like gelatin does when it jiggles.

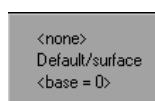


NOTE: The effect of **Hold-structure** is uniform throughout the surface, while with **Sub-structure** it is non-linear, which will often result in a more natural look. You can use a combination of both to achieve just the right result.

**FX Button-** Allows you to use a **Vertex Map** to control the value of an attribute (spring, viscosity, etc) or a function (delay, wave size, compress bump).

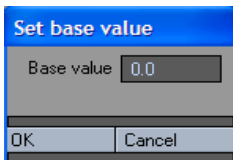


FX Drop Down menu buttons



FX Drop Down Menu showing available options for surfaces to control the value of an attribute.





Base value requestor

From the **FX** drop down menu the user can select a **Vertex Map** as well as change the base value.

**Vertex Map** – Will be displayed by name and type.  
Example: "LeftArm/Weightmap"

**Weight Map** – The weight value of each point is defined in Modeler when you create a **Weight Map**. In Layout you cannot edit **Vertex Map** values. But the **FX** button allows you to add a modifier that, when combined with the original values of the **Vertex Map**, provides a way to adjust the values of the selected attribute/function without changing the original **Vertex Map** value.

**Pointset** – A pointset is created in Modeler. In Layout you cannot add or subtract points from a pointset. But the **FX** button allows you to add a modifier that, when combined with the pointset (value of 100%), provides a way to adjust values of the selected attribute/function.

**Surface** – Surfaces are created in Modeler. In Layout you cannot add or subtract points from a surface. But the **FX** button allows you to add a modifier that, when combined with the surface (value of 100%), provides a way to adjust values of the selected attribute/function.

**UV** – UV maps are created in Modeler. In Layout you cannot add or subtract points from a UV. But the **FX** button allows you to add a modifier that, when combined with the UVs value, provides a way to adjust values of the selected attribute/function.



NOTE: You can always check value information for points in the **Point Info** windows.

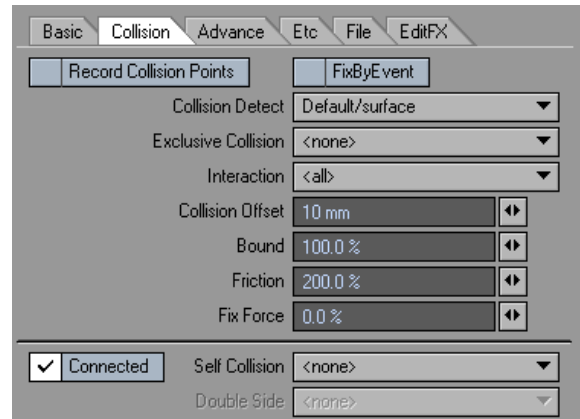
**Base** – Selecting the **<base = 0>** option from the FX Drop Down menu will open a small requester that allows the user to edit the base value. Changing the base value changes the effective range. Example: Base=0% (0 – 100), Base=20% (20 – 100). The base value is 0% by default.



NOTE: For those interested, the equation used to calculate the vertex point value is:  $V_p = V_{map} * value + (1 - V_{map}) * base$ . Another way to look at the equation is:

$V_p = (value - base) * V_{map} + base$ .

## Collision Tab



Collision Tab

**Record Collision Points**- When this is activated, ClothFX will record the time and location of a collision, which will trigger a particle emitter that has been set up and **that is parented to the ClothFX object**.



NOTE: Record CP works with ParticleFX and Nozzle=Parent collision, but you cannot use it with FX Collision and Parent Recorded CP (FX emitter>playback mode).

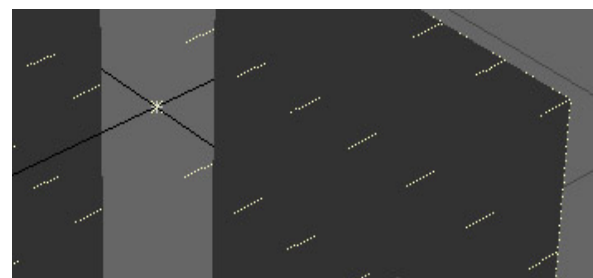
**Fix By Event**- This function will fix points that pass through a collision object that is set to "Event" mode.

**Collision Detect**- allows other objects to influence the motion of a dynamic object. This lets you create complex motions caused by obstacles in the path of the dynamic object. The simulation is performed by taking an object that collides (the collision object) with the elastic body model (dynamic object) into account in the calculation.

**Exclusive Collision**- When a collision object is selected from the list in this control parameter, it becomes the sole collision object. All other collision objects are excluded from the calculation.

**Interaction**- By Default you would want Interaction set to "All" so that all geometry will interact with each other. However you have the option of limiting the interaction by **Vertex Maps**.

**Collision Offset**- By default the collision offset is set to 0 so that the collision object will collide with the surface of the target object, allowing the target object to deform. Increasing the **Collision Offset** will result in the collision object deforming the target object at the distance specified in the **Collision Offset** field.



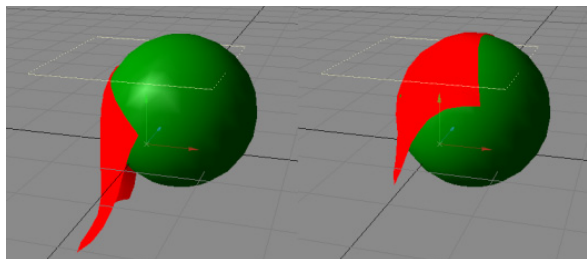
Collision offset



You should understand that the *collision* is detected using a point on the target object (elastic body model) and a polygon of the collision object. As a result, a polygon of the target object can possibly penetrate the collision object—usually an undesirable result. Use the **Collision Offset** setting to avoid undesired surface penetration.

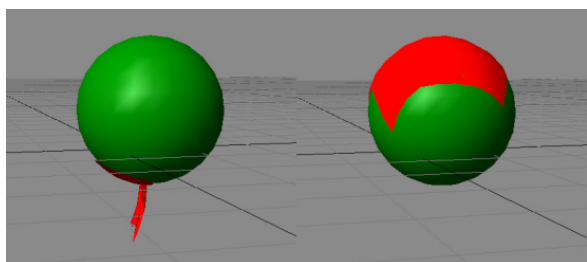
**Bound-** adds a *rebounding* speed change at collision.

**Friction-** Like real-world friction, the **Friction** parameter makes the surface of the dynamic object less slippery. So, if you want the Dynamic object to tend to slip off the collision object, set **Friction** to 0. If you want it to stick more, increase the value.



The settings on the right image have a higher friction causing the cloth to stick to the collision object.

**Fix Force-** causes the dynamic object to stick to the surface and not slide around.



The image on the right has a higher **Fix Force** setting.

**Self Collision-** The **Self Collision** setting helps prevent a soft or cloth object's mesh from passing through itself when the object is transformed during the simulation. **Self Collision** is calculated based on a point on the target object and a polygon on the collision object. If a surface has **Self Collision** and **Collision Detection** active, **Self Collision** is computed on the same surface.

**Self Collision** works in much the same way **Collision Detection** does, and there is no collision detection for polygon edges. As such, you may also want to adjust **Collision Offset** to avoid errant point penetrations.

**Connected-** **Connected** is a Self-Collision option that derives its name from the **Select Connected** option in Modeler. When activated, a contiguous mesh within an object can self-collide. When deactivated, adjacent meshes within a single object can "self-collide" with each other, but any given mesh will not be able to collide with itself.

An example would be a character object that has been modeled with a complete body mesh and with clothing as additional meshes that are not part of the body mesh, such as a T-shirt. The T-shirt and the body of the character would be separate contiguous meshes -- that is, in Modeler, selecting a few polygons on the T-shirt and hitting **Select Connected** will select all of the polygons of the T-shirt, but not select

the polygons of the body. Selecting a few polygons on the body and hitting **Select Connected** will select all of the polygons of the body, but not the T-shirt.

In ClothFX, if **Self-Collision** is on but **Connected** is de-activated, the T-shirt can collide with the body, but not with itself. If **Connected** is activated, the T-shirt can collide with itself and with the body of the character. Deactivating **Connected**, then, can speed up a self-collision calculation at the cost of accuracy. Use only when separate meshes contained within a single object layer must collide against each other, but do not have to collide with themselves. Do *not* deactivate this option when dealing with curtains, windsocks, or any other situation where the material *must* self-collide with areas actually connected to itself.

**Double Side-** Collision is detected even if the motion is from behind the polygon.

## Advanced Tab

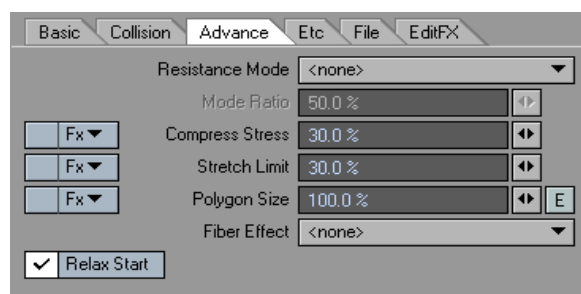


Figure 16: Advanced Tab

**Resistance Mode-** **Resistance mode** tells the **Resistance** setting in **Basic Tab** how to react. With some amount of Resistance applied, you can see as you change **Mode** settings how the polygons will resist wind effects.

**None-** points move based on wind direction.

**Polygon-** points move based on wind direction + point normal.

**Polygon (front)-** points facing the wind move based on wind direction + point normal.

**Polygon (back)-** points facing away from wind move based on action + point normal.



NOTE: **Resistance Mode** settings only apply to wind.

**Mode Ratio-** sets the amount of **Resistance** applied to specific effects set by **Resistance Mode** versus standard. 75% sets 75% of the **Resistance Mode** and 25% of the standard resistance.

**Compress Stress-** controls the amount of compression a surface exhibits as a result of stress. A soft fabric like cotton can be made using a large **Spring** value and a low **Compress Stress** setting. A stiffer fabric would use a higher **Compress Stress** setting. An example for use: you could easily make the apparent thickness of drapery vary with this setting.



Note: **Compress Stress** affects the strength of **Spring** and **Substructure spring** forces when they become compressed during the calculation. Suppose a polygonal edge, which represents a **Spring** force, is 1m in length, and the **Spring** strength is set to 1000 and the **Compress Stress** to 30%. If the material is stretched so that the edge becomes more than 1m in length, the **Spring** force will snap back with its full force of 1000. If



the material is compressed so that the edge becomes less than 1m in length, the Spring force will push the points back with a force of only 300 (30% of 1000). The same Compress Stress percentage applies to Sub Structure forces when a Sub Structure spring is compressed below its original length.

**Stretch Limit-** To prevent a surface from stretching like rubber, lower the Stretch-limit below the default value of 100%. This restricts the amount that the surface can be stretched. Setting Stretch Limit to 0% turns it off completely. For free-moving cloth like flags and curtains, one might set it as low as 1%. However, give clothing some "breathing room" in the Stretch Limit, and do not lower it below 10% for clothing items like sleeves, shirts, and pants. To reduce sagging in clothing, reduce the Polygon Size, not the Stretch Limit.



**NOTE:** Although you could also increase the Spring value to reduce stretching, increasing that value can make the behavior of the surface too complex and cause problems such as weird folds.

**Polygon Size-** Changes the surface to a specified size. For example, a value of 90% reduces the size of the surface to 90%. Use **Polygon Size** to reduce the looseness of a dress or to create frills.

**Fiber effect-** Allows you to set an axis for the traditional **Fiber effect**, basically making a certain axis "stronger" than the others. For example, setting Y as the axis makes the built-in folds of the dress (dress\_fx002) stronger so the folds aren't disregarded during calculation.

**Relax Start** - When this option is on, the ClothFX object starts out with no "Stress State", meaning that the object is considered to be in its "true" or "base" state. Thus, if any displacements are applied to the object on the first frame of the calculation, the resulting form will be considered the "true" or "base" form of the object. This includes any Stretch/Size, bones, displacement maps, or morphs present on the first frame of the calculation. ClothFX will respect the size of each polygonal edge as the "true" length of each polygonal edge, and it will stick close to the distorted proportions of the object during the calculation.

When **Relax Start** is turned off, ClothFX considers the undistorted state of the object as it would be seen in Modeler to be the "true" or "base" state of the object. If the object is deformed by Stretch/Size, bones, displacement maps, or morphs on the first frame of the calculation, the object is considered to be "stressed" and will regain its "true" proportions during the calculation.

For example, take a sheet of polygons aligned with the XY plane and fix the uppermost row of points. If this object is stretched to a value of 4 on the Y axis, and **Relax Start** is turned ON, then the stretched version is accepted as the "true" version, and the calculation will be for a tall version of the object, without any great fuss. If **Relax Start** is turned OFF, then the unstretched version is regarded as the "true" form of the object. Because the object is stretched to four times its original size on the first frame of the calculation, the effect is as though the object begins by being stretched like a rubber band. As soon as you hit Calculate, the non-fixed points of this non-relaxed object will SNAP back into distances between themselves and their neighbors that more closely resemble the distances between them in the undistorted version of the object as would be seen in Modeler.

## Etc. Tab

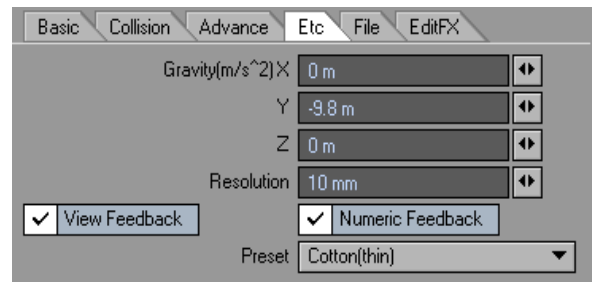


Figure 17: Etc. Tab. Note that the entry shown would be the setting to use for Earth-normal gravity.

**Gravity(m/s<sup>2</sup>)-** Specify the direction and intensity of gravity along the world axes for this particular dynamic object. Note that when you set gravity in the **Etc Tab** for your dynamic object, it is not necessary to add a gravity controller separately. Conversely, if you add a gravity controller and set your gravity characteristics properly there, it will not be necessary to set gravity for the dynamic object on the **Etc Tab**. If you have a number of objects in the scene that you wish to have affected by gravity, then it might save effort to add a gravity controller and properly set it up and size it to affect all the necessary objects.



**HINT:** The gravitational acceleration constant on earth is 9.81m/s<sup>2</sup>. On planets with a less strong gravitational pull, (such as the moon) it is much lower, whereas on planets with a stronger pull (such as Jupiter) it is much higher. In LightWave's coordinate system, you would set gravity by entering a negative value in the Y entry field; -9.81 m for Earth-normal gravity.

**Resolution-** Set the accuracy of calculation by how big the maximum error is.

When it approaches 0 the accuracy increases and errors will decrease, but calculation workload will also increase so it will take more time to calculate.

**View Feedback-** Set the feedback in Layout when controlling ClothFX. When the display is poor or the display speed is slow, you uncheck this.

**Numeric Feedback-** Set the numerical feedback in Layout when controlling ClothFX. When the display is poor or the display speed is slow, you uncheck this.

**Presets-** Similar to surface presets, ClothFX has 5 built in presets that give you a good place to start. Presets include: Cotton Thin, Cotton Thick, Silk, Rubber, Jelly.



## File Tab

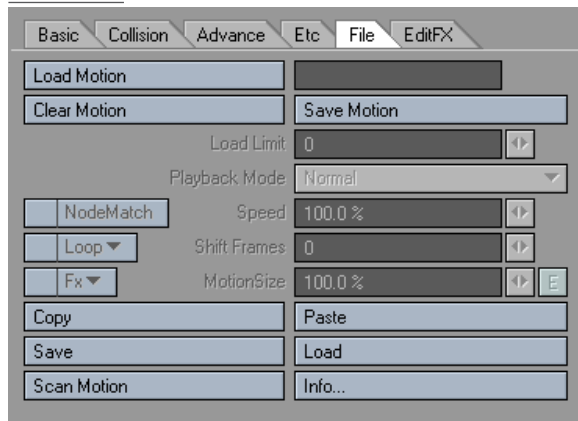


Figure 18: File Tab

**Load Motion-** Loads a previously generated Motion file.

**Clear Motion-** Clears a loaded Motion file.

**Save Motion-** saves the generated motion to a file, which will be used subsequently to deform the object.



*NOTE: After a motion file is saved, the name of the file will automatically be entered into the load field.*

**Load Limit-** Loads a limited range of Frames. *Example:* If you have a MDD file with 2000 frames of data you can limit loading to just 100 frames to make it less memory and processor intensive.

### Playback Mode

**Normal-** Plays back all **Motion** settings of the .mdd file.

**Local-** Plays back based on the local translations (move, rotate and scale) of the object. Allows a .mdd deformation on an object while respecting new motion applied to that object.

**Distance Map-** This adjusts the motion data to the movement of the object. When the speed is 100%, 1s (second) of motion data is used to move the object a distance of 1m.

**Speed-** When Playback Mode is set to Local or Distant the user is able to control the playback speed. One can also play the MDD file backwards from by setting Speed to -100% and inputting the frame-length of the MDD file into Shift Frames. For example, if the user recorded a 300-frame MDD file, he could play it backwards by setting the Playback Mode to Local, the Speed to -100% and the Shift Frames value to 300

**Node Match-** When Node Match is activated, the motion data can be played back on the object's local position. Node Match allows one to play back MDD data recorded from an object with a different point count and order as long as the points share the exact same XYZ coordinates in Modeler. For example, suppose you had a castle model with a flag on top. One could run ClothFX on the whole castle, but this would result in a large MDD file for such a tiny detail, the flag. So, in Modeler, delete the castle polys and save the flag as a different

object. Record the ClothFX calculations on the flag, and save the MDD file. Load the castle into Layout, apply ClothFX, load the MDD file calculated from the flag, set the playback to "Local" and activate Node Match. Node Match will remap the MDD data recorded from the flag-only object to the XYZ equivalents on the castle-with-flag geometry

**Shift Frames-** The user can shift the time the motion starts. Use **Shift Frames** if you need the motion to start on an earlier or later frame. By default, the number entered is regarded as a number of seconds to shift the .mdd file. If you prefer to use frames, enter the number of frames followed by the letter "f"; "-400f", for example, to shift so that the motion in the .mdd file starts at frame -400.

**Loop-** The user has the option to loop the motion file in two ways.

*Repeat-* Plays the motion from start to finish and then repeats the motion from start to finish continually.

*Oscillate-* Plays the Motion from start to finish and then plays it from the end frame backwards to the start frame and then repeats the cycle. (Ping Pong motion)

**Motion Size-** Motion Size increases the distance relationship between points in a motion file.

**Copy-** Use the **Copy** button to copy the current settings between dynamic objects.

**Paste-** Use the **Paste** button to paste the current settings between dynamic objects.

**Save-** Saves all the settings contained in the ClothFX properties.

**Load-** Loads a saved Settings file. This is similar to using a Preset.

**Scan Motion-** Scans the motion of the object and applies the scanned results (.mdd file) to the object.

This means you don't have to apply another plug-in to scan the motion (including bones, etc). It eliminates the steps of loading the motion file back in, and also enables you to quickly scan an object's motion to allow for the use of EditFX on an object even if it's not using dynamics.

**Info-** Clicking the **Info** button brings up a window that gives you the version # of ClothFX and the MDD file information.





## EditFX Tab

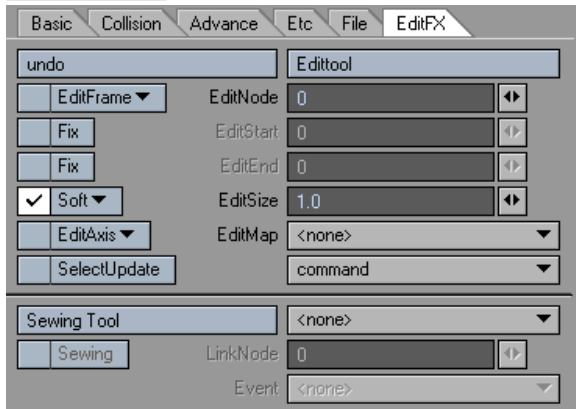


Figure 19: EditFX Tab

**EditTool-** Activates the ability to edit points. You will be able to visually see that you're in **Edit Mode** in the viewport. All the points that make up the object will become highlighted and all the **EditTool** functions will become active.

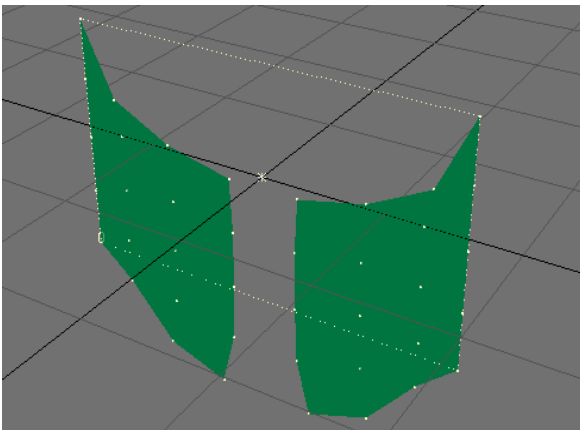


Figure 20: Sew function applied

**Undo-** The **EditTool** has its own undo. Click this button to undo the last edit that was made. You can only undo one edit.

**EditNode-** Specify the ID number of the point to be edited. If the **EditTool** is activated, you can select the target point from the Layout viewports by clicking on the point.

**EditFrame-** There are 4 modes to edit the Node's (point) motion path.

*All-* will edit the entire Motion Path.

*After-* will edit the current frame and everything after it.

*Before-* will edit the current Frame and everything before it.

*Current-* will edit only the current Frame.

**EditStart/EditEnd-** Use the **EditStart** and **EditEnd** Fix options to set the range of frames that the user will be able to edit. All other frames will be fixed (locked).

**EditSize-** (Edit Range) This field changes the influence range.

**Edit Falloff-** This drop down is located to the left of **EditSize**. This will change the **Falloff** settings to either **None**, **Soft**, **Linear**, or **Hard**. As shown in Figure 19, by default the setting is **Soft**.

**None-** All vertices are influenced; in other words the whole mesh moves.

**Linear-** Deformation in the range of EditSize is attenuated linearly.

**Soft-** Deformation is attenuated smoothly in the range of EditSize.

**Hard-** Deformation occurs uniformly in the range of EditSize, and is not attenuated.

**EditAxis-** This sets the axis/axes in which the Node (point) can be edited. (All, X, Y, or Z)



**NOTE:** You do not have to re-calculate when you move the path or keyframe data with **EditAxis**.

**EditMap-** This option lets you limit the portion of the object that is editable by giving the user the option to edit by **Vertex Maps**.

**SelectUpdate-** When activated, you can re-calculate only the parts selected by EditNode. To re-calculate, click **Calculate** with **SelectUpdate** checked.

**Command-** The **Command** drop-down offers two commands for use during the editing process, **Smooth** and **Makepath**.

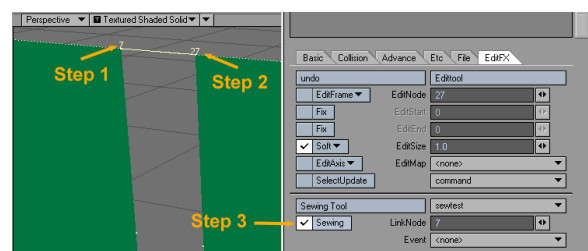
**Smooth-** Referring to the current frame, the movement of the parts that were selected by EditNode will be smoothed. When EditStart and EditEnd are not specified, the movement of the selected parts to the current position will be smoothed across all the frames. When EditStart and EditEnd are specified, it will make the position information smooth from the specified frames to the current frame.

**Makepath-** This command will create a Null object whose path is identical to the motion path of the selected part.

**Sewing Tool-** Enables **Sewing Mode**. **Sewing Mode** gives the user the ability to sew Nodes (points) together.

**Sewing-** This button is the third step in a three-step process of sewing Nodes together using the **Sewing** tool.

**Steps to Sewing:** "Left Click" on the first point and "Right Click" on the second point. Then click on the **Sewing** button.



**Current Sew Object-** Drop down list located to the right of the **Sewing** tool enables the user to choose which object is being sewn.

**Link Node-** Displays the Node that has been selected.

**Event-** This drop down menu determines whether the linked nodes will be Cut by an event or sewn by an event.

**Cut by Event-** When an Event object passes through the Nodes (points), they will break apart and become separated.

**Sewn by Event-** When an Event object passes through the Nodes (points) they will be sewn together (Merged).

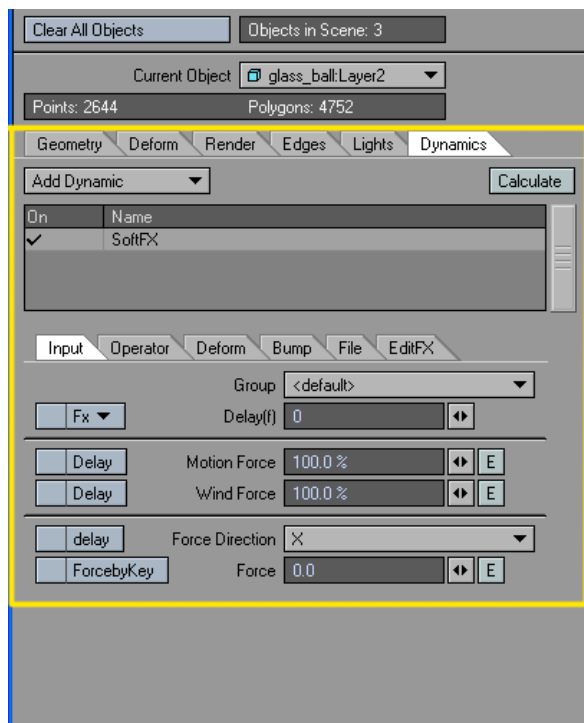


## Soft Body Dynamics

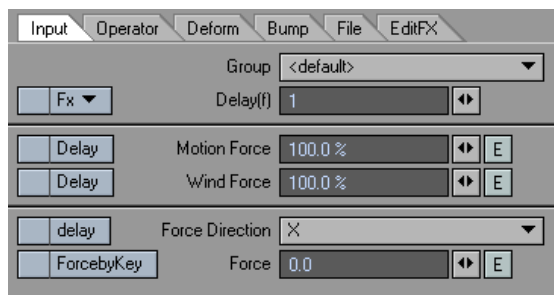
Soft Body Dynamics (**SoftFX**) are used to set physical behavior characteristics and to calculate physical simulations for an object that is intended to have an elastic nature through all or a portion of its volume. Generally a "soft body" maintains its volume, but the shape will vary based on motion and on forces being applied in the scene, including collision with other objects. Some examples:

- Jello wiggling on a plate
- The Pillsbury Dough Boy's tummy jiggling as he jumps around
- A flat tire flopping as the vehicle comes to a stop
- A ball deforming as it bounces on a floor

**SoftFX** is located in the **Dynamics Tab** of the **Object Properties Panel**. Choose **Soft** from the **Add Dynamic** drop down list and then follow that by clicking on **SoftFX** in the dynamics list window to open up the **SoftFX** options.



## Input Tab



**Group**- you can group Dynamic objects with a user-defined name to prevent unwanted interaction. This also works with ParticleFX controllers. This becomes very handy when you are working on a complex scene and you want certain Wind emitters to affect only certain objects.

**Delay**- Delays the deformation by a set amount of frames. Delay will not take effect until the Delay check box is selected for motion force, wind force, or wind force direction.

**FX Button**- Allows you to use a **Vertex Map** to control the value of an attribute (spring, viscosity, etc) or a function (delay, wave size, compress bump).

From the **Effects** drop down menu the user can select a **Vertex Map** as well as change the base value.

**Vertex Map** – Will be displayed by name and type.  
Example: **LeftArm/Weightmap**

**Weightmap** – The weight value of each point is defined in Modeler when you create a **Weight Map**. In Layout you cannot edit **Vertex Map** values. But the **FX** button allows you to add a modifier, that combined with the original values of the **Vertex Map** provides a way to adjust values of the selected attribute/function without changing the original **Vertex Map** value.

**Pointset** – A pointset is created in Modeler. In Layout you cannot add or subtract points from a pointset. But the **FX** button allows you to add a modifier, that combined with the pointset (value of 100%) provides a way to adjust values of the selected attribute/function.

**Surface** – Surfaces are created in Modeler. In Layout you cannot add or subtract points from a surface. But the **FX** button allows you to add a modifier, that combined with the surface (value of 100%) provides a way to adjust values of the selected attribute/function.

**UV** – UVs are created in Modeler. In Layout you cannot add or subtract points from a UV. But the **FX** button allows you to add a modifier, that combined with the UVs value provides a way to adjust values of the selected attribute/function.



**NOTE:** You can always check value information for points in the Point Info windows.

**Base** – The base value is 0% by default. Changing the base value changes the effective range.

Example: Base=0% (0 – 100), Base=20% ( 20 – 100).



**NOTE:** For those interested this is the equation used to calculate the vertex point value.  $Vp = Vmap * value + (1 - Vmap) * base$ .

SoftFX has 3 inputs: Motion, Wind and Force.

**Motion Force**- Uses the **Delay** setting based on the motion of the object. The user has the ability to change the amplitude as well as apply an envelope.

**Motion Force Delay check box**- Activates the **Delay** setting for Motion Force.

**Wind Force**- Uses the **Delay** setting based on wind present in the scene. The user has the ability to change the amplitude as well as apply an envelope.

**Wind Force Delay check box**- Activates the **Delay** setting for Wind Force.





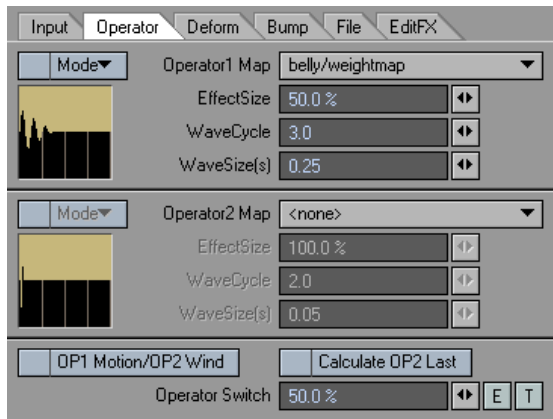
**Force Direction**- Forces the object to move along the axis selected (X, Y, or Z).

**Force Direction Delay**- Activates the **Delay** setting for Direction Force.

**Force By Key**- Only works when an envelope is applied. Takes the keyframes in the envelope and applies a force for every key.

**Force**- The value that Force Direction uses. The user has the ability to apply an envelope.

## Operator Tab



**Operator 1 and 2 Map** – a deformation can be applied to nothing, everything or a particular **Vertex Map**.

**Mode**- How the **Operator Map** affects the objects deformation.

*None* – default (no change)

*Square* – Dampens the **Operator Map** so that the map is affecting the object deformation 0% at the start of the map to 100% at the end of the map.

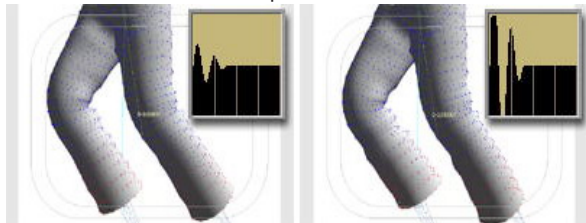
*Invert*- Reverses the **Operator Map's** original effect on the objects deformation.

*Quad*- Divides the map's effect on deformation into 4 equal regions.

*Inv Square* – Reverse of "Square". 100% at the start of the map and 0% at the end.

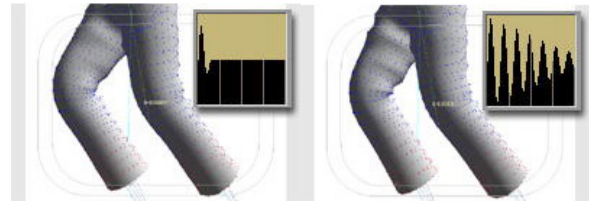
*Inv Quad* – Reverse of Quad.

**Effect Size**- increases the amplitude of the deformation.



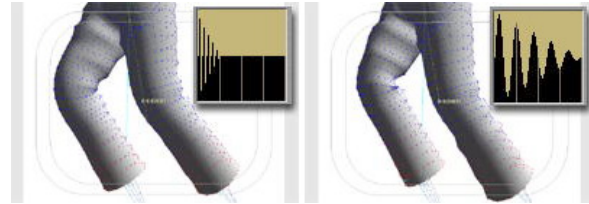
Effectsize 50%, Effectsize 200%

**Wavecycle**- number of cycles inside the deformation.



Wavecycle 1.0, Wavecycle 8.0

**Wave Size(s)**- the frequency of the deformation.



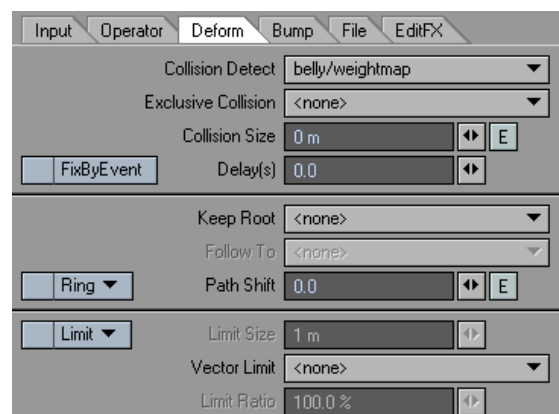
Wavesize 0.1, Wavesize 0.4

**Op1 Motion/Op2 Wind**- Limits the operator 1 map to the motion of the object and limits the **Operator 2 Map** to the wind. By activating these, the motion input affects the **Operator 1 Map**, and the wind input affects the **Operator 2 Map**.

**Calculate Op2 Last**- Applies operator 2 after everything in **SoftFX** has been calculated.

**Operator Switch (blend operator)**- if set to 50% it uses both **Operator 1** and **Operator 2 Map** equally. If set to 0%, it uses 100% of **Operator 1 Map**. At 100%, it uses 100% of **Operator 2 Map**. You can also have a **Texture Map** which will control this blending.

## Deform Tab

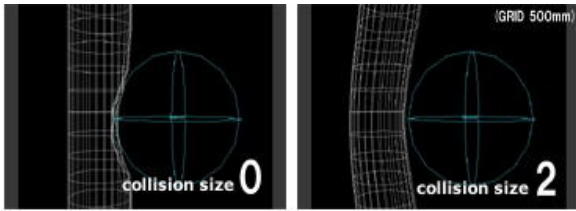


**Collision Detect**- reflects the influence of other objects upon the motion of an object. This lets you create complex motions caused by obstacles. The simulation is performed by taking an object that collides (the collision object) with the elastic body model (dynamic object) into the calculation.

**Exclusive Collision**- Selecting an item from this list will exclude all other collision objects from the calculation.



**Collision Size-** Determines how the force of the collision is distributed across the SoftFX object.



**Delays-** Delays the reaction of the object returning to its original shape.

**Fix By Event-** This function will fix points that pass through a collision object that is set to "Event" mode.

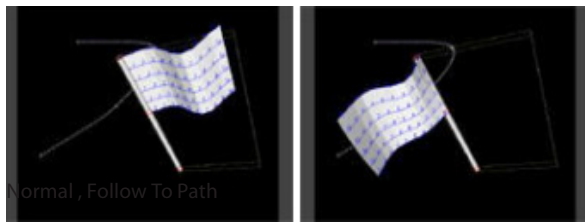
**Keep Root (keep original size)-** object will keep its original size through SoftFX calculations. For example, you start with a 12" ruler. With Keep Root unchecked, this ruler might stretch to 14". If Keep Root is checked it will maintain 12".

**Follow To-** Uses these modes:

*Path* - Deforms the object along its motion path.

*Wind* - Deforms the object in the direction of the wind.

*Gravity* - deforms the object based on the gravity emitter.



**Path Shift-** shrinks the root size (original size).

**Ring-** Ring does for a cylinder what Keep Root does for a straight line. You are limited to only 4 rings for calculation but can use **Hard link** or **Meta link** to apply it to objects with more than 4 rings.



NOTE: A Circle is one Ring. A Cylinder contains 2 Rings.

**Limit Size-** Drop down menu- When activated:

*Max Size*-Will limit the size of the deformation by clipping it to the limit size.

*Resize*- Will limit the size of the deformation by shrinking it to the limit size.

**Limit- Amplitude** setting for the Limit.

**Vector Limit-** Constraining the movement to a particular axis, Normal, and object radius.

**X-** Becomes deformed only in the x direction.

**Y-** Becomes deformed only in the y direction.

**Z-** Becomes deformed only in the z direction.

**X (local)-** Becomes deformed only in the local x direction of object.

**Y (local)-** Becomes deformed only in the local y direction of object.

**Z (local)-** Becomes deformed only in the local z direction of object.

**Xy-** Becomes deformed only in the XY direction.

**Yz-** Becomes deformed only in the YZ direction.

**Zx-** Becomes deformed only in the ZX direction.

**Xy (local)-** Becomes deformed only in the local XY direction of object.

**Yz (local)-** Becomes deformed only in the local YZ direction of object.

**Zx (local)-** Becomes deformed only in the local ZX direction of object.

**Object radius-** Becomes deformed only in the spherical direction, which centers pivot.

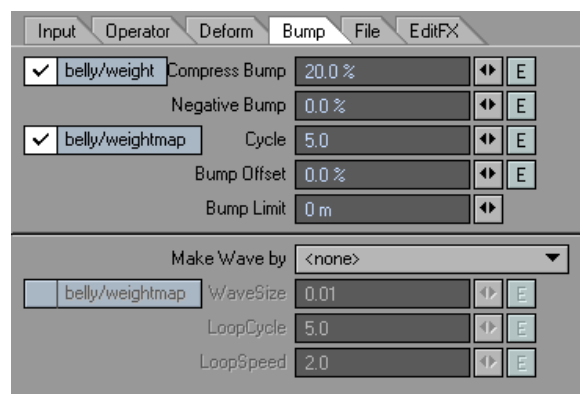
**Normal vector-** Becomes deformed only in the normal direction.

**NV-front-** Becomes deformed only in the direction of normal grain side.

**NV-back-** Becomes deformed only in the direction of normal reverse side.

**Limit Ratio**-percentage of the Vector Limit.

## Bump Tab



**Compress Bump-** Applies bump (bulge) to polys that are under compression (push) and maintains the object's volume.

**FX Button-** Allows you to use a **Vertex Map** to control the value of an attribute (spring, viscosity, etc) or a function (delay, wave size, compress bump),

From the **Effects** drop down menu the user can select a **Vertex Map** as well as change the base value.

**Vertex Map** - Will be displayed by name and type.



Example: "LeftArm/Weightmap

**Weightmap** – The weight value of each point is defined in Modeler when you create a **Weight Map**. In Layout, you cannot edit **Vertex Map** values. But the **FX** button allows you to add a modifier, that combined with the original values of the **Vertex Map** provides a way to adjust values of the selected attribute/function without changing the original **Vertex Map** value.

**Pointset** – A pointset is created in Modeler. In Layout, you cannot add or subtract points from a pointset. But the **FX** button allows you to add a modifier, that combined with the pointset (value of 100%) provides a way to adjust values of the selected attribute/function.

**Surface** – Surfaces are created in Modeler. In Layout you cannot add or subtract points from a surface. But the **FX** button allows you to add a modifier, that combined with the surface (value of 100%) provides a way to adjust values of the selected attribute/function.

**UV** – UVs are created in Modeler. In Layout you cannot add or subtract points from a UVs. But the **FX** button allows you to add a modifier, that combined with the UVs value provides a way to adjust values of the selected attribute/function.



NOTE: You can always check value information for points in the Point Info windows.

**Base** – The base value is 0% by default. Changing the base value changes the effective range.  
Example: Base=0% (0 – 100), Base=20% ( 20 – 100).



NOTE: For those interested this is the equation used to calculate the vertex point value.  $V_p = V_{map} * value + (1 - V_{map}) * base$ .

**Negative Bump**- Applies negative bump (dented) to polys under tension (pull).

**Cycle Bump**- Generates bump(s) across the **Vertex Map** selected. The higher the cycle the more bumps generated.

**Cycle**- Number of bumps generated by Cycle Bump.

**Bump Offset**- Allows you to create a bump/negative (bulge//dent) without any compressed polys.



NOTE: there must be a compressed bump or negative bump value for Bump Offset to function.

**Bump Limit**- Limits the maximum effect of the bump value(s).

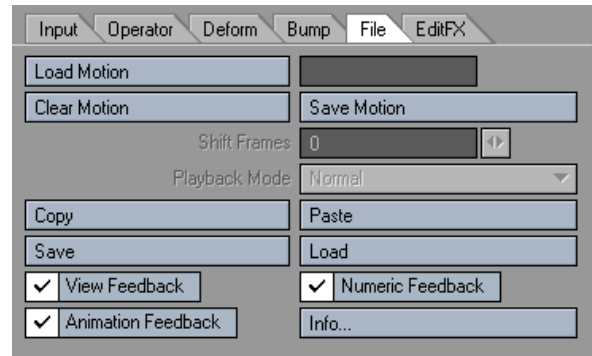
**Make Wave By**- Uses the **Vertex Map** information to create a rolling wave.

**Wave Size**- Amplitude of the wave created. At a higher value the total height of the wave increases.

**Loop Cycle**- Effects number of waves generated.

**Loop Speed**- Effects how fast the waves move across the object.

## File Tab



**Load Motion**- Loads a previously generated motion file.

**Clear Motion**- Clears a loaded Motion file.

**Save Motion**- saves the generated motion to a file, which will be used subsequently to deform the object.

**Shift Frames**- Shifts the starting playback frame of SoftFX data. By default, the number entered is regarded as a number of seconds to shift the motion of the .mdd file. If you prefer to use frames, enter the number of frames followed by the letter "f"; "-400f", for example, to shift so that the motion in the .mdd file starts at frame -400.

Playback Mode

**Normal**- Plays back setting all motion to .mdd file

**Local**- Plays back based on the local translations (move rotate and scale) of the object. Allows an .mdd deformation on an object while respecting new motion applied to that object.

**Copy**- Use the **Copy** button to copy the current settings between dynamic objects.

**Paste**- Use the **Paste** button to paste the current settings between dynamic objects.

**Save**- Saves all the settings contained in the **SoftFX** properties.

**Load**- Loads a saved settings file. Similar to using a Preset.

**View Feedback**- Set the feedback in Layout when controlling HardFX. When the display is poor or the display speed is slow, you uncheck this.

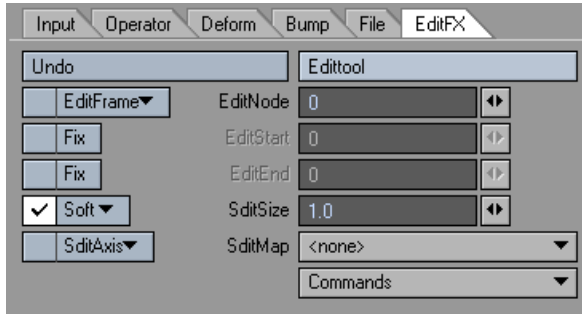
**Numeric Feedback**- Set the numerical feedback in Layout when controlling HardFX. When the display is poor or the display speed is slow, you uncheck this.

Animation Feedback

**Info**- Clicking the **Info** button brings up a window that gives you the version # of ClothFX and MDD file information.



## EditFX Tab



**Edit Tool-** Activates the ability to edit points. You will be able to visually see that you're in **Edit Mode** in the viewport. All the points that make up the object will become highlighted and all the **Edit** tool functions will become active.

**Undo-** The **Edit** tool has its own undo. Click this button to undo the last edit made. You can only undo one edit.

**Edit Node-** Specify the ID number of the point to be edited. If the **Edit** tool is activated, you can select from the Layout by clicking on the point.

**Edit Frame-** There are 4 modes to edit the Node's (point) motion path.

*All-* will edit the entire Motion Path.

*After-* will edit the current frame and everything after it.

*Before-* will edit the current Frame and everything before it.

*Current-* will edit only the current Frame.

**Edit Start/Edit End-** Use the **Edit Start** and **Edit End** Fix options to lock off frames from being edited. This is the range of frames that the user will be able to edit. All other frames will be fixed (locked).

**Edit Size-** (Edit Range) This field changes the influence range.

**Edit Falloff-** This drop down is located to the left of Edit Size. This will change the **Falloff** settings to either **None**, **Soft**, **Linear**, or **Hard**. By default the setting is **Soft**.

*None-* It has an influence on all apexes, in other words the whole mesh moves.

*Linear-* Deformation in the range of editSize is attenuated to linear.

*Soft-* Deformation is attenuated smoothly in the range of editSize.

*Hard-* Deformation occurs uniformly in the range of editSize, and is not attenuated.

**Edit Axis-** This limits the axis in which the Node (point) can be edited. (All, X, Y, or Z)

**Edit Map-** This option lets you limit the portion of the object that is editable by giving the user the option to edit by **Vertex Maps**.

## Command

**Smooth-** Referring to the current frame, the movement of the parts that was selected by editNode will be smooth. When editStart and editEnd are not specified, the current position will be copied to all the frames. When editStart and editEnd are specified, it will make the position information smooth from the specified frame to the current frame.

**Makepath-** This command will create a Null object whose path is identical to the motion path of the selected part.



## Hard Body Dynamics

Hard Body Dynamics (also called Rigid Body Dynamics or Geostatic Simulation) are used to set the physical interactions and characteristics of bodies that are solid, or fixed in shape. Some examples:

- Bowling Pins
- Windowpane breaking
- Objects exploding into pieces
- Dominoes
- Pinball machine

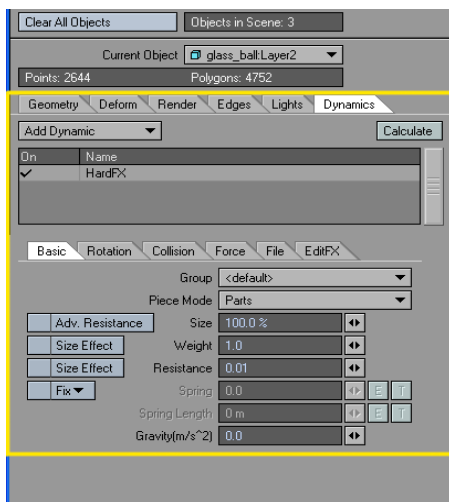
**HardFX** will easily create rigid body characteristics and behavior when applied to a goal object in a scene. You then need to set the **Weight**, **Reflection**, and **Collision** settings and calculate.

The object that has an impact will rotate and move according to its shape and size. It will automatically calculate the result so that the outcome will be more realistic.

You do not have to create individual objects for the pieces that break into parts for calculation. If the points are connected as a continuous surface, or if it is a group, it will be determined as parts. Also, it is possible to create the events as a trigger.

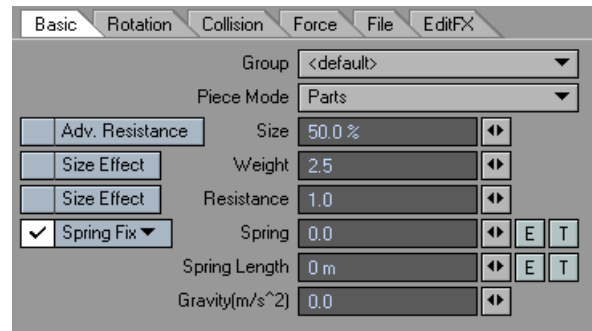
One of the big features is the **Edit tool** which allows you to adjust the path of the objects by simply dragging the mouse, and to move, rotate, and smooth the motion itself. With this addition, you do not have to calculate every time you make some changes. As a result, you can easily create the movement that you wish.

**HardFX** is located in the **Dynamics Tab** of the **Object Properties Panel**. Choose **Hard** from the **Add Dynamic** drop down list and then click on **HardFX** in the list to open up the **HardFX** options.



## Basic Tab

You can set the basic settings like air resistance, weight, and size.



**Group-** you can group Dynamic objects with a user-defined name to prevent unwanted interaction. This also works with ParticleFX controllers. This becomes very handy when you are working on a complex scene and you want certain Wind emitters to only affect certain objects.

**<default>** Includes all groups.

**<new group>** Creates an individual group and can associate functions within a group.



**NOTE:** You can associate with ParticleFX's functions using group function. If it is in the same group, you can associate settings. Also, just by removing from the group, you can break the association.

**Piece Mode-** Setting for how HardFX will treat the object.

**1piece-**Make all polygons as one piece.

**Parts-** Make polygons that have common points as one piece.

**1piece>parts (collision)-**When it starts, it will be in 1 piece mode, but with Collision, it will change to parts mode.

**1piece>parts (event)-**When it starts, it will be in 1 piece mode, but with a Collision event, it will change to parts mode.

The event's detection is made by each part's pivot.

**Adv. Resistance-** Changes the calculation of the air resistance to an Advanced Mode. Usually, the value for air resistance is set only in Resistance; however, by turning **Adv. Resistance** on, it will include the object's XYZ ratio. This helps create realistic physical movement like leaves falling.



**NOTE:** If the value for the resistance is 0 or the ratio of each part is the same, even if you turn on Adv. resistance, you will not get any effect. In the case of a sphere (with same radius), since XYZ ratio are the same, you will not get any effect.

**Size-** used to set the size of the object that will be used for collision. The size will be the percentage of each part.

**Size Effect (weight)-** Proportionate to each part's weight to size. By turning it on, you can set weight proportional to the size (scale).





**Weight-** Sets the mass for each part. A higher value will increase the velocity from the items origin, while a lower value slows the velocity.

**Size Effect (Resistance)-** Controls the relationship of each part's resistance to size. By turning it on, you set resistance proportional to size (scale).



NOTE: If the value for the resistance is 0 or the ratio of each part is the same, even if you turn on Size Effect (resistance), you will not get any effect.

**Resistance-** This one is not a characteristic of the object or part itself, even though it can be set per object, per group or per part. **Resistance** controls the amount of resistance of the environmental medium in which the simulation is taking place. For example, this affects how quickly an object or part moves through the virtual air in the scene. The default is 1.0; raising this number increases the resistance to the item's movement and lowering to a number between 1.0 and 0.0 decreases the resistance. This allows the simulation of a wide range of gaseous or even liquid media as the environment for the physical simulation. To simulate behavior within a normal atmosphere at sea level, a value of .14 would be appropriate. A value of 1.0 gives the medium a character close to that of molasses. This can be used as a way to differentiate the behavior of differing parts or objects within a given scene.



Note: If you are coming to LightWave from Maya, the function that is most similar from that application is "Damping."

## Fix

**Sets a Fix** each part's pivot.

*Free-* Will not fix.

*spring fix-* Fixes with a spring, causing a back and forth motion.

*fix-* Fixes.

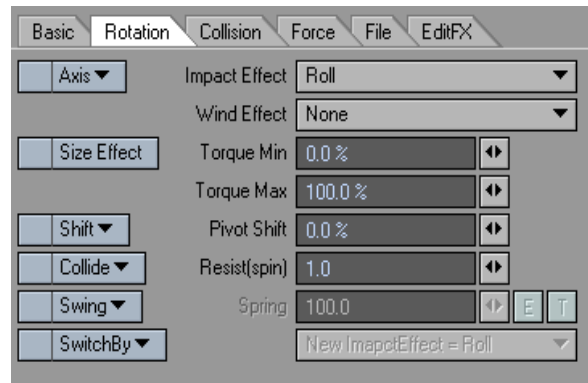
**Spring-** When Spring Fix is selected, this represents the power of the Spring, which acts like a rubber band.

**Spring Length-** Setting for length when spring power is 0. Spring length is the length of the spring and with the length of spring offset of the pivot, it moves freely. Depending on the value of spring, spring length will vary.

**Gravity-** Setting for downward (-y) gravity.

## Rotation Tab

The **Rotation Tab** will set the parameters for rotation.



**Axis-** Restricts rotation along the specified axis.

*Free-* Not restricted

*x axis-* Allows rotation only along the X axis.

*y axis-* Allows rotation only along the Y axis.

*z axis-* Allows rotation only along the Z axis.

*xz axis-* Allows rotation only along the X and Z axis.

*xy axis-* Allows rotation only along the X and Y axis.

*yz axis-* Allows rotation only along the Y and Z axis.

**Impact Effect-** Settings for the effect on rotation when there is an impact.

*Roll-* Creates rotation depending on the size and velocity.

*Force-* Creates rotation depending on the impact. This is useful for regular physical movement.

*None-* No effect to the rotation.

*Stop-* Stops the rotation.



NOTE: If the Impact effect mode is roll, you cannot assign Y axis rotation to the Axis. Roll refers only to X and Z rotation.

**Wind Effect-** Setting for rotation from the effect of Wind.

*None-* Does not effect the rotation.

*Accelerate-* Strengthens the current rotation.

*Roll-* Creates perpendicular rotation to wind and moving direction.

*Spin-* The object will rotate around the axis that it is moving along..

**Size Effect (torque)-** Makes a ratio to each part's torque to size. By turning it on, you can set torque proportional to the size.

**Torque Min-** Sets the minimum value for force for the rotational





direction. By combining it with torque Max and giving a different torque to each part, you obtain a different rotation for each part.

**Torque Max-** Sets the maximum value for force for the rotational direction. By combining it with torque Min and giving a different torque to each part, you obtain a different rotation for each part.

**Shift-** Sets the direction of the pivot shift.

*Random-* Moves the pivot point in a random direction.

*x shift-* Moves the pivot point in the X direction.

*y shift-* Moves the pivot point in the Y direction.

*z shift-* Moves the pivot point in the Z direction.

*object pivot-* Sets the part's pivot as the object's pivot.

**Pivot Shift-** Moves the pivot of the rotation towards the direction set in shift. The amount is the percentage of the part's size.

**Collide-** Restricts the movement of Resist (spin).

*Free-* Resist (spin) will always be on.

*Collide-* Resist (spin) will turn on only when a collision occurs.

*Collide (axis)-* Resist (spin) will turn on only when collision occurs and the axis is perpendicular to the collision surface.

**Resist (spin)-** Applies resistance to the rotational direction. This is useful when speed decreases by friction.

**Swing-** Fixes each part's position. Useful when you want to create a *jiggly* movement.

*Free-* Will not fix the position.

*Swing-* Turns on swing for fixing the position.

*Swing (wind)-* Turns on swing and also adds wind to the jiggle. When you have roll or spin selected in wind effect, the effect of the wind will be invalidated.

**Spring-** Sets the restoring force when swing is on.



NOTE: The value for spring power is for each rotational direction.

**Switch By-** Changes the setting by the specified trigger.

*Changed group-* When the group is changed in Collision, settings will change.

*Event-* When an event is given by Collision, settings will change.

**New Impact Effect-** Changes the mode of impact effect when the trigger set in "switch by" is given.

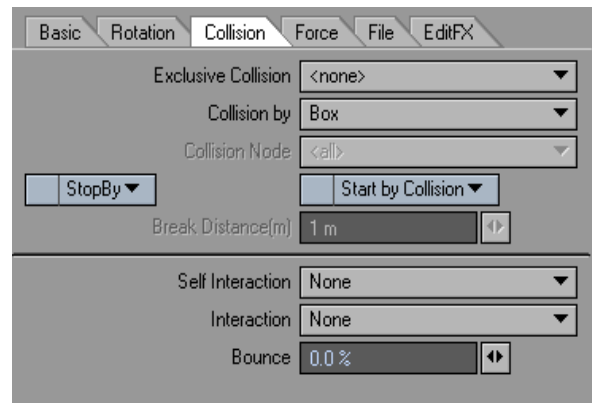


NOTE: Supplement: By default, HardFX sets the axis of the calculated parts to the object's pivot, unless it is changed by Shift or Pivot Shift.

Set shift to Object Pivot to make the object's pivot point the center of the axis.

## Collision Tab

Settings for collision.



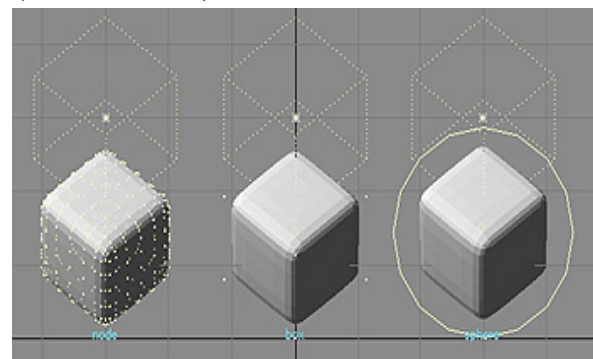
**Exclusive Collision-** Selecting an item from this list will exclude all other collision objects from the calculation.

**Collision By-** Sets the detection method for collision.

*Node-* Detects the object's vertex.

*Box-* Detects with a box vertex including the object.

*Sphere-* Detects the sphere set, in size, in **Basic Tab**.



From left to right: node, box, and sphere

**Collision Node-** When **Collision By** is set to *Node*, you can select the vertex of collision detection by using **Vertex Maps**.

**Stop By Event-** With **Collision: Event**, the movement will stop. It will move again with **startByCollision** or **RestartByCollision**. Event is determined by each part's pivot.

Start By Collision

**Start By Collision-** Usually, an animation starts from the start of the calculation, but when you select **StartByCollision**, the animation is made after the collision is determined.

**Restart By Collision-** The animation is made after the collision is determined. The difference with **StartByCollision** is that even if there is no collision, the animation is still created.



NOTE: StartByCollision is valid only with Collision. It will not work with two HardFX.



**Break Distance-** Sets the condition for Start By Collision's collision factor. The calculation will not start if the distance is shorter than the specified distance.

**Self Interaction-** Sets the interaction within the same object.

*None-* No calculations made for interaction.

*Sphere-* Detects the sphere set, in size, in **Basic Tab**.

*Box-* Detects with a box vertex including the object.

**Interaction-** Sets the interaction between objects.

*None-* No calculations made for interaction.

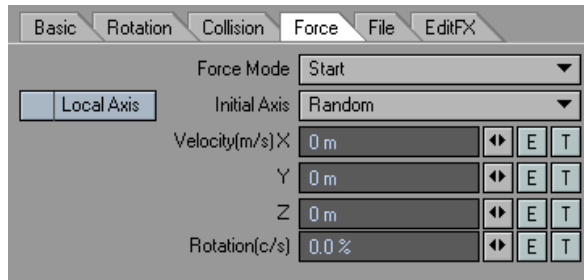
*Sphere-* Detects the sphere set, in size, in **Basic Tab**.

*Box-* Detects with a box vertex including the object.

**Bounce-** Sets the power of the interaction. When the relative speed is kept at 100% after collision, the energy of the collision is not lost.

## Force Tab

Applies settings for amount and behavior for the types of force that will be applied.



**Force Mode-** Sets the type of force that will occur.

**Start-** Force will occur at the beginning or when started by StartByCollision.

**break(1piece>parts)-** In Piece mode, when you select **1piece>parts**, force will occur when it starts to break.

**Event-** Force will occur from FX\_Collision's event mode. The detection will be each parts' pivot.

**Initial Axis-** Set the axis of the rotational power.

**Random-** Sets the axis to a random direction.

x- Sets axis to X direction.

y- Sets axis to Y direction.

z- Sets axis to Z direction.

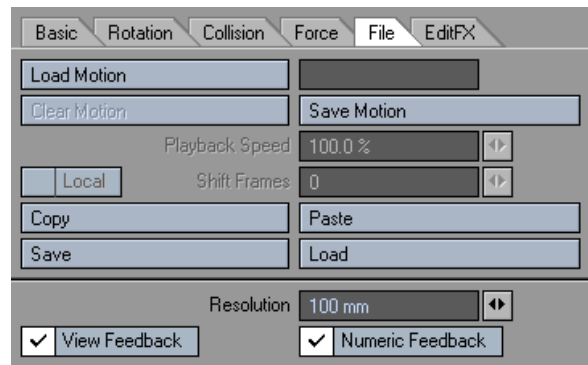
**Local Axis-** When activated, specification for Velocity(m/s) is set to the object's local axis.

**Velocity(m/s)-** Sets the velocity of the force along the specified axis.

**Rotation (c/s)-** Sets the rotational power by rotational speed (cycles/sec).

## File Tab

Manages copy, paste, and HardFX's motion data.



**Load Motion-** Loads a previously generated motion file.

**Clear Motion-** Clears a loaded Motion file.

**Save Motion-** Saves the generated motion to a file, which will be used subsequently to deform the object.

**Playback Speed-** This will set the percentage of the playback speed of HardFX data about to be played.

**Shift Frames-** Shifts the playback starting frame of HardFX data. By default, the number entered is regarded as a number of seconds to shift the motion of the .mdd file. If you prefer to use frames, enter the number of frames followed by the letter "f"; "-400f", for example, to shift so that the motion in the .mdd file starts at frame -400.

**Local-** Plays back based on the local translations (move, rotate and scale) of the object. Allows an .mdd deformation on an object while respecting new motion applied to that object.

**Copy-** Copies the current settings between dynamic objects.

**Paste-** Pastes the current settings between dynamic objects

**Save-** Saves all the settings contained in the HardFX properties.

**Load-** Loads a saved settings file. Similar to using a Preset.

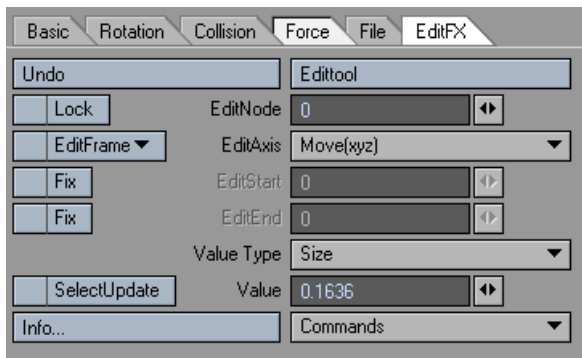
**Resolution-** Sets the accuracy of calculation by how big the maximum error is. When it approaches 0 the accuracy increases and errors will decrease, but the amount of calculation will also increase, resulting in longer processing times.

**View Feedback-** Sets the feedback in Layout when controlling HardFX. When the display is poor or the display speed is slow, uncheck this.

**Numeric Feedback-** Sets the numerical feedback in Layout when controlling HardFX. When the display is poor or the display speed is slow, uncheck this.



## EditFX Tab



**Edit Tool-** Activates the ability to edit Nodes. To edit, you will have to calculate HardFX, or load HardFX data.

**Undo-** The **Edit** tool has its own undo. Click this button to undo the last edit. You can only undo one edit.

**Edit Node-** Specify the number of the parts to be edited. If the **Edit** tool is activated, you can select from the Layout by clicking the parts.

**Lock-** Displays whether the parts specified by editNode are in Lock state or not. When parameters like size is changed by Value, it will be in Lock state. In this state, even if the size parameter is changed from the original, the Locked parameter will not change. To activate the size parameter, uncheck the Lock.

**Edit Frame-** Specify the frame(s) to be edited.

*All-* All frames will be editable.

*After-* Can edit frames before the count.

*Before-* Can edit after the count.

*Current-* Can edit only the current frame.

**Edit Axis-** Specify the direction to edit.

*move(xyz)-* Move the selected parts.

*move(x)-* Move the selected parts to X direction.

*move(y)-* Move the selected parts to Y direction.

*move(z)-* Move the selected parts to Z direction.

*rotation(x)-* Rotate the selected parts to X axis.

*rotation(y)-* Rotate the selected parts to Y axis.

*rotation(z)-* Rotate the selected parts to Z axis.

*bend(x)-* Rotate to X axis by making the center as the location data and rotation data from the selected parts.

*bend(y)-* Rotate to Y axis by making the center as the location data and rotation data from the selected parts.

*bend(z)-* Rotate to Z axis by making the center as the location data and rotation data from the selected parts.

**Fix Edit Start-** Activates **EditStart**. When activated, the current frame value is inputted in EditStart. When this is not specified, all the frames before the current frame will be edited. When you specify **fix**, you can get a smooth edit effect with EditStart's frame, but when you press Ctrl key while dragging, you will get a linear edit effect.

**Edit Start-** Fixes the specified frame value and you can get a smooth edit effect between the current frames.

**Fix Edit End-** Activates **EditEnd**. When activated, current frame value is entered in EditEnd. When this is not specified, all the frames after the current frame will be edited. When you specify **fix**, you can get a smooth edit effect with EditEnd's frame, but when you press Ctrl key while dragging, you will get a linear edit effect.

**Edit End-** It will fix the specified frame value and you can get a smooth edit effect between the current frames.

**Value Type-** Select the parts data to edit using value.

*Size-* Changes the size parameter for the parts selected in editNode.

*Weight-* Changes the weight parameter for the parts selected in editNode.

*Resistance-* Changes the resistance parameter for the parts selected in editNode.

*Torque-* Changes the torque parameter for the parts selected in editNode.

*pivot(x)-* Changes the pivot's X axis for the parts selected in editNode.

*pivot(y)-* Changes the pivot's Y axis for the parts selected in editNode.

*pivot(z)-* Changes the pivot's Z axis for the parts selected in editNode.

**Value-** Sets the value for selected item in Value type. When edited, Lock will be checked. In this state, even if the original parameter is changed, the Locked parameter will not change. To activate the size parameter, uncheck the Lock.

**Selected Update-** When activated, you can re-calculate only the parts selected by editNode. To re-calculate, activate **Calculate** with **SelectUpdate** checked. You do not have to re-calculate when you move the path or key frame data with editAxis

### Command

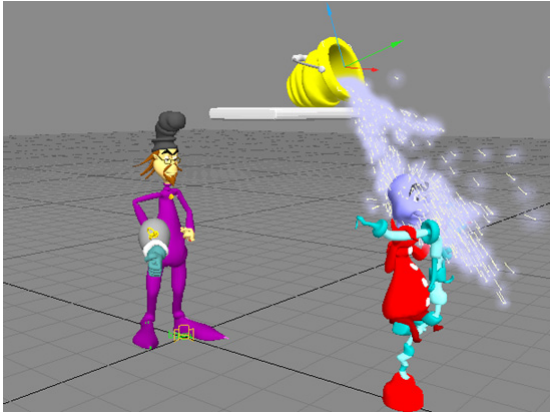
**Smooth-** Referencing the current frame, the movement of the parts that was selected by editNode will be smooth. When editStart and editEnd are not specified, the current position will be copied to all the frames. When editStart and editEnd are specified, it will make the position information smooth from the specified frame to the current frame.

**Make Path-** This command will create a Null object whose path is identical to the motion path of the selected part.



## ParticleFX (Emitter)

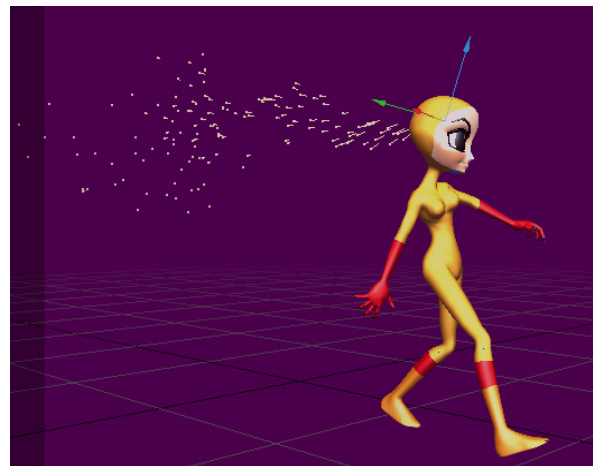
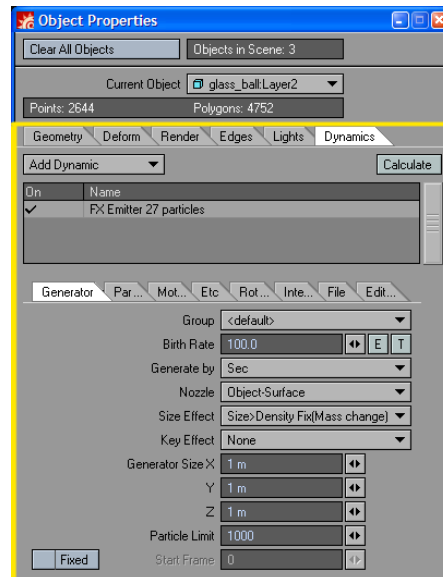
With LightWave's integrated particle system, ParticleFX, you can create scintillating effects like sparks, explosions, liquids, smoke, and much more. Because the system is integrated, there is no need for a secondary viewing interface or duplication of existing items like cameras. Everything is handled within the normal Layout interface, reducing the learning curve tremendously.



ParticleFX features several basic controllers: Emitters, Wind, Collision, and Gravity. You can add one or more of these controllers to your scene. Then, each can be set with its own parameter settings to create just the effect you are looking for.

## Adding a Particle Emitter

If you wish to add a new **Particle Emitter** to your scene, choose **Items > Dynamic Object>Particle**. You can also add a particle emitter by choosing **Emitter** from the **Add Dynamic** drop down menu on the **Dynamic Tab** in the **Object Properties Panel**.



The main difference between adding an emitter from the **Dynamic Objects** drop down menu is that you have the option of creating a **Partigon Emitter** by choosing it from the **Emitter Type** drop down menu.



## Partigons

"Partigons" are a special object type for particles. Partigons provide a means for particle systems (like ParticleFX) to automatically generate single-point polygons as needed. You can apply surface attributes to single-point polygons and, thus, make them visible when they render.



NOTE: Use the *Particle2Partigon* (Utilities>Additional>Particle2Partigon) function to convert a Particle Emitter into a Partigon Emitter.

## Emitter Types

There are two emitter types: **HVEmitter** and **PolygonEmitter**. Either type automatically uses the `FX_Emitter` custom object plug-in to create the particles. The difference between the two selections is the type of object that each uses in Layout, which affects how they render.

HVEmitters use null objects that are invisible by themselves. To render particles from HVEmitters, you must add the **HyperVoxels** volumetric plug-in on the **Volumetrics Tab** of the **Effects Panel**. Then, on the **HyperVoxels Panel**, activate the HVEmitter object. If you use gradients with **HyperVoxels** parameters, special particle-related **Input Parameters** will be available.

PolygonEmitters, on the other hand, use partigon objects—special objects designed specifically for use by particle systems. Essentially, they generate single-point polygons on the fly. Single-point polygons have a surface and you can use normal surfacing techniques. The surface will even be visible in shaded OpenGL viewports.

You can achieve many effects, like fireworks and sparks, by using PolygonEmitters alone (i.e., without **HyperVoxels**). You can save significant rendering time compared to HVEmitters, which require **HyperVoxels** for rendering.



NOTE: PolygonEmitters can use HyperVoxels just like an HVEmitter can. As such, you may want to use PolygonEmitters exclusively to avoid confusion.

If you do not wish to see the particles (e.g., when HyperVoxels are transparent), just deactivate the object on the Scene Editor. This will not affect the rendering of the HyperVoxels.

## Using an Object as an Emitter

You can use a LightWave object as an emitter by adding **Emitter** from the **Add Dynamic** drop down menu on the **Dynamic Tab** in the **Object Properties Panel**. You must also use one of the **objectNozzle** settings discussed below under **Emitter Controller**.

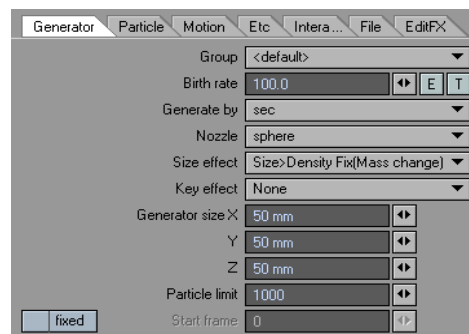


NOTE: The custom object bounding box will continue to appear.



## Generator Tab

On the **Generator Tab**, you set parameters relating to the creation of the particles.



### Group

When you work with multiple controllers, you may want to isolate specific groups. That is, perhaps you want Emitter (1) affected by Wind (1), but you want Emitter (2) to be affected only by Wind (2). This is easily accomplished by using groups.

The **<default>** group is a special global group that disregards independent groupings. Thus, if an emitter is set to a defined group (e.g., `new_group`), a wind controller set to **<default>** will still affect it.

**<default>** Includes all groups.

**<new group>** Creates an individual group. Can associate functions within a group.

Once you add a new group, it appears in the **Group** pop-up menu, where you can select it, as desired, to associate groups of controllers.



NOTE: The Group feature works in conjunction with the Other Dynamic tools! Thus, if you add an emitter, a wind controller, and a soft body (Dynamic Object) element to the same group, they are all affected by the wind!



NOTE: You can use the Group feature to disable a controller by simply setting it to a group not associated with an emitter.

The **Birth rate** works in conjunction with the **Generate by** setting. Essentially, it determines how many particles are born per the **Generate by** setting. The texture **T** option generates particles where the texture exists.

**Generate by** has several possible settings. With the **frame** and **sec** options, every new frame or second, the number of particles set as the **Birth rate** are created.

When **Generate by** is set to **speed**, it creates the particles based on the speed of the emitter, caused by moving it. You need to keyframe some movement for the emitter to use this setting. The faster the movement, the faster the particles are created. The effect is similar to salt coming out of a saltshaker—the harder you shake it, the more salt comes out.

When **Generate by** is set to **collision event**, the particles are created when a collision event occurs. You need to use a collision controller whose **Mode** is set to **event**.





When **Generate by** is set to **wind**, particles are created if the wind speed exceeds the **threshold1** setting (**Motion Tab**). (**Threshold2** has no impact here.) The force of the wind has no effect on the birth rate. The **windspeed** setting is similar, but the birth rate is affected by the wind force. The greater the wind force, the greater the number of particles born.

The **Nozzle** setting determines the shape of the emitting source. With **box**, particles are emitted from inside a box. With **sphere**, particles are emitted from inside a sphere. With **cone**, particles are emitted from inside a cone.

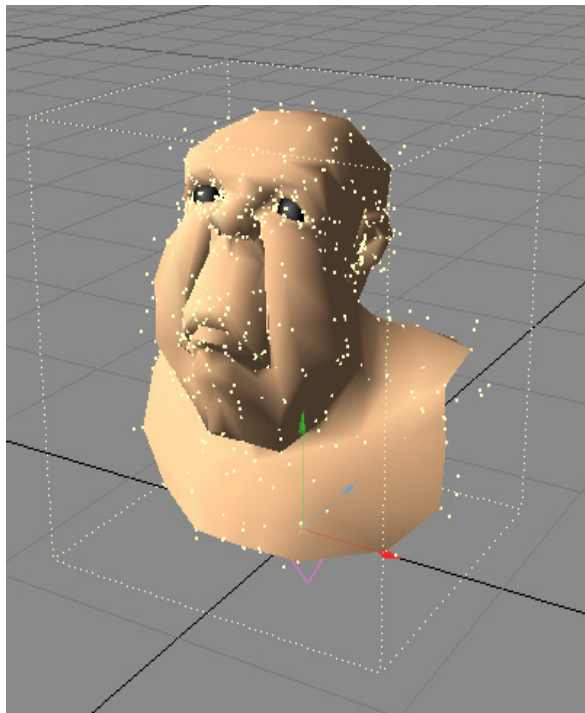


NOTE: For best results when using cone as the nozzle type, some Velocity and Vibration settings (none-zero) should be added in the Motion Tab.

You can use geometry as an emitter, by adding the **FX\_Emitter** custom object to any loaded object. This will allow you to use the **Object-vertices**, **Object-normal**, **Object-surface**, and **Object-line** nozzle types. When you use these nozzles, you can offset the center position by changing the **Center Position** values on the **Etc. Tab**.



HINT: Setting Particle Life Time to 0 means the particle will never die.



When you use the **parent-emitter**, **parent-emitter(end)**, **parent-collision**, and **child-pivots** nozzle types, they use one controller's relationship to another in order to create multiple or cascading effects.

To use **parent-emitter**, you need to create two emitters and parent one to the other. Then, set the child's **Nozzle** to **parent-emitter**. The parent's particles will spawn particles, essentially becoming emitters themselves. **Parent-emitter(end)** is similar, but emits the particles after the parent's particles die—perfect for exploding fireworks. **Parent-collision** doesn't emit the particles until the parent particle has a collision. (Make sure you click **Start** to compute the motion.)

**Child-pivots** aligns the emitter's particles with the pivot point of child objects (i.e., objects parented to the emitter). The child objects must have the **FX\_Link** motion modifier added. This allows you to use particle collision detection to move (e.g., break up) a group of objects.



HINT: Usually, you use the object modes in conjunction with an Explosion value (**Motion Tab**).

The **Size effect** setting determines how keyframing a Size change for the emitter affects the particles.

The **Key effect** option causes particles to be created when a keyframe is encountered. The **key** setting differs from **key-env**, in that **key-env** creates them in a smoothed non-linear fashion. Set to **none** to turn this option off.

The XYZ **Generator size** fields set the default size of the emitter.

**Particle limit** sets an overall limit for the number of particles that are emitted.

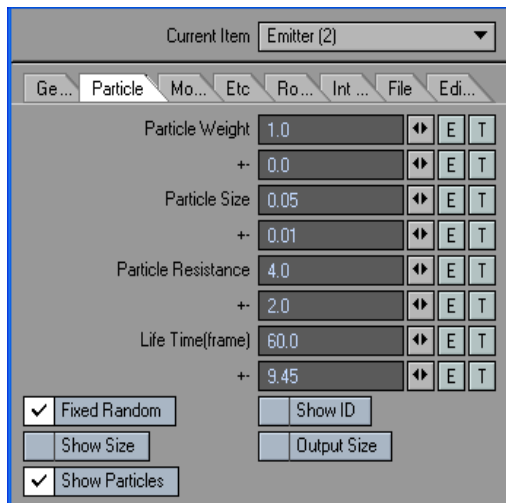
The **Start frame** field is normally an informational field that indicates the frame the particles will start being born. By default, this is the first frame setting in Layout. You can manually change this starting frame by checking the **Fixed** option and entering the desired frame in the **Start frame** field.





## Particle Tab

On the **Particle Tab**, you set parameters that describe the created particles.



**Particle weight** sets an arbitrary weighting value that will influence how properties like gravity affect the particles.

The plus(+) and minus(-) fields randomize the preceding field by adding or subtracting a value between 0 and the number entered. So if Particle weight was 1 and its randomize value was .1, the Particle weights would range from .9 to 1.1. A total particle value of zero or less will be zero.

**Particle size** affects the outer boundary used for collision detection. It can also be used by volumetric plug-ins like **HyperVoxels**.

**Particle resistance** adds an air resistance effect. Particles will move slower as you increase this value.

**Life time (frame)** sets the life of the particles in frames. Once a particle is born, it lasts only this long.

If you activate **Fixed Random**, random calculations are constant, so they yield more predictable results.

By activating the **Show Id** option, each particle will display a number in the Layout viewport indicating the index number of the particle.

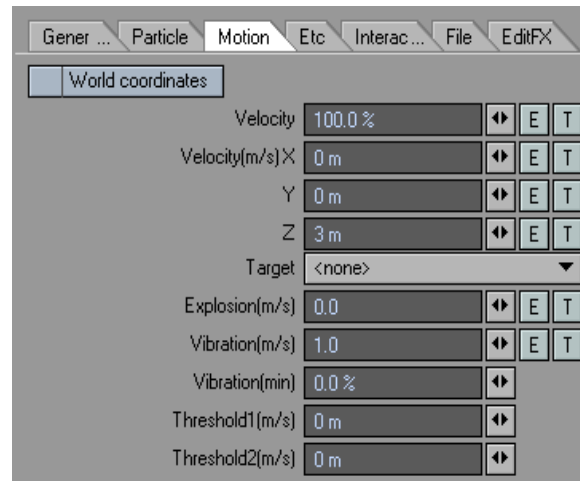
The **Show Size** option draws a wireframe sphere around each particle representing the particle's current size.

If the Output Size is not checked, the size will always be 1.

The display of particles in the OpenGL windows can now be turned ON/OFF in the Particle Tab of the Particle Property Window.

## Motion Tab

On the **Motion Tab**, you set how the particles are placed into motion.



Use the **Velocity** setting to scale the overall speed of the particles. A setting of 100% is normal. Lower values will slow down your particles and higher values will speed them up.

The **XYZ Vector** settings determine the initial direction and speed of the particle motion. Other factors, like gravity, wind, and so on, will impact the actual result, however. You may override the **Vector** settings and point the particles to an object in the scene by selecting it in the **Target** pop-up menu.

If you want the **Vector** settings to relate to World coordinates instead of the emitter's local coordinates, activate the **World coordinates** option.

The **Explosion** setting makes the particles move out from the center of the nozzle with an initial velocity equal to the value that you set.

**Vibration(m/s)** randomizes the initial trajectory of each particle. You can achieve a fountain-like effect by reducing the size of your nozzle (e.g., XYZ=.1), adding some vector speed and increasing the Vibration value.

**Vibration(min)** sets a minimum vibration amount.

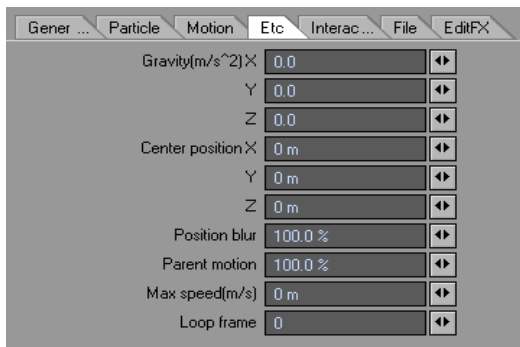
The **threshold1** and **threshold2** values set up a vibration effect range. If a particle's initial speed is under the **threshold1** value, no vibration will occur. If it exceeds the **threshold2** value, the vibration is applied. The vibration is phased in for speeds between **threshold1** and **threshold2**. Use this for effects like water coming from a garden hose, where the water tends to fan out as more water comes out.

**Threshold1** is also used when particles are generated by wind. When the wind speed exceeds the **threshold1** value, particles are generated.



## Etc Tab

On the **Etc Tab**, you set miscellaneous parameters for the emitter.



In most cases, if you want gravity, you set the **Gravity Y** value to something like -1. However, you can set up inverse gravity by using positive values, and you can apply gravity on other axes.

The **Center Position** XYZ values set the center for particle effects like Explosion (**Motion Tab**). Sometimes you may not want the particles exploding from the center of the emitter.

The **Position blur** value randomizes the initial particle position by using velocity. If you set this option to 0%, the particles are created side by side.

Use the **Parent motion** setting to control how much of the emitter's motion is applied to the particles. If it is set to 0, the particles are emitted the same no matter how the emitter may be moving.

Setting the **Max Speed (m/s)** field regulates the top speed of the particle.

The **Loop Frame** setting is used to repeat the particle generation and motion in the set number of frames. For example, if you enter a Loop Frame value of 30 frames, in a two second animation two particles will be generated with the same birth location and motion.

## Rotation Tab

The settings on this tab affect the rotation properties of each particle.

### Rotation Menu:

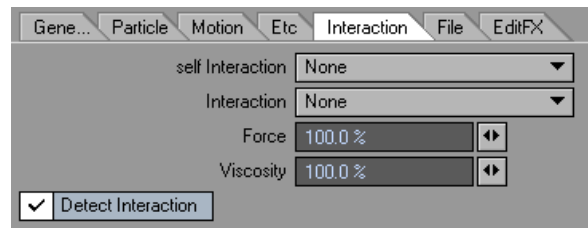
None-No rotation will be calculated for the particles.

Align to Path-The particle will align to the rotation of the path it is travelling along

Random Rotation&Scale-Will align, spin, and scale the particle randomly as it travels along the path

## Interaction Tab

The settings on this tab affect the way particles from both the same and other emitters of the group interact with each other.



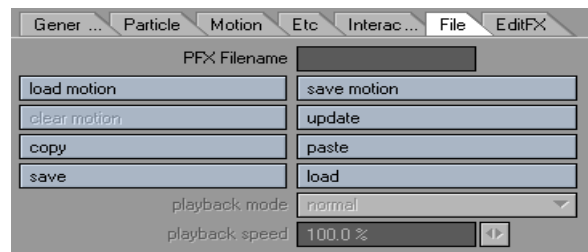
The **Self Interaction** drop-list determines how particles react to other particles of the same emitter. The **Interaction** drop-list determines how particles react to other particles from different emitters (within the same group). Options include having colliding particles **push** emitters, **bounce off**, **drag**, and **crowd** around each other.

The **Force** setting adjusts the strength of the interaction effect. Adjust the **Viscosity** setting to change the amount of resistance which particles receive when they interact with each other. This setting is useful when creating different types of liquid effects.

Turning on and off the interaction effects is done by activating and deactivating the **Interaction Detect** option.

## File Tab

The **File Tab** contains options for handling the clipboard and file commands.



You can save the particle motion of an emitter using the **Save Motion** button. The motion will take into account wind, gravity, and other options. It effectively freezes the particle motion; the particles will move just as they did when the motion was saved. Changing the **Emitter** settings, clearing/adding wind and gravity controllers, and so on, will have no impact.

**Save motion** is a great time-saver for network- rendered scenes or scenes that require heavy calculations for collisions. Once you've saved the motion, it doesn't need to be calculated again.

To use this feature, first, get your particles moving the way you want. Then, make sure your scene plays back without any pauses—this means that ParticleFX has finished its calculations. Now, you can click the **Save Motion** button to save out the .pfx file. (If you look at the emitter's **Object Properties Panel**, you'll see the .pfx file referenced in the FX emitter custom object entry.) Now save the scene and you're done. Remember, you can even clear wind, collision, and gravity controllers without affecting particle motion.



Hint: You can save time by advancing to the last frame, letting the frame refresh, then saving the PFX file.



When this feature is active, the **Save Motion** button will be ghosted. To clear a loaded motion, click **clear motion**. You can also load the motion file into another emitter using **load motion**.

The **playback mode** works with the motion and determines how it will playback. The **Normal** mode indicates that it will playback normally, starting at frame zero. With **Key** the motion will be started every time a keyframe is encountered for the emitter. **Parent-Key** is similar, but uses the parent item's keyframes.

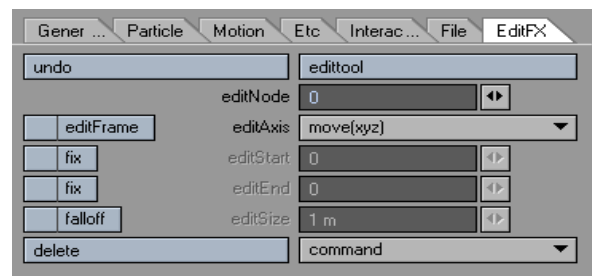
**Parent Particles** and **Parent Particles (end)** work like the identically named **Nozzle** options. Of course, the particles come from the motion file instead of **Emitter** settings. Make sure that the parent's **Particle size** is set greater than 0.

**Parent Recoded CP** uses the **Recode CP** option on the Collision controller, which must be the parent of the emitter. Playback will occur when collisions are detected. Use this for effects like splashes in water from rain drops, where one emitter provides the rain and another emitter (with this option set) creates the splashes from the detected collisions with a ground plane.

The **Copy** and **Paste** clipboard buttons allow you to copy the settings of a controller and apply them to another. Use the **Save** and **Load** buttons to add custom saved ParticleFX controllers to your scene.

## EditFX Tab

The **EditFX Tab** contains options for real-time editing of your Particles.



**Edit Tool**- Activates the ability to edit each individual particle (Node).

**Undo**- The **Edit** tool has its own undo. Click this button to undo the last edit. You can only undo one edit.

**Edit Node**- Specify the number of particles to be edited. If the **Edit** tool is activated, you can select from the Layout by clicking the parts.

**Edit Frame**- Specify the frame(s) to be edited.

*All*- All frames will be editable.

*After*- Can edit frames before the count.

*Before*- Can edit after the count.

*Current*- Can edit only the current frame.

**Edit Axis**- Specify the direction to edit.

*move(xyz)*- Move the selected parts in any direction.

*move(x)*- Move the selected parts along the X direction.

*move(y)*- Move the selected parts along the Y direction.

*move(z)*- Move the selected parts along the Z direction.

*rotation(x)*- Rotate the selected parts along the X axis.

*rotation(y)*- Rotate the selected parts along the Y axis.

*rotation(z)*- Rotate the selected parts along the Z axis.

*rotation(View Axis)*- Rotate the selected parts.

**Fix Edit Start**- Activate editStart. When activated, current frame value is inputted in editStart. When this is not specified, all the frames before the current frame will be edited. When you specify fix, you can get a smooth edit effect with editStart's frame, but when you press Ctrl key when dragging, you will get a linear edit effect.

**Edit Start**- It will fix the specified frame value and you can get a smooth edit effect between the current frames.

**Fix Edit End**- Activate editEnd. When activated, current frame value is inputted in editEnd. When this is not specified, all the frames after the current frame will be edited. When you specify fix, you can get a smooth edit effect with editEnd's frame, but when you press Ctrl key when dragging, you will get a linear edit effect.

**Edit End**- It will fix the specified frame value and you can get a smooth



edit effect between the current frames.

**Edit Size-** (Edit Range) This field changes the influence range.

Command

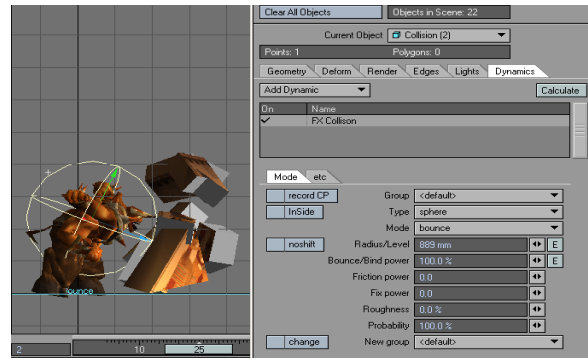
**Smooth-** Referencing the current frame, the movement of the particle that was selected by editNode will be smooth. When editStart and editEnd are not specified, the current position will be copied to all the frames. When editStart and editEnd are specified, it will make the position information smooth from the specified frame to the current frame.

**Make Path-** This command will create a Null object whose path is identical to the motion path of the selected particle.

**Delete-** You can delete individual particles from the Emitter.

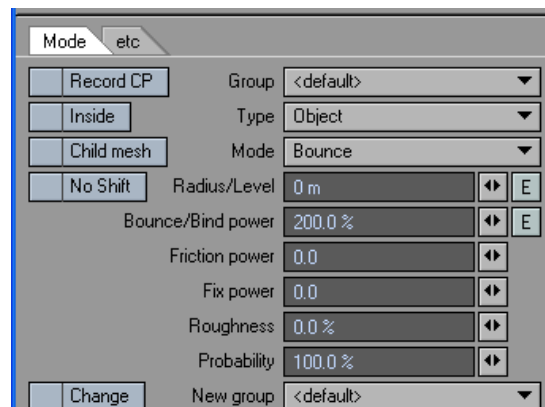
## Dynamics: Collision Controller

The Collision controller lets you add an element for the particles to bump into.



Collision controllers (**FX Gravity**) can be added in the **Dynamics Tab** of the **Object Properties Panel** or on the **Items Tab** in the **Dynamic Objects** drop down menu.

## Mode Tab



**Group-** you can group Dynamic objects with a user-defined name to prevent unwanted interaction. This becomes very handy when you are working on a complex scene and you want certain Collision objects to only affect certain objects.

**<default>** Includes all groups.

**<new group>** Creates individual group. It can associate functions within a group.



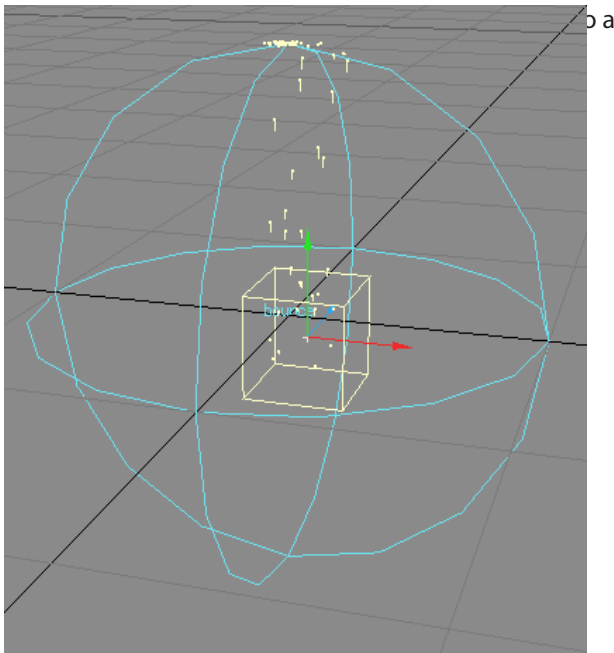
**NOTE:** You can associate with ParticleFX's functions using group function. If it is in the same group, you can associate settings. Also, if you do not want to associate, you can disregard the association by removing from the group.

The **Type** setting controls the shape of the collision object. The **Sphere**, **Box**, and **Plane** options can be used for basic primitive collision shapes. The **object**, **object-subdiv**, and **object-advanced** settings are used when you use a LightWave object for collisions. Use the **Infinite** setting for an infinite-sized collision area. This is particularly useful with the **Scatter** and **Attract** modes. (Note that the **Bound** and **Stick** modes will continue to respect the **Radius/Level** setting.)

Use the **Recode CP** option to set up an event-like trigger to be read by a child emitter's **Playback Mode**. As particles cross the collision plane created by the collision item, this event triggers the playback of a saved motion.



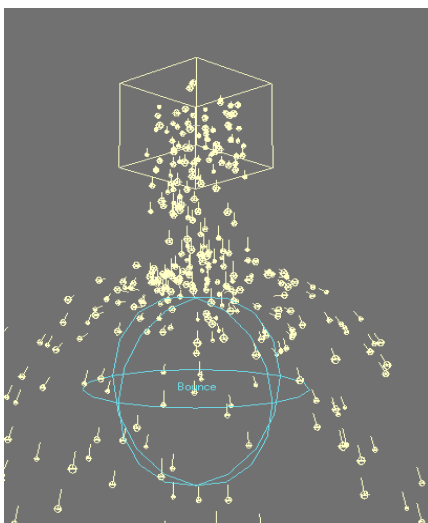
The **InSide** option causes the collision to occur inside the collision object.



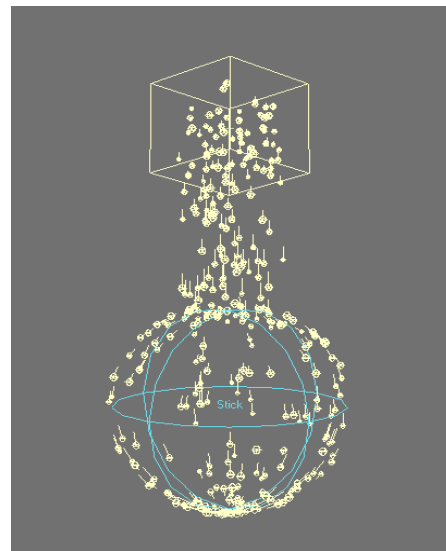
The **Noshift** option disables any offset the particles may generate in the collision process. This ensures that the particles will collide directly on the collision surface.

By activating the **Change** option, particles passing through the collision item can switch to a different group, to be affected by a different controller. The **New Group** defines which group the particles will switch to.

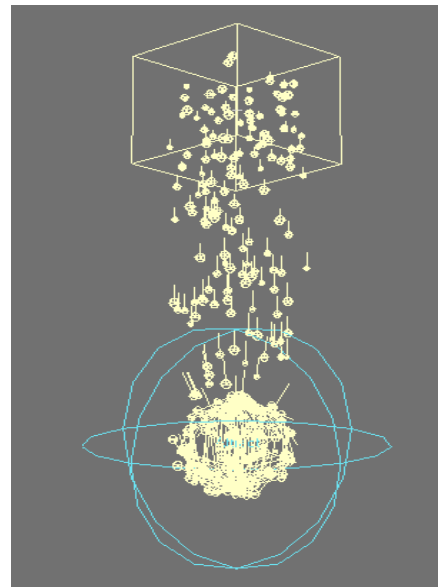
The **Mode** setting determines what happens when a collision occurs. The **Bounce** option causes the collision to change the direction and velocity of the particles. With **Stick**, the particles simply adhere to the surface. **Erase** kills the particle upon contact. The **event** mode is used when the emitter has its **Generate by** option set to **collision event**: it creates particles at the time of collision. In **Scatter** mode, the collisions occur within the interior of the collision object and not at its surface. This causes the particles to scatter. The **Attract** mode causes the particles to be attracted to the center of the collision object.



Bounce



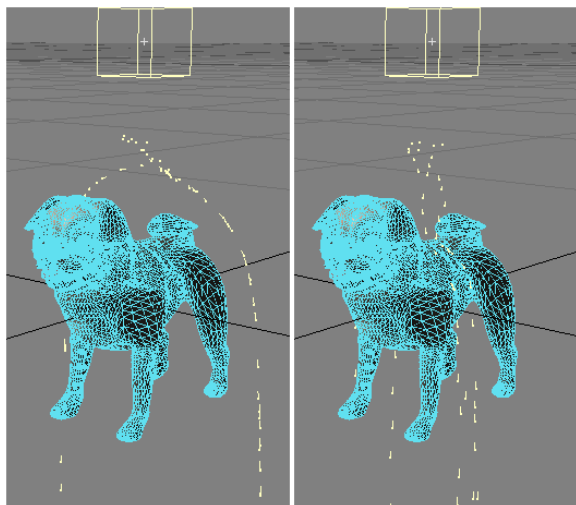
Stick



Attract

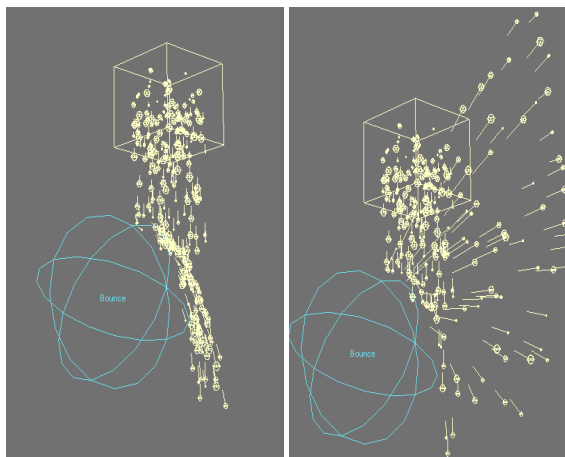


The **Radius/Level** setting sets the size of the collision object based on this radius. If **Type** is set to **Plane**, this setting controls the level of the plane. If you are using a LightWave object, this setting controls the distance from the surface where the collision occurs.



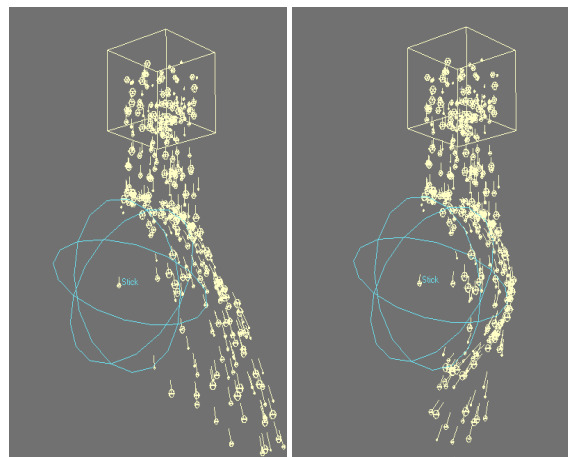
Left: Radius = 3m, Right: Radius = 0m

The **Bounce/Bind power** setting controls how particles bounce off the surface, when **Mode** is set to **Bounce**. The **Stick** mode controls the amount of attachment the particles have.



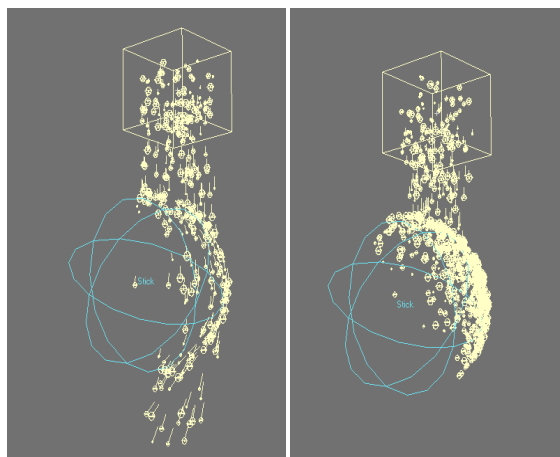
Left: Bounce Mode, Bounce/Bind: 50%  
Bounce Mode, Bounce/Bind: 400%

Right:



Left: Stick Mode, Bounce/Bind: 10%  
Right: Stick Mode, Bounce/Bind: 90%

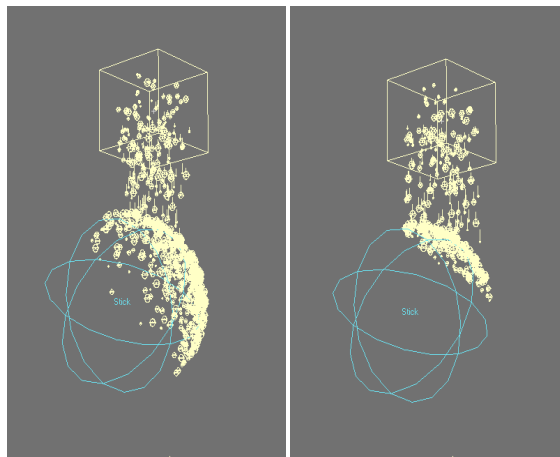
**Friction power** adds friction to the collision surface when you use the **Stick** mode, which slows the momentum of particles sliding over the surface.



Friction: 0  
20

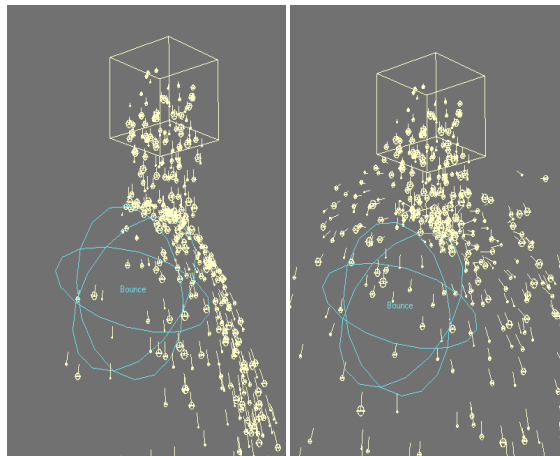
Friction:

**Fix power** causes the particles to stick to the surface and not slide around.



Left: Fix: 0; Right: Fix: 1.0

Increasing the **Roughness** setting roughens up the collision surface. **Roughness** changes particle motion after collisions, but it also depends on the **Mode** setting.

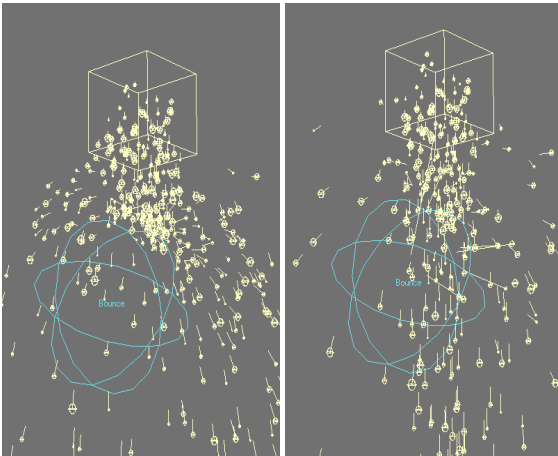


Roughness: 0  
Roughness: 200





If you reduce the **Probability** setting below 100%, you reduce the chance of a particle being treated as having collided with the collision object.



Probability: 100%

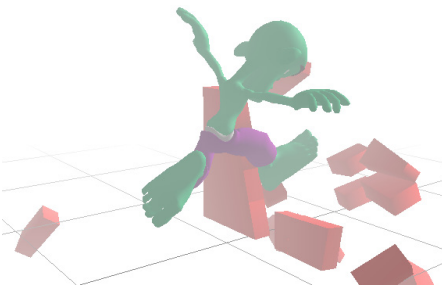
Probability: 10%

## Etc Tab

On the **Etc. Tab**, you can **Copy**, **Paste**, **Load**, and **Save** settings for the **Collision** controller.

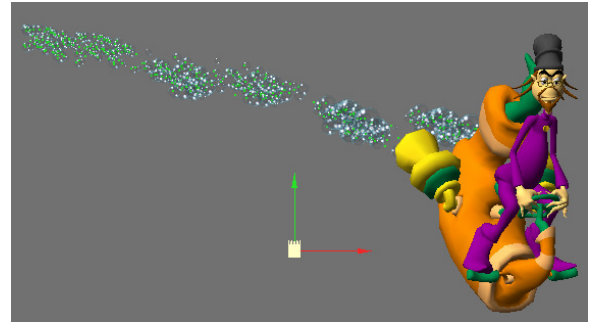
## Using an Object for Collisions

You can use a LightWave object as a Collision object by adding **Collision** from the **Add Dynamic** drop down menu in the **Object Properties Panel, Dynamic Tab**. You must set the **Type** to **object** on the object's **Collision Controller Panel**. If you use a SubPatch object, you set the **Type** to **object-subdiv** to use the subdivided mesh instead of the base polygonal object. Use the **object-advanced** type to get even more accurate collision detection.

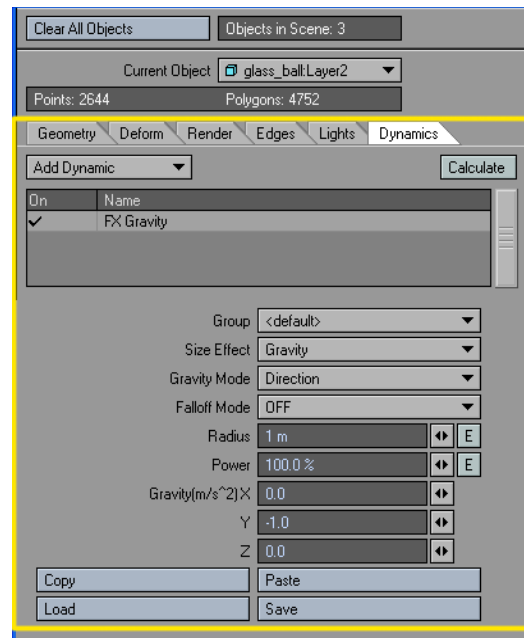


## Dynamics: Gravity Controller

A Gravity controller may be added to your scene to produce gravity-like effects on your particles and dynamic objects.

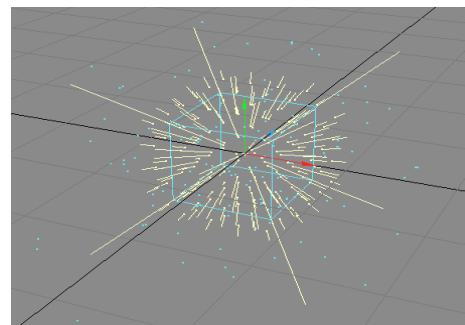


Gravity controllers (**FX Gravity**) can be added in the **Dynamics Tab** of the **Object Properties Panel** or on the **Items Tab** in the **Dynamic Objects** drop down menu.



**Size effect** determines how scaling the Gravity controller in Layout affects the gravity. The **Gravity** option scales the gravity effect. **Region** scales the overall size of the effect area.

The **Direction Gravity** mode applies the gravity in a single direction using the **Gravity** XYZ values. The **Point** mode causes the center of the Gravity controller to attract/repel the particles. Negative values attract and positive values repel.



Point Mode



The **2Pole Gravity mode** creates two attraction points, one meter on either side of the center. It can create magnetic or electric field effects.

The **Falloff Mode** setting determines how the gravity's effect tapers off at its region borders. **Linear** means the effect is linear throughout the controller's region. The **Inverse Distance** setting uses a one divided by the radius of the controller region to taper off the gravity's power from the center of the region. **Off** means no falloff. With this setting, the particles do not need to be within a controller region to be affected.

The **Radius** setting sets the radius of the controller region.

The **Power** setting scales the overall force of the gravity.

The **Gravity XYZ** settings determine the direction and force of your gravity when **Gravity Mode** is set to **Direction**. Normally, you'll want to set Y to something like -1.

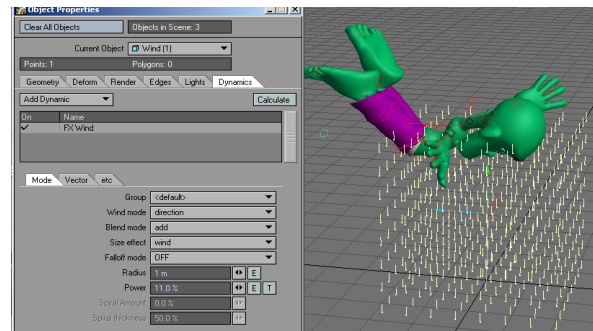
Use the **Copy**, **Paste**, **Load** and **Save** options to re-use **Gravity** settings.



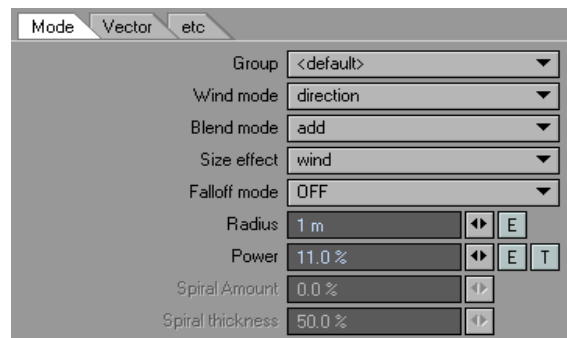
NOTE: Cloth and Hard dynamics have Gravity settings as well.

## Dynamics: Wind Controller

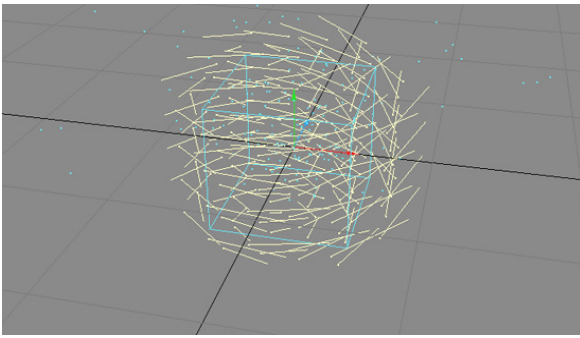
The Wind controller lets you add wind to blow your particles and dynamic objects around. The Wind controller appears in a Layout viewport as a cloud of wind indicators—a bunch of short lines with pivot point dots at one end. The length of the lines indicates the force of the wind: the longer the line, the more powerful the wind. The angle of the line indicates the direction of the wind at the line's location.



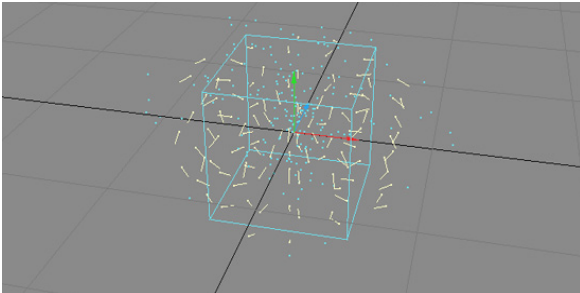
Wind controllers (**FX Wind**) can be added in the **Dynamics Tab** of the **Object Properties Panel** or on the **Items Tab** in the **Dynamic Objects** drop down menu.



The **Wind mode** setting determines the type of wind and how it changes over time. You can get a feel for what the setting does by looking at the wind indicators. There are a few special indicators, however. If you select **Turbulence**, you can set the size of the effect and direction using the **Turbulence** settings on the **Vector Tab**. If you select **Direction** as the **Wind mode**, you can use a **Texture Map** (**Texture** button on **Vector Tab**) to control the wind power and turbulence. Animate the texture to animate the wind. If you select **rotation(y)** or **doughnut**, use the **Spiral** setting at the bottom of the panel to adjust the intensity of the rotation.



rotation (y)



doughnut

**Animation Path** is a bit different. It uses handles which can be used to control the path of the wind effect. The path can be controlled by the position and rotation of the handle.



Hint: One way to add extra handles is to add a null and parent it to the wind controller. Or, you can clone one of the other handles.

The **Blend mode** sets how the wind blends with other overlapping winds. **Add** is additive. **Max** means the highest wind controls. With **overwrite**, the wind is replaced with itself. **Heavy-wind** disregards particle weight.

If winds have different **Blend mode** settings, obviously only one can control the blend. The priority order from highest to lowest is: **overwrite**, **max**, **add**, and **heavy-wind**. So, if Wind1 is set to **add** and Wind2 is set to **max**, the overlapping area will use max blend because it has the highest priority.

The **Size effect** setting determines how scaling the Wind controller in Layout affects the wind. Setting it to **Wind** scales the wind power. Setting it to **Region** scales the overall size of the effect area.

The **Falloff mode** setting determines how the wind's effect tapers off at its region borders, if at all. **Linear** means the wind is the same throughout its borders. The **Inverse Distance** setting tapers off the wind's power from its center. The **Distance** setting makes the wind falloff towards its center. **OFF** means no falloff. With this setting, the particles do not need to be within a controller region to be affected.

The **Radius** setting sets the radius of the controller region.

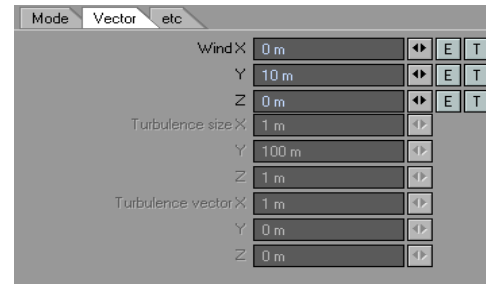
Use the **Power** setting to scale the overall force of the wind.

The **Spiral Amount** setting is used for the **rotation(y)** or **doughnut** Wind modes.

The **Spiral Thickness** setting controls the thickness of the doughnut wind.

## Vector Tab

On the **Vector Tab**, you set the direction and force characteristics of the wind controller. The vectors are scaled per second.



The **Wind XYZ** settings determine the basic direction and force of your wind.

If you are using the **Direction**, **Rotation**, **Cylinder-Explosion**, or **Hemisphere** Wind mode, use the **Envelope E** button to animate the wind direction and force. To remove the envelope, Shift+ Click the **E** button.

If you are using the **Direction** Wind mode, use the **Texture T** button to add a 3D texture that affects wind direction and force. Note that the texture is added to the base **Wind XYZ** settings. To remove the texture, Shift + click the **T** button.



NOTE: Textures are three-dimensional. Thus, adding a texture adds it to all channels. Also, for envelopes, make sure you have the correct channel in the channel bin.

If you are using the Turbulence Wind mode, the **Turbulence Vector** sets the direction and power of wind turbulence. **Turbulence size** sets the wavelength of the turbulence.





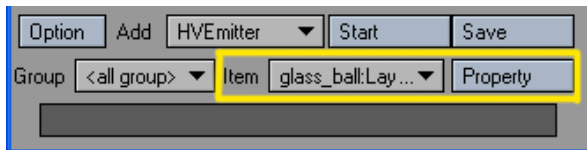
## Additional Dynamics Utilities

### Deleting an FX controller

Select and clear, just as you would any other Layout item (**Items > Clear Selected Item**).

#### To open a controller's property panel:

Choose the controller from the **Item** pop-up menu and click the **Property** button to open the **Properties Panel** for the selected controller.



NOTE: You can also display the **Property Panel** for the currently selected controller by choosing **Utilities > FX\_Property** or by accessing the **Dynamics Tab** in the **Object Properties Panel**.

### The Start Button

Some effects require a "pre-calculation" before they can be previewed in your animations. These are instances where you have particles interacting or have post-deformed geometry. By activating the **Start** button, the FX will begin solving any simulations that are currently setup in the scene. The progress of the simulation is displayed in an output pane that opens at the bottom of the **FX Browser** when the **Start** button is activated. A shortcut to this is on Layout's toolbar at **Modify > IKB Calculate**.

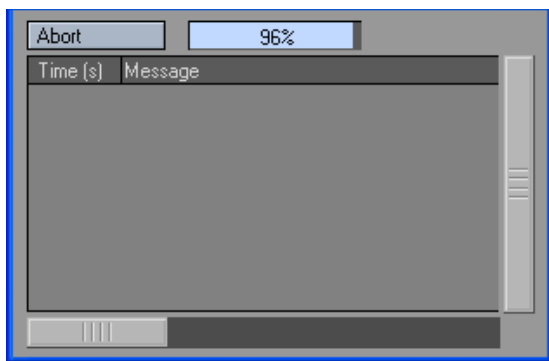


Figure xx: Dynamics Window during calculation



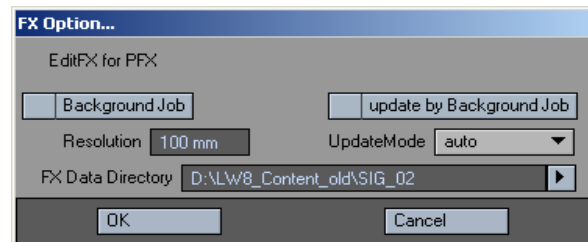
NOTE: Many scenes don't need this step, but clicking **Start** won't do any harm if you are unsure.

### The Save Button

Clicking the **Save** button saves all controller motions, using individual files, to your FX directory (defined on the **Option** dialog). These can be loaded using the **Load Motion** option on the **Controllers Properties Panel**.

### Options Dialog

The **Options** dialog has some options that affect the manner in which LightWave computes controllers.



The **Background Job** option activates multithreading. The **update by Background Job** option will update Layout when background tasks are complete. In most cases, you should leave these options in their default state of off.

The **Resolution** setting adjusts the parameter used when performing the physics simulations. The smaller the **Resolution** setting is, the more accurate the simulation will be. However, this also has a direct effect on the rendering times.

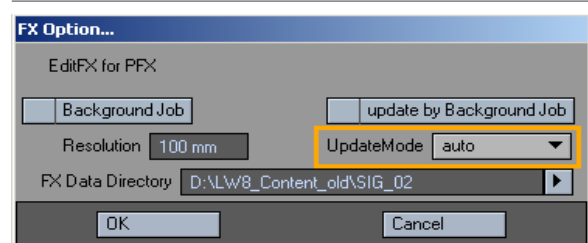
In this dialog you can also set the **FX Directory**. This is the default directory used when saving **Controller** settings.

### Real-time Display

Just click the **Play** button in Layout and as you tweak the various controller parameters, you get real-time updates of how your changes affect the particles.



NOTE: Scenes with particles interacting or with post-deformed geometry require you to click the **Start** button first to pre-compute some motions. Clicking **Start** won't do any harm if you are unsure.



**Auto** updates all emitters as parameters are modified. Use this for fast systems or in low-particle count simulations.

If you have multiple emitters and are using the **Select** update, then only the emitter showing in the **Select** pop-up menu, if any, will be updated.

**Adaptive** dynamically scales the number of visible particles based on your CPU performance. This setting attempts to keep interactive performance at a useful level.



User turns off particle updating completely. You may force an update by either clicking the **Start** button on the Particle FX Browser, or the **Update** button on the **File Tab** of the **Emitter Controller Panel**. This setting is useful in situations with heavy particle counts.

## Groups

You can group Dynamic objects with a user-defined name to prevent unwanted interaction. This also works with Particle FX controllers. This becomes very handy when you are working on a complex scene and you want certain Wind emitters to only affect certain objects.

<All Group> Includes all groups.

<default> Includes all items in scene.

<new group> Create individual group. Can associate functions within a group.



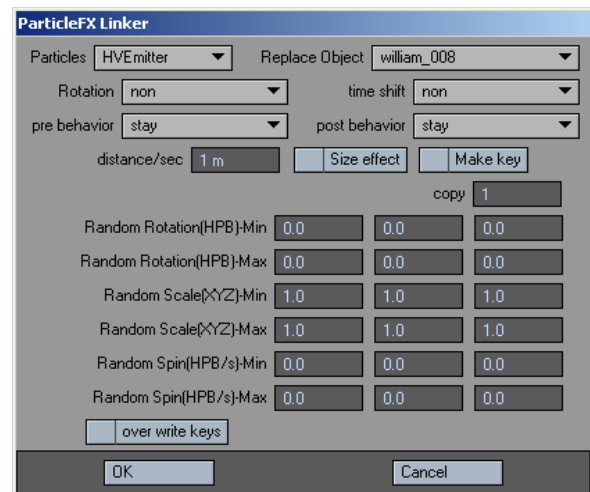
NOTE: You can associate with particle FX's functions using group function. If it is in the same group, you can associate settings. Also, if you do not want to associate, just by removing from the group, you can disregard the association.

## FX Linker



WARNING: Using **FX Linker** is a one-way shot. You can't undo what it does after you click **OK**. Therefore, you should definitely save your scene before running this modifier.

**FX Linker (Utilities > Additional > FX Linker)** automates FX\_Link setup on multiple particles. It basically clones the specified object and adds the FX\_Linkmotion modifier using the settings you specify. This is a great tool for creating a flock of bats, or for creating a quick crowd simulation.



### To use FX Linker:

- 1 - Save your scene!
- 2 - Load the target object.
- 3 - If you plan to use the Size effect option, keyframe the size on this object. Also, make sure your particles have some amount of size (Emitter properties).
- 4 - Choose **Utilities > Additional > FX Linker** and set the copy field to the number of instances you want to create.
- 5 - The **Random, Time Shift, Pre/Post Behavior, Distance/Sec, Size effect**, and **Make key** options correspond to the same options on FX Link, see above. Check them if you want those options set.
- 6 - If you are using **Random** or **Size effect**, you may set maximum and minimum values in the provided fields.
- 7 - Click **OK**. Depending on how many copies you make, it may take a while before FX Linker finishes.



WARNING: Using **FX Linker** is a one-way shot. You can't undo what it does after you click **OK**. Therefore, you should definitely save your scene before running this modifier, or better, press **shift + s** to save a scene increment.

## FX Property

You can display the **Property Panel** for the currently selected FX controller by choosing **Utilities > Plug-ins: Additional > FX Property**.



NOTE: You can also access the Controllers Property Panel by using the Dynamics Tab in the Object Properties Panel.

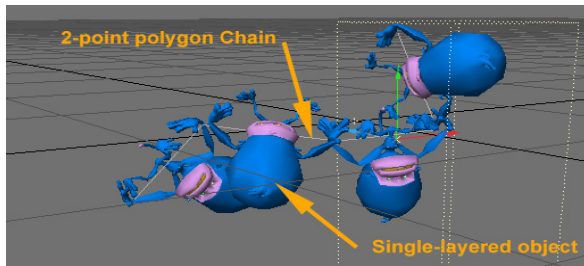


## FX\_Hardlink

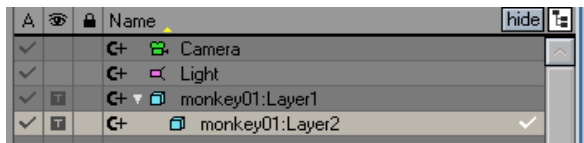
**FX\_Hardlink** is a very powerful tool that allows you to use the dynamic motion from one object and apply it to another. In most cases the dynamic object contains very few polygons (2 point polygons) while the "HardLink" object has a higher polygon count.

In the example below, **Cloth Dynamics** is applied to a 2-point polygon chain made up of (x4) 2-point polygons. It is the parent of a single layered object that is made up of 4 monkeys. When you use **FX\_Hardlink**, the monkeys take on the motion of the 2- point polygon chain.

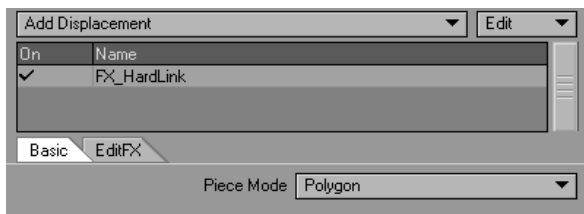
**Hardlink** keeps the geometry rigid and will displace it based on the **Piece Mode** setting.



**NOTE:** The HardLink Object must be parented to the Dynamic object in order for FX\_Hardlink to work.

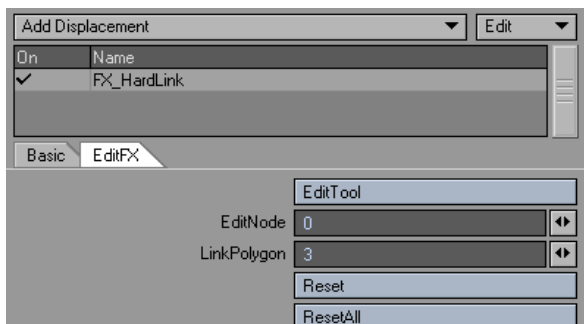


### FX\_Hardlink Properties



Under the **Basic Tab** you can choose what **Piece Mode** you would like to use. **Polygon** treats each segment as its own object. **1 Piece** will make the entire object a solid and considers it one piece. **Point Set** will read selection sets that you could create in Modeler.

The Edit FX Tab allows you to change what pieces are linked to what nodes. This is handy if the tool doesn't quite link what you thought it would on complex objects.



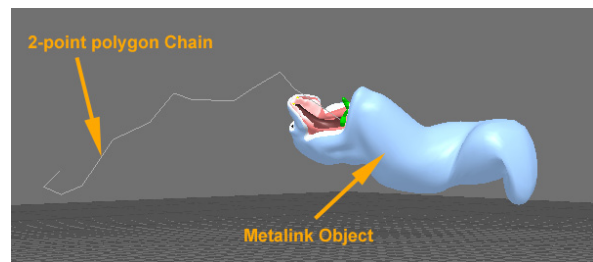
## FX\_MetaLink

**FX\_Metalink** is a very powerful tool that allows you to use the dynamic motion from one object and apply it to another. In most cases the dynamic object contains very few polygons (2 point polygons) while the "MetaLink" object has a higher polygon count.

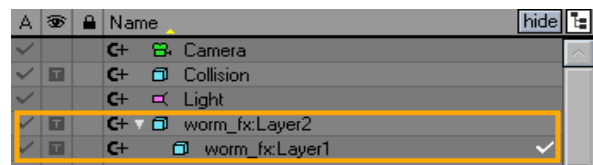


**NOTE:** You may run into problems if the Dynamic Object is too rough. Subdividing the Dynamic Object may help.

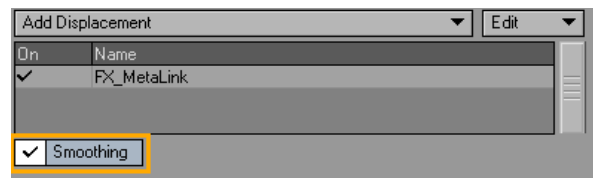
In the example below, a 2 point polygon chain with Cloth dynamics applied to it is the Parent of the Worm (MetaLink Object). The Work takes on the dynamic motion of the 2-point polygon chain with zero calculation time.



**NOTE:** The MetaLink Object must be parented to the Dynamic object in order for FX\_Metalink to work.



The **Smoothing** option attempts to smooth the reshaping of the Metalink object. If unchecked, the reshaping can pass through the vertex of the Cage Object (dynamic Object).



**NOTE:** If you want to use other Displacement Maps and not have them be ignored by MetaLink also apply **MetaLink\_Morph**.





## FX Metalink Morph

**FX\_Metalink\_Morph** is a displacement plug-in that enhances the functions of **FX\_Metalink**. By itself, **FX\_Metalink** cannot use normal morphing information because it ignores bones, **Morph Mapping**, and **Displacement Maps**. However, if you use **FX\_Metalink** with **FX\_Metalink\_Morph**, you can use normal morphing data.



NOTE: The **FX\_Metalink\_Morph** plug-in can be added before or after **FX\_Metalink**.

**FX\_Metalink\_Morph** has one pop-up menu called **Morph Mode**. Set this to **One time morph** to execute morphing only one time. This mode is appropriate when the morphing is from **Morph Mapping**. Use **Every time morph** to execute morphing for each displacement process. This mode is appropriate when the morphing varies, like the **Displacement Map** of waves. The **Non morph** setting simply disables this plug-in.

## ParticleFX DynamicLinker

Allows you to replace particles with other objects.

Particles--Choose the particle emitter to do the replacing.

Replace Object--Choose the object to replace the particles

Rotation--Give the item a rotation, once it is replaced

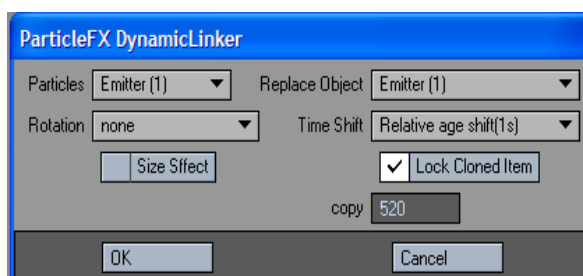
Time Shift--The amount of time to wait to replace the object

Size effect

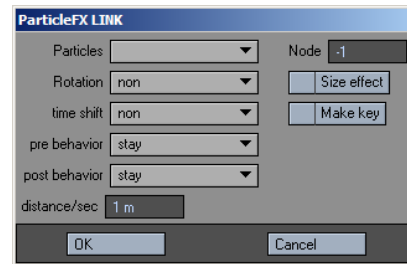
**time shift** - Contains: start shift, end shift, start adjust (distance) and end adjust.

Lock Cloned Item--Locks the current item to be the same cloned item

Copy--The number of replacements to make



## ParticleFX\_Link



### Particles

Choose the particle group.

### Node

**Rotation** - The choices in rotation are: random, align to path (h) and align to path (hp).

**Random** will rotate the item starting from a random rotational position.

**align to path (h)** Will align the item to a path using heading.

**align to path (hp)** Will align the item to a path using heading and pitch.

### Size effect

**time shift** - Contains: start shift, end shift, start adjust (distance) and end adjust.

### Make Key

**pre behavior** - Contains: original, stay and size dissolve.

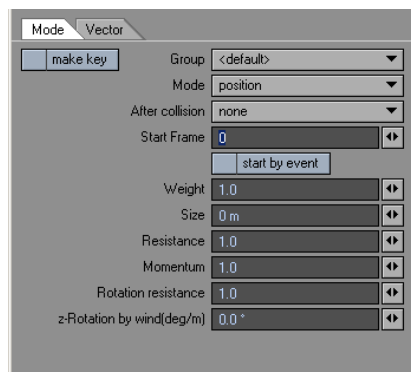
**post behavior** - Contains: original, stay and size dissolve.

distance/sec



## FX\_Motion

### Mode Tab



**make key** - Generates a key for every frame.

**Group** - Select an existing group or create a new one.

#### Mode

**After collision** - Choose an action post collision: none, reverse, random, stop.

**Start Frame** - Choose what frame to start the effect.

**Start by event** - Use this option if you want to trigger the motion upon collision.

#### Weight

This option sets an arbitrary weighting value that will influence how factors like gravity affect the item.

#### Size

This option affects the outer boundary used for collision detection. It can also be used by volumetric plug-ins like **HyperVoxels**.

#### Resistance

This option adds an *air resistance* effect. Items will move slower as this value is increased.

#### Momentum

This option adds in a momentum or increased mass effect. Items will tend to resist being stopped or slowed down.

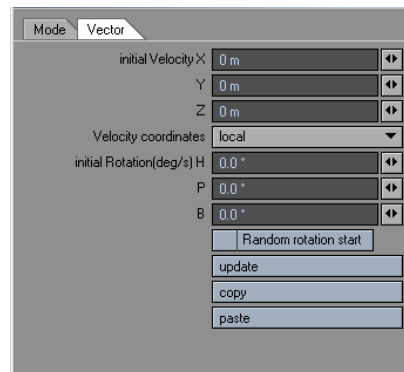
#### Rotation Resistance

This option has the same effect as Resistance except that it affects the items' rotational channels.

#### z-Rotation by wind(deg/m)

This option rotates the item in its bank channel as it moves or is affected by the wind.

### Vector Tab



#### Initial Velocity

This option sets the initial direction and force.

#### Velocity coordinates

This option sets whether these settings use the item's local axes or world coordinates.

#### Initial Rotation(deg/s)

This option sets the items' initial HPB rotation.

#### Random rotation start

This option randomizes the initial HPB rotation.

#### Update

This option updates/refreshes the current settings in the Layout viewport. This is good to do with a heavy scene.

#### Copy

This option copies the current settings.

#### Paste

This option pastes the current settings.



## Motion Designer Controller

### Motion Designer (Utilities > Plug-ins: Additional> MD\_Controller)

is a legacy tool that has been replaced with LightWave's dynamics tools (**Hard FX**, **Soft FX**, **Cloth FX**, etc.) and is available for use with older scene files.



NOTE: The newer dynamics can read old **Motion Designer** files.

## Introduction

**Motion Designer** is LightWave's legacy soft body dynamics engine that reshapes objects according to wind, gravity, and object motions, by using *elastic body models* to create animation data. You can easily achieve realistic effects like draping cloth over objects, material flapping in the wind, or jiggling jello, with **Motion Designer**. Because it is integrated into LightWave, **Motion Designer** can automatically take into account motions caused by displacement, bones, IK, and so on.



WARNING: Your Bounding Box Threshold (**Display Options Tab** of the **Preferences Panel**) must be set high enough to display your Motion Designer targets. If not, you may encounter a can't scan objects... message dialog.

### Elastic Body Models

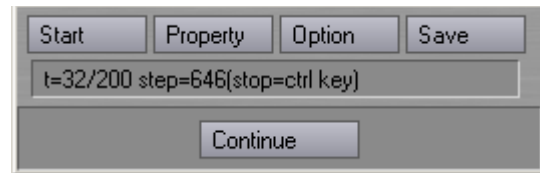
A standard LightWave object normally consists of points, polygons and surfaces. Motion Designer adds other properties to create its *elastic body models*. Points can be influenced by gravity and air resistance, proportional to speed. Polygon edges are deemed *springs* that also influence points. Surfaces have additional parameters that further define the elastic body model. The *spring force* is calculated continuously to reproduce its motions.

### Operating Motion Designer

You begin using Motion Designer by choosing **Utilities > Additional> MD\_Controller** to access the MD Controller. The following panel is displayed.



Click **Start** to apply your **MD** settings to your LightWave scene. This button is used after you have set all of the MD properties to your satisfaction. Press your CTRL key to abort the calculation.



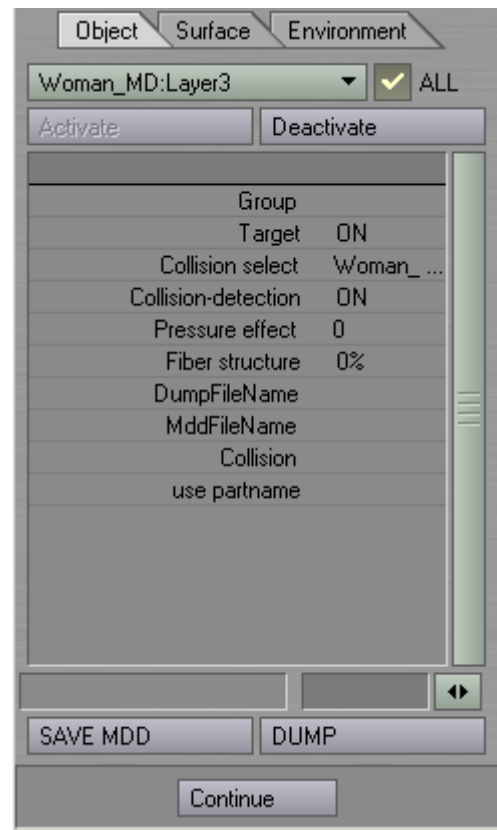
Click **Property** to bring up the **MotionDesigner Property Panel**. Here you set the details of how MD affects the items in your scene.

Click the **Option** button to set various global options, which are discussed later.

Click **Save** to manually save the MD data to a MDD file.

### Setting Parameters

To change a parameter, select it in the list. The selected parameter's name will appear in the field beneath the list window. You change the related value by entering a number in the input field to the right.



You can click some parameters in the list to toggle their state on or off. For filename parameters, you can double-click or press ENTER to bring up a file dialog.



NOTE: For On/Off parameters, a value of 1 means On and 0 means off.



## Property Panel: Object Tab

The **Object Tab** is where you specify which objects to use and how Motion Designer treats them. Select the object from the object list pop-up menu and click the **Activate** button. You can then apply MD parameters to it. Click **Deactivate** to deactivate an MD item.



**NOTE:** *Activate and Deactivate essentially add and remove the MD\_Plug Displacement Map plug-in automatically for you.*

Initially you'll want the **ALL** option active so that all objects are listed in the pop-up menu. Once you have defined the MD objects, you can uncheck ALL to list only defined objects. This may be useful if you have a lot of objects loaded into Layout.

**Group-** You can group MD objects with a user-defined name to prevent unwanted interaction. This also works with Particle FX controllers.

**Target On-** means the object is an elastic body model and can be influenced by other Target or Collision objects.

**Collision Select-** Normally, MD executes the collision calculation among all target and collision objects. The **Collision Select** function lets you specify which object the target collides with. Double-clicking the setting displays a dialog where you select the collision object. This function will improve the calculation speed.

Use the **Collision-Detection** function to specify collision detection on an object-by-object basis instead of surface-by-surface. If this parameter is OFF, all of the object's surfaces are treated as if the **(Surface Tab) Collision-detection** setting was OFF.

**Pressure effect** causes the object to maintain its volume during a motion, as if it is filled with a stiffer gel, instead of, say, water. This works best when the object is a closed volume and spherical in shape.

When **Fiber structure** is set to a value other than zero, the object seems to be made of a fiber material. This function essentially weakens the force when the motion direction is different from that of the virtual fiber. When the value is 100%, the force is zero, except along the fiber. This setting also weakens the force from the sub-structure.

**DumpFileName** is a special file that sets up an initial pose for the target object. This file is discussed below.

**Collision On-** means the object is a collision object that will influence target objects, if it collides with one.



**NOTE:** *You cannot have Target and Collision both set to ON for the same object.*

**MddFileName-** Generally, this setting is for backwards compatibility with prior versions of MD, which required you to manually specify motion data files.

**Motion Files-** For Target objects, click **SAVE MDD** to save the generated motion to a file, which will be used subsequently to deform the object.

Click **DUMP** to save the final pose (position, speed) of a Target object to a file, based on the last frame calculated. In a subsequent session, you can specify this file as the **DumpFileName** parameter for a Target object and set a starting pose for the target. This is useful to have a

realistic beginning state for the target.

### To use a dump file:

Set your Last Frame in Layout to the pose you want to start with.

Start the calculation.

After the calculation, click **DUMP** and save the Dump file.

Specify the saved Dump file for the **DumpFileName** parameter of the Target object. The Target object will not begin in this state.



**NOTE:** *If you do not want to use the Dump file, make sure you clear it from DumpFileName.*

## Using Motion Designer with SubPatch Objects

In order to use Motion Designer with SubPatch objects, you need to set Subdivision Order to Last on the **Object Properties Panel**. However, understand that Motion Designer does not directly interact with the underlying smooth SubPatch surface. Basically, it is a displacement plug-in and can only deform the cage.

By postponing the subdivision until after all displacement deformations take place, Motion Designer can appear to work with the smooth SubPatch surface. Make sure the cage conforms closely to the underlying surface as much as possible--this will likely mean you will need to use a more complex cage than you would normally. This is particularly important when collision detection is involved.



## Property Panel: Surface Tab

Apply physical characteristics to the surfaces of Target objects or Collision objects on the **Surface Tab**. To specify characteristics, select the surface from the surface list and assign parameters.



### Basic Settings

**Weight** defines the weight of the material.

Use the **Weight +-** parameter to add randomness to the **Weight** parameter.

The **Spring** coefficient controls the *springiness* or stiffness of the material.

**Viscosity** controls the impact of a collision. If an object bounces, a higher **Viscosity** value will have less bounce motion because the bouncing force is *absorbed* by the **Viscosity**.

**Resistance** controls the amount of air resistance.

**Parallel Resistance** controls the amount of air resistance parallel to the surface. An object, such as a floor tile, will fall quicker if turned on its edge. The default value of 100% for parallel resistance yields normal results and should be used for items like falling leaves and flags. If for some reason you do not desire this effect, reduce the setting.

**Back-resistance** specifies the air resistance for the back of a surface (i.e., opposite the surface normal's direction). At 100%, the *back side* is affected by wind the same as the front. At a value lower than 100%, the back side is affected more by wind. This function can be effective when your wind becomes weak due to overlapped objects.

### Structure Settings

**Fixed**- The shape of polygons with **Fixed** On remains intact.

**Sub-Structure**- The surface of a pure, elastic body model has a high degree of freedom, which causes the object to easily distort. To prevent distortion, you can apply an auxiliary MD form, called a **Sub-structure**, to restrict the instability of the surface. This can improve your results with a two-dimensional object (i.e., one with no thickness) making it act as if it has some thickness. However, this reinforcement can require a high **Sub-structure** setting that can take longer to calculate.

When **Sub-structure** is used on a square mesh, it creates something that looks like two triangular polygons placed on top of each other in opposite directions. The sub-structure mitigates the tendencies to bend and to resist bending in certain directions.

When the Sub-structure value is non-zero, the auxiliary form is applied. The higher the value, the more the form will have a tendency to keep its shape.

**Hold-Structure**- Because a pure elastic body model simulates only the surface structure, it can be limited to two-dimensional motions, like with a bed sheet. In order to simulate elastic three-dimensional motions, such as some gelatin, a **Hold-structure** needs to be used. This parameter causes a surface to tend to maintain its original shape, like gelatin does when it jiggles.



NOTE: The effect of Hold-structure is uniform throughout the surface, while with Sub-structure it is non-linear, which will often result in a more natural look. You can use a combination of both to achieve just the right result.

**Smoothing**- To smooth out the transition between, say, a **Fixed** surface and **Hold-structure**, apply some level of **Smoothing** to both surfaces. This can help prevent unwanted creases and wrinkles occurring between the two areas.

**Stretch-limit**- To prevent a surface from stretching like rubber, lower the **Stretch-limit** below the default value of 100%. This restricts the amount of the surface to be stretched.



NOTE: Although you could also increase the Spring value to reduce stretching, increasing that value can make the surface too complex and cause other problems, like weird folds.

**Compress Stress** controls the amount of compression a surface exhibits as a result of stress. A soft fabric like cotton can be made using a large **Spring** value and a low **Compress stress** setting. A stiffer fabric would use a higher **Compress stress** setting.



NOTE: You can easily adjust the apparent thickness of a drape by adjusting Compress Stress.

**Shrink** reduces the surface to a specified size. For example, a value of 90% reduces the size of the surface to 90%. Use **Shrink** to reduce the looseness of a dress or to create frills.



## Collision Settings



**HINT:** Try to use **Self Collision** and **Collision Detection** only on necessary surfaces to minimize calculation time.

The **Self Collision** setting can prevent an object from crossing itself when the object is transformed. **Self Collision** is calculated based on a point on the target object and a polygon on the collision object. If a surface has **Self Collision** and **Collision Detection** active, **Self Collision** is computed on the same surface.

**Self Collision** works in much the same way **Collision Detection** does, and there is no collision detection for polygon edges. As such, you may also want to adjust **Skin Thickness** to avoid errant point penetrations. Note that unlike **Collision Detection**, **Self Collision** is detected even if the motion is from behind the polygon.

**Collision Detection** reflects the influence of other objects upon the motion of an object. This lets you create complex motions caused by obstacles. The simulation is performed by taking an object that collides (the collision object) with the elastic body model (the target object) into the calculation.

The direction of the surface normal plays an important part in collision detection. The target object must collide with the surface of the collision object. If it contacts the surface from the rear (i.e., away from the surface normal), the target object will simply pass through. As such, if you need collision detection inside a container, for example, you'll need to have interior facing surface normals.

When **Single-sided** is ON, surfaces with normals facing the same direction will not collide. If the collision is between surfaces with opposing normals, you can reduce calculation time by turning this ON. If you will have collisions between surface normals facing the same direction, you should turn this OFF.

**Skin Thickness**- You should understand that the *collision* is detected using a point on the target object (elastic body model) and a polygon of the collision object. As a result, a polygon of the target object can possibly penetrate the collision object—usually an undesirable result.

Use the **Skin Thickness (Surface Tab)** setting to avoid undesired surface penetration like the example below. **Skin Thickness**, set in meters, creates a gap between the collision and target objects, within which a collision is deemed to have occurred. Make sure you specify **Skin Thickness** for the surface of the collision object, not the surface of the target object.

**Friction**- Like real world friction, the **Friction** parameter makes the surface less slippery. So if you want the target object to tend to slip off the collision object, set **Friction** to 0. If you want it to stick more, increase the value.

**Bound force** adds a *rebounding* speed change at collision, if set greater than 0. At 1, the speed is the same as the collision object. At 2, it rebounds at the collision speed.

**Action force** decides whether the colliding object receives the reaction force at the time of the collision, or not. This function does not affect the reaction force that the collision object receives.

**Bind force** causes the target to adhere to the collision surface.

If you have some level of **Bind force**, **Fix Force** will also be available.

**Fix Force** causes the target object to stick to the surface and not slide around.

## Other Surface Controls

**Hide**-When **Hide** is unchecked, MD displays all of the Surface parameters that you can specify. When **Hide** is checked, MD hides all parameters at their default values. This display mode helps you *zero in* on just the settings you have tweaked.

Use the **COPY** and **PASTE** buttons to copy and paste the current settings between surfaces. Use the **SAVE** button to save the current surface settings to a file on your hard drive. Use the **LOAD** button to load the settings from a previously saved file.

If you save the files to the SURFACES subdirectory in your Content directory, they will appear in the Material Library pop-up menu, discussed below.

### Material Library

This pop-up menu (located near the **Hide** button) contains pre-defined sets of surface parameters.

### Property Panel: Environment Tab

The **Gravity** and **Wind** settings are specified on the **Environment Tab**. All of these settings affect Target objects.

Object	Surface	Environment
Gravity(m/s <sup>2</sup> )	0.0	-1.0 0.0
Wind1(m/s)	0.0	0.0 0.0
Wind2(m/s)	0.0	0.0 0.0
Turbulence(m/s)	0.0	0.0 0.0
Wavelength(m)	1.0	1.0 1.0
Wind Mode	Cycle	
Random-ratio(%)	0.0	
Cycle-length(s)	2.0	
Gust-start(s)	0.0	
Gust-length(s)	1.0	
<div>SAVE LOAD</div> <div>Continue</div>		

### Gravity

Specify the direction and intensity of gravity along the world axes.

### Wind1/Wind2

Specify the direction and intensity of two winds along the world axes.





## Turbulence

Specify the direction and intensity of turbulence along the world axes.



NOTE: The strength of turbulence is affected by the wind.

## Wavelength

Specify the wavelength of the turbulence.

## Wind Mode

The **Wind Mode** controls the repeating pattern for **Wind1** and **Wind2**. **Random** randomizes the pattern according to the specified **Random-ratio**. The **Cycle** setting swaps **Wind1** and **Wind2** every **Cycle-length** period, specified in seconds. **Gust** uses **Wind2** only for the duration beginning with **Gust-start** (in seconds) for the **Gust-length** (in seconds). **Wind1** is used at other times. Default ignores the **Wind1** setting and uses only **Wind2**.

## SAVE/LOAD

Use the **SAVE** button to save the current **Environment** settings to a file on your hard drive. Use the **LOAD** button to load the settings from a previously saved file.

## Helpful Tips

With Motion Designer, you can adjust motions intuitively since MD is based on physical models. You can achieve heavy motions by increasing the **Weight** value, and you can create light motions by decreasing **Weight**.

Reducing the **Spring** coefficient creates soft motions, while raising **Spring** produces motions with a stronger repelling force. Setting the coefficient to an extremely large value creates stiff motions.

Since Motion Designer uses elastic body models as calculation models, based on mass, spring, and resistance, unexpected vibration may appear in the calculated motions. One of the following adjustments may help correct the problem:

Increase **Resistance** to make moving more difficult.

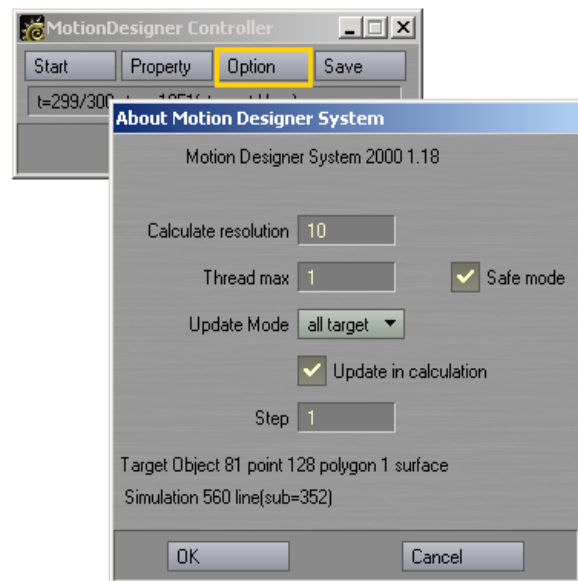
Reduce the **Spring** coefficient. This will reduce the force that causes vibration.

Increase the **Weight** to stabilize the surface.

Abnormal motion, including vibration, may be the result of (*gasp!*) calculation errors. Remember, the motion is approximated based only on numeric values. This problem may be solved by adjusting the accuracy of the calculations using the **Calculate Resolution** setting on the **Motion Designer Options Panel**, discussed below.

## Options Panel

The **Motion Designer Options Panel** is displayed when you click on the **Option** button on the main **Motion Designer Panel**.



The **Calculate Resolution** setting controls the accuracy of the motion calculations. The default is 10. Increasing this value improves the accuracy of calculation, but as you might expect, it also increases the amount of calculation. Extremely large values will result in unusually long calculation times.

If you are using a multi-processor machine, specify the number of CPUs in **Thread max** field.

When the **Safe Mode** option is unchecked, some internal calculation restrictions are eased. This allows faster calculations, but may result in unwanted vibrations, as well as different results for calculations based on the same parameter values.

You can control how MD updates by using the **Update Mode** pop-up menu. **Auto** updates only targets that have had parameters changed. Use **select only** if you have multiple targets, then only the selected targets will be updated. **All targets** updates all targets. Use this if your system is fast.

If **Update in calculation** is checked, Layout is updated while MD calculates.

**Step** is a great time-saving feature. It sets the frame increment for MD calculations. Since LightWave will obviously interpolate movement between frames, it is usually not necessary to set this to 1. You may often get away with 3, 5, or even higher settings, depending on the scene.





---

## Chapter 22: Environments

---



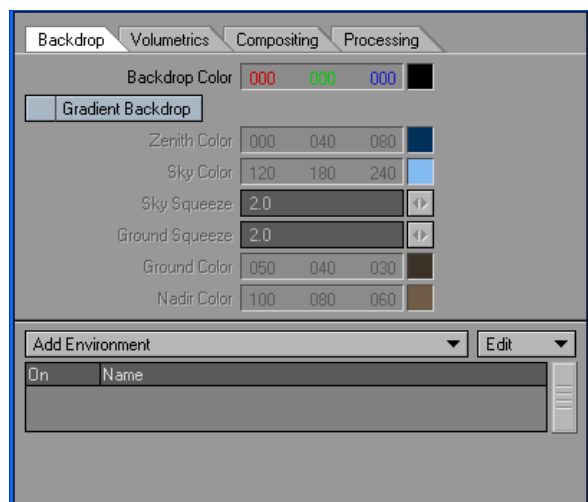
## Environments

Fire! Smoke! Flowing Magma! This is where you add these types of dynamic effects. You can also create fog effects, or even enhance your backgrounds with the effects controls.

## Backdrop Options

(default keyboard shortcut **Ctrl F5**)

LightWave lets you change the colors of the backdrop behind all of the objects in a scene. The settings are on the **Backdrop Tab** of the **Effects Panel (Window > Backdrop Options)**. You can elect to have a solid color using the **Backdrop Color** setting or a gradient color backdrop. By default, the backdrop is solid black.



**NOTE:** If you have any reflective surfaces and you want the backdrop included in the reflection, set the **Reflection** options on the **Surface Editor's Environment Tab** to one of the backdrop options.

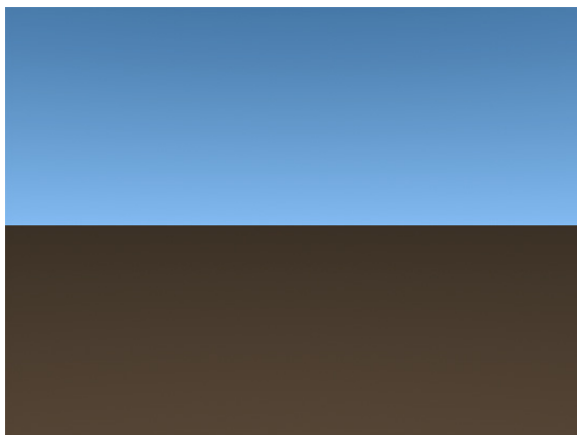
## Gradient Backdrops

LightWave's **Gradient Backdrop** settings essentially provide a quick earth-like environment background. Note that no shadows can be cast on the backdrop since it really isn't there.

Although you're unlikely to use it much for real-life imagery, it is great for faking a sky behind mountain objects, instructional animations, and logo animations. Use it when you need something to stick in the background.



There are actually two gradient areas. Think of it as a huge sphere cut into two halves. One half rests on top of the Y axis plane, and the other half sits directly beneath it.



The **Sky Color** begins all around the edge of the top half of the sphere and gradually turns into the **Zenith Color** moving towards the top. The **Ground Color** begins all around the edge of the bottom half of the sphere and gradually turns into the **Nadir Color** moving towards the bottom. Note that there is no gradual change of color between the **Sky Color** and **Ground Color**.

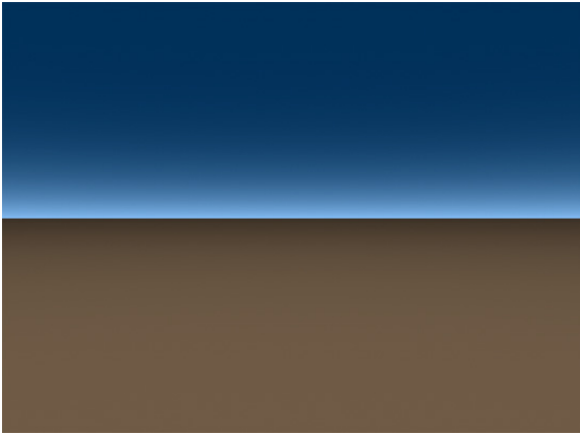


**HINT:** If you want the **Gradient Backdrop** to blend everywhere with no sharp color change, make the **Sky Color** and **Ground Color** the same.

The camera is always positioned right in the center of the sphere. Thus, if you move the camera, the Gradient Backdrop will always look the same. However, if you rotate the camera, you will see the various colors.



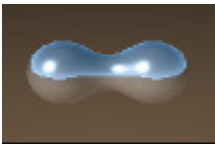
The **Sky Squeeze** value determines how the Zenith and Sky colors are blended. The **Ground Squeeze** value determines how the Ground and Nadir colors are blended. The default value of 2.0 yields a nice spread between the various colors. A lower value will spread the change in colors over a greater area and a higher value will compress the change.



Sky and Ground Squeeze both at 20.0

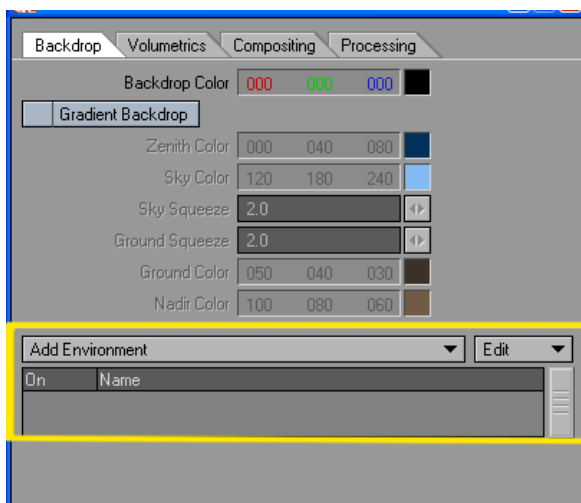


HINT: The default gradient colors are useful when simulating a reflective chrome surface.



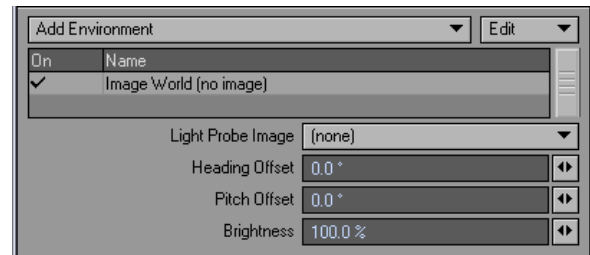
## Environments

Several plugins may be used to create exciting backgrounds for your animations. You access these plugins by adding them on the **Add Environment** pop-up menu on the **Backdrop Tab** of the **Effects Panel**.



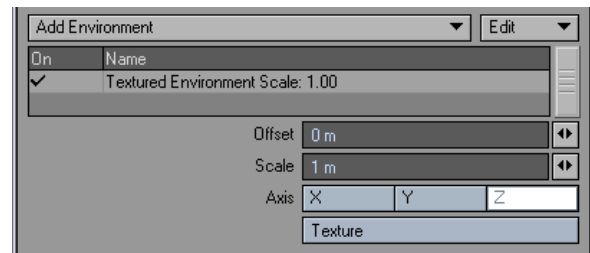
## Image World

The Image World Environment lets you apply an image as a spherical environment wrap; this approach is perfect for high dynamic range images. You can offset the heading and pitch of the image, as well as adjust its brightness.



## Textured Environment

The Textured Environment lets you apply a texture as a spherical environment wrap; this approach is perfect for high dynamic range images. Clicking the **Texture** button launches the standard **Texture Editor** giving you access to gradients, procedurals, and the use of images to create an endless variety of textures.



The texture is not *stuck* to the camera background, like a normal background image. Therefore, as you move the camera, you will pan over the environment.

An interesting application would be to use a gradient and select the (camera) Heading or Pitch as the Input Parameter. This lets you vary a glorious sunset sky based on the rotation of the camera.

You might also use this to add a nebula behind your starfield using a simple Turbulence or Fractal Noise procedural texture.

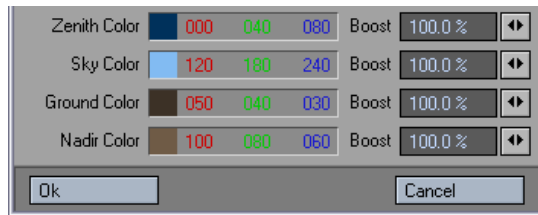


NOTE: For more information on the **Texture Editor** see the **Texture Editor** section in Chapter 36: Surface Editor, starting on page 971.



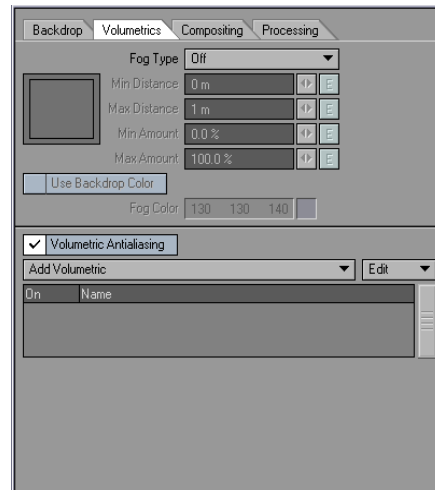
## Background Booster

The **Background Booster** tool (**Window>Background Booster**) gives you the ability to easily edit the **Background Gradient** settings. Use the Boost function to multiply each of the RGB values by the boost percentage.



## Volumetrics and Fog Options

(default keyboard shortcut **Ctrl F6**)



LightWave features volumetric effects — essentially lighting and particle effects with physical volume. These effects are common in everyday life and can play a key role in creating dramatic realistic environments. However, they are difficult to reproduce using standard polygonal models.



Copyright Pixel Magic

## Background

A very common example of volumetric lighting is the atmosphere: the color of the sky comes from the scattering and absorption of light in the different layers of the atmosphere.

The combination of light scattering and absorption is the very core of volumetric lighting effects. Additionally, other parameters must also be taken into account, like volume size and shape, density distribution inside the volume, lighting conditions, and behaviour of light inside the medium. Adjusting these parameters lets you create a wide range of natural effects.

The sky, for example, is usually blue because light attenuation depends on wavelength and distance. When the sun is at the zenith, its light crosses fewer layers of atmosphere than when the sun is on the horizon. Blue light is stronger at the zenith than at sunset because the thinner atmosphere does not interfere with its shorter wavelength. In red sunsets, the thicker layers of atmosphere attenuate blue light, but the longer wavelength of red light passes through the atmospheric layers.





Fog is another good example. It is a medium composed of vaporised water where density is distributed in a non-homogeneous manner. The water particles in the fog cause a dispersion/absorption phenomenon that causes the lighting effect, while the density distribution gives the global appearance of the fog. If you want a thick fog lying on the ground and fading with altitude, you will have to use a density distribution that makes the density high at lower altitudes and low at higher altitudes. If you want to add turbulence in the fog (to have a more cloudy appearance), you can add fractal noise, which creates a 3D density field.

## Computational Issues

Volumetrics are calculated by integrating all the scattering/absorption contributions along a ray (which comes from the camera). When you use a 3D fractal density field, the integration must be made numerically with a limited number of sampling points. In this case, the values will be calculated at each sampling point, which means that for 50 sampling points, the algorithm calculates 50 density field values, 50 lighting values, and 50 scattering/absorption values. All those values can take a lot of time to compute. Using fewer sampling points will result in a faster rendering but will introduce numerical errors: this is volumetric aliasing. Volumetric shadows can be obtained this way by measuring the lighting conditions at each sampling point along the ray.

When you work with a normal density distribution, you can make the integration literally, which gives a much faster rendering. But in this model (which we call fast model), it is not possible to measure lighting conditions along the ray, and as a consequence it is not possible to get volumetric shadows.

Another important note about numerical issues is how to adjust values to get the desired effect. The intensity of the effect is always related to the length of the medium crossed by the ray. This is obvious if you compare cigarette smoke to smoke from a large fire — the size of the volume has a big influence on the result. The behaviour of light may also change completely when the volume size changes, because absorption may overpower scattering, and vice versa. A good example of this is clouds.

When you look at clouds, you see that small thin clouds are bright and totally white, while big clouds have dark grey areas and a thin white border. The dark grey color comes from the absorption of light inside the cloud. Even the scattering of light emitted inside the cloud is absorbed from the point of scattering to the boundaries of the cloud. In this example, absorption takes precedence over scattering when thickness gets bigger. However, under other circumstances, the opposite could occur. When you use high absorption and scattering values, you can create explosion-like effects, where there is very high contrast between bright and dark areas. In conclusion, when adjusting parameters, you must be aware of the scale of the object you are working on.

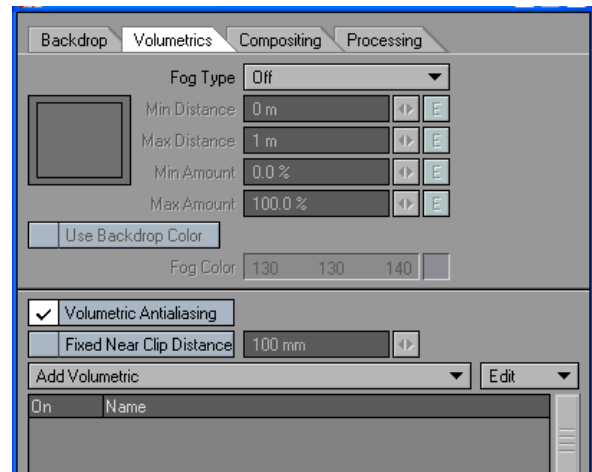
### About Particles

When you work with particles, a sphere of gas is associated with each particle. As a result, a particle cloud is really just a union of spheres. Computing the effect for every particle can be computationally intensive, particularly when their spheres overlap one another. The solution is the automatic particle sizing option, which evaluates a particle size so that each particle is close to another. The result is a dense cloud optimised for numerical integration and lower rendering times.

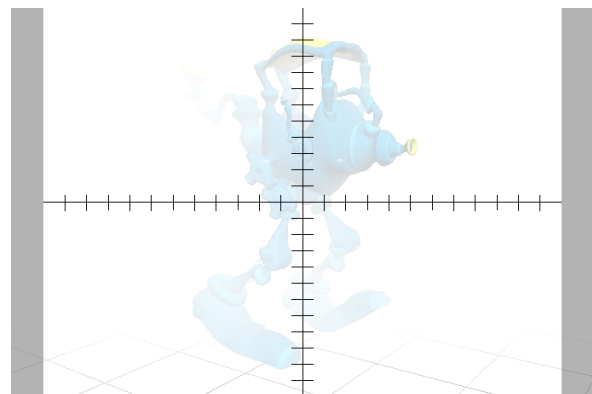
## Normal Fog

LightWave can generate a quick fog effect that is useful for many special effects. Just as objects seem to blend into real fog with distance, the effect fades the objects away into the fog color you set. Fog surrounds the camera in all directions, as though the camera were in the middle of a huge spherical fog bank extending in all directions.

The **Fog** settings are on the **Volumetric Tab** of the **Effects Panel** (**Window > Volumetrics and Fog Options**). In setting up fog, you will set a minimum and maximum distance from the camera. Within and beyond this range, objects will take on some amount of the **Fog Color**. You can also specify the percent of fog color that objects take on at the minimum and maximum distances.



The **Fog Type** pop-up menu sets the characteristics of your fog. **Off**, obviously, turns off the fog effect. The other fog types differ in how the effect falls off toward the camera. **Linear** falls off in a linear manner. **Nonlinear 1** is somewhat more realistic in appearance, since the fog will appear to grow thicker with distance. **Nonlinear 2** has a steeper falloff curve. **Realistic** generates realistic fog that works correctly with objects that have reflection, refraction and/or transparency that are inside the fog. Unlike the other fog modes, it has no exact limit to the fog range.



It's important to realize that the Fog feature doesn't actually calculate a wispy volumetric fog around objects, but rather changes the color of the objects to that color chosen as the **Fog Color**. As such, the backdrop will receive no amount of fog. For volumetric 3D fog, use the Ground Fog volumetric, discussed later.

A negative **Minimum Distance** will start the fog behind the

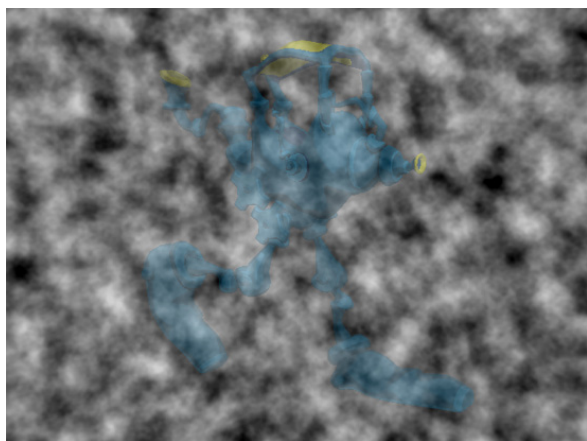


camera. You can even enter a larger minimum amount than the **Maximum Distance** amount, which results in an effect whereby objects will render in more of the fog when closer to the camera.

By default, fog is applied linearly between the Minimum Distance and Maximum Distance. Non-linear options are also available. They apply the fog amount more rapidly as they are moved away from the camera, then less so as they approach the maximum distance. The small graph to the left of the setting gives you an indication of the fog application over distance.

The **Use Backdrop Color** option causes an object to blend in with whatever backdrop you have set, including a background image. This can have the effect of making objects appear slightly transparent. Use this option to simulate the effects of an underwater environment or a hazy, foggy day with an appropriate **Backdrop Color** like bluish green for underwater and greyish white for a foggy day.

If you add the Texture Environment environment (**Window > Backdrop Options**) and also activate **Use Backdrop Color**, your fog (and backdrop) can use a texture.



Textured Fog

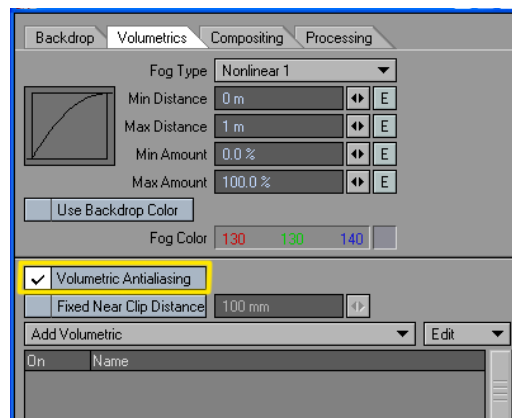


NOTE: The rendering speed of the Fast Fog Render Type for the Ground Fog volumetric comes at a price. Because it is not a full

volumetric effect, it will not always blend accurately with other volumetric effects, like volumetric lights. This may result in visible artifacts in your rendered images.

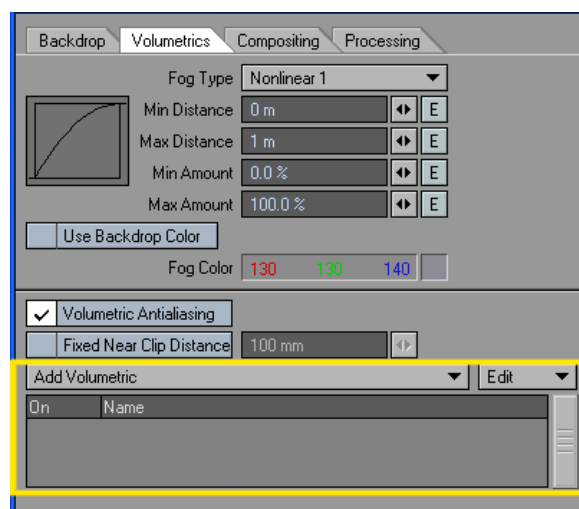
## Volumetric Antialiasing

The **Volumetric Antialiasing** option activates or deactivates the antialiasing of volumetric plugins and volumetric lights. If this option is off, the volumetric effects from the first rendering pass are stored and reused in later passes, instead of being re-rendered in each pass. Obviously, this can save rendering time, but will require more memory and may cause problems when used in conjunction with motion blur or depth of field.



## Volumetric Plugins

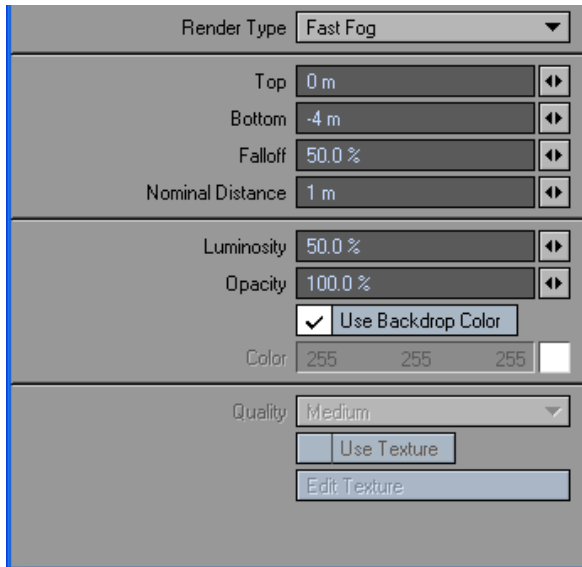
Volumetric Plugins are added on the **Volumetrics Tab** of the **Effects Panel**.





## Ground Fog

Use the **Ground Fog** volumetric plugin to add three-dimensional fog to your scene. The fog has actual physical size so, for example, the camera can move in and out of the fog. This differs fundamentally from normal fog, discussed previously, where the camera is always within the fog.



With the **Render Type** pop-up menu, you can choose between two different types of Ground Fog. **Fast Fog** is a quick-rendering fog with a uniform thickness. It is basically the three-dimensional version of normal fog. **Ray Marcher** adds an uneven fog, particularly when you use a texture. Since this fog varies by precise physical position, like real fog, it is computationally intensive, but generally yields a more accurate and realistic result.

When using **Ray Marcher**, you can throttle the amount of computations using the **Quality** pop-up menu at the bottom of the panel. Add a texture by activating the **Use Texture** option. Clicking the **Edit Texture** button will bring up the standard **Texture Editor**. A fog texture will make your fog more interesting and less flat.

The **Top** and **Bottom** settings control the altitude of the fog, that is, where your fog starts and stops vertically. **Falloff** determines how the fog decreases to zero, from the **Bottom** to the **Top**. The higher the value, the more the fog will decrease its density. Note that the **Ray Marcher** mode tends to fall off quickly at the fog's edges, while **Fast Fog** has a uniform linear fall off.



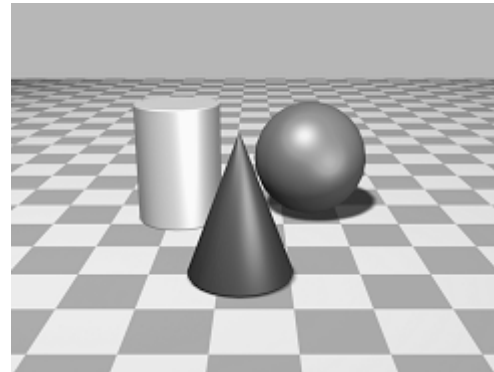
**NOTE:** Make sure you know where your camera is when using GroundFog. The effect is really best seen from outside of the fog. If your camera is inside the fog, changing settings may appear to have little effect. In fact, if your camera is always within the fog, you may want to just use the (faster) normal fog.

The **Luminosity** and **Opacity** values are the values where the fog is at its thickest.

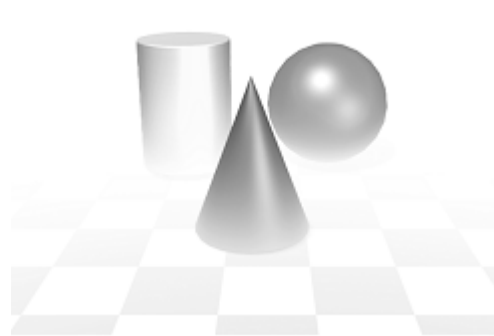
**Nominal Distance** is the distance at which the fog has a medium

effect — it is not like standard Fog's **Minimum Distance**. You will want to use small values for small-scale scenes. Large-scale scenes may require higher values to keep close objects from getting too affected by the fog.

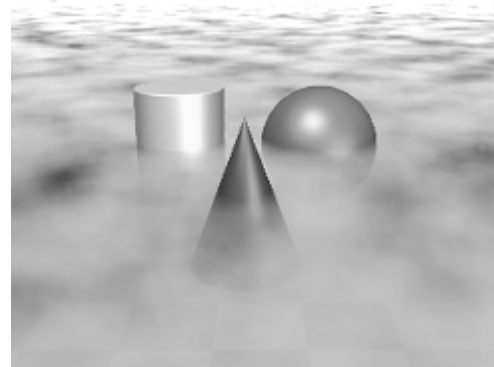
You can set the color of the fog with the **Color** setting or you can just use the backdrop color.



No Fog



FastFog



Ray Marcher Fog using a Texture

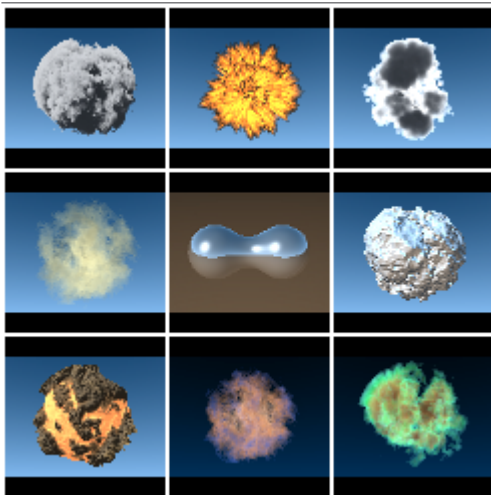


## HyperVoxels

With ordinary polygonal objects, realistic effects like liquids, smoke, clouds, and fire are difficult, if not impossible, to achieve without **HyperVoxels**. It simplifies the creation of volumetric rendering effects such as photo-realistic clouds, flames, explosions, dust, nebulae, contrails, fluids, smoke, ash, pyroclastics, gelatin, electro-microscopic images, rusted materials, detailed solid and rocky surfaces, and much, much more.



**NOTE:** The way Hypervoxel sprites are calculated has changed. They now work as you would expect by the values as entered. 100% means 100% and not 50%. The texture amplitude is now only applied if there is a hypertexture. 100% amplitude is full contrast not 50% contrast as before. The amplitude is also applied in a more logical fashion and clipped to 0 to prevent negative values. This means that if you apply a texture map to your sprite, it will appear at 100% intensity as you would expect if the luminosity and density are 100%. All of this will change the look of Hypervoxel sprites considerably. Please note that there is an ambient color that is applied to sprites. This can only be set in the Advanced shading tab for Volume Hypervoxels. It will make sprites a bit brighter than you would expect but it is being calculated correctly and can be turned off.



**HyperVoxels** have computed mass, as opposed to the modelled mass of normal LightWave objects. This opens the door to effects of greater complexity. You can now transform objects dynamically, like merging and slicing objects, without Modeling multiple geometries. A common example of this is the action of the substance inside lava lamps.

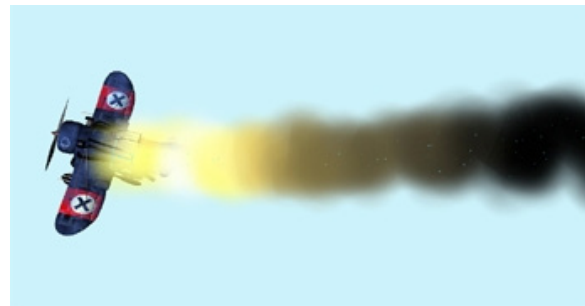
**HyperVoxels** can be a surface, volume, or sprite. Surface **HyperVoxels** are similar to standard LightWave objects. They have a defined surface, but no interior. If you go inside such an object, there is nothing. Volume **HyperVoxels**, on the other hand, have computed volume. You would use this option to create effects like gases, flames, clouds, explosions, or even short hair or fur. Your camera can go inside these **HyperVoxels** and still see the texture. The **Sprite Mode** is a "slice" of a volume **HyperVoxel**. It renders more quickly, but lacks much of the volume mode's 3D quality.

The **HyperVoxels** volumetric filter is usually applied to a points-only object or a null object. The reason is that polygons may be visible after rendering and usually this is not desired. However, having polygons may make the object easier to see in Layout. In such cases, you can use the **Unseen by Camera** object property.

**HyperVoxel** textures are 3D algorithmic textures, unlike normal surface **Bump Maps**, which only appear to have depth. This means you can actually get very close to the surface and the textures will look three dimensional. **HyperVoxels** feature sub-pixel displacement, which results in surface details no matter how close you get.

You can determine the position of **HyperVoxels** by using a null object or an object's points. Moving the object will move the **HyperVoxel** object. Changing the position of an object's internal points will also change the look of a **HyperVoxel** object. **HyperVoxels** will appear around each point and their proximity to each other affects how the object appears on the whole.

A particle animation system is not required. You can achieve many spectacular effects using null objects or points animated by conventional means. However, to achieve realistic dynamic liquid effects, you will probably need a LightWave particle system, like **Particle FX**.



**NOTE:** HyperVoxel objects exist just as normal objects do. Thus, they cast shadows, are reflected, and so on. However, currently, plugins do not have access to all lighting information. As a result, **HyperVoxels** will continue to receive shadows, even if the **Receive Shadows** option on the **Object Properties Panel** is disabled.

## HyperVoxels and Transparent Surfaces

Because volumetrics are ray-traced, to see a HyperVoxel behind a transparent surface, you must ray trace transparent surfaces. You can do this by simply activating the **Ray Trace Transparency** option on the **Render Options Panel**. However, this is not necessary if the **Refraction Index** of the object's transparent surface (**Surface Editor**) is set greater than 1.0 (even 1.001) and **Ray Trace Refraction** on the **Render Options Panel** is enabled.

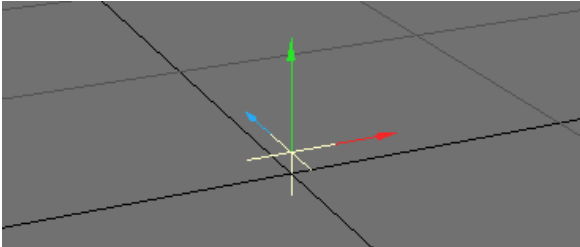
Also, if you want to see an object behind a transparent **HyperVoxel** surface, you need to set the **Refraction Option** to **Ray Tracing + Backdrop** on the **HyperVoxels Shading, Environment Tab**.



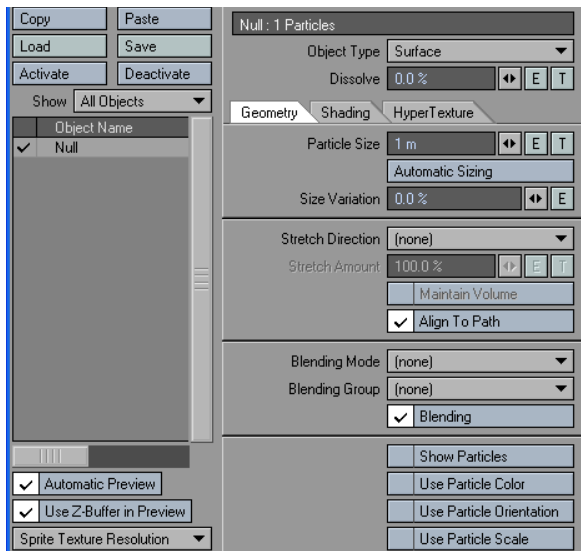
## Jump-starting with HyperVoxels

Here are some short exercises to jump-start you with **HyperVoxels**. This exercise gives you a quick hands-on feel for how to use **HyperVoxels**.

**Step 1:** Add a null object to an empty scene in Layout.



**Step 2:** Add **HyperVoxels** from the **Add Volumetric** drop down menu (**Window > Volumetrics and Fog Options**) or hit **Ctrl F6**. Open the **HyperVoxels Settings Panel** by double clicking on it in the list.

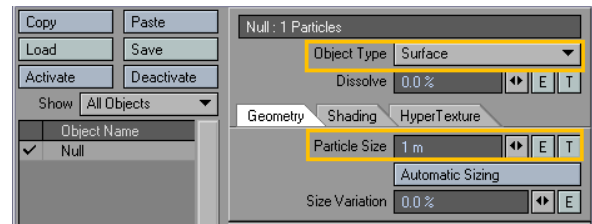


**Step 3:** On the **HyperVoxels Panel**, double-click on the null in the list window. This will activate **HyperVoxels** for this object. A check will appear to the left of its name. (You could also select the object and click the **Activate** button.)

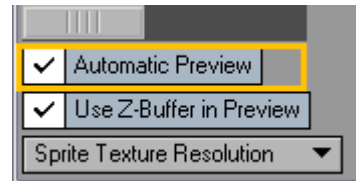


**Step 4:** **Object Type** should be set to **Surface**. When you

activate an object, the automatic size function is performed. The value 1m should appear in the **Particle Size** field.



**Step 5:** Make sure the **Automatic Preview** options are active so we can see the results of your efforts. **Automatic Preview** updates the preview as you make changes, so you don't have to keep clicking **Render**.

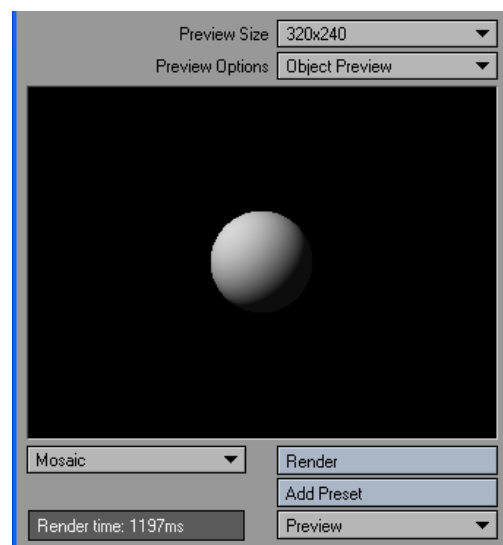


Open VIPER (there is a button for it on the Render tab, or you can hit **F7**). You should already see a rather dull ball in the VIPER window—of course it's a lot more interesting than a rendered null.



particles.

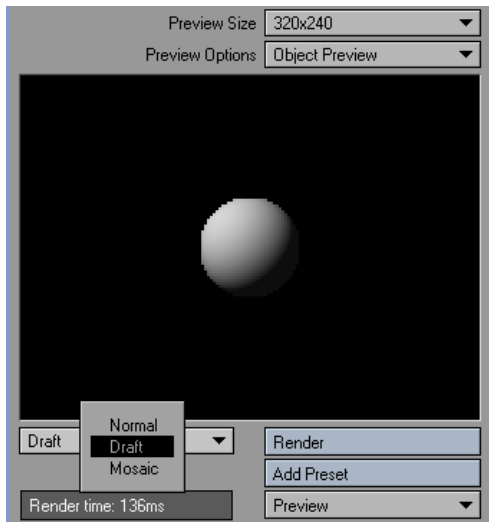
NOTE: You can preview animated textures with VIPER. Remember that VIPER uses the camera view, so make sure the camera can see your



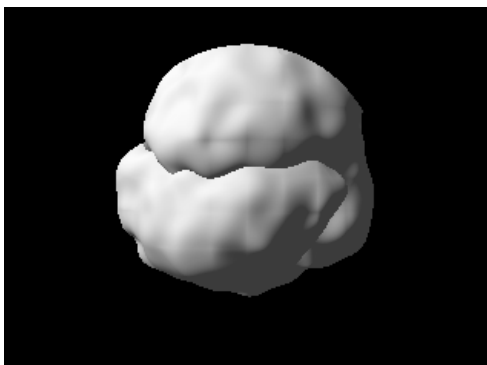
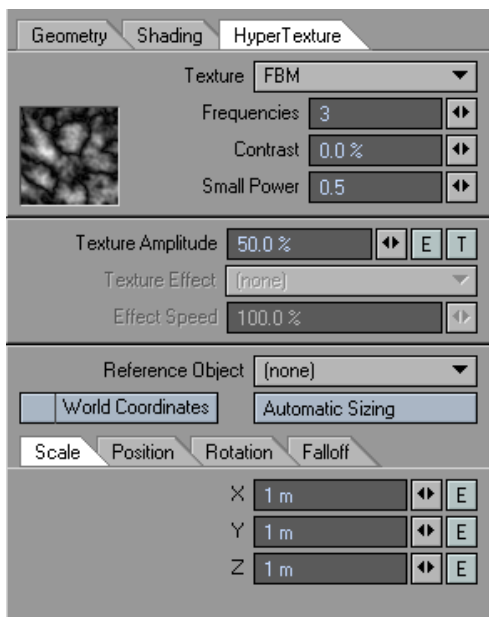




**Step 6:** Change the **Particle size** to 2. The ball should be larger now. Activate the **Draft Mode** to speed VIPER up.



**Step 7:** Click on the **HyperTexture Tab**. This is where you define the surface texture—essentially the terrain of the surface. Select **FBM** from the **Texture** pop-up. Notice that this is similar to setting a surface procedural texture.



## VIPER Render

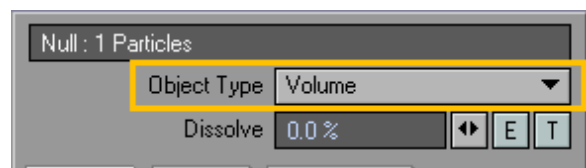
Go ahead and play around with the various settings on this tab and watch the **HyperVoxel** change.

You can actually apply all of the standard surface attributes to **HyperVoxels** on the **Shading Tab**. All of the settings should be familiar to you since they are identical in name and operation to the normal LightWave **Surface** settings. Try changing the various values and see their effects on **VIPER**.

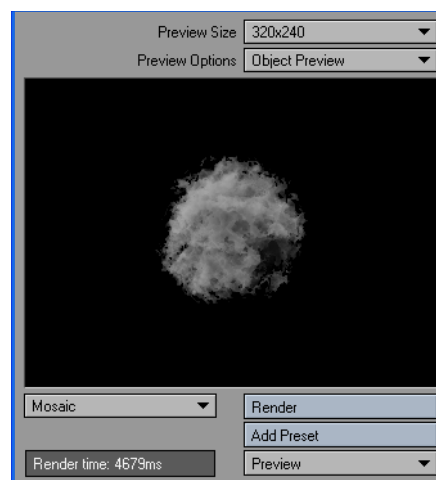
### Exercise: HyperVoxel volumetrics

This exercise will give you a flavour of some of the volumetric features.

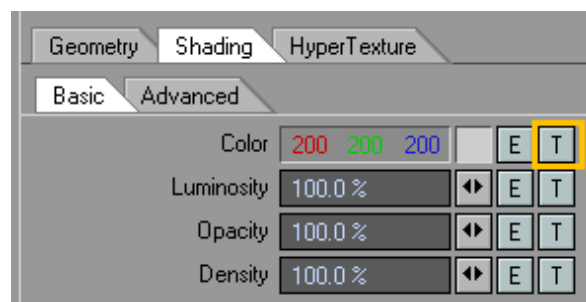
**Step 1:** Using the HyperVoxel from the previous exercise, change the **Object Type** from **Surface** to **Volume**.



**Step 2:** Go to the **HyperTexture Tab** and select **Turbulence** as the **Texture**. You should now see a nice wispy volumetric cloud. This object is totally three dimensional. You could fly the camera into it and continue to see all of the details.



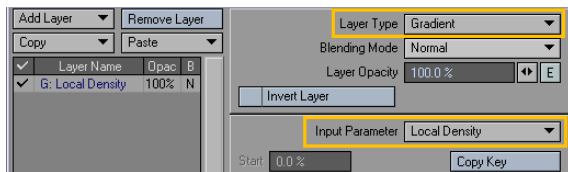
**Step 3:** Let's add some color. Go to **Shading Tab > Basic Tab** and click the **Color Texture** button.



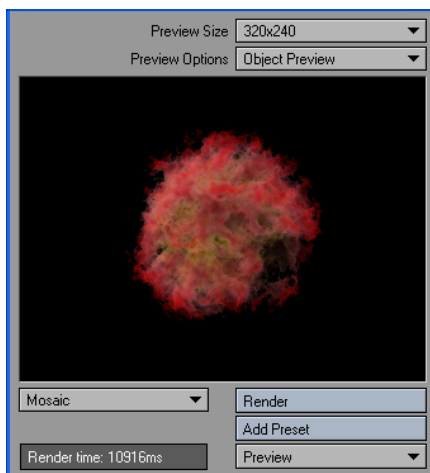
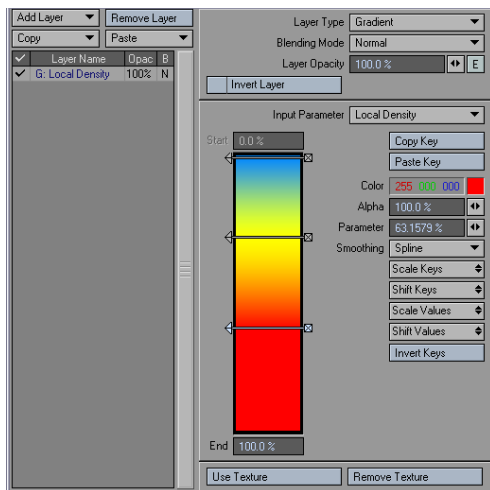




**Step 4:** Change the default **Layer Type** to a **Gradient** and use **Local Density** as the **Input Parameter**. This will apply the gradient based on the density of the cloud.

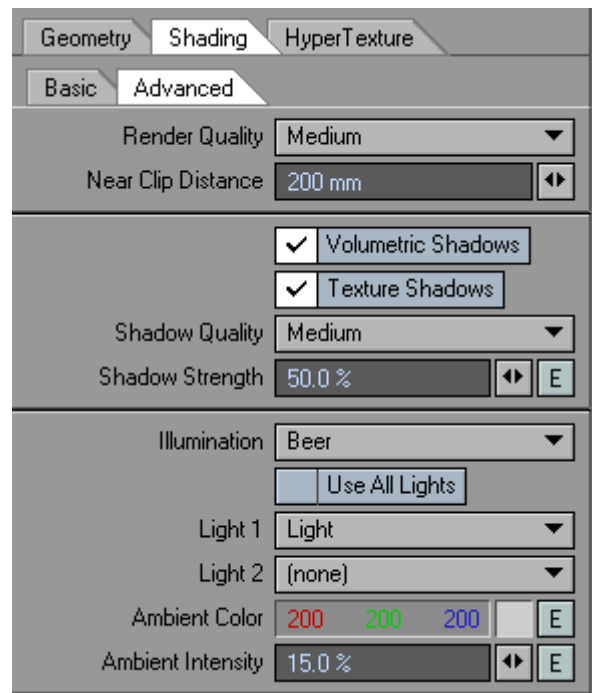


**Step 5:** Create three keys: blue, yellow, and red, top to bottom. To create these keys, just click in the gradient bar. You will see VIPER updating the preview as you make changes. Click **Use Texture** to close the panel when you are done.



**Step 6:** The options on the **Shading Tab > Basic Tab** control the basic look of your volumetric HyperVoxel. You can adjust the color, opacity, and density. The lower half of the tab controls the fractal look of the object. Try playing with the settings to see their effects.

**Step 7:** Click on the **Advanced Sub Tab**. As you might expect, this tab contains more advanced options. Included are options to define how the **HyperVoxel** object is lit, as well as self-shadow options.



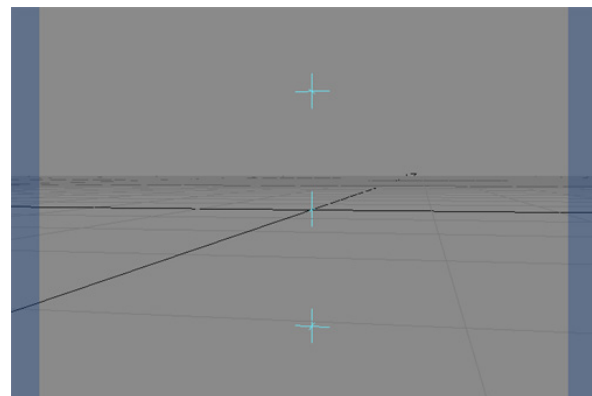
**Step 8:** Try playing with the various settings, including the **Sprite Mode**.

### Exercise: Blending HyperVoxel Objects

This exercise shows how different **HyperVoxel** objects can interact with one another.

**Step 1:** Add three null objects to an empty scene in Layout.

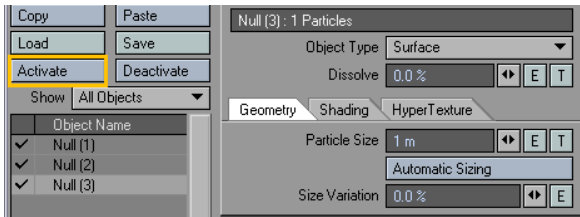
**Step 2:** Move and keyframe Null (1) to Y = 1m and Null (3) to Y = -1m.



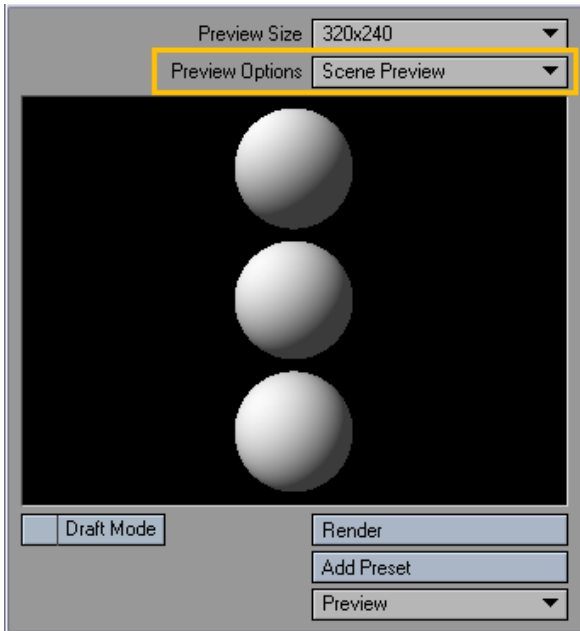
**Step 3:** Choose Window > Volumetrics and Fog Options to open its panel.



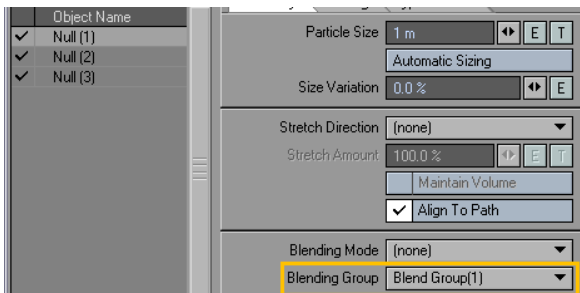
**Step 4:** Activate **HyperVoxels** for all three objects.



**Step 5:** Open VIPER and set **Preview Options** to **Scene Preview**. This will give us a preview of all of the HyperVoxel objects in the scene.

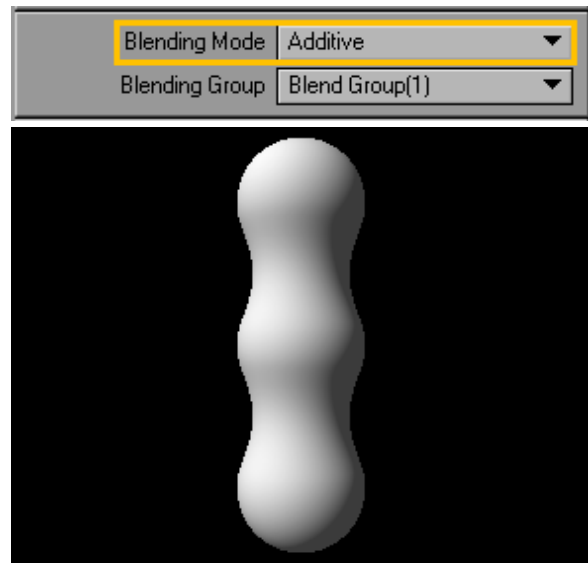


**Step 6:** Select Null (1) and choose **New Group** from the **Blending Group** pop-up menu on **Geometry Tab**. Accept the default name Blend Group (1) by clicking **OK**.

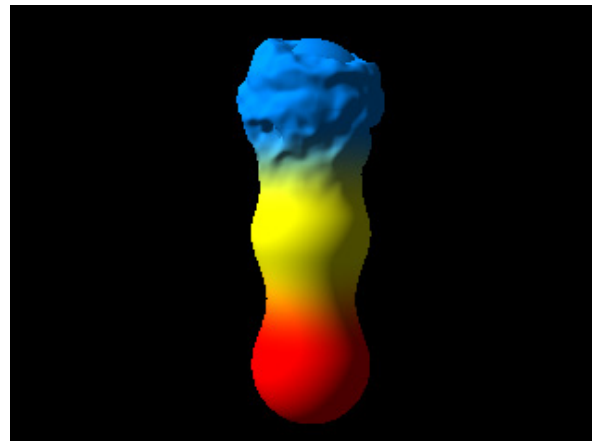


**Step 7:** For each HyperVoxel set the **Blending Group** to Blend

Group (1) and the **Blending Mode** to **Additive** (**Geometry Tab**). By placing all of the **HyperVoxels** in the same group, they can interact with each other.

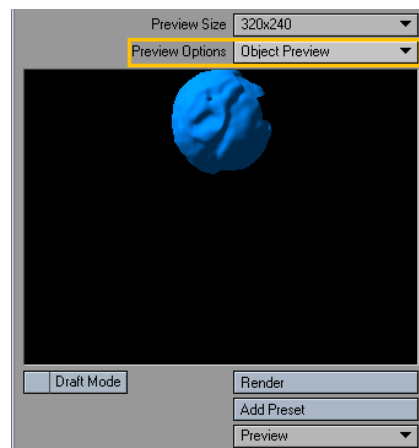


**Step 8:** Try changing the **Color** (**Shading Tab > Basic Tab**) of each null and giving them **Texture** (**HyperTexture Tab**).



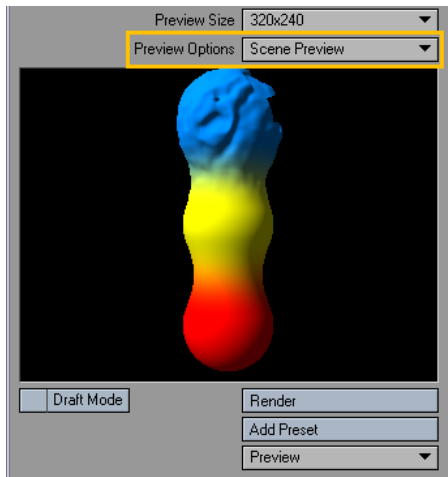
## Preview Options

When used with **HyperVoxels**, the **Preview Options** pop-up menu on the VIPER window has some options. **Object Preview** gives you a preview of the selected HyperVoxel object only from the camera's perspective.

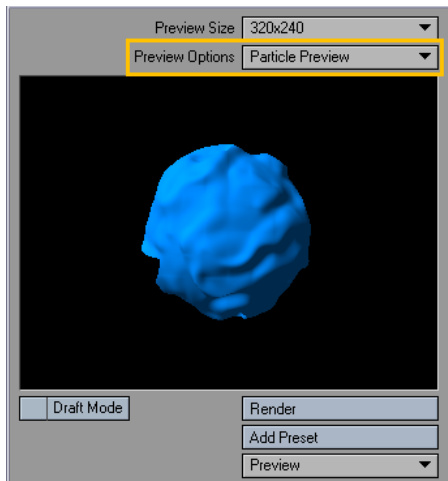




**Scene Preview** will show you all of the HyperVoxel objects which are visible from the camera's perspective.

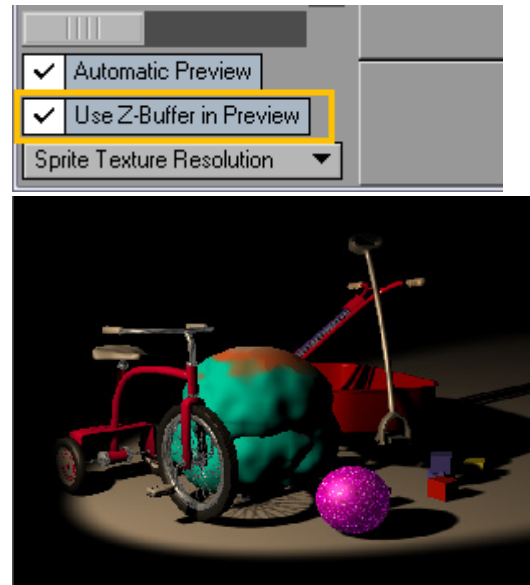


**Particle Preview** will give you an enlarged view of a single particle, which is particularly helpful if your HyperVoxel object consists of many particles.



## Use Z-Buffer

The **Use Z-Buffer in Preview** option (on the main **HyperVoxel Panel**) allows you to preview your **HyperVoxels** in the last rendered frame. (Note: **Object Preview** or **Scene Preview** modes only.) This requires that the **Enable VIPER** option be active on the **Render Options Panel**.

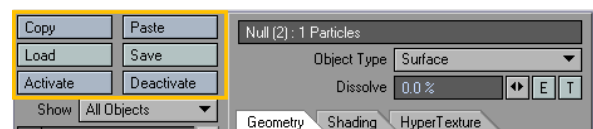


## Sprite Texture Resolution

The **Sprite Texture Resolution** setting (on the main **HyperVoxel Panel**) is a **Display** setting and does not affect your rendered **HyperVoxels** nor **VIPER**. It allows you to set the quality of textures used on **HyperVoxel** particles in Layout.

## HyperVoxels Setting Management

**Load/Save** buttons allow you to retrieve and store **HyperVoxels** setting files. This can be used to create a library of attributes.



Use the **Copy** button to copy the settings for the selected **HyperVoxel** object to a memory buffer. These can be pasted to the currently selected **HyperVoxel** object using the **Paste** button.

You can enable/disable **HyperVoxels** for an activated object by clicking in the checkmark column and you will not lose your settings. The **Activate** button will activate the selected object and set all settings to their defaults. The **Deactivate** button places the object in an inactive default state. With **Activate** and **Deactivate**, you will lose any existing settings.

If the **Show** pop-up menu is set to **All Objects**, then all objects in the scene will be listed. **HyperVoxels Only** will list only **HyperVoxel** objects; however, this includes disabled **HyperVoxel** objects.



## Sprites

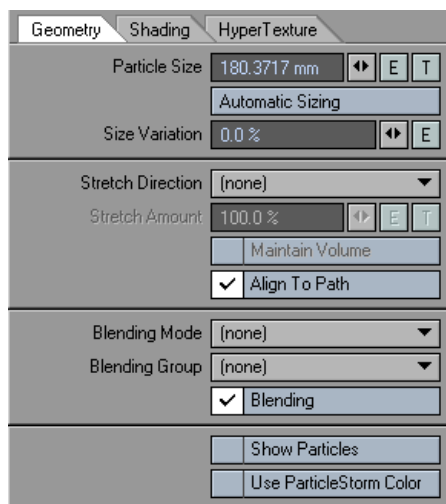
The **Sprite** mode is a simplified two-dimensional slice (or slices) of a HyperVoxel, so it can quickly produce results to give you a fast sketch of the HyperVoxel's rendered appearance. You might think of it as a slice of a **Volume** that has been mapped on a polygon that always faces the camera.

You may want to use Sprite to accelerate the setup process for Surface or Volume modes. It's even great for instances where you don't need a full 3D volumetric effect, like a smokestack in the distance.

Although Sprites are two-dimensional in nature, you can still fly a camera through them. The sprite will begin to dissolve when the camera gets into it and eventually fully dissolve when the camera hits the slice.

## Geometry Tab

Enter a specific radius for the individual particles in the **Particle Size** field. Click **Automatic Sizing** to compute an average size — this is a handy way to find a starting point. The average particle size is based on the shape of the entire object — **HyperVoxels** surfaces will not intersect at this setting.



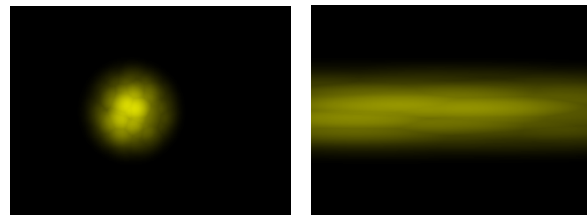
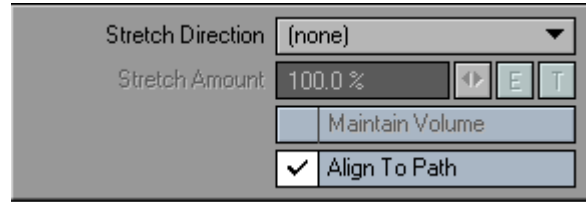
NOTE: When you first activate an object, automatic sizing is performed for you.

Rendering time is related to how much the particles intersect each other — the more intersections you have, the longer the rendering time. For fastest rendering, always try to use the smallest number of particles and the smallest **Particle Size** that achieves acceptable results.

**Size Variation** sets the maximum percent that the particle size can vary. For example, a particle size of 1m with a variation of 100% can be as small as 1m and as large as 2m. Using a variation of 50% would yield particles ranging from 1m to 1.5m in size. Use this setting to create random disturbances in the surface.

## Stretching and Rotating HyperVoxels

You can stretch **HyperVoxels** on a selected axis or based on velocity (**Stretch Direction**). The amount of stretching is determined by the **Stretch Amount** setting, which may be animated. When using nulls, you may stretch **HyperVoxels** by animating normal Layout XYZ size values.



Left: Without Stretch Direction, Right With Stretch Direction

**HyperVoxels** also respect the rotation of the object.

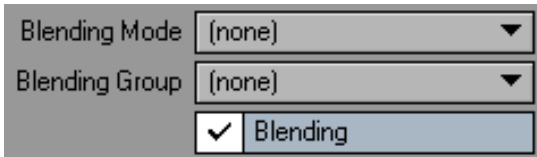
The **Maintain Volume** option on the **HyperVoxel Panel Geometry Tab** will cause the **HyperVoxels** to maintain their volume. This means that when they get squashed, they will maintain the same area.

## Align to Path

**Align to Path** causes the **HyperVoxels** to rotate as they follow a path. This can be particularly important when using particles.



## Blending HyperVoxels



Separate **HyperVoxel** items can interact with one another. You can individually set how the selected object interacts with other **HyperVoxel** objects. In order for **HyperVoxels** to interact:

All must belong to the same **Blending Group**.

All must have a **Blending Mode** selected.

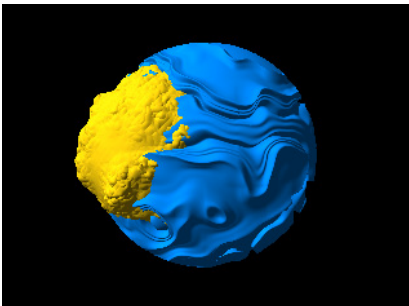
To view with **VIPER**, you must have **Scene Preview** set as the **Preview Options**.

Here, the two objects are shown separately:



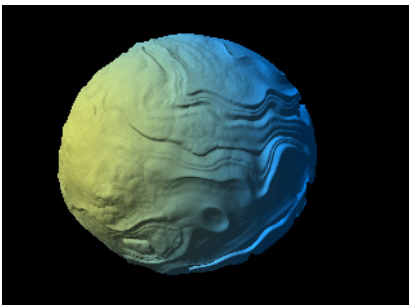
Left: Object 1, Right: Object 2

**(none)** means it will not interact with other HyperVoxel items.



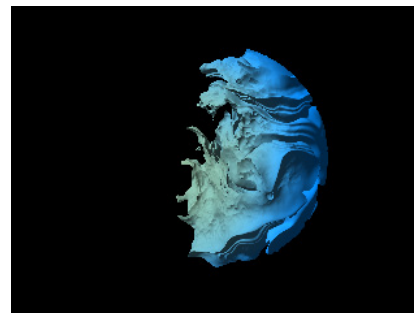
No Blending

**Additive** will smoothly blend **HyperVoxels**.



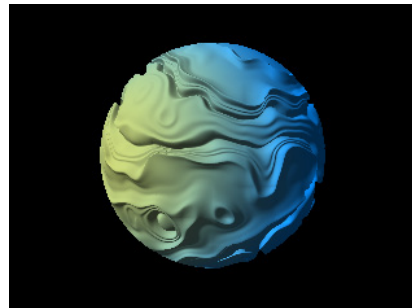
Object 1 Additive, Object 2 Additive

**Negative** will make the object push away portions of other **HyperVoxel** items that it comes in contact with — sort of like an animated boolean effect.



Object 1 Additive, Object 2 Negative

**Effector** is kind of a cross between **Additive** and **Negative**.

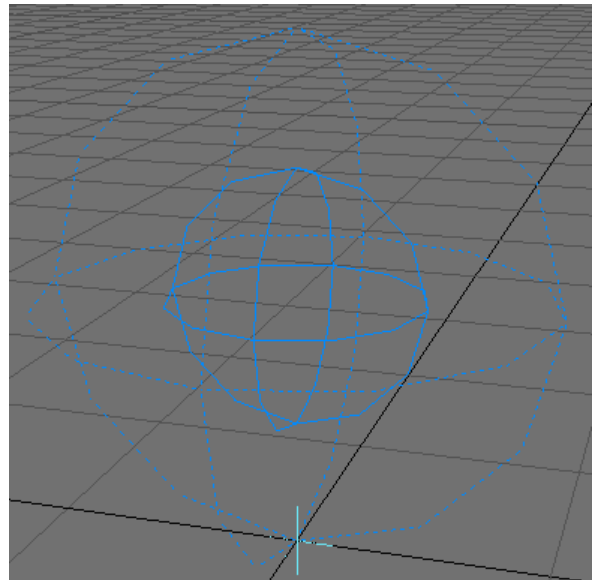


Object 1 Additive, Object 2 Effector

A **Blending Group** is the group of HyperVoxel objects which you want to interact with each other. You can select an existing group or create a new one.

## Show Particles

The **Show Particles** option will display a bounding sphere in viewports.



The Blending Box will enable/disable the surface tension of HyperVoxels. Checked, Blending will result in the HyperVoxels connecting in close proximity. Unchecked, the HyperVoxels will not connect in close proximity.

With Blending off, you can create things like sand, gravel, and other loose particulate matter.



## Use ParticleStorm Color

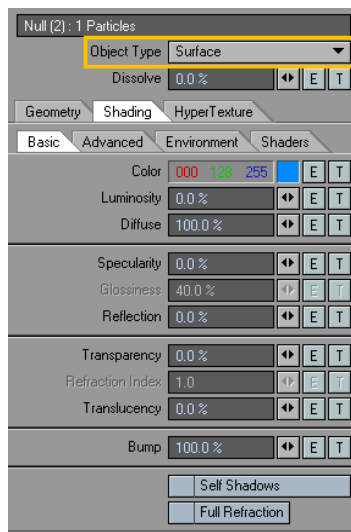
The **HyperVoxels** will use the color from Particle Storm particles, particles, if available. Particle Storm (now retitled Napalm) is a plugin available from Dynamic Realities.



NOTE: The **HyperVoxels Particles Displacement** plugin lets you set the **HyperVoxels** particle base color to that of the **Vertex Color Map**, if one exists.

## Shading Tab: Surface Mode

The parameters on the **Shading Tab** work just like their cousins on the **Surface Editor**.



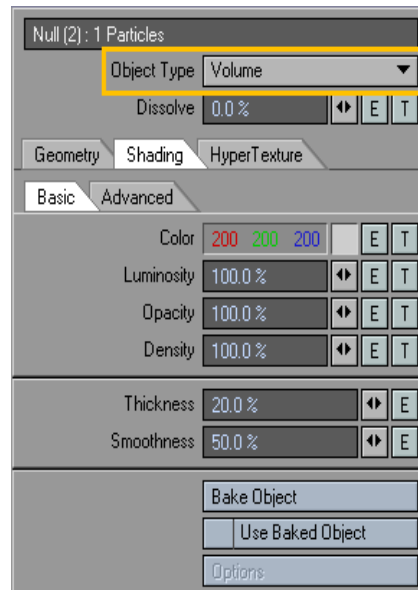
**Self Shadows** lets the HyperVoxel cast shadows on itself. (The HyperVoxel will cast shadows on other objects in the normal way.) **Full Refraction** forces rays to bounce each time a new refracting (i.e., transparent) surface is found. If this option is inactive, one refraction bounce is computed, which is enough most of the time. Full Refraction will, of course, increase render time, but will result in more sophisticated effects.



HINT: You might try the (faster-rendering) corresponding bump textures on the **Shading Tab** without HyperTextures, if you don't need the true surface details. Remember this: "Don't get hyper if bumping will do" or "A bump in the night is not worth getting hyper over."

## Shading Tab: Volume Mode

The options on the **Shading Tab** are different for **Volume HyperVoxels**, which are gas- or cloud-type effects. Use **Color**, **Luminosity**, **Opacity**, and **Density** settings to change those parameters for the volumetric object.



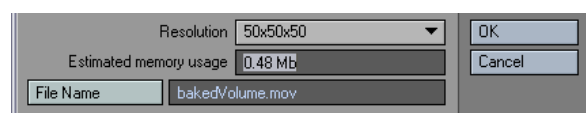
The **Thickness** setting, which defaults to 50%, lets you adjust the general heaviness of the volume. Higher **Smoothing** values yield less of a hard-edged look and volumes will appear more wispy.

### Baking HyperVoxels

The baking feature subdivides **HyperVoxels** into a 3D grid of subvolumes (a space subdivision). This dramatically reduces the necessary computations and allows preprocessing of the **HyperVoxel** effect, including shadows, textures, and so on. The result is essentially a stack of **Image Maps**.

Click the **Options** button to display the baking options. The size of the "image map stack" is set on the **Resolution** pop-up menu. The first two numbers define the size of the images and the third defines the number of frames.

Enter the name of the file to save in the **Filename** field. The animation codec (**File Type**) used when saving the volume data is very important. By default it uses QuickTime uncompressed because this format preserves the alpha channel. This is important because the alpha channel encodes the volume opacity information. Many codecs don't handle the alpha channel properly and the opacity information will be compromised. (Note that you could edit the file externally in another program or even create one from scratch.)



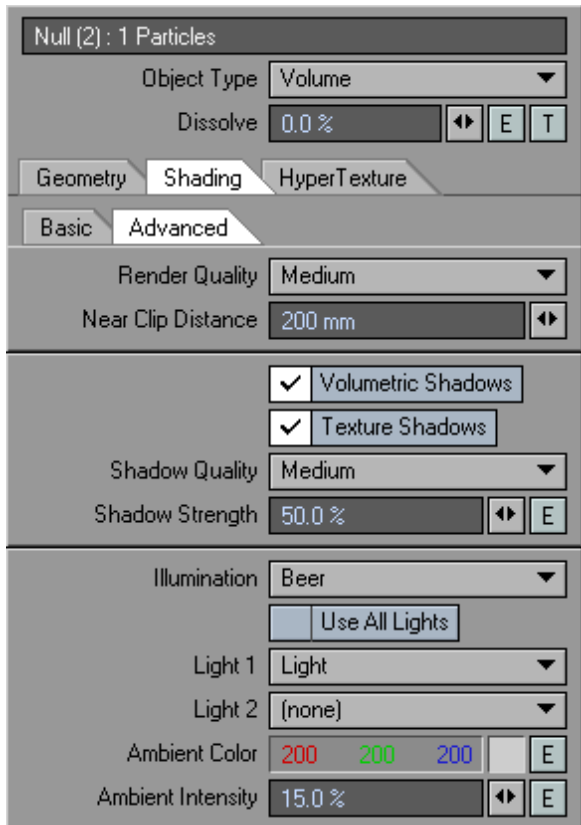
Click **Bake Object** to execute the baking computation. When complete, you should notice that most of the normal settings will be ghosted except for some basic ones like **Color**, **Luminosity**, and so on. These can be adjusted to change the look of the baked object.





## Volume Mode Advanced Sub-tab

The various items on the **Render Quality** pop-up menu determine the level of detail **HyperVoxels** uses to render the volume. The default is **Medium** and should be adequate for most situations. **Near Clip** is the distance from the camera where the rendering starts. The default value is normally fine. A lower value will render finer details, but take longer to render. You might adjust this setting if the camera was inside a cloud. It really depends on the scale of the **HyperVoxel** objects and the scale of the details you want to see.



The **Volumetric Shadows** option creates shadows within the volume, which add detail and realism. However, it can increase render time significantly, especially when you are trying to simulate very dense clouds, like volcanic smoke. The **Texture shadows** option yields a very realistic shading of the hypertexture, but increases rendering time significantly. If you disable this option, you will miss all of the hypertexture shadow details.

Use the **Shadow Quality** pop-up menu to choose the render quality of the shadows. The **Shadow Strength** sets the darkness of the shadows.

The **Illumination** pop-up menu determines how the **HyperVoxels** volume is lighted. **Self** acts as if the light comes from the viewer. It can be useful to see more details on the object. With **Constant**, the scattering of light is constant, that is, the amount of light scattered at one point is proportional to the amount of light received at that point.

**Rayleigh** simulates a strong forward scattering of light within the volume. Light scatters in the direction of the light relative to the viewer, which means that illumination will be at its maximum when the light faces the viewer. This is perfect for back-lit clouds, smoke, and so on.

**Beer** uses the pattern's sample density amount as the illumination value, dramatically reducing contrast between lighted areas. The setting uses the physical model for scattering of light inside a gas, and is a good setting for clouds. For a more "cartoony" look, try **CelShade**.

You can choose to have all lights affect the **HyperVoxel** with **Use All Lights**, or choose one or two specific lights.



**HINT:** Try to avoid using these **Light** options, if you are ray tracing shadows. Ray-traced shadows on volumetrics is incredibly computationally intensive and will often result in unacceptable rendering times. A single pixel in volumetrics requires 20 or more samples and a shadow is ray traced for every sample. Try to fake lighting with Ambient Intensity and surface Luminosity or use **Shadow Map Spotlights**.

**Ambient Color** lets you specify the ambient light color of the volume. This setting works hand-in-hand with the **Color** setting on the **Shader Tab**.

**Ambient Color** is similar to ambient light in Layout, except that it applies only to the selected HyperVoxel object. Basically, the shading process is very similar to what happens with regular object surfaces.

Because of the way illumination is calculated inside volumetrics objects, **Ambient Intensity** can be higher than 100% without overexposing the image.

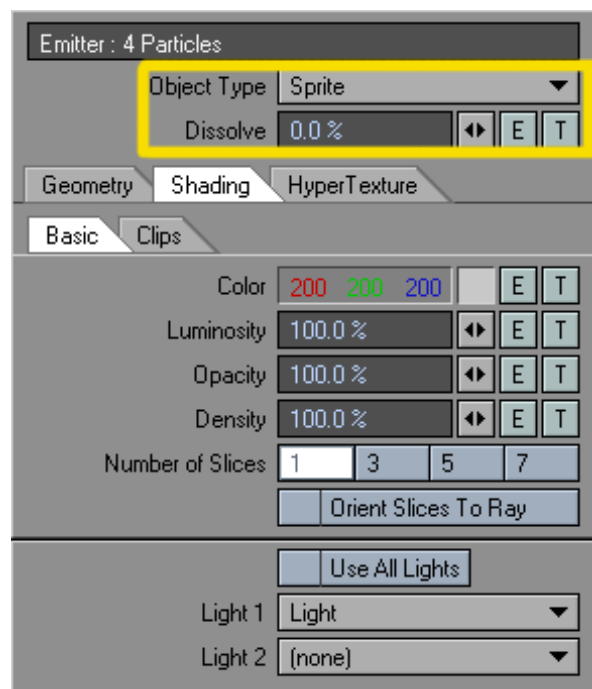


## Shading Tab: Sprite Mode

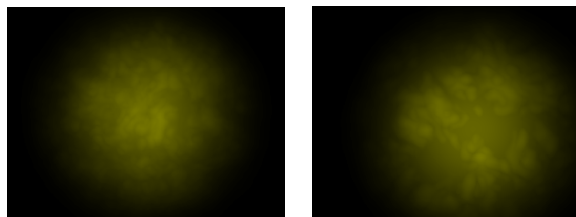
The options on the **Sprite Mode Basic Subtab** are similar to the options available for the Volume mode. This makes sense since the Sprite mode is just a slice of a Volume mode HyperVoxel.



**Density** on sprites is a multiplier for both **Opacity** and **Luminosity**. It is essentially the opposite of **Dissolve**.



With the **Number of Slices** setting, you can “thicken up” the effect by increasing the number of slices used for sprites from the default setting of 1. This allows you to get something a little closer to the **Volume Mode** while still retaining the speed of the **Sprite Mode**.



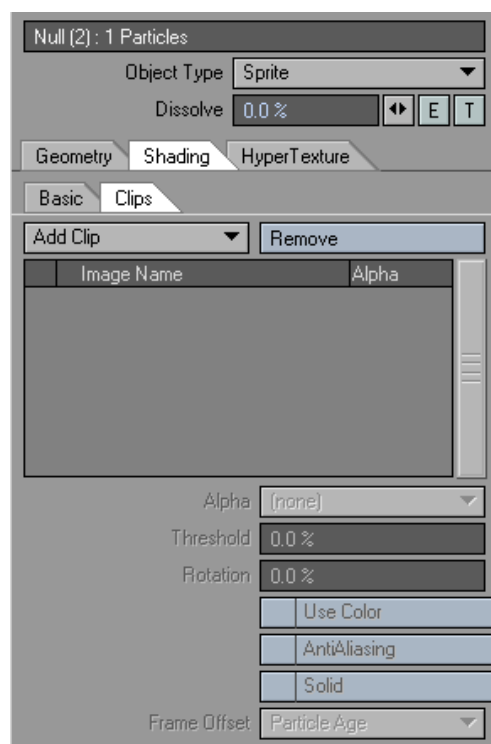
Left: 1-Slice, Right: 3-Slice

You may choose to light the Sprite using all lights in your scene or up to two specific lights.

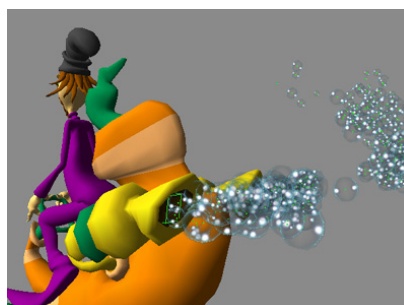
**Orient Slices to Ray** orients the sprite's coordinate system with the ray, rather than the default of aligning with the camera.

On the **Clips Subtab**, you can map an image directly onto a sprite using the **Add Clip** button.

Any image available to the scene may be selected. Use the **Image Editor** to load an image not already available.



Clips are added to the HyperTexture. If you want to see only the clip, set the **Texture** option to **(none)** on the **HyperTexture Tab**.





Some options on the **Alpha** pop-up menu let you cut away parts of the image you don't want to use. If the image file includes an alpha channel, choose **Embedded** to use the alpha channel. Use **Luminosity** to leave brighter parts of the image. The **Black** mode will cut away black areas. The **Threshold** level lets you increase the range for what is considered black.

The **Rotation** setting sets a rotation speed for your clips. This is perfect for effects like billowing smoke and explosions. The higher the setting, the faster the rotation. Positive values make the clip rotate clockwise. Rotation always faces the camera.

If you do not want to use the color information in the image, uncheck **Use Color**. Then, only the greyscale information will be used. **Antialiasing** will yield better-looking edges, but it significantly increases render time and memory requirements.

When **Solid** is checked, the sprite is rendered as a solid object, preventing the normal additive result of overlapping sprites.

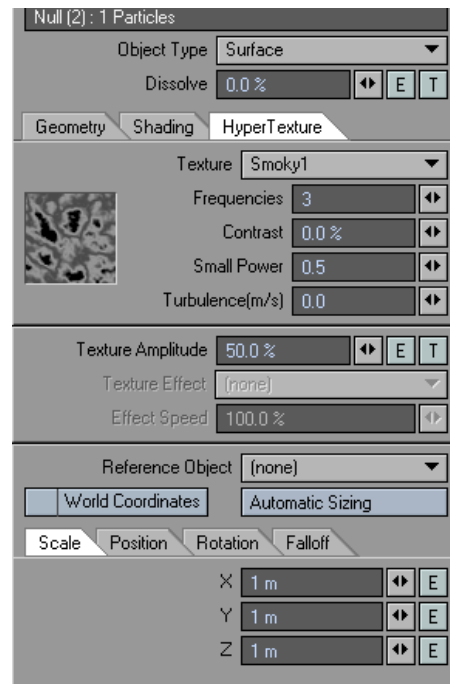
## Sprite Clip Frame Offset

If you use **HyperVoxels** with particles, you can use multiple clips. The **Frame Offset** setting determines how you want to map an image sequence to a given particle. **Particle Age** starts the sequence for a particle at birth. Thus, the image will likely be different for particles at a given frame. **Uniform** will use the layout time as the time in the sequence. Thus, the image used is the same at each frame. When the bottom of the list is reached, it starts again from the top. **Random** will assign a random offset to each particle.

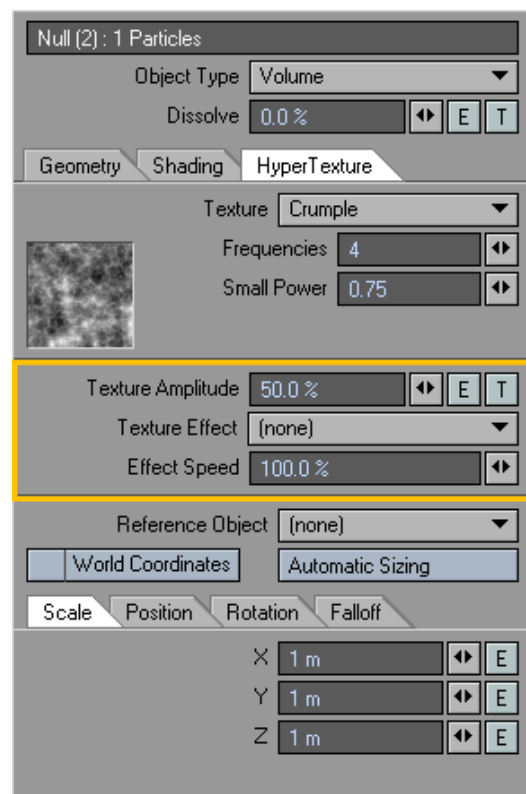
**Frame Offset** is an important control for sprite animations. It allows you to decide how to attach images to particles. **Particle Age** is probably the most useful. You can have literally hundreds of particles, each mapped with distinct image sequences. For example, you could have particles with independent water-splash image sequences, creating a complex and rich visual effect.

## HyperTexture Tab

The **Texture** pop-up menu determines what the texture will look like. The textures in this menu correspond to the identically named procedural textures, although not all of them will be available here.



The big difference is when you are using a **Volume (Object Type)** HyperVoxel. In this case, you can animate the texture effect.





In the **Texture Effect** pop-up menu you have several options. **Turbulence** makes the texture move towards the viewer. It is like animating a texture with the texture velocity parameter. **Billowing** animates the texture by rotating outward along a velocity vector (à la Dante's Peak). It is very good for explosions. **Displace** adds variations to the texture. The effect is best seen when the texture is moving. **Dissolve** gradually lowers density, making particles disintegrate.

**Velocity Translate** moves the texture in the direction of the particle velocity. Since each particle has a different velocity, this can result in a cool and complex texture motion. **Dissolve & Expand** might be used on explosions. It dissolves the effect as it expands. Some trial and error may be necessary to get the desired look.

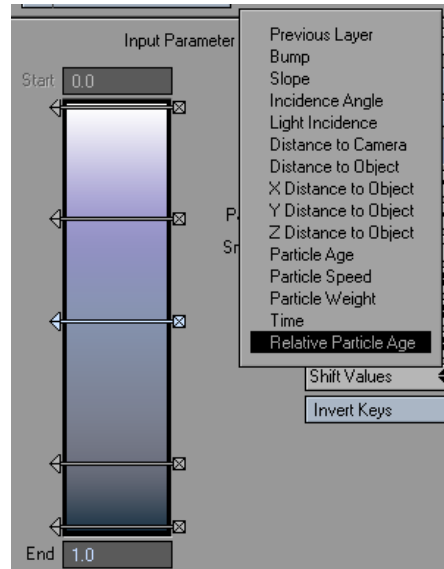
The **Effect Speed** setting is the duration of the looping cycle. The value is equal to a percentage of Layout's **Default** units, on the **General Options Tab of the Preferences Panel**, per frame.



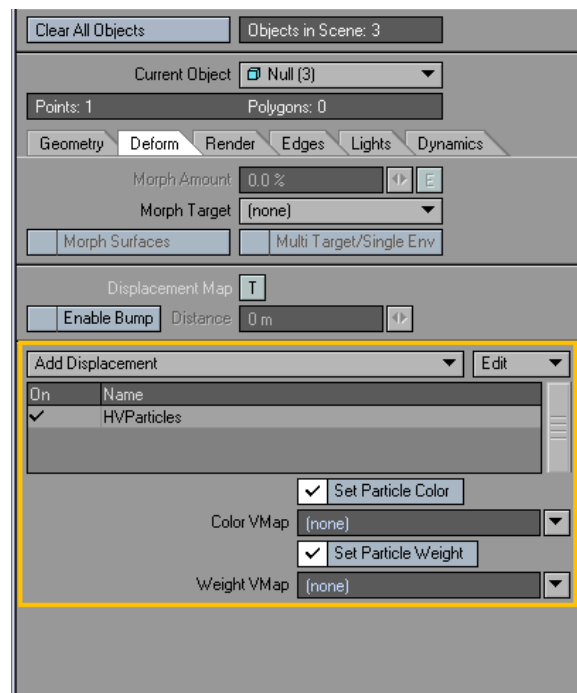
NOTE: Animated textures can be previewed with VIPER.

## Gradient Input Parameters

Special **Input Parameters** will be available for texture gradients on HyperVoxel parameters. **Particle Age** is simply the age of (Particle FX) particles, in frames. **Particle Speed** is the particle speed in default LightWave units (usually meters) per second. Use **Time** to change the setting over a range of frames.



The **Particle Weight** Input Parameter is used with regular objects that use **HyperVoxels** and requires them to have a **Weight Map**. The object must also have the **HyperVoxels Particles Displacement** plugin added. **Set Particle Weight** should be checked and the **Weight Map** selected.

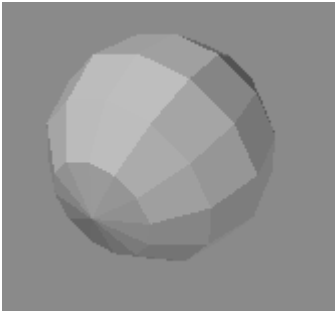


For **HyperTexture** and **Shading** settings, gradient textures can use a distance to particle (along a specific axis) input parameter. This is the surface distance from the actual particle position along the selected axis.

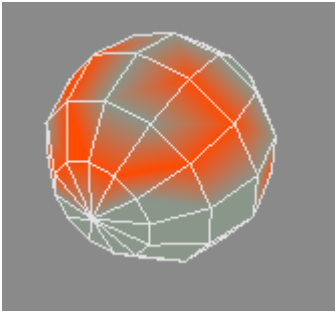


### To use the Particle Weight Input Parameter:

**Step 1:** Create an object in Modeler.



**Step 2:** Create a **Weight Map** and add some random weighting to the points.



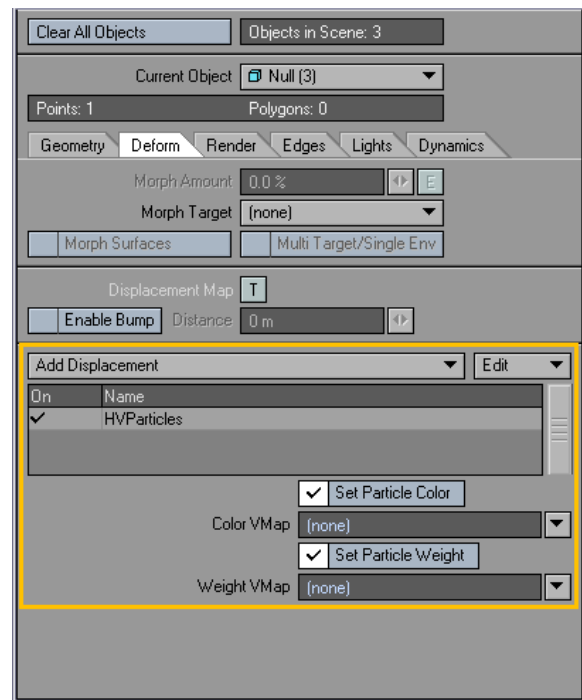
**Step 3:** Load the object into Layout.

**Step 4:** Choose **Window > Volumetrics and Fog Options**.

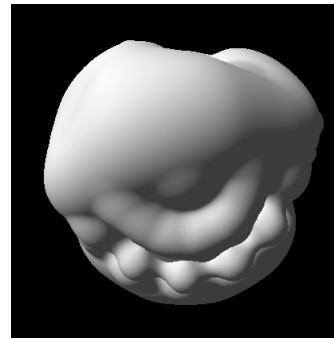
**Step 5:** Open the **HyperVoxels Panel** and activate the object.

**Step 6:** Open the object's **Properties Panel**, add the **HyperVoxels Particles Displacement** plugin.

Open its options, activate **Set Particle Weight** and choose the **Weight Map**.



**Step 7:** On the **HyperVoxels Panel**, add a texture to the **Particle Size** and set the **Layer Type** to **Gradient**. Choose **Particle Weight** as the **Input Parameter**. Create some keys.





## SkyTracer2

### SkyTracer

The **SkyTracer Environment** was designed to create sophisticated atmospheric effects using real-world parameters.

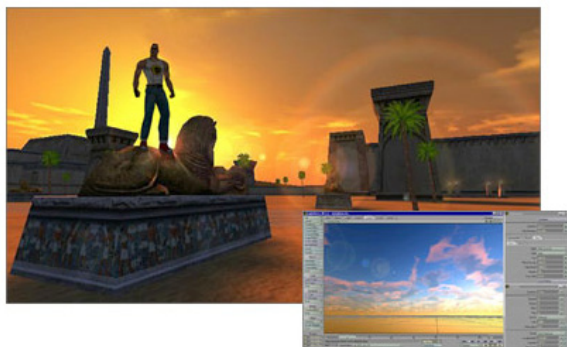


NOTE: This is a legacy tool replaced by **SkyTracer2** and NewTek recommends that you only use this to read older LightWave scene files.

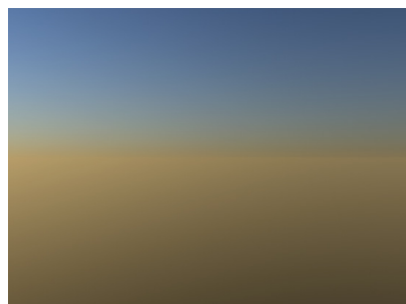
**SkyTracer2** offers all these features and is far more robust.

### SkyTracer2

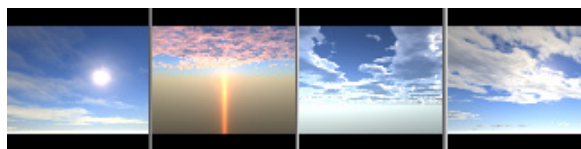
The **SkyTracer2 Environment** was designed to create sophisticated atmospheric effects using real-world parameters. You can adjust the **Atmospheric**, **Cloud**, and **Sun** settings within the interface to create a variety of beautifully rendered sky images. These skies can be rendered (volumetrically or as a 2D effect) within an existing scene, or saved as image files to be composited or used as **Texture Maps**. The **SkyTracer2 Environment** was designed to create sophisticated atmospheric effects using real-world parameters. You can adjust the **Atmospheric**, **Cloud**, and **Sun** settings within the interface to create a variety of beautifully rendered sky images. These skies can be rendered (volumetrically or as a 2D effect) within an existing scene, or saved as image files to be composited or used as texture maps.



**SkyTracer2** simulates light scattering and light absorption in the atmosphere. The settings for this simulation are real-life parameters such as atmosphere thickness, cloud altitude, sun position and so on. Because **SkyTracer2** uses volumetric lighting techniques, you can even get spectacular effects like the sun casting light rays through the clouds. You can adjust both the current **Time** and **Location** settings to accurately simulate the sun rising or setting, anywhere in this world or another.

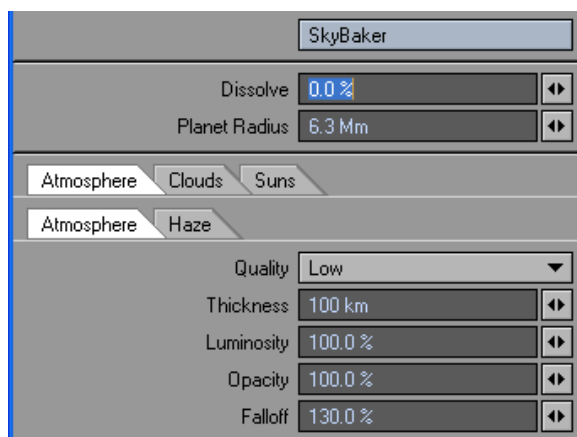


Default SkyTracer2 settings

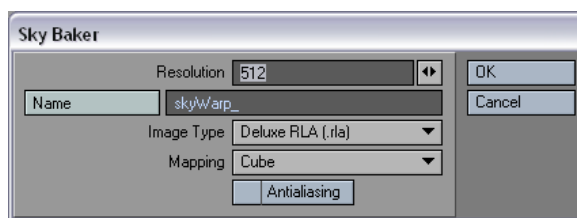


Several Presets

The interface is split into two sections: global controls are at the top of the panel and controls to adjust atmospheric effects are at the bottom of the panel.



The **SkyBaker** button brings up the **SkyBaker Panel**.



The **Dissolve** setting adjusts the transparency of the **SkyTracer2** effect and the **Planet Radius** adjusts the size of the planet, and thus the size of the atmosphere. The default value of 6.3Mm represents the size of the Earth.





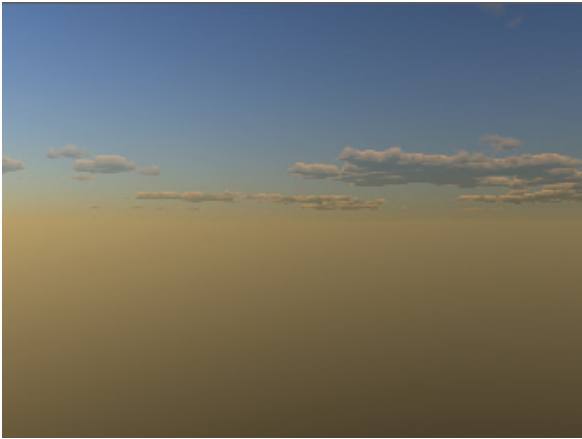
## The Atmosphere Panel

Think of the sky as a stack of two layers, Atmosphere and Haze. These layers have identical controls, but each can have different values for scattering, absorption, and density distribution.

**Quality** is the level of sampling performed along the camera's viewing direction. Higher **Quality** levels give more accurate renders, however, this setting also affects rendering times. Therefore, the **High** setting should be reserved for very precise effects.

The **Thickness** setting controls the thickness of the atmosphere and haze layers. **Thickness** is measured in kilometers and meters respectively, with 100km and 1000m as default values.

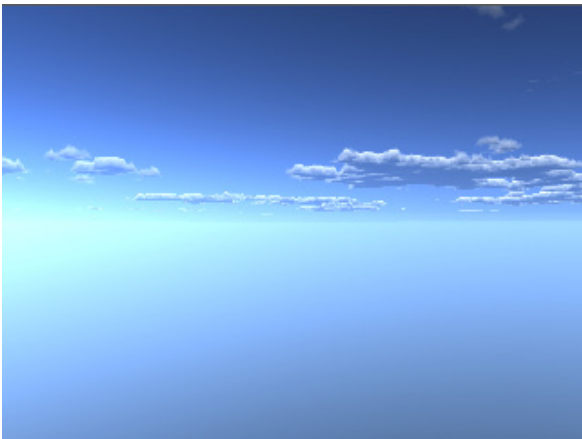
The **Luminosity** parameter adjusts the percentage of light that scatters within the layer. This scattering intensity is greatest in front of the sun and lower elsewhere.



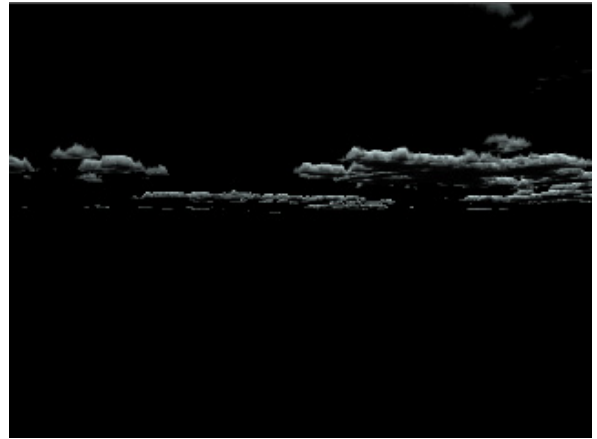
Thickness 100%



Luminosity 100%



Thickness 50%



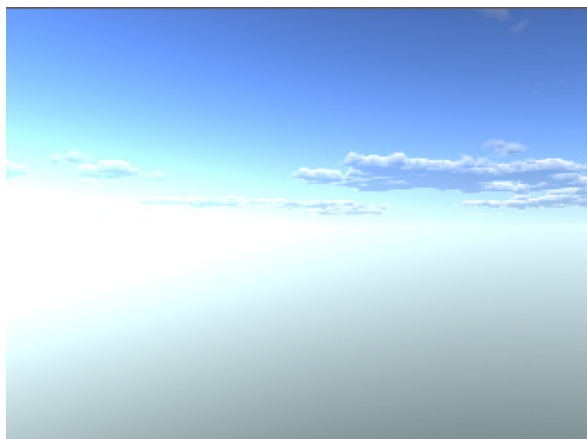
Luminosity 0%



**Opacity** measures the percentage of light absorbed within the layer. High values attenuate light, and objects disappear at the horizon. When you use opacity with haze, you can create pollution-like effects. However when you adjust the opacity within the **Atmosphere Panel**, you can create strange, out-of-this world skies.



Opacity 50%

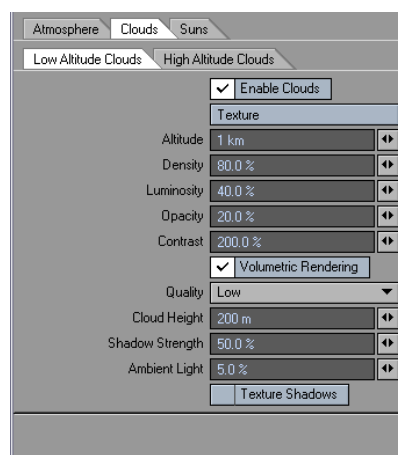
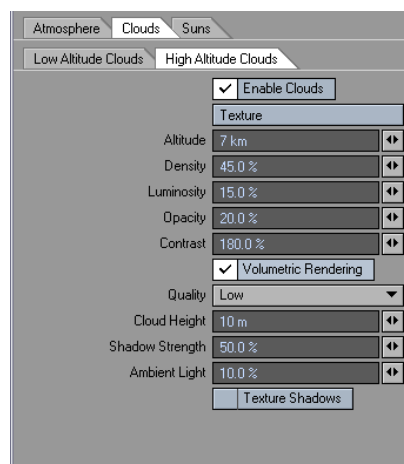


Opacity 200%

**Fall Off** modifies how density is distributed throughout the layer. Values higher than 100% decrease density rapidly with altitude, while a value of 0% has no decrease at all. Negative values in the **Haze Falloff** setting will invert the density distribution.

## The Clouds Panel

No sky would be complete without some clouds! By controlling the clouds' altitude, density, luminosity, opacity, and contrast parameters, **SkyTracer2** can create an incredible range of cloud effects. The two layers in **SkyTracer2**, low and high-altitude clouds, have identical parameters; you can enable these layers separately for drastically different effects or together for complex sky patterns.



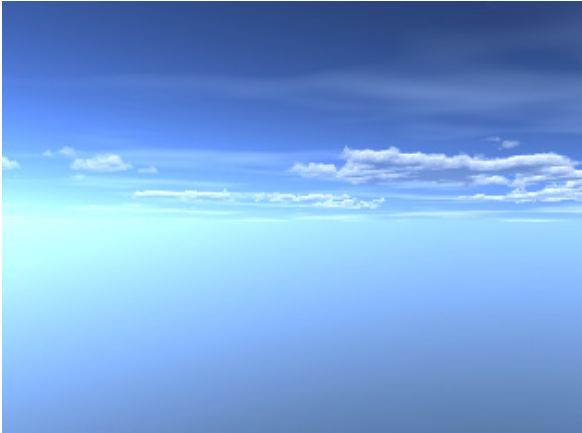
Use the **Enable Clouds** switch to turn the two cloud layers on or off.

The **Texture** button opens the **Texture Editor Panel**, which is where you specify the type of cloud to render and adjust its coverage parameters. (see below)

**Altitude** sets the height in the sky where the cloud layer will start.



The **Density** setting controls how dense the cloud layer is. Think of this setting as adjusting the probability of rainfall for the area. A **Density** value of 10% means that the clouds are fairly devoid of water. This gives a much thinner cloud layer, which allows more light to travel through it. A value of 75% has a much thicker appearance, and shows a higher probability of rain.



Density 20%



Density 200%

**Luminosity** controls the scattering intensity of light through the clouds. With high values, you create bright clouds. The light scattered from the cloud is attenuated by the **Atmospheric** and **Haze** settings.



Luminosity 10%



Luminosity 100%



**Opacity** controls how much light is allowed through the layer. A setting of 100% renders a completely dark sky, while a value of 0% renders a much brighter sky.



Opacity 10%

The **Contrast** setting affects the coverage of the cloud layer. A value of 10% gives a constant sheet of clouds, with little variation within the layer. While the default value of 200% shows subtle variation within the cloud, it also reduces the cloud's coverage.



Contrast 100%



Opacity 50%



Contrast 200%



Enable the **Volumetric Rendering** option to specify how **SkyTracer2** will render the layer. When **Volumetric Rendering** is enabled, **SkyTracer2** renders the clouds as a volume that has both density and mass. When **Volumetric Rendering** is disabled, **SkyTracer2** renders the layer like an image projected on a piece of geometry in the sky. Disabling this effect will greatly reduce rendering time, but it will also generate a less convincing cloud layer.

If enabled, **Volumetric Rendering** also makes the cloud quality, height, shadow strength, and ambient light parameters available. These options do not apply to a cloud layer when you disable Volumetric Rendering.



Activated



Deactivated

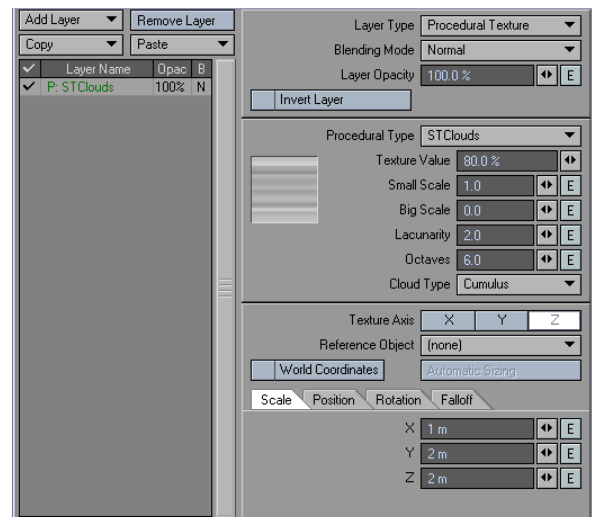
**Textured Shadows** activates the cloud layer's self-shadowing option, which generates more accurate shadows within the layer, but increases rendering times. **Textured Shadows** is similar to HyperVoxel's self-shadowing feature.

Like in reality, you can get an under-lit effect when the sun is very low on the horizon. In this case, the clouds are lit from beneath, creating spectacular red lighting effects. The effect is best accomplished by using volumetric clouds with textured shadows.

## The Texture Editor Panel

**SkyTracer2** uses a texture defined in LightWave's **Texture Editor** to control the style, coverage, and placement of the cloud layers. Although you can use a **Procedural**, **Image**, or **Gradient** texture layer, the most common texture used will be the STClouds procedural texture. This procedural texture is specifically designed to work with **SkyTracer2** and contains many of the settings needed to create realistic clouds.

Use the procedural preview window in **Texture Editor** to quickly see results when you adjust settings for cloud style and coverage. Simply adjust one of the procedural's settings and watch how it affects the computed fractal.



The

**Texture Value** controls the strength of the cloud layer. The higher the percentage, the more cloud coverage in this layer.

The **Small Scale** and **Big Scale** values are two fractal scale parameters for the appearance of the clouds.

**Lacunarity** sets the turbulence within the cloud layer. A value of 1 renders the clouds nice and smooth, while a value of 5 or higher breaks up and distorts the cloud to give it a much more natural appearance.

The **Octaves** setting controls the amount of frequencies or detail your fractal receives. The higher the value, the more frequencies are used when rendering. This will create more details within the cloud, but it also causes longer rendering times.

**Cloud Type** lets you select from cloud styles. Cumulus clouds are big and puffy low-altitude clouds; Cirrus clouds are long and thin and are generally found at higher altitudes. You can even simulate the streaming effect that jets create when flying at high altitudes by selecting the Jet Trails Cloud Type.

**SkyTracer2** applies textures to an imaginary cloud layer that is parallel to the XZ plane. Therefore when you apply textures with a planar projection, you should use the Y axis. You can create realistic cloud effects like attenuating the clouds in the distance using **X** and **Z Falloff**, or create pyramid-shaped clouds (as in Cumulus) by using a texture **Falloff** on the Y axis.



NOTE: The **SkyTracer2** default texture has a **Falloff** setting already applied.



Instead of using a procedural approach, you can specify the clouds' exact coverage by loading a fractal-like image on the Y axis. To get a convincing effect, you should use a smoothly repeating texture.

Use a gradient in any layer to modify the previous layer's effect. The gradient **Input Parameters** are **Heading** (ray heading angle), **Pitch** (ray pitch angle), and **Height**. The height is the height inside the cloud layer.



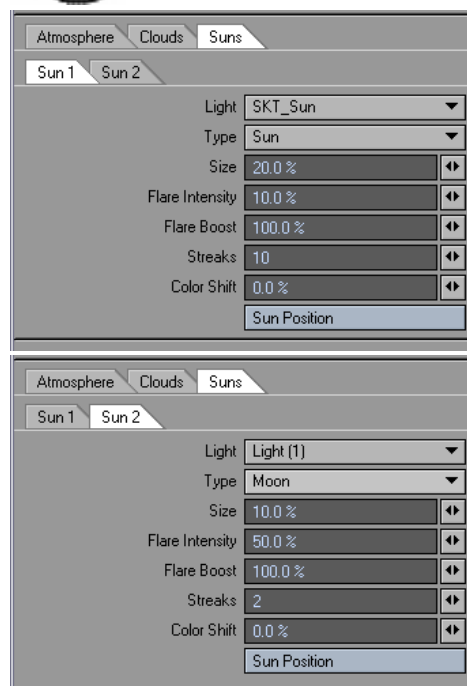
NOTE: That is 0 in the bottom and 1 at the top.

## The Suns Panel

**SkyTracer2** uses one or two lights to render the clouds and atmospheric effects. You can use either your own lights or the SKT\_Sun lights added by **SkyTracer2**. An added bonus: you can also set up these lights to render as either a sun or a moon flare.



NOTE: By default only one Sun light is added.



You can select which light(s) **SkyTracer2** uses to calculate the appearance of the clouds in the **Sun1** and **Sun2 Light** drop down list. The **Type** lets you choose to render the light as a sun or a moon. You can change the appearance of the sun or moon by adjusting the flare's **Size**, **Flare Intensity**, **Flare Boost**, **Streaks**, and **Color Shift** values.





## Sun Position

By pressing the **Sun Position** button, you access the **Sun Spot Motion Modifier**. You adjust the sun's position by specifying the exact day and time for the light to be positioned.

Although you may never see it, the position of the SKT\_Sun light rotates properly as time passes. To increase the speed of the animating sun, edit the **Time Lapse** field. To complete the time lapse effect, animate the cloud's texture parameters.

## The Sky Baker Panel

Once you create your sky, you can either render it on every frame along with your scene or **Sky Baker** can generate image files to map it onto geometry and fake the effect. Several reasons are listed below for why you would want to bake a sky:

1. You can see your sky in real-time using OpenGL.
2. The sky is saved as an image, so you can edit it with the **Image Editor** controls (e.g., hue, saturation, gamma, etc.).
3. The solution is already figured out, so the sky renders extremely fast.

The **Sky Baker Panel** lets you select the resolution of the images, their image type, mapping projection style and a file prefix name. The **Antialiasing** option activates and deactivates the Antialiasing pass when rendering the images.



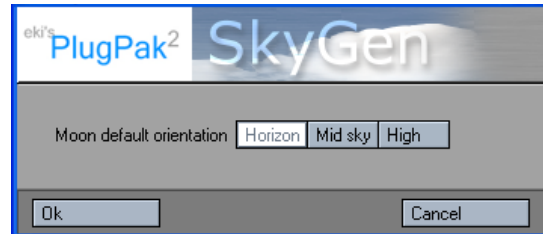
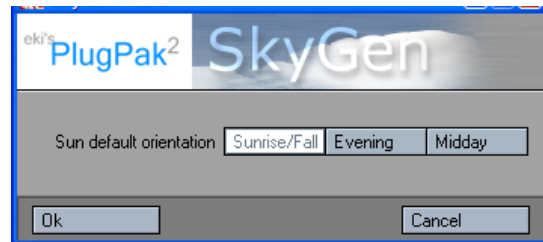
**NOTE:** The spherical projection has the advantage of being more compact and not distorted when compared to the cubic projection.

By pressing the **OK** button **Sky Baker** will start to render all images. A progress bar shows you how much **Sky Baker** has completed. Once **Sky Baker** completes the job, you can choose to generate, position, and map the geometry needed to complete the effect.

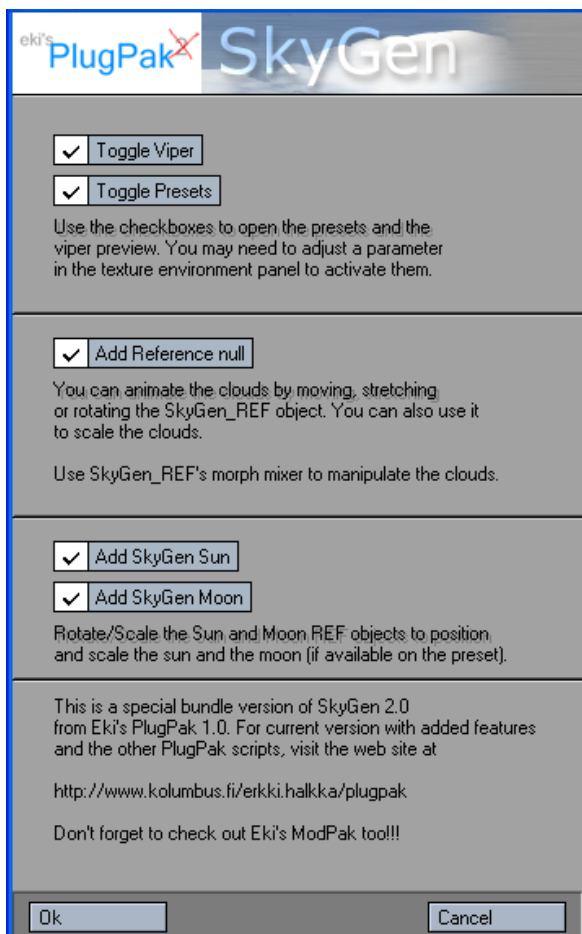


## Skygen

**SkyGen (Utilities> Plugins: Additional> Skygen)** is a set of pre-made procedural texture environments. From realistic to cartoon the sky samples cover daytime, evening and night. Complete with procedural clouds and the ability to animate each parameter. SkyGen scripts will also add all necessary control objects and images.



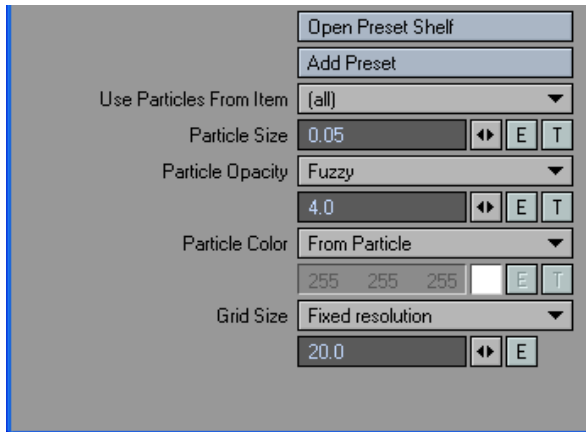
The initial **Skygen Panel** gives you the option to toggle on the **Presets Panel** as well as **VIPER**. Also, you can add a reference null to be used in animating the clouds. Do this by selecting Add Reference Null on the panel.



Use the **Add Skygen Sun** and **Moon** options to add a Sun and Moon to your sky environment. Once you click OK, if these options are selected you will be given options for each for the default orientation.



## Pixie Dust



Pixie Dust is a Volumetric Sprite Particle Renderer that is in most cases faster than HyperVoxels while rendering loads of particles. This does however doesn't replace HyperVoxels as there is no lightmodel within the particle renderer, which means Pixie Dust uses ambient light and does not take information from other lights in the scene..

### Controls:

#### Open Preset Shelf:

Opens the Preset Shelf panel, which allows the use of saved Pixie Dust presets.

#### Add Preset:

Adds the current configuration of Pixie Dust to the Preset Shelf.

#### Use particles from item:

All: Uses all particle systems in the scene

Item: Uses particles within an item plus items parented to it

#### Particle size:

Numerical, envelope and texture control

#### Particle Opacity:

Fuzzy: Controls the opacity with a nonlinear gradient from the center to the edge with Numerical, envelope and texture control.

Custom: Controls the opacity with Numerical, envelope and texture control

#### Particle Color:

From Particle: Get color from the particles themselves(if available,

Custom: Control the color with Numerical, envelope and texture control

### Gradients:

#### Standard Gradients:

Distance to Camera

Distance to Object

Particle Age

Relative Age

Time

Previous Layer

#### Pixie Dust Specific Gradients:

Particle Birth Time: The time at which the particle was created

Relative Particle Birth Time: The particle birth time as a fraction of the current time.

Ray to Particle Distance: How far the ray passed from the particle center.

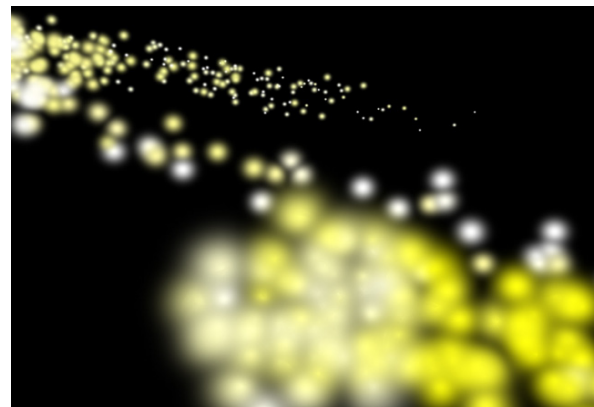
Relative Ray to Particle Distance: Same as above but normalized so that 0 is through the center and 1 is just touching the edge of the particle.

Grid Size: Speeds up rendering by putting the particles on a grid, and then searching the grid for particles that intersect a given ray. There is a per case tradeoff depending on size, amount of particles, density and opacity, having a fine grid (lots of grid squares) means fewer per-particle tests need to be done, but more grid squares have to be tested.

Fixed Spacing: Set each grid square to the given size with Numerical or Envelope.

Fixed Resolution: Set the grid resolution directly(n by n grid squares) with Numerical or Envelope.

Spacing per Particle Size: Sets the size of each grid square as the given multiple of the particle size with Numerical or Envelope.







## Chapter 23: Customizing Layout



## Customizing Layout

- Edit Drop Down Menu
- View Tab
- Customize Color

## Edit Drop Down Menu

### Undo

(default keyboard shortcut **Ctrl Z**)

The undo function (**Edit> Undo**) will undo motion changes (**Move**, **Rotate**, **Size**, or **Stretch** tools).



**HINT:** If you are not using the Auto Key feature, you can quickly reset all aspects of a frame (i.e., position, rotation, etc.) if you haven't created the key yet. Just press the Right cursor key and then the Left cursor key. This advances the current frame and then goes back to the original frame. The frame will return to the last keyframed state, or if the frame is not a keyframe, to its in-between state.

The number of undos is determined by the **Undo Levels** setting found in the **General Options Panel**.



**Redo** (default keyboard shortcut **Z**)

The Redo function (**Edit> Redo**) will redo motion changes (**Move**, **Rotate**, **Size**, or **Stretch** tools) that were previously undone.

The combination of **Undo** and **Redo** is a powerful one, allowing you to step backward and forward through the evolution of a motion.



**NOTE:** The Undo/Redo functions can also be found in the lower left corner of Layout's interface.



## Edit Menu Layout

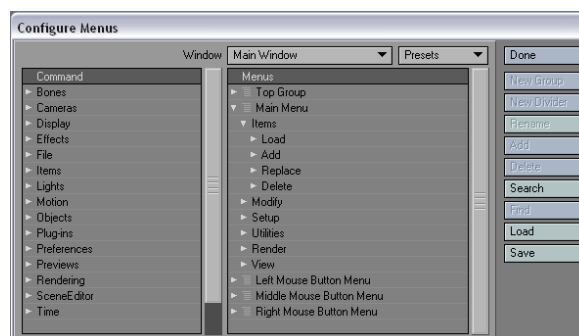
(default keyboard shortcut **Ctrl F10**)



**WARNING:** We strongly suggest that you keep the default menu organisation intact. Otherwise, technical support and using the documentation may become difficult. A better solution is to create a new menu tab and/or groups and place your frequently used tools in them.

LightWave menus are customisable. You can add, remove, group, and reorganize commands.

Choose **Edit> Edit Menu Layout** to open the Configure Menus Panel.



In the (left) Command window, a list of available commands will appear. These will be grouped by type.

In the (right) Menu window, several main sections have sub-items, which are indented to show the hierarchical relationships. **Top Group** contains the items that are always visible no matter which tab is selected. **Main Menu** items are the main tabs and related buttons for the main interface toolbar. **Left**, **Middle**, and **Right Mouse Button Menu** (when available) are the menus that appear when the **Shift + Ctrl** keys are held down along with the corresponding mouse button. There may be other menu sections defined.

In the Menu window, dots indicate commands while arrows indicate groups of commands/sub-groups beneath it. If the arrow points to the right, the group's sub-menu items are collapsed and not visible. To reveal the sub-menu items for a collapsed group, simply click on its right-facing arrow. To collapse an open group, click on its downward-facing arrow.

### Finding Assignments and Commands

If you have a command selected in the left window and click **Find**, the command, if any, on the right will become selected. Selecting a command in the right window and clicking **Find** will select the matching command in the left window.

Clicking **Search** allows you to type in a search phrase to find a command. Search is case-sensitive. This will find the first instance when first run, and can be used iteratively to find additional instances by just hitting the button again, as it retains the string until you type in a new one.

### Maintaining Menu Layouts

The **Load** and **Save** buttons let you retrieve and store *menu layouts* that you develop.

Any available preset menu layouts can be chosen from the **Presets** pop-up menu. The **Window** pop-up menu lets you select different menus to edit (e.g., **Graph Editor**), if available.





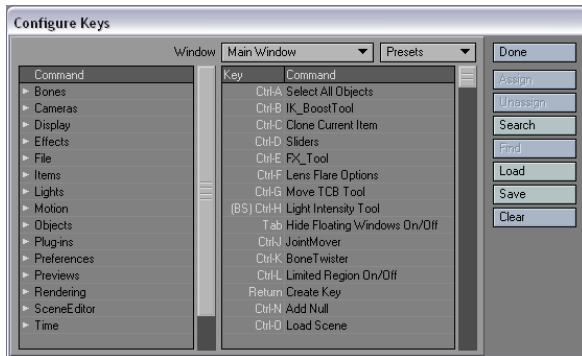
## Edit Keyboard Shortcuts

(default keyboard shortcut **Ctrl** **F9**)



**WARNING:** We strongly suggest that you keep the default keyboard mapping assignments and make new assignments only to unmapped keys. (Most of the function keys are open.) Otherwise, technical support and using the documentation may become difficult.

Like the menus, keyboard shortcuts can be configured to suit your own needs. To display the **Configure Keys Panel**, choose **Edit > Edit Keyboard Shortcuts**.



The window on the left contains a complete list of all assignable commands grouped by type. The right window shows a complete list of all keystrokes and assigned commands, if applicable.

### To assign a command to key:

Select a command in the left window.

Select the target key in the right window. (You can hit the desired key or keystroke combo to quickly select the key.)

Click the **Assign** button. This will overwrite any existing assignment. Alternatively, you may drag the command to the right window.

### To unassign a command to key:

Select the target key in the right window.

Click the **Unassign** button.

### Finding Assignments and Commands

If you have a command selected in the left window and click **Find**, the command, if any, on the right will become selected. Selecting a command in the right window and clicking **Find** will select the matching command in the left window.

Clicking **Search** allows you to type in a search phrase to find a command. Search is case-sensitive. This will find the first instance when first run, and can be used iteratively to find additional instances by just hitting the button again, as it retains the string until you type in a new one.



**NOTE:** Keyboard shortcuts can be assigned to only one command. Assigning a command that is already assigned to a different key will assign it to the new key, but remove it from the old one.

## Maintaining Key Mapping Sets

The **Load** and **Save** buttons let you retrieve and store *key mapping sets* that you develop. The **Clear** button clears out all assignments, so use with caution.

Any available preset key mapping sets can be chosen from the **Presets** pop-up menu. The **Window** pop-up menu lets you select different menus to edit (e.g., **Graph Editor**), if available.



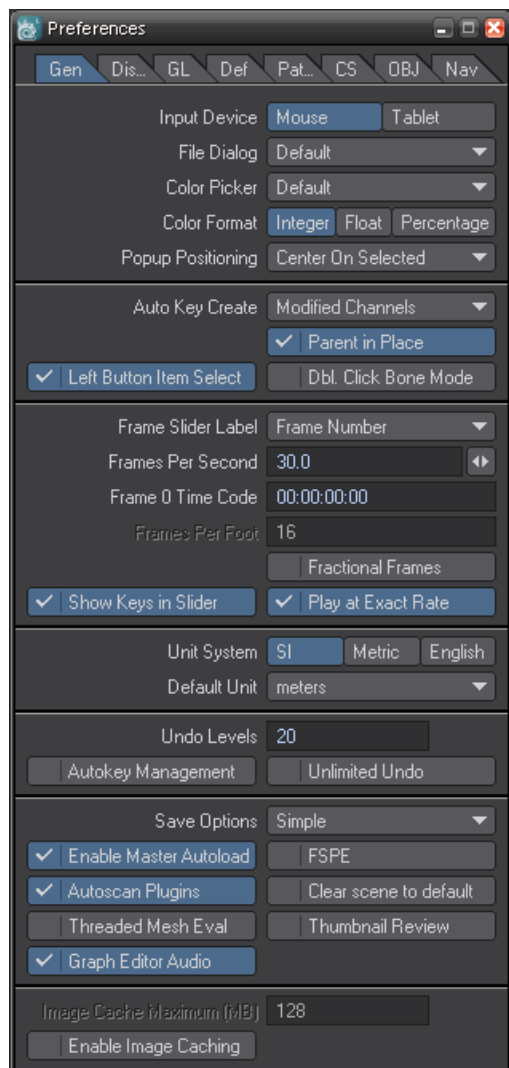
## Window Config Loader

This command is a quick way to launch a window configuration that was generated by the **Window Configure** tool.

## Layout General Options

(default keyboard shortcut **O**)

Choose **Edit > General Options** to bring up the General Options Tab on the Preferences Panel.



## Clear Scene to Default



When this option is enabled, if a default scene exists in the current contentpath, it will be loaded into Layout whenever the user successfully performs a "Clear Scene".



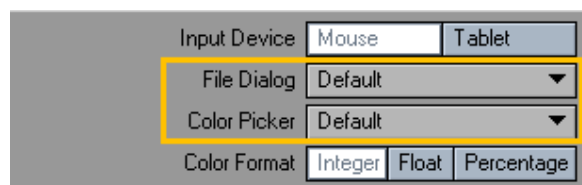
Note: Layout now recognizes a "default.lws" scene. If you save a scene named "default.lws" in the root of your Content Directory, each time you open Layout it will open this scene.

## Input Device

The **Input Device** pop-up menu lets you select either a **Mouse** or **Tablet** as the primary input device. When you choose the input device, the default system drivers for the appropriate device are used.

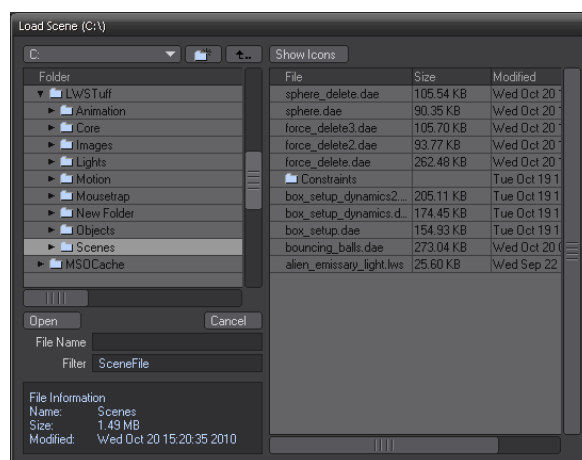
## File Dialog and Color Picker Selection

The **File Dialog** and **Color Picker** pop-up menus let you use custom LightWave dialogs for file loading/saving and picking colors. Selecting **Default** will use your standard system dialogs.



## The Visual Browser

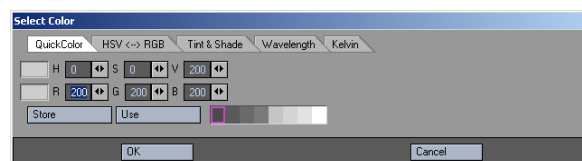
You can make the LightWave visual browser the default file dialog for Layout by choosing **VB File Requester** on the **File Dialog** pop-up menu. This appears on the **Interface Tab** of Modeler's **Display Options Panel** (**Edit > Display Options**).



## LW\_ColrPikr

The custom Color dialog provides many additional features not available with the standard system dialogs.

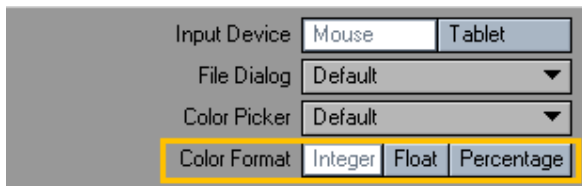
The **LW\_ColrPikr** color picker is a color selector with more features, such as a wavelength and temperature-guided layout of color.





## Color Format

The **Color Format** setting determines the scale used where the color selector appears next to the color swatch in Panels and Dialogs. **Integer** uses values 000 to 255, **Float** uses .00 to 1.00, and **Percentage** uses 0% to 100%.



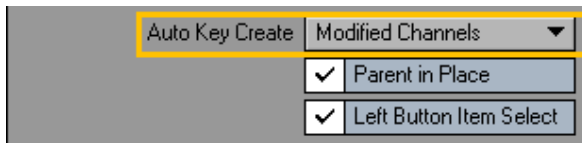
## Popup Positioning

**Center On Selected:** this is the method where the selected item is always centered on the popup button. This can cause the menu to be partly truncated if it falls partly outside of the screen, which means that the user may have to scroll through the menu to get at the desired item.

**Keep All On Screen:** this is the method which ensures that the whole popup menu is always shown on screen. This may require the menu to be moved so that the selected item is no longer centred on the popup button.

## Automatically Creating Keyframes

The **Auto Key** button on the lower part of the main interface turns the overall auto key mode on or off. When active, existing keys are automatically adjusted if an item is modified.



The pop-up menu on the **General Options Tab** has three settings:

**Off** — Off will modify only the settings of an existing keyframe for the current item.

**All Modified Channels** — All Modified Channels creates a keyframe, but for only the modified motion channel group (e.g., XYZ or HPB). So, for example, if you change only an item's heading, a key is created only for the current item's heading, pitch, and bank at the current frame.

**All Motion Channels** — All Motion Channels creates a keyframe for all motion channels.

This feature can be a nice time-saver when compositing still images or initial keyframes, but can be a detriment when changing a feature like a position, as a test to see if you like it.

## Parent in Place

When **Parent in Place** is active, an item will maintain a constant position, size, and orientation when it is parented or unparented.

## Left Mouse Button Item Select

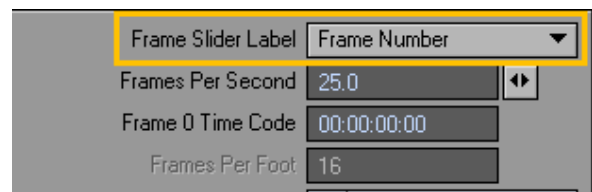
When **Left Mouse Button Item Select** is active, you may click on an item with the LMB to select it in a viewport. (Those of you with three-button mouses can always use the MMB.)

## DbL. Click Bone Mode

With Double Click Bone Mode activated, bone mode is selected when a bone is double-clicked on.

## Frame Slider Label

The **Frame Slider Label** pop-up menu lets you choose a type of measurement on the frame slider, which is the *ruler-like* control on the main interface.



You can select between **Frame Number** (the default), **SMPTE Time Code** (HH:MM:SS:FF, where HH is hours, MM is minutes, SS is seconds and FF is frames), **Film Key Code**, or **Time in Seconds**.



Frame



SMPTE



Film Key Code



Time in Seconds

## Frames Per Second

The **Frames Per Second** rate is specific to the calculation of texture velocities and other internal information that can be accessed by LightWave's plugin architecture. If you had based a moving texture velocity so that it appeared to repeat every 25 frames, but your output was going to film (at the film rate of 24 frames per second), you would want to set this value to 24 so that the repeating nature of the pattern repeated properly. Change this frame rate if you are working on film or another medium that has a frame rate different than video (25 fps).

## Frame 0 Time Code

Enter a starting SMPTE time code for frame 0 in this input field if it is other than 00:00:00:00. This can be important, even if you are not using **SMPTE Time Code** as your **Frame Slider Label**, but are using it in the **Data Overlay** setting on the **Render Options Panel** (**Render > Render Options**).



## Frames Per Foot

When using **Film Key Code**, you can specify the **Frames Per Foot**.

## Fractional Frames

When **Fractional Frames** (formerly Allow Fractional Frames) is turned on, the frame slider can be placed at non-integer frame values, either by adjusting the frame slider, jumping to the previous or next key, or typing into the **Go to Frame** requester.



**NOTE:** Changing this value does not move keyframes. Keyframes are located based on time rather than frames so they're always allowed to lie on fractional frames. If you want to move fractional keyframes to integral frame values, use the **Snap Keys** function in the **DopeTrack**.

## Show Keys in Slider

When Show Keys in Slider is active, a line will appear in the slider where a keyframe exists for the selected item(s).



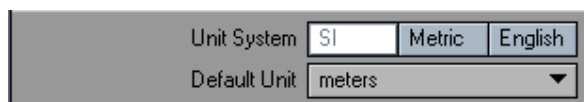
## Play at Exact Rate

Play at Exact Rate causes forward and reverse playing of the scene to ignore the frame step and instead play in real-time, waiting or skipping frames as needed. It has no effect on preview animation playback.

## Measurement Unit System

LightWave supports several units of measurement. You can choose to work in different notations; however, you may input a value using a unit of measurement other than the default, and it will be converted on the fly. For example, you may be using meters, but if you type in "5 ft" LightWave converts it to 1.524 m. (**Metric** is basically the same as **SI** except that it also uses centimeters.)

The **Unit System** setting determines what units of measurement Layout uses and displays.



**SI** — **SI** is the International System of Units. (**SI** is the abbreviation of the French "Le Système International d'Unités.") Unit measurements in Layout will now use a base system of meters. Grid sizes and distances can be measured in megameters, kilometers, meters, millimeters, micrometers, and nanometers.

**Metric** — **Metric** is the same as **SI** with the addition of centimeters.

**English** — The **English** system refers to measurements in miles, feet, and inches.



**NOTE:** We strongly suggest that you use SI or Metric since that measurement type is generally assumed for exercises and tutorials.

## Setting the Default Unit

If a unit is not specified in an input field, the default is used. You can select the default unit using the **Default Unit** pop-up menu.



**WARNING:** We strongly suggest that you use the SI or Metric Unit System and meters as the default unit. Use other settings

only for special situations. Both systems are based on 10, which will make your Modeling and animation life easier. The Default Unit of meters is assumed for exercises.

## Undo Levels

The number of available undos is determined by the **Undo Levels** setting found in the **General Options Panel**. You can set a specific number of levels by typing in a number in the **Undo Levels** entry field, or you can click to activate the **Unlimited Undos** option.



The **Autokey Management** option will delete the keyframe when an undo is performed.

When **Autokey Management** is deactivated, the last action will be undone, but a keyframe will be left in place.

## FSPE

A toggle has been added for FSPE (Full Scene Parameter Evaluation), which was previously a hidden feature. You can find it in the General Options panel in Layout. FSPE is a legacy plugin which was intended to take IK data into motion plugins.



**Warning:** It is recommended to keep FSPE disabled. A continuous loop can be added if a child object is dependent on the motion of the parent object.



**Note:** FSPE and Fixed Near Clip Distance are set back to the default settings upon Clear Scene or New Scene.

## Autoscan Plugins

If checked on, Autoscan Plugins scans the plugins folder and creates a file "Extension Cache" which contains a list of all of your plugins. If you leave Autoscan Plugins off, you can fill the "LWEXT10" configuration file by running the Scan Directory.

Autoscanning is a process to look for plugins in specific locations for the current session. To speed up this process in subsequent sessions, the new extension cache file remembers the previous autoscanning and allows comparisons with the new session to see if these specific plugin locations have added or removed plugins from their folders. Plugins found through this approach are tagged internally as 'non-persistent', since they may not be around the next session. Manual scanning of plugins (the edit plugins panel, either adding a single plugin or scanning a folder) makes the system aware of 'persistent'



plugins, which are plugins you have specifically chosen to use regardless of any non-persistent (autoscanned) plugins. Knowledge of persistent plugins is kept in the usual “Extensions 10” file. Persistent plugins supercede non-persistent plugins if they should ever conflict, thus giving the user more control of what plugins are used. It is recommended you have autoscan enabled and manually add plugins that are not in the pre-determined plugin locations. If you turn off autoscan, you are still able to manually add persistent plugins, but no non-persistent plugins will be available.

## Clear Scene to Default

Checking this open the default scene when a scene is cleared.

## Threaded Mesh Evaluation

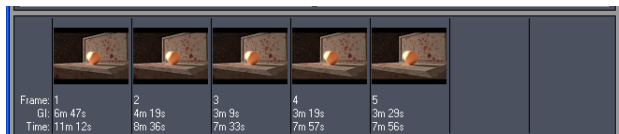
Threaded Mesh Evaluation will use however many cores you have and may help speed up mesh deformations..

The multithreading only happens in Layout for interactive work, not when rendering. Currently the following parts of the mesh evaluation have been multithreaded:

- Morphing
- Bone deformation
- Motion (application of item move/rotate/scale to mesh)

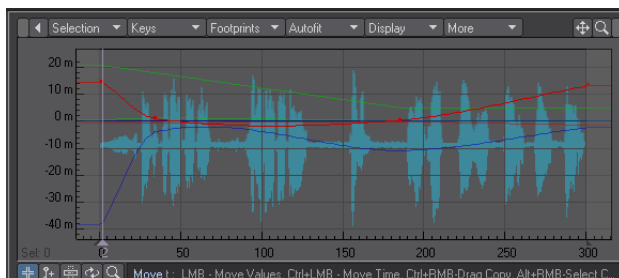
## Thumbnail Review

When this is checked and the Render Scene (F10) function is running, a series of thumbnails will appear in the Render Status Panel, showing the last several renders in the sequence..



## Graph Editor Audio

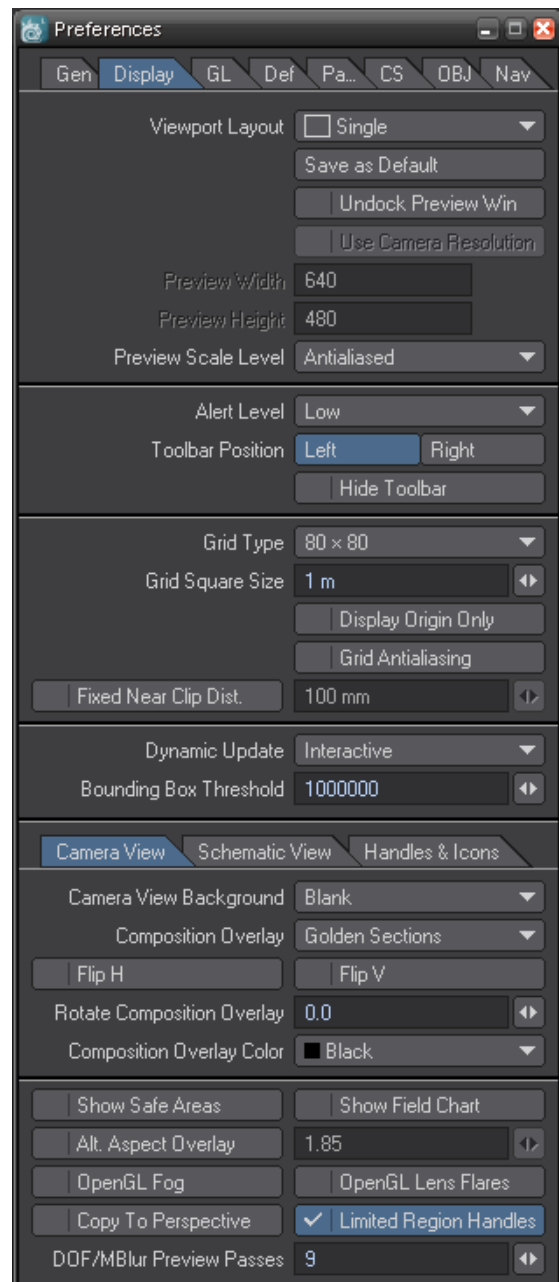
This option enables the graphical representation of audio in the Graph Editor



## Layout Display Options

(default keyboard shortcut **D**)

The **Display Options Tab** on the **Preferences Panel (Edit> Display Options)** controls the settings for display options that are global in nature.

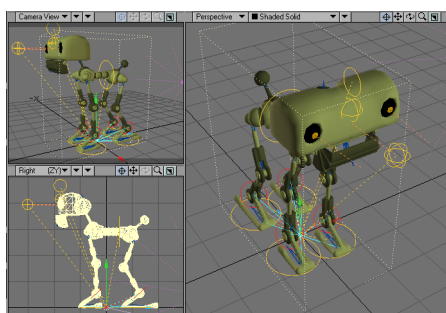
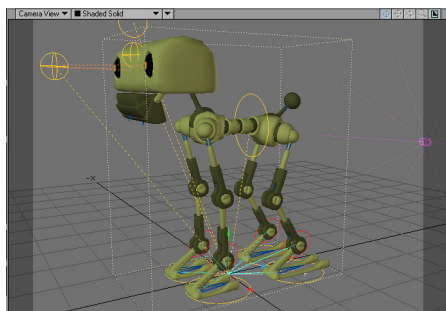
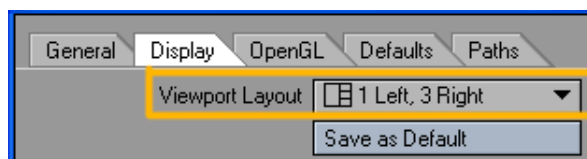


NOTE: The display always uses OpenGL; it is not an option. Also, many of the display options are approximations and your actual rendered results may be different.

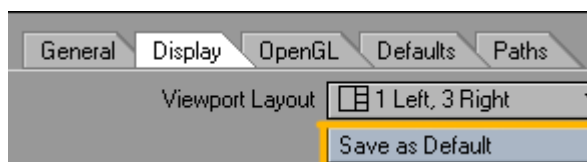


## Viewport Layouts

You can display more than one viewport by changing the **Viewport Layout** pop-up menu. All of the normal viewport options can be set independently and all viewports will update simultaneously.



Click the **Save as Default** button to use the selected **Viewport Layout** for future sessions. Otherwise, it will affect only the current session.

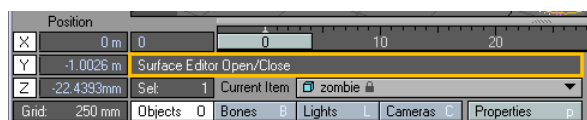


You can customise the relative sizes of multiple viewports by dragging the borders between them.

**Undock Preview Window** will make an undocked preview window when Make Preview is used.

## Alert Level

The **Alert Level** setting controls how error, warning and informational messages are displayed. When set to **Beginner**, these messages are displayed as dialog boxes, which you need to manually dismiss. The **Intermediate** level moves warning and information messages to an information line near the bottom of the interface and **Expert** moves all messages.



Layout Information Line

## Toolbar

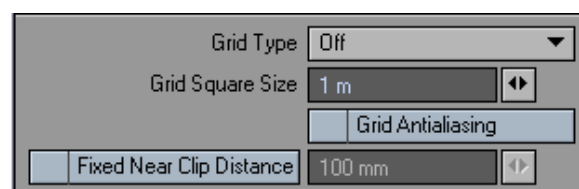
The **Left** or **Right** settings determine on which side of the interface the Toolbar appears. (If you are left-handed, select **Right**.) If you are a keyboard shortcut expert, you can activate the **Hide Toolbar** option, which makes the Toolbar and Tabs invisible.



NOTE: To redisplay the toolbar, press the letter **(O)** key to bring up the **General Options Tab** of the **Preferences Panel**.

## Grid Settings

Use the **Grid Type** pop-up menu to set the size of the overall grid or turn it off. Adjust the **Grid Square Size** to change the size of each square. **Display Origin Only** will display only the origin lines of the Grid. The **Grid Antialiasing** option will smooth out *jaggies* when active.



## Fixed Near Clip Distance

The *near clipping distance* is the point at which surfaces disappear in your OpenGL display. Normally, the distance to the near clipping plane is set automatically to one-tenth of the grid size, multiplied by the camera zoom factor. You may change the base value by activating the **Fixed Near Clip Distance** option and entering a value in the input field.

If near clipping becomes a problem for you, you may find it easier to just adjust the grid size using the **[ and ]** keyboard shortcuts.



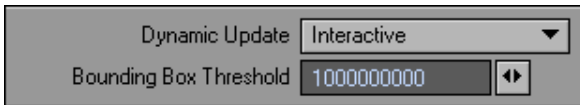
NOTE: The near clipping distance only affects your OpenGL display and has no impact on your rendered images. There is also a far clipping distance that is always set to 10,000 times the near clipping distance. Normally you never see its effects unless you zoom out extremely far or have very distant objects, like stars.





## Dynamic Update

The **Dynamic Update** pop-up menu affects how the Layout window is updated with respect to changes on various panels. **Interactive** will continuously update the Layout window while changes are made. **Delayed** will update the Layout window when you release the mouse button. **Off** will update the window only when you close the panel.

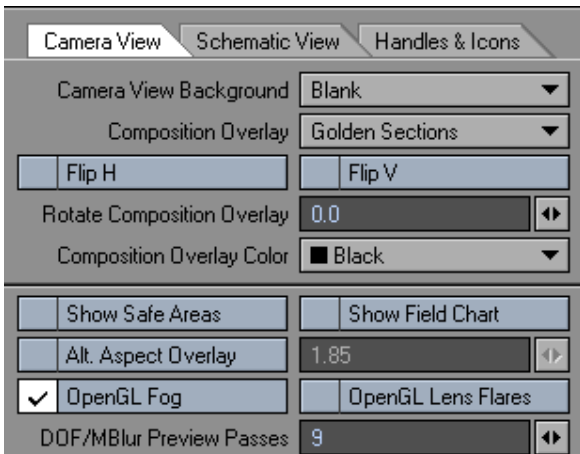


Which setting you use will depend on many factors, like your processor speed, available processing resources, scene complexity, and so on. You may want to try **Interactive** and then back off to a lower setting, if updates become too sluggish.

## Bounding Box Threshold

This value is an absolute limit for the number of points or polygons an object must have for it to automatically change to a bounding box during editing (move, rotate, size, etc.). For example, if the value is set to 5000, any object that has fewer than 5000 points or polygons will remain in solid or wireframe mode even as the object is manipulated. This can result in jerky movements depending on the speed of your computer and the quality of the video card. Any object with more points or polygons than the limit will automatically switch to bounding box mode so that the object can be manipulated smoothly. You should experiment and find a level that your computer can display at a decent rate and then leave that as your default.

## Camera View Tab



## Camera View Background

LightWave lets you choose how you want the Layout interface background to appear when using the Camera view (only).

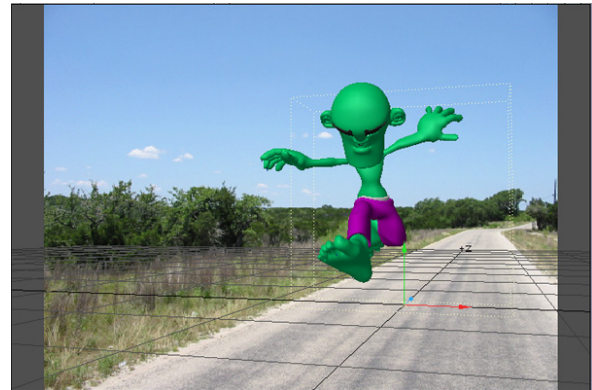
**Blank** is the default choice and will show nothing but the normal blank grey background. **Backdrop Color** uses the color selected as the **Backdrop Color** on the **Backdrop Tab** of the **Effects Panel** (**Scene > Effects: Backdrop**).



NOTE: This setting is independent of how the background will appear in a rendered image. That is, if the **Backdrop Color** is orange and there is a

**Background Image** set, the viewport background will still appear orange if this option is set to **Backdrop Color**.

**Background Image** lets you see the image set as the **Background Image** on the **Compositing Tab** of the **Effects Panel** (**Window > Compositing Options**) in the **Camera** view. You can use it to help you align objects that need to be positioned properly with respect to the background image. It will also show when you make preview animations using the **Camera** view.



LightWave also allows you to use a preview animation as a background, if you use the **Preview** option. This requires you to have a preview animation loaded into memory of course. This can be one that was just generated or one that was loaded.

The preview animation will load starting at the upper-left corner of the Layout window. Make sure to use the same size interface as you did when you generated the preview for easy reference. It will also load beginning at frame 0 no matter what the frame was originally when the animation was created.

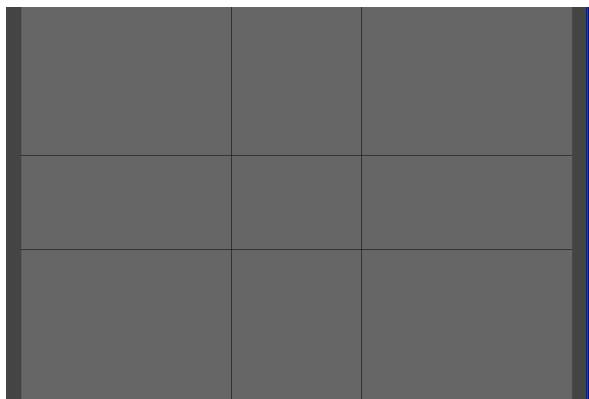
This feature is invaluable for creating moving composite shots and mattes. Using an image sequence composed of live footage as the background image, create a preview animation of that sequence with no objects or grid. Save the preview and load it in later when you need it. The preview animation will update much faster than using the original images as a background.



## Composition Overlay

The Composition Overlay divides up the screen in a number of ways and is used to help with the aesthetics of a composition. Some of the overlays use the Golden Ratio (sometimes called the Golden Cut or Gilded Proportion) which is represented by the Greek  $\phi$ , or phi, which is a constant of about 1.6180339887. In mathematical terms, using a line as an example, the ratio of "a to b" is the same as the ratio of "a+b to a", "a" and "b" being lengths on the line.

**Golden Sections:** The camera view is divided by  $\phi$  in both directions for X and Y directions.



**Golden Triangle:** The camera view is divided by  $\phi$  using the sides of triangles for the ratio.



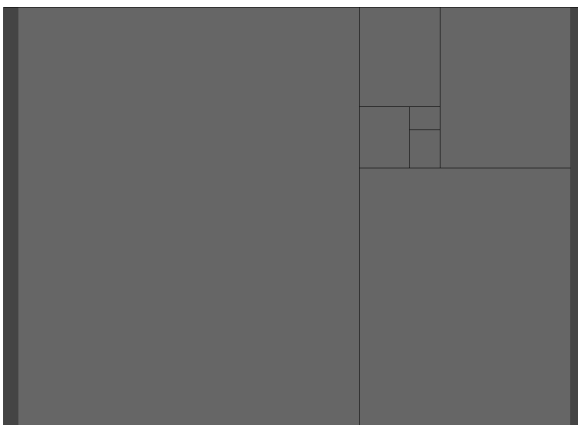
**Harmonious Triangles:** Similar to the rules for Golden Triangle, it draws from opposite sides.



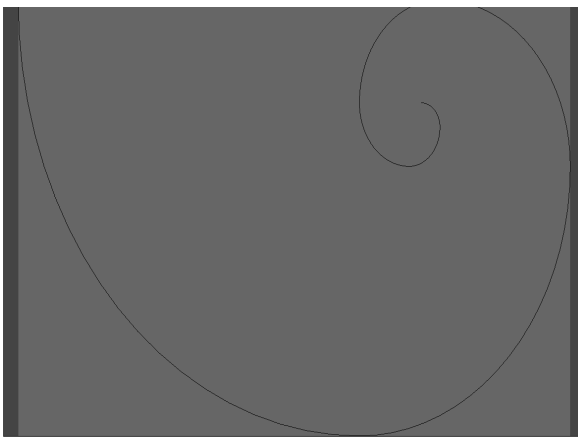
**Rule of Thirds:** Divides the screen into three equal parts in X and Y.



**Spiral Sections:** Using the rule of  $\phi$ , the screen area is divided into a series of rectangles in the form of a spiral pattern.



**Golden Spiral:** Similar to the Spiral Sections, the screen is divided by a curve using proportions of  $\phi$ .





**Flip H**-Rotates the Horizontal orientation 180 degrees.

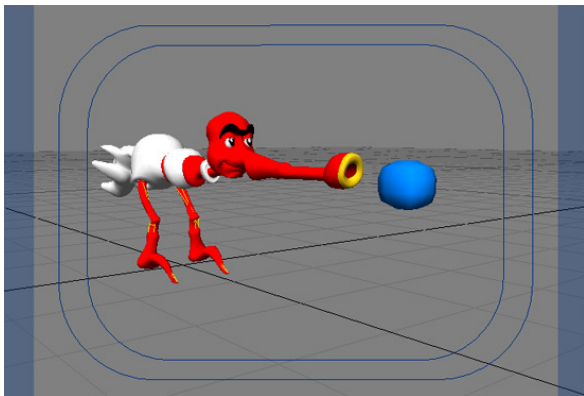
**Flip V**-Rotates the Vertical orientation 180 degrees.

**Rotate Compositional Overlay**- Rotates the overlay by the specified amount.

**Compositional Overlay Color**- Select the desired color for the overlay.

## Show Safe Areas

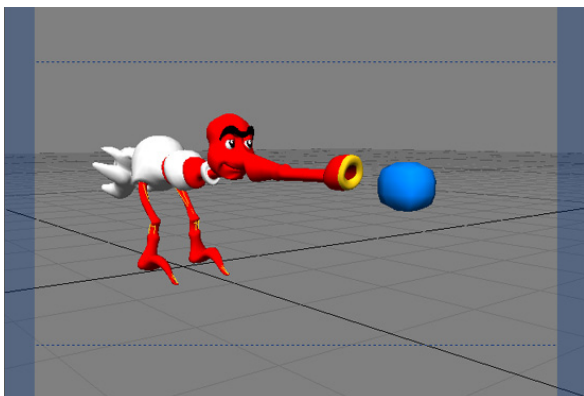
If your animation will be viewed on a television, be aware that viewers will not see the entire image. Select **Show Safe Areas** to toggle outlines in a Camera view that indicate the *safe text* (the inner border) and *safe action* areas (the outer border) for television. To allow for the display differences of different brands and designs of television sets, you should use this guide to ensure that text and action elements in your animations remain on screen when it is important that they be seen.



NOTE: These are reference guidelines and are not absolute positions.

## Alternate Aspect Overlay

The **Alternate Aspect Overlay** setting lets you overlay the borders of an alternate frame aspect ratio, within the normal frame in camera viewports. This *information-only* display mode can be used to simultaneously compose shots for both film and television, for example. The default alternate aspect ratio is 1.85.

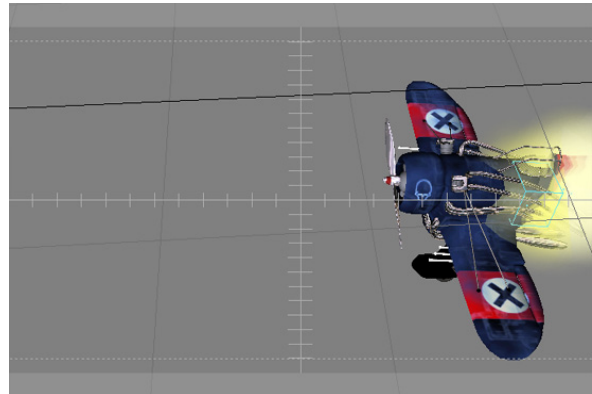


## OpenGL Fog

If you have the **OpenGL Fog** option active, you can see an approximation of your **Fog** settings from the **Camera View** as well.

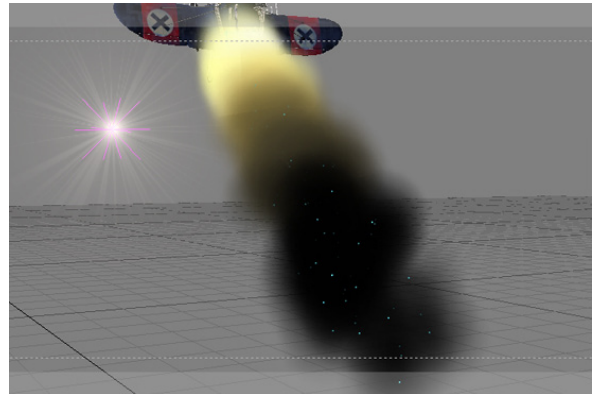
## Show Field Chart

The **Show Field Chart** option turns on a crosshair chart similar to a 12-field chart as used in the film industry. It divides the camera view into quadrants, with each quadrant having 12 segments.



## OpenGL Lens Flares

The **OpenGL Lens Flares** option lets you see lens flares right in a viewport while you use the Camera view. If you have the **Lens Flares Options Panel** open you can interactively make changes to the settings.



## Copy to Perspective

The Copy to Perspective option will use the settings in the Camera for the Perspective view.

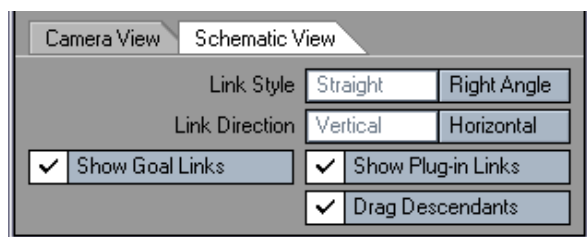
DOF/MBLur Preview Passes

The DOF/MBLur Preview Passes setting sets the number of passes for Depth of Field and Motion Blur previews in the viewport.

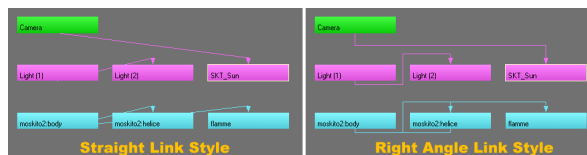


## Schematic View Tab

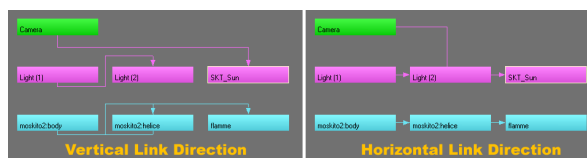
Several display options for the Schematic view appear on the **Schematic View Tab**.



Choose between **Straight** or **Right Angle** from the **Link Style** toggle.



Also, choose between **Vertical** or **Horizontal** from the **Link Direction** toggle.



If the **Show Goal Links** option is active, item links to goals are shown in dotted lines.

If the **Drag Descendants** option is active, moving a parent will move all its children as well.

If the **Show Plug-in Links** option is active, motion dependencies from motion and channel modifier plug-ins are shown as dotted black lines.

## Handles And Icons Tab

Handle, Camera, Light and Bone Icon sizes can now be changed. Bone sizes can also be changed individually in the Bone Properties Panel.

Draw Bones Filled will draw filled in bones.

Handle Wire Smoothing will set the amount of smoothing for handles in the viewport.

## OpenGL Options

### Display Characteristic Settings



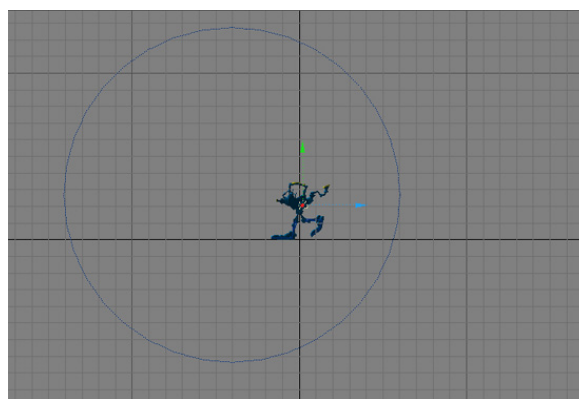
The **Show Motion Paths** option toggles the visibility of the motion path and keyframes for the current active item in Layout.

A motion path looks like a line with small white (+) symbols at each keyframe. Motion paths are subdivided into smaller segments corresponding to the number of frames between keyframes. With **Show Motion Paths** active for an object that is stationary during an animation, the graph will display a single keyframe symbol only, indicating that this is its only keyframe position. It has a stationary motion path.



*Motion Path for the Right Hand Bone*

When you want to see the extent of your fog's **Max Distance** and **Min Distance** on the **Volumetrics Tab** of the **Effects Panel** (**Scene > Effects: Volumetrics**), activate the **Show Fog Circles** option and use one of the orthogonal views. You will see circles representing the two fog radiuses, a result of the minimum and maximum fog values. Just as the backdrop gradient *sphere* is centered about the camera, so is fog.



If you activate fog and the **Show Fog Circles** option, but do not see the indicator, check for the following factors:

Verify that **Fog Type** on the **Volumetrics Tab** of the **Effects Panel** is **not** set to **Off**.

Verify that you are using an orthogonal view.

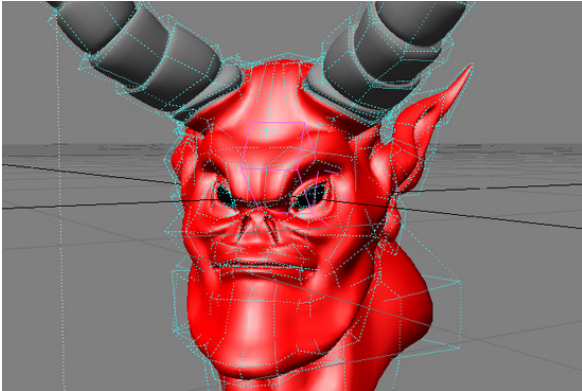
Verify that you should be able to see the fog circles. Are you too close or too far from the camera for the circles to be visible? Use the grid



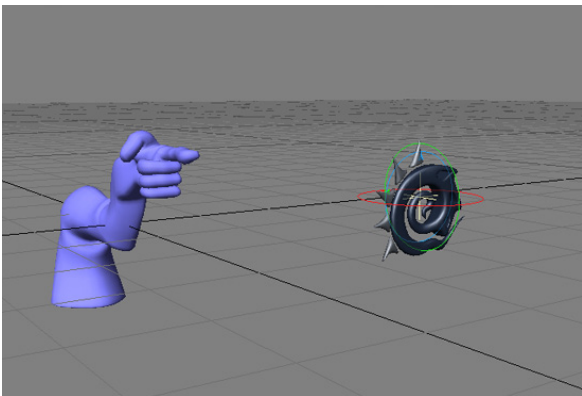


as a guide. Also, is the Overlay color too similar to the object color? Try changing to a different display mode, or try changing the Overlay color (discussed below).

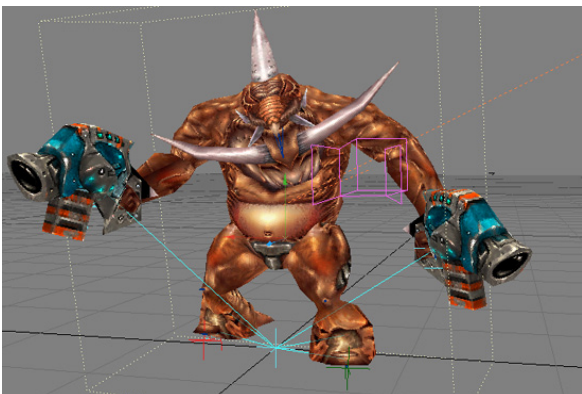
When using SubPatch objects, you may want to see the SubPatch cage. Activate **Show SubPatchCages**, if this is the case.



The **Show Handles** option will display reference handles for the current item when you are moving, rotating, or stretching. These are based on the item's local axes at its pivot point.



The **Show IK Chains** option will display a solid line for the IK chain and a dotted line for the direction of the item reaching for the goal object.



The **Show Target Lines** option turns on/off the line that connects to targetted items.

## Overlay Color

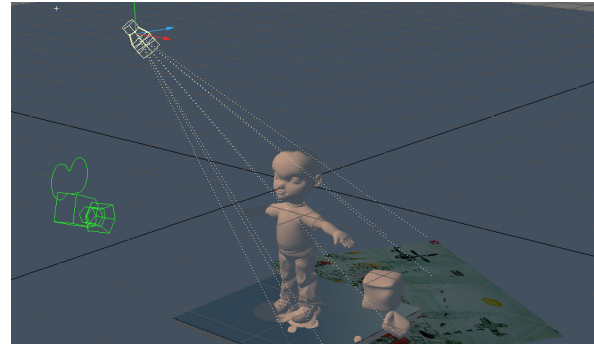
The color of the overlays for the field chart, limited region, fog circles, and so on, can be set to any of the standard wireframe colors using the **Overlay Color** pop-up menu.

## Shaded Display Options

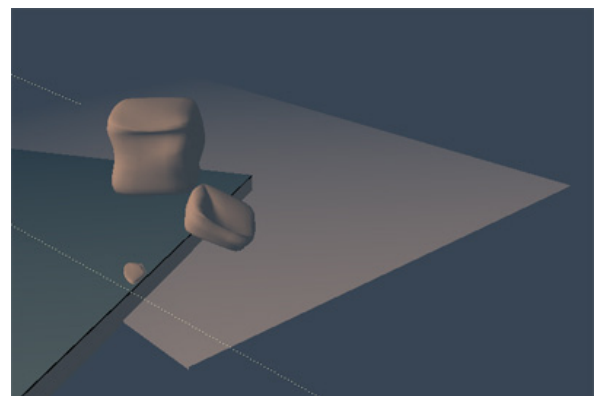
The following options affect the shaded OpenGL display characteristics.



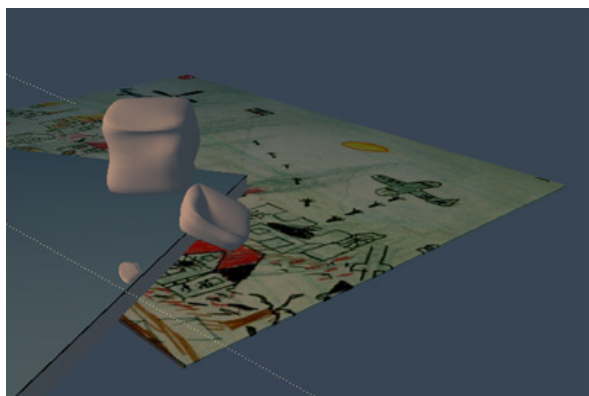
In the **Max OpenGL Lights** field, enter the maximum number, up to eight, of light sources you want used in the display. This lets you see their effect right in the viewport in real-time.



Activate **OpenGL Textures** to show image-mapped textures (not procedural) in the viewports. Use the **Texture Size** pop-up menu to select the pixel resolution (e.g., 128 x 128). Lower settings will update faster and use fewer system resources.



*OpenGL Textures Off*



OpenGL Textures On

Usually the first Color or Diffuse image map layer is shown in the **Texture Mode** viewports.

Enable **Show Texture Editor Layer** to show the **Texture Editor**'s current layer instead, if applicable.

The **OpenGL Pixel Blending** option will activate a smoothing display function.

The **OpenGL Reflections** option lets you see the effects of image-mapped reflections (not ray-traced) in a Layout window



NOTE: The **Reflection Mapping** will be visible only when the surface **Reflection** value is greater than 50 percent.

The **Faster Highlights** option makes the display of (specular) highlights faster, but less accurate. Note that the difference may not always be visible and will vary depending on circumstances.

The **Frame Buffer Object** option applies the z buffer to the viewport object.

The **OpenGL Transparency** option activates a *surface transparency* feature in viewports. This lets you see through transparent surfaces in shaded viewports. (Of course, this is only an approximation of your actual rendered result.) This setting also controls Modeler's display of transparent surfaces, if the Hub is running. Modeler will remember the last used setting, if you aren't using the Hub.

## OpenGL MipMap

Mipmapping is similar to what is used in today's games to avoid graininess of textures in a distance or at a flat angle. Basically lower-res versions of the texture are generated in realtime and blended in. This feature is supported in hardware by most of today's graphics cards. This feature also works if Multitexturing is turned off. Please note however that due to the nature of this filtering method, low-resolution textures may appear a bit blurry.



## OpenGL Multi- Texturing

De- /Activates Multi-texturing – multiple textures layers per polygon in OpenGL.

Depending on the settings that you activate (see below), the following combinations of texture layers are possible:

- A. Two color-layers with one diffuse-layer, one luminosity-layer and one reflection map (5 textures/polygon).
- B. One color-layer with one transparency-layer and one reflection map (3 textures/polygon).

Currently the Multitexturing is made to work with graphics-cards with at least two texture memory units.

## GLSL HW Shading

GLSL HW Shading (OpenGL Hardware Shading) is now among the selections in the Display Options panel. Support for the OpenGL 2.0 hardware shader technology in newer video cards has been added to Layout to provide very close approximations of render functions in the viewport displays. Light falloff, surface blending, gradients, and many procedurals can now be displayed in the OpenGL viewports in Layout when GLSL HW Shading is turned on.

## Geometry Acceleration

Determines how the graphics card displays OpenGL. Streaming renders the mesh immediately to screen, using the lowest amount of memory at the cost of speed. Buffered(VBO) will attempt to store the geometry in graphics card memory, allowing for the highest speed, at the cost of memory. In cases where the mesh or shading changes with every frame no caching is possible, a fallback to the Streaming method will result, for example with animated meshes and reflection maps. Smooth shaded geometry will benefit the most from the Buffered(VBO) mode. If the mesh is buffered in graphics card memory the performance you will get as much performance as your graphics card can give you.





## Color Channel

De- /Activates the display of textures in the Color Channel if Multi-texturing is on.

## Diffuse Channel

De- /Activates the display of textures in the Diffuse Channel if Multi-texturing is on.

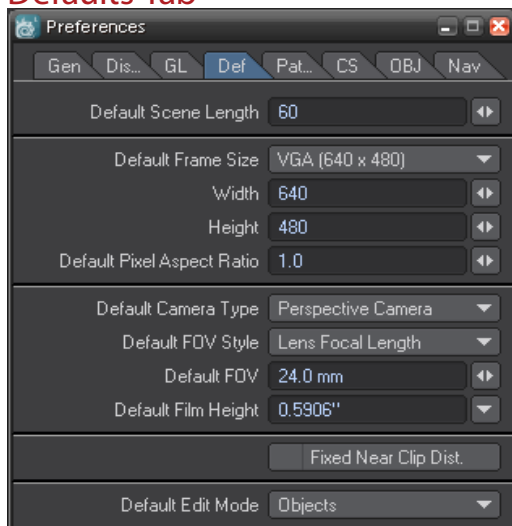
## Transparency Channel

De- /Activates the display of textures in the Transparency Channel if Multi-texturing is on.

## Luminosity Channel

De- /Activates the display of textures in the Luminosity Channel if Multi-texturing is on.

## Defaults Tab



The Defaults Tab allows you to set default settings for a number of settings, including Scene Length, Frame settings, Camera settings, and the maximum number of items allowed in a scene.

**Default Edit Mode** will allow you to specify which of the four edit modes (Objects, Bones, Lights, Cameras) to make active whenever the scene is initialized. This happens, for example, when the application starts, or when a scene is loaded.

## CS (Color Space)Tab

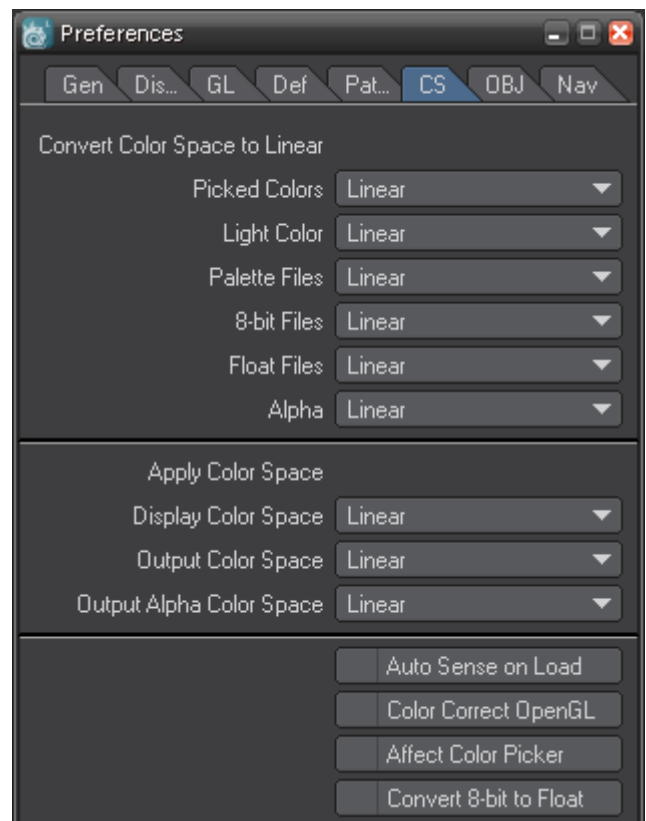
Color space conversion is now performed in 4 places in LightWave.

1. On loading, an image can be converted from its native color space format to linear.
2. When sent to the Image Viewer, an image can be converted from linear to another color space.
3. When saved from the Renderer, an image can be converted from linear to another color space.
4. When picked from the Color Picker, a color can be converted from and then to linear color space.

The color space defaults are set up on the Layout Preferences panel.

When an option is selected in the pop-up it becomes the current item for that selection. Color tables are added to the selections, as if they were built-in.

A color table can be loaded by using Load Table from the pop-up. The color tables are stored in the project directory, in a directory called ColorTables.



In the Image Editor, there is a check box, Treat as Alpha. This means use the alpha channel from the Preferences Default setting, not the alpha channel from the color space in the Image Editor. This is an override on the alpha color space selected in the Image Editor.

One can set the color correction for the Viewer, files with palettes, 8-bit files, floating point files, alpha channel and the color picker from the preference default panel and the Modeler General Options panel.

When an image is loaded, if the file setting for that image is set to default, then an attempt is made to look at the metadata setting for that image. If the metadata settings have the color space the image was saved in, then it is used. For example, jpegs have metadata setting for sRGB and Adobe 1998 linear format.



The built-in color spaces are:

1. Linear, LightWave linear color space.
2. sRGB, Standard RGB color space.
3. rec709, BT.709, HDTV
4. Cineon, Eastman Kodak Co.

The color lookup tables come in 2 formats:

1. LightWave color table format.
2. 3D LUT format.

*LightWave color table format is as follows:*

*Code:*

*EGA*

```
17, -0.5, 1.5
; Convert on load.
-0.5, -0.5, -0.5, -0.5
-0.375, -0.375, -0.375, -0.375
-0.25, -0.25, -0.25, -0.25
-0.125, -0.125, -0.125, -0.125
0.0, 0.0, 0.0, 0.0
0.125, 0.125, 0.125, 0.125
0.25, 0.25, 0.25, 0.25
0.375, 0.375, 0.375, 0.375
0.5, 0.5, 0.5, 0.5
0.625, 0.625, 0.625, 0.625
0.75, 0.75, 0.75, 0.75
0.875, 0.875, 0.875, 0.875
1.0, 1.0, 1.0, 1.0
1.125, 1.125, 1.125, 1.125
1.25, 1.25, 1.25, 1.25
1.375, 1.375, 1.375, 1.375
1.5, 1.5, 1.5, 1.5
; Convert on save.
-0.5, -0.5, -0.5, -0.5
-0.375, -0.375, -0.375, -0.375
-0.25, -0.25, -0.25, -0.25
-0.125, -0.125, -0.125, -0.125
0.0, 0.0, 0.0, 0.0
0.125, 0.125, 0.125, 0.125
0.25, 0.25, 0.25, 0.25
0.375, 0.375, 0.375, 0.375
0.5, 0.5, 0.5, 0.5
0.625, 0.625, 0.625, 0.625
0.75, 0.75, 0.75, 0.75
0.875, 0.875, 0.875, 0.875
1.0, 1.0, 1.0, 1.0
1.125, 1.125, 1.125, 1.125
1.25, 1.25, 1.25, 1.25
1.375, 1.375, 1.375, 1.375
1.5, 1.5, 1.5, 1.5
```

*Where:*

EGA is the name of the color space that appears in the pop-up.  
 17 is the number of entries in the color table.  
 -0.5 is the lower range of the look-up table ( black ).  
 1.5 is the upper range of the look-up table ( white ).  
 -0.5, -0.5, -0.5, -0.5 are the rgba values for the table entry.  
 Blank lines and lines beginning with comments, are skipped over.

3D LUT format tables are read in and a reverse lookup table is made.  
 The name of the color space is the name of the file on the pop-up.

Auto Sense on Load detects the Color Space settings and uses those when a scene is loaded.

Color Correct OpenGL will color correct the OpenGL viewports.

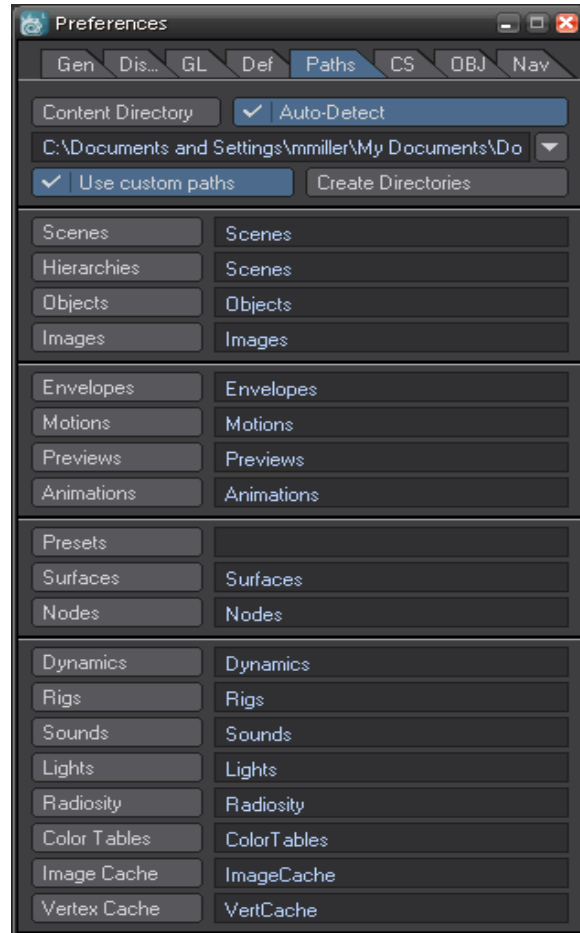
Convert 8-Bit to Float: If the image is 8-bit, this option will convert it to float. This option can be used if you are seeing banding in your images.

## Paths Tab

### Set Content Directory

(default keyboard shortcut **Alt F12**)

This is a shortcut command used to quickly change Content Directories.





## OBJ Tab

The OBJ tab contains options for importing and exporting OBJ file objects.

**OBJ One Layer** imports the object as a single layer.

**OBJ One VMap** imports the object with a single vertex map.

**OBJ Pivot at Center** creates the pivot of the object at the center of the object.

**OBJ Write Normals** writes the normals associated with the object when saved.

**OBJ Merge Points** merges points sharing the same space.

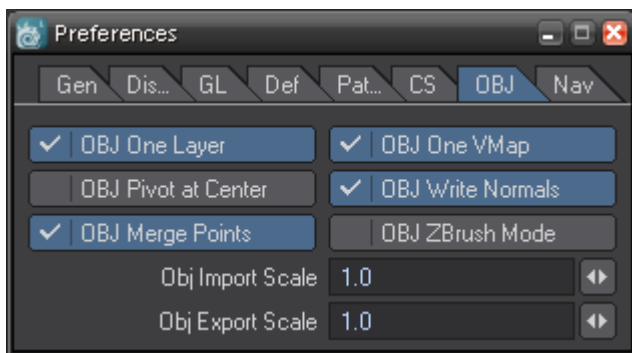
**OBJ ZBrush Mode** exports the object to work with ZBrush.

**OBJ Import Scale** sets the scale of an object when it is imported.

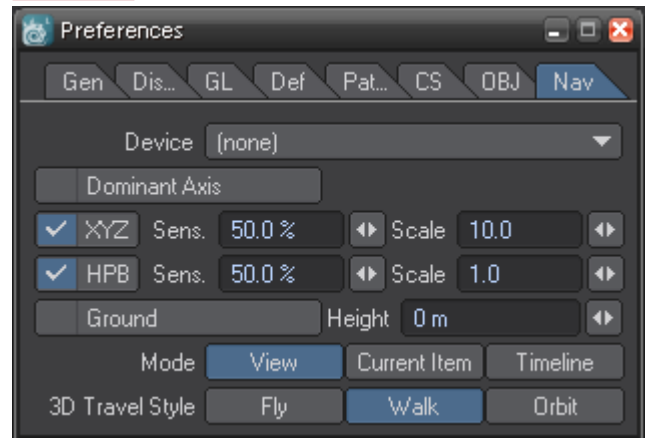
**OBJ Export Scale** sets the scale of the object when it is exported.

OBJ ZBrush Mode

Exports the selected object as an OBJ with multiple UV maps, if available. This option is found in the General Options panel, OBJ tab.

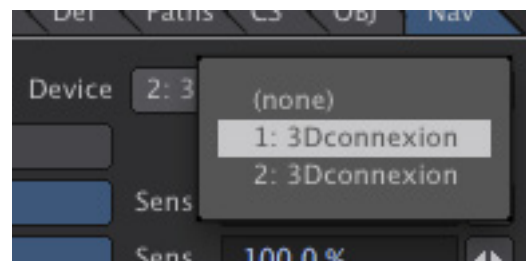


## Nav Tab



The "Nav" tab is short for Navigation, which refers to using a 3D mouse device to manipulate the user interface in ways different than with a standard 2D mouse. The current implementation supports 3Dconnexion 3D mice products such as SpacePilot, SpaceNavigator, SpaceTraveler. Layout currently supports navigation for 3D view ports and scene time scrubbing. These settings are remembered for use in your next Layout session.

**Device:** choose which 3D mouse device to use for navigation features. Currently, only 3Dconnexion products will be shown. When multiple devices are available, they are each given an identifiable number/name. When a device is chosen and subsequently removed, another available device will be automatically chosen. If no device is chosen and a new device is added, an available device will be automatically chosen. If no device is chosen, and a device is removed, the choice will remain "(none)". Care should be taken when using a 3D mouse for both navigation and Virtual Studio at the same time to avoid unwanted behaviors of the same device affecting both feature sets.



**Dominant Axis:** The 3D mouse is a six degree of freedom controller. When the dominant axis feature is enabled, only the axis having the most affect will be applied. This helps avoid jittery movements when only a single direction of control was desired.

**Translation:** When enabled, the three translation degrees of freedom, X Y Z axes, will be used. Also, a sensitivity adjustment is available, with 100% being normal.

**Rotation:** When enabled, the three rotation degrees of freedom will be used. Also, a sensitivity adjustment is available, with 100% being normal.

Sensitivity defines how sensitive the subtle movements of the 3D mouse puck are. A value of 100% means that the movements close to center are just as sensitive as movements far from the center. Smaller values make movements close to center less effective. This allows users of various skill levels to make subtle adjustments without losing the ability to make extreme adjustments.

Scale defines what the extreme movement will be while moving the puck to its extreme in a particular direction. For translation, this refers to the number of grid units per second. For rotation, this refers to the number of half spins (180deg) per second.



**Ground:** When enabled, the viewer of a view port is prevented from going below the specified ground height. This can help in architectural walkthroughs to keep the viewer from entering the abyss. This is only applicable in a non-time context.

**Context:** The context determines what the 3D mouse will be manipulating. The available choices are: View, Current Item, and Time. The context will cycle through these choices whenever button 0 from the 3D mouse is pressed.

**View:** This context represents the viewer in a 3D view port. This includes viewers based on the current camera or current light scene items. Since multiple 3D view ports are allowed, the view being manipulated is the last known one being hovered by the 2D mouse pointer.

**Current Item:** This context represents the currently selected scene item, which can be an object, bone, light, or camera. The translational movement is relative to the viewpoint of the 3D view port being hovered by the 2D mouse pointer.

**Time:** This context represents the scene time-line. Rotating the cap about the its y-axis will progress through frames in the scene. Quickly banking the cap about its z-axis will trigger forward or backward tracking through scene time.

**3D Travel Style:** This determines how to apply viewer manipulations when in the “view” context. The style will cycle through these choices whenever button 1 from the 3D mouse is pressed.

**Fly:** This is similar to flying (without the fear of falling when not moving). No limitation to any of six degrees of freedom is imposed. Forward is relative to the point of view of the view port viewer.

**Walk:** This is similar to walking as a person, where the head can only tilt left/right/up/down only so far. This helps avoid strange conditions while performing scene walk-throughs.

**Orbit:** This is similar to revolving around a point of interest in the view port.

#### Caveats:

- The precise name of each 3D mouse device is not extracted, so they all appear as “3Dconnexion” instead of “Space Pilot”, “Space Pilot Pro”, “Space Navigator”, etc.
- The buttons of each device are not of much use right beyond button 0 and 1, which happen to be the ‘menu’ and ‘fit’ buttons of the newest SpacePilotPro device, but which are different for each device. Further development efforts will make better use of the device buttons.
- Devices only report up to 8 buttons on the Mac, while up to 32 on Windows. This is a 3Dconnexion framework limitation; it may be possible to circumvent by bypassing the provided framework and working with the USB device in a more generic manner.
- Time scrubbing may or may not show frame-by-frame view port updates depending on the complexity of the scene and how often the 3D mouse device is operated. The events from the 3D mouse can come in rather often, and, to avoid losing or ignoring any events, they must be processed. The view port refreshing can sometimes be delayed or interrupted before seeing the results.
- The Windows Layout Navigation preferences are not affected by any system control panel settings.
- The Mac Layout Navigation preferences can be affected by system preferences for 3Dconnexion devices, namely the overall sensitivity value.

## Content Directory

LightWave defaults to looking in certain directories under the **Content Directory** when you load scenes, objects, surfaces, images, envelopes, motions, previews, etc. This is usually the directory you installed the LightWave software into. The Content Directory is LightWave’s master directory; LightWave expects to find all of the appropriate subdirectories within this master directory.

The Content Directory allows you to create a truly portable LightWave scene, including all object and image files. It essentially acts as a pseudo root directory. By saving all your object and image files in subdirectories below the Content Directory, your LightWave scene and related files can be moved from drive to drive, from system to system, and even platform to platform, and still load properly.

Portability is important because LightWave scenes are often rendered on multiple machines or shared for education or fun.



**WARNING:** If you use LightWave in a network environment, it is imperative that you use the Content Directory correctly.

#### Relative Links

When you save a scene, LightWave tries to save only a relative link to image and object files. So an object stored on your hard drive as C:\MyProjects\StretchPrincess\Objects\Jo.lwo where the Content Directory was C:\MyProjects\StretchPrincess, would be saved in the scene file as only Objects\StretchPrincess.lwo.

If you use objects or images outside of the Content Directory, those links are *hard-coded* (e.g., F:\Stretch\Princess\Jo.lwo). If you never move the scene and support files to another computer, the scene will load fine, but this isn’t the way you should do it.



**NOTE:** If you load a scene where an object/image file cannot be found, a file dialog will appear, letting you manually locate the file.

#### Object File Links

Like scenes, objects can also have linked files. These are usually **Image Maps** used for surface textures. The Content Directory concept is also relevant here. Using image files below the Content Directory will avoid loading problems.

If you make any changes to object surfaces, you must save the object file — **a step separate from saving a scene.**



## To set the Content Directory:

Open the **General Options Tab** of the **Preferences Panel (Edit > General Options)** and click the **Content Directory** button. A file browser will open and you can navigate to and select the folder you wish to use as your Content Directory.

You can also choose **Set Content Directory** from the **Edit** button, or choose from among the Content Directories that you've used lately by using **Edit > Recent Content Directories**, or by using the keyboard shortcut **Alt + F12**.

You can also change it in Modeler, on the **General Options Panel**. (If the Hub is active, Layout and Modeler will sync any changes to this setting.)

## Ways to Use the Content Directory

Here are a few ways you might utilise the Content Directory feature:

Use a separate directory as the Content Directory for every project. You'll need to create subdirectories for Objects, Images, and Scenes beneath it. All files relevant to the project would be stored here. As you change from project to project, you must also change your Content Directory. (Note that your project could contain a multitude of scenes.)

Create subdirectories called MyProjects in the Objects, Images, and Scenes subdirectories that are created when you installed LightWave. (e.g., C:\Lightwave\ Objects\ MyProjects, C:\Lightwave\ Images\ MyProjects, etc.). Then, for each project, create identically named subdirectories in each of the MyProjects subdirectories and store your files accordingly.

## Production Data Files

Subdirectories other than Images, Objects, and Scenes (e.g., Surfaces, Motions, etc.) are generally important only during the production stage. The information from these files is incorporated in the scene or object files and is not tracked independently. For example, when you apply that cromulent silver surface file to your spaceship's skin, the settings are saved in the object file. The surface is not referred to again, unless you use it again.



**NOTE:** The generic plugin Content Manager, discussed later in this section, can be used to collect a scene's supporting files and ensure correct compliance with your content directory.

## Recent Content Directory

Choose **Edit > Recent Content Directory**, and select a content directory from the submenu.

## Auto-Detect

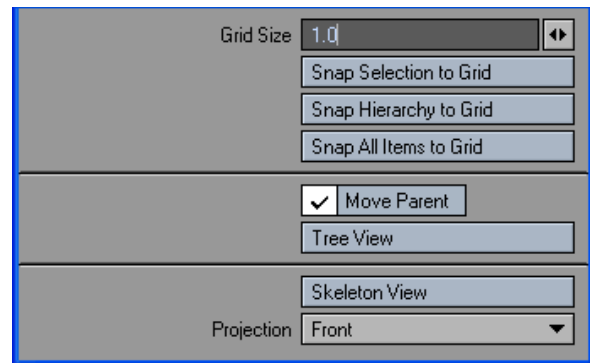
With Auto-Detect enabled, when you open a scene, Layout will attempt to automatically detect the paths for items in a scene.

## Use custom paths

With custom paths enabled, you can specify a separate path for each category.

## Schematic View Tools

This tool helps you organize the scene items when using the **Schematic View**.



You can set the **Grid Size** by typing an entry in the field or by using the sliders to the right of the entry field. **Snap Selection to Grid** will snap the selected items to the specified **Grid Size**. **Snap Hierarchy to Grid** is similar, but only works on the hierarchy from the selected root item. **Snap All Items to Grid** snaps everything.

The **Tree View** button will organize a hierarchy in a "tree" fashion. You must select the parent item before clicking the button. If **Move Parent** is active, the parent will be repositioned along with the children. If inactive, only the children are moved.

The **Skeleton View** button will organize a hierarchy based on the positions of the items at frame 0, from the perspective set on the **Projection** pop-up menu. You must select the root item before clicking the button.



## View Tab

## Viewports

### Fit All

(default keyboard shortcut **A**)

Automatic pan and zoom in Layout's viewports can be achieved with the Fit All command (**View > Fit All**). This will pan and zoom the viewport to fit all the items in your scene. Use **Fit Selected** to pan and zoom the viewport to fit the current item.

### Fit Selected

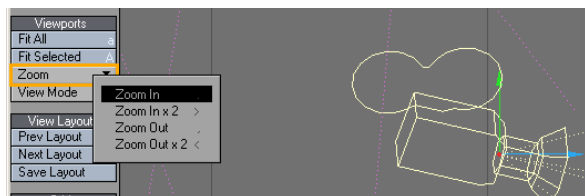
(default keyboard shortcut **Shift A**)

Automatic pan and zoom in Layout's viewports can be achieved with the Fit Selected command (**View > Fit Selected**). Fit Selected will pan and zoom the viewport to fit the current item. To pan and zoom the viewport to fit all the items in your scene use **Fit All**.

## Zoom In and Zoom Out in Layout

Layout provides many ways to zoom in and out of your viewports. First, you can adjust viewport magnification by pressing the period key (.) to zoom in or the comma key (,) to zoom out. (These are shortcuts for **View > Zoom In / Zoom Out**.) Holding the **Shift** key while pressing either key (i.e., the > and < keys on a US keyboard) will double the zoom amount.

These tools are located under the **View Tab** in Layout if you choose not to use the shortcut keys.



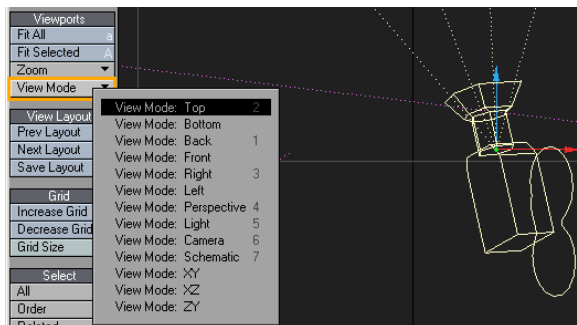
### OpenGL Display Clipping

In some cases, if you zoom in too far, **OpenGL** will *clip out* points and polygons—essentially causing them to disappear. Unfortunately, this is a feature of OpenGL. (Actually, anything that is not in the exact center of the viewport gets clipped. If you center your viewport on an

item, it continues to be visible). If you are still losing your objects from view, you can change the near clipping distance in the Display Options (default keyboard shortcut **D**). For more info, see the Display Options section of Chapter 4 - Edit Drop Down Menu.

### View Mode

You can choose between several different points of view (POV) for each viewport using the **View Mode** drop down menu (**View > View Mode**). Manipulating items in virtual 3D space on a 2D display (i.e., your monitor) can be difficult at times, so you will switch between nearly all of these as you edit your scene.



It is sometimes easier to work in just two dimensions at a time. The options with the axis notations (e.g., **Top (XZ)**) are the "orthogonal" views, which let you move items in only two dimensions (horizontally or vertically), along the XY, XZ, or ZY axes. The Perspective view gives you a three-dimensional look at your scene.

Use the numeric keys as a shortcut to toggle between views.

- 1 — Back
- 2 — Top
- 3 — Right
- 4 — Perspective
- 5 — Light (Current Light)
- 6 — Camera (Current Camera)
- 7 — Schematic



**NOTE:** You can choose between several different points of view (POV) for each viewport using the View Type pop-up menu at each viewport's top-left corner as well.

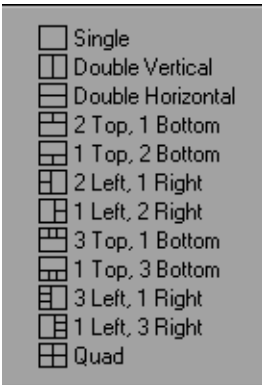




## View Layout

(default keyboard shortcuts **F3** and **F4**)

Layout options control the global arrangement of the viewports, the number of viewports and some other global **Display** settings. Use the **Previous Layout** (**View > Prev Layout**) and **Next Layout** (**View > Next Layout**) buttons to toggle through the various viewport layout presets.



Click the **Save Layout** button (**View > Save Layout**) to use the selected Viewport Layout for future sessions. Otherwise, it will affect only the current session.



NOTE: These options can also be adjusted in the Display Options Panel (**Edit > Display Options**).

## Layout Grid Tools

A grid of squares — cleverly known as the *grid* — is visible in any of the orthogonal views, as well as the **Perspective** view. The grid serves as a visual reference when you move items around, but it will never render in a final image (no matter how much you wish it to). The grid lines are darker every tenth square for visual reference. The Origin is located at the center of the grid.

The **Grid Size** (**View > Grid Size**) determines the incremental change as you drag your mouse. Thus, a smaller size lets you edit your object's position with greater accuracy than using a larger one. If you find that you can't edit an object with the accuracy required, try lowering the Grid Size. However, this will also affect the orthogonal and Perspective view modes.



Use the **Increase Grid** (**View > Increase Grid**) and **Decrease Grid** (**View > Decrease Grid**) buttons to adjust the grid size in small increments.

When you start creating a new scene, your **Grid Size** will automatically adjust itself upwards only, if necessary. This can be problematic when you use objects that differ significantly in relative size, like planets and spaceships. Objects may seem to disappear, when in reality they are just too small or too big to see in the viewport.

Once you manually set the **Grid Size** or save and reload a scene, the automatic sizing adjustment is deactivated. As such, you may want to load the smaller objects first and then manually change the **Grid Square Size to the same value**. Then, load the larger objects.

The size of lights and cameras (that is, how they appear in the viewport) are relative to the size of the grid squares. If you have very large grid squares, you will also have very large lights/camera compared to objects, and vice versa.



NOTE: The size of the lights/cameras do not affect their functionality.



## Layout Selection Tools

Usually, you work on one item at a time, the *current item*, and you need to tell LightWave which item it is. But before you learn how to do that, you need to know that Layout items are grouped into four different types: objects, bones, lights, and cameras. When you work on any item, the **Edit Mode** buttons along the bottom (i.e., **Objects**, **Bones**, **Lights** or **Cameras**) are set to the current item's type.

There are several ways to select an item in Layout:

1. **LMB** or **MMB** click on the item in a viewport;



NOTE: You can select items in a viewport by clicking on any polygon edge rather than just on a pivot point..

2. Click on the item's name in the **Scene Editor Panel (Scene Editor)**; or
3. Manually select the **Edit Mode** and then select the item from the **Current Item** pop-up menu. Note that you cannot select a locked item (a little lock icon appears next to name).
4. Use the Item Picker
5. Use the **Selection** tools in the **View Tab**

## All Drop Down Menu

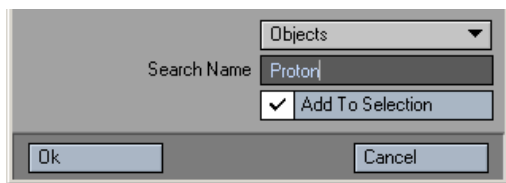
Select All Objects (default keyboard shortcut **Ctrl** **A**) — Selects all objects in the current scene.

Select All Bones of Current Object — Selects all bones in the current object.

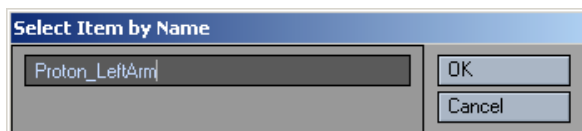
Select All Lights — Selects all Lights in the current scene.

Select All Cameras — Selects all Cameras in the current scene.

Search By Name — Allows you to search for objects and lights by their name and add them to the selection.



Select Item By Name (default keyboard shortcut \*) — Selects an object based on what name is typed in the field. This is handy for very complex scenes.



## Order Drop Down Menu

Select Previous Item (default keyboard shortcut **Cursor Up**) — Selects the next item up in the item list of the current edit mode.

Select Next Item (default keyboard shortcut **Cursor Down**) — Selects the next item down in the item list of the current edit mode.

Select First Item (default keyboard shortcut **Shift** **Cursor Up**) — Selects the first item in the item list of the current edit mode.

Select Last Item (default keyboard shortcut **Shift** **Cursor Down**) — Selects the last item in the item list of the current edit mode.

### Related Drop Down Menu

Parent — Selects the item that the current item is parented to.

Child — Selects the first child of the current item.

Children — Selects all the children of the current item.

Hierarchy — Selects the entire hierarchy of the current item.

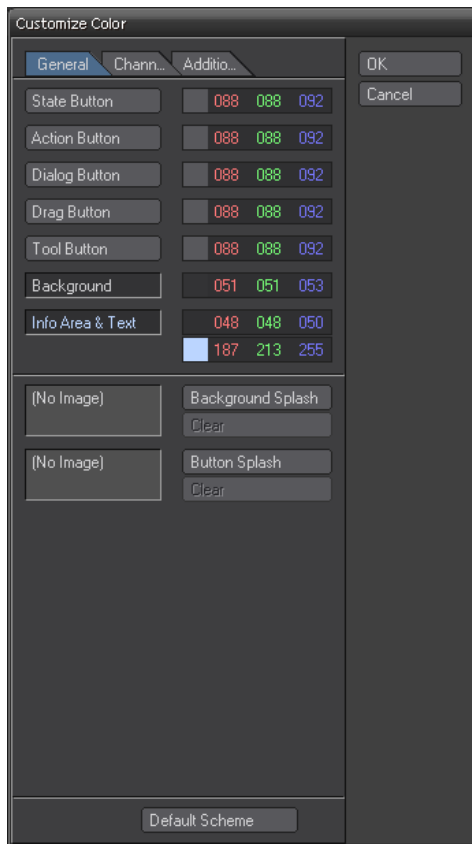
Select Previous Sibling (default keyboard shortcut **Ctrl** **Cursor Up**) — Selects the next item at the same hierarchical level up in the list of the current object.

Select Next Sibling (default keyboard shortcut **Ctrl** **Cursor Down**) — Selects the next item at the same hierarchical level down in the list of the current object.



## Customize Color

Click on **Customize Color** in the Hub Properties Window to open the **Customize Color Panel**.

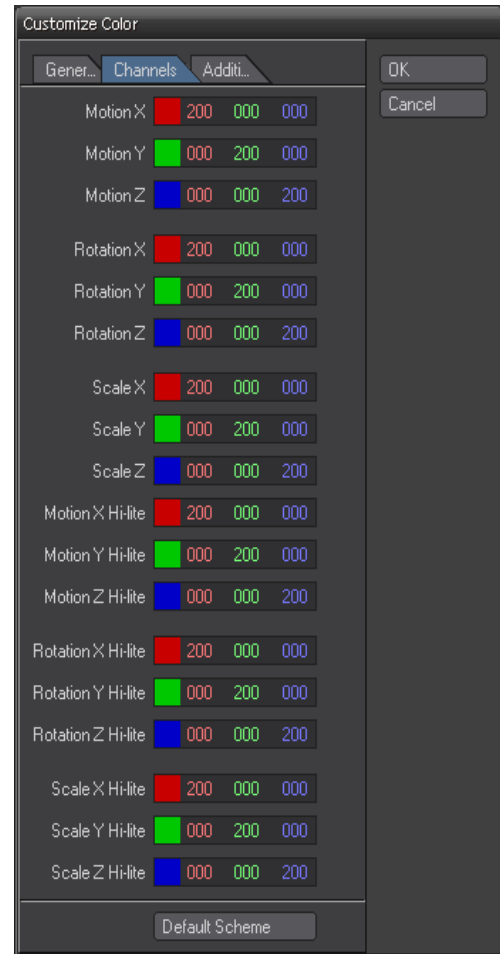


### General Tab

You can customize the color of each button type here. The background color can also be adjusted. If you have an image you would like to use as the background, choose **Background Splash**, and the image will be applied to the background of the interface. You apply the same to all buttons if you choose **Button Splash**.

**Default Scheme** will reset all options to the default LightWave settings.

### Channels Tab



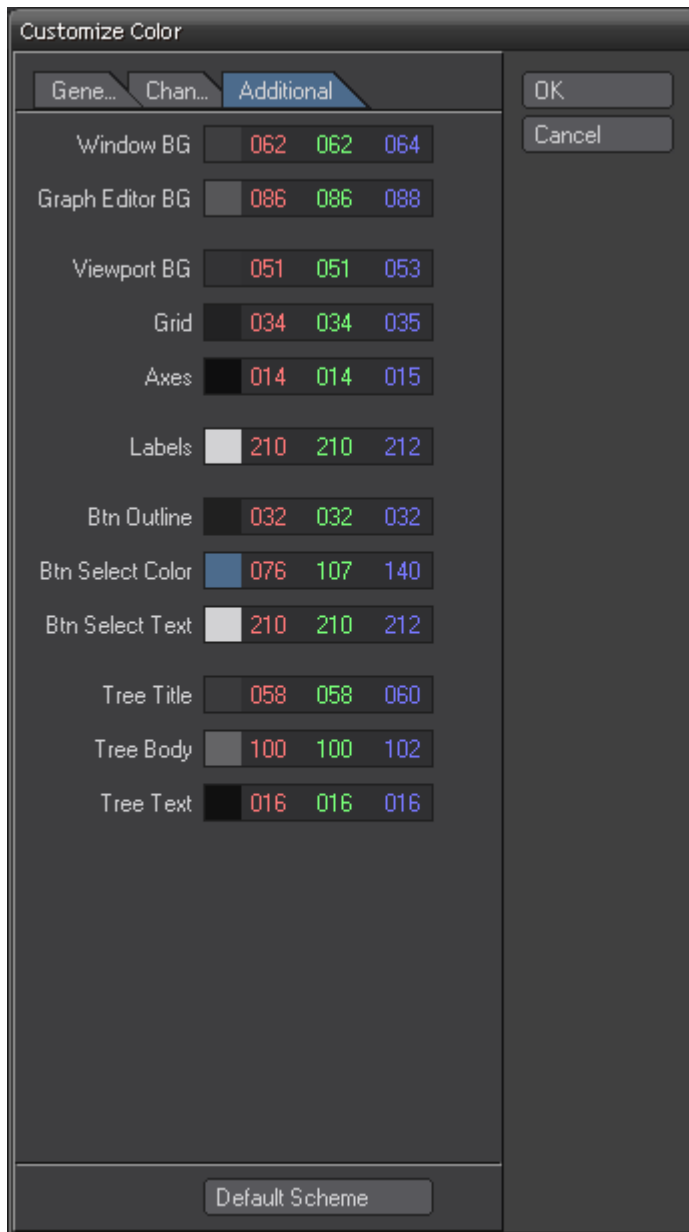
Here you can change the color of the XYZ handles for Motion, Rotation, and Scale for items in the viewports.

**GE Background** will change the color of the Graph Editor background.



## Additional Tab

Here you will find color options for items not covered in the other tabs.





---

## Glossary

---



## Glossary of 3D Terms

**1-Sided** — When a polygon is created, unless otherwise set up, it will have only one side. If you were to look at a playing card, it has a front and a back. A 1-sided polygon only has a front, and therefore only one surface normal.

**2D Map** — Two-dimensional map consisting of either a bitmap or a procedural map. An object using a 2D map needs texture coordinates. See **UV** for further details.

**2-Sided** — Like a playing card. A polygon that has a front, and a back, is 2-sided. A 2-sided polygon will have two surface normals, facing opposite directions.

**3D** — A three-dimensional medium, display, or performance, especially a cinematic or graphic medium in three dimensions.

**3D Accelerator Card** — A graphics card specifically designed for 3D graphics. LightWave uses a system of 3D graphics called OpenGL, and your graphics card must support this.

**3D Map** — Three-dimensional map either built up from multiple layers of bitmaps or, more often, generated in three dimensions with a procedural texture. These are algorithms that can generate 3D maps resembling marble or wood, and when applied to an object, the grains of the marble, and the fibres of the wood, will be correctly mapped to the surface in all three dimensions. If you split a 3D-mapped cube in two halves, the cross section surface will match the neighbouring faces. A 3D map does not require texture coordinates.

**3D Object** — Anything with a position and representation in 3D space. Some objects have a special role, for instance a camera or a light, while others serve as controls for other objects, for instance splines or manipulators. The most common 3D objects are geometric objects, classified according to whether they are polygon meshes, surfaces, curves, implicit objects, or nulls.

**3D Shutter Glasses** — 3D glasses made with electronic liquid crystal shutters. They are powered by the computer they are attached to and use this power to turn on and off the liquid crystal in each of the lenses creating a 3D effect, instead of the usual 2D display a computer monitor can offer.

**3DS** — Aged file format used by Autodesk 3D Studio and discreet 3d Studio max for three-dimensional scenes. It contains geometry, textures, lights and cameras as well as animation data, but polygons cannot contain more than three points.

**Absolute Coordinates** — The location of a point in terms of distances and/or angles from a fixed origin. See **Relative coordinates**.

**Adaptive Supersampling** — Way of antialiasing an object's surface by decreasing the oversampling rate for those pixels that do not require the oversampling. The results of adaptive supersampling are slightly more localised, and computing time is often shorter than other sampling methods.

**Additive Mixing of Colored Light** — There are two sorts of mixing of colors. One is called additive, or sometimes transmissive, and refers to the fact that the more red, green and blue you

add together the nearer to white your final color will be. This is the normal light scheme for LightWave or other graphics packages with output mainly through the medium of a screen. Subtractive mixing indicates that the fewer colors you mix the nearer to white you are and is used for reflective color, such as printed material.

**Additive Opacity** — Type of opacity that adds the background color to the material color of the transparent object.

**Aggregate Object** — An object that is made up of a number of other objects. A normal aggregate object will be made up of primitives. A more complex aggregate object may be made up of primitives, other aggregate objects, or both.

**Algorithm** — A problem-solving method that involves using a multi-step process.

**Aliasing** — When referring to pictures, aliasing is the effect that occurs when a line looks jagged instead of smooth because of a contrast in colors. Usually, you can tell when this happens because the line between the colors looks very jagged, as if it were a flight of stairs, in fact it is often referred to as a "stairstepping" effect. For contrast, see **Antialiasing**.

**Alpha Channel** — One of the four channels (or components) of information that make up every pixel in an image. There are three channels for red, green, and blue (RGB) and one alpha channel. The alpha channel is really a mask — it specifies the transparency of each pixel, which allows portions of the foreground image to reveal the background when two images are overlaid.

**Alpha/Matte Image** — Generally refers to an image where the brightness of each pixel is used to cut or partially dissolve out another image. These are generally greyscale or black-and-white images, but the brightness values can also be extracted from a color image.

**Ambient Component** — Part of the reflection-illumination model. A common surface shader parameter that adds consistency to the color of an object's surface to simulate an ambient light that reaches all points in a scene. An ambient value is determined for individual surfaces. Scene ambience is multiplied with an object's ambient color. If the scene ambience is set to black, nothing alters the ambient color of an object except, of course, a light. The careful balance of ambient and direct light sources is the key to convincing lighting. Global illumination is an alternative to ambient light that is more accurate but takes longer to render.

**Ambient Map** — Allows manipulation of the ambient component of an object's reflection-illumination model. Usually the ambient component is given a value near that of the diffuse component.

**Ambient Light** — All-directional light illuminating every object uniformly from all sides.

**Anaglyph** — Moving or still pictures in contrasting colors that appear three-dimensional when superimposed. Anaglyph works well for printed matter or computer display, but color problems inherent in television displays (both PAL and NTSC) result in poor 3D broadcasts.

**Anamorphic Distort** — An option referring to the width of a lens flare. When selected, the larger the distort factor, the wider the lens flare will become.





**Angle of Incidence** — The relative angle between a lit surface and the light source. The more the surface is turned away from the light source, the less light it receives and the darker it becomes. When the angle of incidence is 90 degrees, the light shines directly on the surface and it is illuminated with maximum intensity.

**Animate, Animation** — The movement of elements through time and space. Also, the process of creating and recording images that change over time. Everything in a scene is represented by numeric values and, as such, animation is also the process of changing these values — position, color, or any other property — over time. A method of creating the illusion of life or movement in inanimate objects or drawings. Through animation the artist's drawing comes to life. The most well known works are cartoon comedies, like *Ren & Stimpy* or *The Simpsons*.

**Animatics** — Preliminary animated versions of a final video or film presentation.

**Animation Channel** — Animation Channel refers to the different position, rotation, and scaling settings an item can have in Layout. It can also refer to other envelope elements like light intensity. See also **motion channel**.

**Annotation** — The process of inserting text or a special note, explanation or to provide relevant detail to a surface, a rig or a point in your scene in LightWave.

**Antialiasing** — A method for blending harsh contours and preventing staircasing or stairstepping. It is achieved by taking the surrounding areas into account when assigning a color value to pixels lying on an object's contour.

**Antisymmetry Surface Restraint** — The restraint of a surface tangent to the surface. This implies that the structure is symmetrical about this plane, and the load on the implied symmetrical part is equal to, but in a direction opposite to, the modelled part.

**Aperture** — The opening size of a camera lens. The greater the aperture, the smaller the depth of field and the greater the amount of light entering the lens.

**API** — Abbreviation for application programming interface.

**Arc** — Curved open element with a constant radius around a single center point. Section of a circle.

**Area Light** — A special kind of point or spotlight. The rays emanate from a geometric area instead of a single point (entire surface uniformly emits light). This is useful for creating soft shadows with both an umbra (the full shadow) and a penumbra (the partial shadow).

**Array** — A set of elements put together into a single entity. A pixel array is an ordered set of colored elements used for display purposes. In a 3D program, the array tool is usually used to create ordered copies of an object within 3D space. This tool is so named because it creates arrays of objects (creates an ordered set consisting of multiple copies of the same object).

**Aspect Ratio** — Description of the proportion of an image by comparing its width to its height. 35 mm slides have the aspect ratio of 4:3 (1.33:1). Images become distorted if forced

into a different aspect ratio during enlargement, reduction, or transfers. It should not be confused with the pixel aspect ratio, explained further on.

**Atmosphere** — Volumetric effect that simulates reduced visibility in air over distances.

**Attenuation** — The strength of light diminishes with distance when it travels through air. The further light travels, the dimmer it becomes. In real life, light attenuates by the inverse square of the distance. If attenuation is turned on for a light, only the geometry in its proximity will be lit. Not only is this more realistic for your renderings, it also helps speed up rendering time since only the geometry close enough to be affected by the light needs calculation time. See also **Decay**.

**AVI** — Audio Video Interleave. A popular animation file format that combines video and audio.

**Axis of Motion** — In 3D space, the line that an object follows during movement.

**Axis of Rotation** — In 3D space, the line that an object rotates around.

**Axis, Axes** — Axis refers to the XYZ coordinates used as the basis for positioning elements in LightWave's 3D space. It is somewhat like the concept of left/right, up/down, and near/far.

**B Rep** — See **Boundary Representation**

**Backface Culling/Elimination** — A process included in most 3D graphics pipelines, backface culling eliminates triangles facing away from the camera. This is most efficiently performed by checking the orientation of the triangle normal in relation to the camera. The technique ignores geometry seen from behind so that only the fronts of objects that are facing the camera are rendered. Both faces of an object are rendered by default; that is, the ones whose normals are facing the camera as well as those that are not. You can choose which faces of the object you want to render as part of the rendering options: front, back, or both faces. Backface culling (rendering only the front) can improve performance because less geometry needs to be rendered.

**Background Color** — The color that occupies all space not occupied by text, images, or any other objects. LightWave's default background color is black.

**Ball** — (Sphere) A 3D circle or oval created by user-defined dimensions and settings.

**Bandwidth** — How much information a network can carry. Think of the network as a highway, and each message as a car. The more lanes in the highway, and the higher the speed limit, the more traffic it can carry. So the wider the bandwidth of a network, and the faster its speed, the more information it can carry.

**Banking** — This is when an object following a path rotates about the path when it changes direction.

**Barn Doors** — The flaps on the front of a movie light to assist in limiting the beam of light.



**Baud** — Bits per second. Hence kilobaud or Kbaud, thousands of bits per second. The technical meaning is 'level transitions per second'; this coincides with bps only for two-level modulation with no framing or stop bits.

**Bend** — To deviate from a straight line or position: The lane bends to the right at the bridge. To assume a curved, crooked, or angular form or direction: The saplings bent in the wind.

**Bevel** — A method of eliminating sharp edges from objects by extending an object's faces to round off corners.

**Bezier Curve** — A technique for creating curves that was attributed to and named after a French engineer, Pierre Bézier, who used them for the body design of Renault's cars in the 1970s.

**Bilinear Filtering** — Blurring the pixels in a bitmap when it is zoomed in so that it seems smoother than it really is.

**Bilinear Intensity Calculation** — A high-speed algorithm for generating shaded faces. Used in Gouraud shading and Phong shading.

**Binary Space Partition (BSP)** — Also known as BSP, this is a technique used in real-time rendering for resolving in which order polygons should be drawn. The technique requires that a so-called BSP tree is built before the scene may be rendered. As this build process is very costly in terms of execution speed, BSP trees cannot usually be calculated in real-time and thus essentially only support highly complex yet static 3D worlds.

**Bit** — The building blocks of computer data. Has either the value of 1 or 0 (current or no current). Bits can be grouped together to carry larger values.

**Bitmap** — Two-dimensional monochrome raster image. A bitmap is a black and white image marking boundaries. It is often used for clip maps in LightWave.

**Blade** — A thin plane placed in front of a light to cast a shadow, taking light off of an object. A device to create a shadow.

**Blending** — Mixing of two (or more) textures into one final texture that is displayed in rendering.

**Blitting** — The copying of a virtual frame buffer to the displaying screen.

**Bone Hierarchy** — Bones can be arranged to build a Bone Hierarchy, also called a Skeleton. The hierarchy defines how the movement of one bone affects other bones (up and down the hierarchy). If you then also add Constraints to the bone hierarchy, you have a Rig.

**Bones** — For any object, you can define a skeletal system composed of bones. By moving bones around, you can change the shape of an object.

**Boolean** — A mathematical system developed by English mathematician George Boole that expresses logical relationships between things. The results in a Boolean operation can be either true or false. Boolean is used in 3D to add, subtract, and other operations that involve Boolean calculations.

**Boolean Operations** — **Modeling** technique that uses two overlapping objects to create a new object. There are three kinds of boolean operations: subtraction, union and intersection. By taking the first shape and subtracting/unifying/intersecting it to the second — a new shape is created.

**Boom Shot** — A camera move. Usually describes a shot in which the camera is mounted on a crane. The camera can move in all axes of movement.

**Boundary Representation (B Rep)** — A polygonal mesh representation. A polygonal mesh is, most commonly, a simplification of a shape using facets to describe curvatures. Its surface, or boundary, is built up from several faces that describe the shape. If it is a polyhedron the polygon model can be identical to the shape, whereas an organic shape is represented by a more or less simplified version that mimics the curvature using facets with variable density.

**Bounding Box** — A cubic shape that exactly circumscribes a (more complex) 3D model and is used to optimise three-space calculations like ray tracing. By representing a more complex shape with a box, possible ray intersections can be investigated much more swiftly. Also used to represent complex objects for proxy animation and setup to speed up operations.

**Bounding Volume** — A way of speeding up ray tracing operations involving intersection calculations, by inscribing a complex mesh in a considerably less complex shape like a box or sphere. Often used when rendering must be done in a short amount of time. Instead of having to check the intersection of a more complex mesh, like a space ship or a teapot, the bounding box works as a stand-in, with the same maximum height, width and length as the mesh it substitutes for. Therefore a possible ray intersection can be either ruled out (if the ray doesn't pass through the bounding box, it doesn't pass through the mesh either), or let a more time-consuming algorithm take over working with the complex mesh instead.

**Box** — (Cube) A six-sided 3D object that can be thought of as a 3D square or rectangle. Boxes are created based on user-defined input as to the dimensions and locations desired.

**BSP** — See **Binary Space Partition**.

**B-Spline** — A free-form curve that is defined with parameters in which each separate vertex on the curve has an influence over a portion of the curve. In 3D, B-splines allow a user to control a curved line on two axes at once.

**Bump Array** — The purpose of a bump array is to create an ordered series of bumps in a surface. This tool means exactly what its name implies - an array of bumps. See also **Array**.



- Bump Map** — Creates the illusion of three-dimensionality of a surface (protrusions and cavities) by recalculating the normals of the object, without changing the mesh itself. It is very common in 3D renderings and suitable for creating effects like wrinkles, creases, crumples, cracks, seams etc. The silhouette of a bump mapped object is a give-away since, in these areas, it is obvious that the mesh is left unaffected (if trying to create an orange by using a perfect sphere with an orange peel texture applied to it for bump mapping it will still have an impeccably round silhouette). In LightWave areas in a bump map that are black are unaffected and areas that are white are raised.
- Byte** — 8 bits. Multiples of bytes make up the terms kilobyte (1024 bytes), megabytes (1024 kilobytes) and gigabyte (1024 megabytes).
- CAD** — Computer Aided Drafting (or Design); A system that lets a designer use a computer screen instead of a drafting table to make plans and blueprints. However, CAD packages don't often have superb rendering abilities, so packages like LightWave are turned to, for their abilities.
- CAM** — Computer Aided Manufacturing; the process of using a computer to create a physical product from a computer-created design. CAM is usually used to control robots that perform tasks that would be tedious or dangerous to human workers. See also **Rapid Prototyping**.
- Camera** — An apparatus for taking photographs, generally consisting of a lightproof enclosure having an aperture with a shuttered lens through which the image of an object is focused and recorded on a photosensitive film or plate. Digital cameras use CCDs to focus light and create a digital picture that can be seen and transferred to a computer immediately. In LightWave, the camera is the conduit through which objects and scenes are turned into still images or animations.
- Capping** — Surface closing the upper and/or bottom side of an object such as a cylinder.
- Cartesian Coordinate** — Mathematical representation of Euclidean space. Every point can be described by three coordinates (X, Y, Z) representing the position along the orthogonal X, Y, and Z axes. The point (0, 0, 0) is called the origin, which is the global center of the 3D world.
- Cartesian Space** — A space in which positions are denoted by a three-coordinate system (x, y, and z coordinates) relating to a central origin (0,0,0).
- Catacorner** — Slanted across a polygon on a diagonal line; "set off in a catty-corner direction across the vacant lot". syn: cata-cornered, catercorner, cater-cornered, catty-corner, catty-cornered, kitty-corner, kitty-cornered.
- Cattiwompus** — Weird. Mixed up. Unusual. Distorted.
- Caustics** — Light pattern created by specular reflection or refraction of light, such as the patterns of light on the bottom of a swimming pool, or light through a glass of wine.
- CD** — Compact Disc storage media. Also the chemical symbol for Cadmium and the sticker on the back of a car with the diplomatic corps.
- Center of Projection** — The point in perspective projection where all projectors intersect.
- Center of the World** — Is the absolute center of a 3D space, represented by X, Y, and Z points (0, 0, 0). Also referred to as the Origin.
- CenterPoint** — A point that represents the center of an object. This point is used in some programs for a point of reference for rotation and position. The center point of a polygon is where the line representing the normal comes out from.
- Chamfer** — To cut off the edges of the geometry with a planar cut, creating a more blunt shape, typically at a 45 degree angle. A bevelled edge or corner between two intersecting lines or surfaces.
- Child** — An object whose movements are influenced by another object, called the "parent".
- Chord Length Parametrisation** — See **Non-Uniform Parametrisation**.
- Clean Modeling** — Refers to the practice of removing geometry from a model that is not wanted or needed. Also refers to the use of proper geometry construction techniques, such as creating continuous surfaces, minimising narrow faces, and avoiding small corner angles, that facilitates downstream processes.
- Clipping** — More often than not, much of the graphics drawn for a specific scene does not fit into the viewport of the camera. Accordingly, those which fall outside of the viewport must be clipped so as they are not drawn. Depending on the nature of the application, there are two kinds of clipping: 2D and 3D. The earlier simply compares each pixel against the extents of the rendering viewport, while the latter technique uses the six sides of the view frustum to determine whether a 3D vertex is inside the viewport or not.
- Clone** — This tool creates copies of an object based on user-defined parameters for offset, motions, morphing, shadows, etc. This tool can be used to make ordered sets of objects, but is different from the array command because not all new objects need be exactly the same as the original.
- Cloud of Points, or Point Cloud** — A set of x-y-z coordinates obtained from a 3D scanner or digitiser. The data can then be turned into a continuous surface and used as a 3D model.
- CODEC** — Short for "COmpressor/DECompressor". This is the term used to reference the way that software programs handle different movie files, such as Quick Time, AVI, etc. The CODEC can control image quality, and can assign the amount of space given to the movie file. First, a multimedia publisher uses a codec to squeeze more sound and video into less file space. These compressed files are easier to fit on a CD-ROM and transfer to your computer. Then, your computer uses a codec to expand these files back to their original size and replay them on screen.
- Coincidence** — Refers to geometry that occupies the same spatial location. For example, coincident vertices are points that occupy the same x, y, and z coordinates. Coincident lines can have differing lengths while one occupies the same location as the other.



**Color Bleeding** — When the color from one surface reflects onto another surface.

**Color Depth** — The number of bits used to represent a color. For example an 8-bit image uses  $2^8=256$  colors. The bits build up the three primary colors red, green and blue.

The bits are divided into red, green and blue (optionally an alpha channel as well).  
For example a 16-bit color could look like this R: 4-bit (16), G: 4-bit (16), B: 4-bit (16), Alpha: 4-bit (16) — together they add up to 16-bits. The number of bits can also be unevenly divided (R:5, G:5, B:5 Alpha:1).

This is why a GIF (max 8-bit=256 colors) only sports 128 colors if it is transparent (1 bit is used to represent transparency in the alpha channel, 7-bits =  $2^7=128$ ).

The following table indicates the number of colors an image can have.

8-bit =  $2^8 = 256$   
16-bit =  $2^{16} = 65536$   
24-bit =  $2^{24} = 16$  million  
32-bit =  $2^{32} = 4.3$  billion

You should also be aware of FP, or HDR images.

**Color Model** — A system used to specify colors. In LightWave, you can set color according to the following color models: RGB (red, green, blue), HLS (hue, lightness, saturation), HSV (hue, saturation, value) or integer values.

**Color Keying** — An old technique of combining two images by replacing one color of the destination image by the corresponding pixels of the source image.

**Column Interleaved Format** — The 3D image format used by the VRex VR-4200 projector ([www.vrex.com](http://www.vrex.com)). Left and right view image data are encoded on alternate columns of the display.

**Compiled Vertex Array (CVA)** — Array of geometry data on a vertex level that is optimized (compiled) for faster access by the graphics card. (Note that this is an OpenGL term, and is known by other names in other APIs.

**Compositing** — A layering technique that places one image on top of another, properly taking into account transparent pixels, apparent depth, shadowing and other elements that make up an image.

**Concentric** — Having a common center or origin point with varying radii.

**Cone Angle** — The angle at the peak of a cone.

**Conic** — Element having the form of a cone.

**Conic Section** — Curve formed by the intersection of a plane with a cone.

**Constraints** — Values in a geometric model that define relationships, i.e. a line is tangent to a circle. Constraints are often used to drive parametric or variational geometry-based systems; the algorithms used to work with constraints are known as constraint management.

**Continuous LoD** — Short for continuous Level-of-Detail, this method is based on the observation that 3D objects located far off in the distance may be approximated by simpler versions without loss of visual quality, thus increasing the rendering performance. The “continuous” refers to having the algorithm constantly recompute the detail level of the 3D object depending on the distance to the camera instead of having a pre-computed set of objects to choose from. Computationally expensive, this method is most often used in height field rendering applications. LightWave can approximate continuous LoD by using an Object list.

**Control Mesh** — A cage of points used to shape SubPatches.

**Convex Volume** — A convex volume can be defined as a volume whose every corner can be visible from all other corners in the same volume. Another way of defining the convexity is that all faces in the volume will be lit by a point light located anywhere within the volume.

**Cookaloris, Cookie** — A device put in front of a light, to break the light up. Common cookaloris look like leaves on trees, or blinds on windows.

**Coons Patch** — A free-form surface that is determined by the four curves that define its edges.

**Coplanar** — Refers to two or more entities that lie on the same plane. Two planar surfaces, for example, that lie on the same 3-dimensional plane are considered coplanar. If these coplanar surfaces share a common edge, it is recommended that they be joined into a single surface.

**Cross Product** — Using two vectors to calculate their normal.

**Cross-Section** — A view of the interior of an object as it is sliced along a plane.

**Cubic Image Map** — One of the many methods of applying a texture to a surface. This method applies the texture to an object as it would apply the texture to a cube. There are many other methods of texturing objects, such as Planar and Cylindrical image mapping.

**Curvature Continuity** — A curvature continuity with smooth transition of the edges of two meeting surfaces (the highlights of the two surfaces blend together seamlessly, forming the illusion of a single shape). If a curve (or surface) has tangent continuity and both the 2D curves (or 3D surfaces) have the same radius a very smooth transition is created with curvature continuity. Curvature is defined as  $1/\text{radius}$ . Hence, small radius equals high curvature.

**Curve** — In computer graphics, there are different ways of representing a curve, such as NURBS and Bezier curves. See also **NURBS** and **Bezier**.

**Curve Parametrisation** — See Parametrisation.

**CVA** — See **Compiled Vertex Array**.

**Cylindrical Image Map** — One of the many methods of applying a texture to a surface. This method applies the texture to an object as it would apply the texture to a cylinder. There are many other methods of texturing objects, such as Cubic and Planar image mapping.





**Decay** — Phenomenon where light intensity decreases with distance. The further away from the light source, the less intense are its rays. In the real world the decay is proportional to the inversed square of the distance (quadratic decay), but there is also directional (one-dimensional) decay (slower than in real life) as well as cubic decay (faster than in real life). See also **Attenuation**.

**Decompression** — Process of returning a compressed file to its full size.

**Default Unit** — The Default unit is the unit of measure (ex. meter, feet, etc.) that is assumed, usually when no unit of measure is entered with the numeric data. In Layout, it is determined by the setting on the General Options Tab of the Preferences panel. In Modeler, the setting is on the Display Options panel.

**Depth Buffer** — See **Z-Buffer**.

**Depth Cueing** — The process of reducing the apparent sharpness of an object the farther away it is from the viewer or camera. This often enhances the perception of depth.

**Depth of Field (DoF)** — The total distance, on either side of the point of focus, which, when viewed from an appropriate distance, appears sharp in the final print.

**Depth Sorting** — Sorting all triangles in the world depending on diminishing depth (lower and lower z-value) so when rendered, the triangle closest to the viewer obscures those behind it.

**Deskew** — Process used to remove skew or distortion through a small angle rotation.

**Diffuse Component** — Part of the reflection-illumination model. The diffuse is concerned with the amount of light that is reflected back.

**Diffuse Light** — A component of the reflective model that is the result of direct illumination.

**Diffuse Map** — Replaces the diffuse component of the reflection-illumination model, basically giving the illusion of being painted onto the surface. To create a material resembling wood or marble, this map is used. Generally, when you talk about the “texture map” in an application, this is the map actually referred to.

**Dimension** — A measure of spatial extent, especially width, height, or length.

**Directional Light** — See **Distant Light**.

**Director of Photography (DP)** — Person on set that determines how to photograph the movie.

**Disc** — Shape that is referred to in mathematics as a cylinder. This shape is composed of two circular or oval-shaped bases and the space contained between those bases. In other words, a disc is like a stack of circles with set parameters defined by you.

**Displacement Map** — Can be used to modify the actual mesh (as opposed to the bump map) to create wrinkles, creases, crumples etc. The displacement map will need a more complex mesh to create the same effect as bump mapping, but has the advantage of allowing more thorough close-ups, since the surface is actually deformed and not just simulated as being so.

**Display Types** — Ways of displaying objects in a viewport. Display types are available only for geometry views. The available display types are Bounding Box, Vertices, Wireframe, Front Face Wireframe, Shaded Solid, Textured Shaded Solid and Textured Shaded Solid Wireframe. Display types do not determine the quality of the final render.

**Distant Light** — A light with color, intensity and direction. All rays emitted from a distant light are parallel, and therefore it has no obvious source. Distant lights can be used to simulate point lights from a great distance (whose rays can be approximated to be parallel), for example the sun. The intensity from a distant light does not decay.

**Dithering** — Creating the impression of having more color on the screen than there actually are by plotting pixels (with a limited amount) of different colors next to each other.

**DoF** — See **Depth of Field**.

**Dolly** — To move the camera along its line of sight (in a straight line following the imaginary path between the actual camera and its target point).

**Dongle** — A hardware lock used to prevent the piracy of LightWave.

**Dopesheet, Dopetrack** — Animation tools in LightWave that allow you to better organise keyframes.

**DPI** — Dots per Inch. In a bitmapped image, the number of dots that exist within each inch of the image. This number remains constant, so when you make an image larger, the quality decreases, but when you make the image smaller, it appears to increase.

**Double Buffering** — This is the process of using two frame buffers for smooth animation. While the image of the first buffer is being displayed, the graphics controller can use the second buffer to build or render the next image. Once the second image is completed, the buffers are switched. Thus, the result is the appearance of smooth animation because only complete images are displayed, and the process of drawing is not shown. You can often now see triple buffering in graphics cards to allow an extra buffer for the next image in case there is a problem.

**DP** — See **Director of Photography**.

**DVD** — A high-density compact disc for storing large amounts of data, especially high-resolution audio-visual material. DVDs used solely for computers are commonly referred to as DVD-ROM.



**DWG** — AutoCAD native file format. It can contain 3D data, but is hard to convert to a LightWave-native format because of its construction. A DWG file is parametric, that is to say it does not contain the objects themselves, but rather instructions on how to build the objects. This makes it hard to translate if you do not possess a licence of AutoCAD. The solution is to either get one or get your client to supply you the object in a different format, preferably OBJ.

**Easing** — Reduction in the acceleration or deceleration of motion to present a smoother, more continuous movement. The shape of a function curve can reflect this when using spline interpolation.

**Edge** — Straight line connecting two points on a polygon.

**Edge Loop** — Particular method of modeling organic shapes with the edges of polygons creating a loop or a flow around circular features, like the eyes and the mouth for example.

**Endomorph** — Object containing one or more Morph Maps.

**Envelope** — Way of animating a particular value over time using a graphical input mode.

**Environment Map** — Map often used to simulate (faking) reflection of the surrounding world without using ray tracing.

**Euler Angles** — Euler angles are one of the simplest methods of representing 3D rotations, and generally also the easiest to visualize. An object's rotation can be specified in terms of its yaw, pitch and roll, or rotation around the Y, X and Z axis, respectively. Euler angles suffer from singularities in the form of so-called Gimbal lock, however, and are also difficult to smoothly interpolate for keyframe animation.

**Expression** — Mathematical expressions that allow you to change the animation of an object. You can also create constraints between objects using expressions or create conditional animation. Expressions are very powerful for creating precise animation and to create automated animation of things such as wheels.

**External Attributes** — The position of the camera and the direction it is pointing in.

**Extrude** — Creating a three-dimensional object from a two-dimensional shape by adding a third dimension to it. You can also do this along a motion path or spline.

**Face** — The shape made up by the bounding point making a polygon. Faces can have as many vertices as wanted, but only polygons having a shape of three or four vertices can be made into subdivision surfaces.

**Face Normal** — Also just known as the normal, this is a line perpendicular to the face that also describes which way the face is pointing in a one-sided polygon.

**Falloff** — The volume starting at the outer rim of a spotlight's hotspot, decaying from full intensity at the start to zero intensity at the outermost rim of the spotlight. The less the difference (in angles) between the hotspot/falloff, the crisper the shadows. If the falloff angle is much larger than the hotspot angle, the boundaries of the area lit up by the spotlight will be fuzzy.

**Field of View (FOV)** — The angle of the view frustum. The wider the FOV, the more you see of the scene. Human eyes have a FOV of about 50 degrees, and normally virtual reality application use similar values to resemble real life.

**Field Rendering** — An option that causes the program to render two interlaced fields of information. This is in contrast to rendering only one (full frame) and makes moving objects appear to move more smoothly. Used for projects that will play back on television monitors that display 50 or 60 interlaced frames per second. Fielded animation is not useful for animations designed to be displayed on computer monitors. See **Fields**.

**Field Sequential 3D Video** — The most common format for 3D video. Left and right image data are encoded on alternate fields of the standard video signal.

**Fields** — Interlaced images (video) consist of two fields combined into one frame. Each field contains half the scan lines (either even or odd) and is a separate pass. This is more common to render to for TV broadcast. Moving items horizontally will strobe without rendering to fields.

**Fill Light** — Additional light sources assisting the key light in a scene. Usually they are less intense than the key light and created using point light or spotlight.

**Fillet** — To round off the edges of an object with a round shape. Think "router", use "Rounder" in Modeler to achieve it.

**Fill-rate** — The amount of pixels from a texturemap (texels) that are rendered per time unit. Measured in texels/second.

**Filter Opacity** — Type of opacity that uses a color to simulate object opacity.

**Finger** — A small strip placed in front of a light to cast a discrete shadow.

**FK** — Forward Kinematics. Positioning a figure by specifying joint angles, like posing a toy action figure.

**Flag** — A large device placed in front of a light to create a shadow, creating a large shaded area.

**Flat Shading** — Shading technique where all individual faces in a mesh are assigned a single color value based on the orientation of their face normals.

**Flatness** — Flatness is used as a threshold in determining if a polygon is non-planar. A flatness of 0 percent means the polygon is absolutely flat. Flatness is computed as a percentage deviation from a triangle (the "ideal plane") formed from the first two and last vertices of a polygon. All of the other points are measured relative to this plane. The largest deviation is divided by the total size of the polygon to get a percentage that is the flatness value. For example, if a polygon is 1 meter wide, .5% flatness means that no point will be outside the ideal plane of the polygon by more than 5 millimeters. (1 x .005)

**Floating Point (FP) Images** — Refers to images that do not use standard color depth models to represent the colors contained in them, but rather an expression of the floating point value of a color changing from 0 for black up to 1 for the brightest point in the image. A mid-grey in such an image would be represented by R: 0.5, G: 0.5, B: 0.5.





- Focal Length** — The distance between the lens and the light-sensitive surface on the backplane (the film in real-world cameras). The lower the focal length, the more of the scene is visible. Focal lengths less than 50 mm is called wide angle, while lengths greater than 50 mm is referred to as telephoto lenses. The longer the lens, the narrower the field of view. Distant details become more visible. The shorter the lens, the wider the FOV. More of the environment is visible in the rendered image. To simulate the human eye, you can use values of about 50 mm.
- Fog** — Simple yet effective tool often used in real-time graphics to obscure the far plane, thus bounding the viewing distance of the application. There are essentially three types of fog: table-based, vertex-based, and volumetric. Fog values may also follow linear or exponential curves.
- Foreground Image** — The image closest to the camera.
- Foreshortening** — The apparent effect of viewing an object on its long axis that makes it seem shorter. For instance, an arm pointing directly at the camera seems to lose its length as does a road going directly away toward the horizon.
- Formula-Defined Shapes** — Refers to shapes that are defined by using one or more equations. This includes complex shapes such as aesthetic bottles, or simple shapes such hyperbolic paraboloids, oblate spheroids, prolate spheroids, or ellipsoids.
- FOV** — See **Field of View**.
- FPS** — Frames Per Second. The main unit of measure that is used to describe graphics and video performance.
- Frame** — One image out of many that define an animation. There are 24 frames per second in film, 25 frames per second in PAL video, and approximately 30 frames per second in NTSC video.
- Frame-Buffer** — The memory a computer uses to hold one or more frames for later use.
- Frame-Rate** — The speed at which a frame of animation is shown, usually expressed in frames per second. European TV is at 25 frames per second, US is typically 29.97 frames and movies are projected at 24 frames.
- Freeze** — To convert from vector or interpolated geometry (splines, NURBS, subdivision surfaces) to pure polygons. Even if the renderer supports NURBS or subdivision surfaces, this freezing happens at render time, and is usually definable to the level of polygon creation by you.
- Frustum** — The part of a solid, such as a cone or pyramid, between two parallel planes cutting the solid, especially the section between the base and a plane parallel to the base. See **View Frustum**.
- FX** — Shorthand term for effects.
- GCore (Geometry Core)** — Engine in LightWave that handles all animation and Modeling tools.
- Generic Primitive** — Simple 3D objects that most 3D programs can create easily. These objects typically consist of spheres, cylinders, cubes, and cones.
- Geometry** — Positional layout of points and polygons for an object. These points are usually seen with objects that can be rendered. For example, a cube's geometry is composed of eight points. A curve has geometry since it is composed of one or more points, whereas nulls have no geometry.
- Gimbal-Lock** — What happens when two axes of rotation line up, thereby making three-dimensional rotation impossible. As an example, take any object with neutral rotation (0 degrees on heading, pitch, and bank) and rotate the pitch 90 degrees. Now, try to rotate the bank. This is gimbal-lock.
- Gizmo** — See **Null**.
- Global Illumination** — Unlike local illumination, this method of generating images supports effects not only linked directly to the light sources themselves. In real life, the intensity of a surface depends not only on direct illumination from the light source itself, but also from indirect illumination from surfaces being lit.
- Ray tracing can cast shadows from an object onto a surface, allowing objects to be reflected in shiny surfaces or refracted in transparent materials. Radiosity is the effect of reflected light. If you have spotlights projected at the ceiling in a white room, the light will bounce back and light up the entire room. However, this can only happen if the renderer supports radiosity (as LightWave does) or other similar techniques.
- Glossiness** — Affects how spread out across a surface a lighting highlight is. Low glossiness makes a spread out highlight while high glossiness creates a more central, pinpointed highlight.
- Glossiness Map** — An image to control the glossiness of a surface. Bright values in the image indicate more glossiness, dark values less.
- Glow** — Optical light effect that looks like a fuzzy disc around the center of a light source.
- Goal** — An object used in IK to create a point where an object will always reach for. This is used to make objects appear to have realistic motion.
- Gouraud Shading** — Developed by Henri Gouraud in 1971, this is a fast incremental shading technique using bilinear intensity calculation to create smooth transitions between the vertices in a triangle. It is most often used for lighting purposes by computing the vertex normals in the polygon, calculating the light values for each vertex, and then Gouraud shading the polygon. Even though it has obvious advantages over flat shading, the facets in the mesh can still be discerned. The placement of the highlight depends on the underlying polygons.
- GUI** — Graphical User Interface. The graphical interpreter between man and computer allows a more intuitive interaction with the computer. The window maker in UNIX, and Windows for the PC are both GUIs. This way you don't have to be computer literate to the same extent as if you should have to type all commands you wanted the computer to perform.



**Greeblies** — English slang. This describes the non-such little details on objects, usually mechanical objects. Those details which can be found on spaceships, in engine rooms, etc. You can also use the words “didges”, “nurnies” and “doohickies”.

**Halo** — Optical light effect that forms concentric circles around the center of a lightsource. Often clearly visible around street lights after a rainy day.

**Hidden** — Any element that is not shown in the current rendering of the scene but that still exists.

**Hidden Surface Removal** — Algorithm for removing obscured polygons in a three-dimensional view space. As opposed to the faster algorithm backface culling, the hidden surface removal algorithm is able to sort out those polygons that are obscured by another object. Another way of finding an obscured polygon is the z-buffer.

**Hierarchy** — A way of defining objects in relationship to each other (using a parent-child or tree analogy). This relationship means that transformations, deformations, and any other property of the parent object affect all child objects. This allows separately modelled objects to be used in a scene as a single functional unit. The movement of a parent affects the movement of the child, but you can move the child without affecting the parent.

**HDRI** — High Dynamic Range Image. An image with a wide intensity range between the brightest and darkest pixels. In typical 8/24-bit images, the maximum possible intensity range is 255 times brighter than the darkest grey pixel (with a value of 1). Natural scenes and images rendered with radiosity can have dynamic ranges from 10 to 10,000 times greater than this. Recording this information requires use of an image format with higher precision - such as LightWave's native .FLX format.

**Highlight** — Reflection of a light source on an object's surface. The size of the highlight (the area that shows the light source reflection) depends on the angle. Consequently, multiple light sources result in multiple highlights. This is also the Specularity.

**HLS Color Model** — Hue, Lightness and Saturation: the three components of the HLS color model. Hue refers to the position of the color in the spectrum, such as red, yellow, or green. Lightness is the amount of white mixed in a color, such as the difference between a pure red and pink. Saturation is the purity of the color, such as the difference between a pure red and a dusty rose - low saturation means that there is grey in the color.

**Hotspot** — The inner intense cone of light emanating from a spotlight.

**HSV Color Model** — Hue, Saturation, Value: the three components of the HSV color model. This color model defines the hue and saturation similar to the HLS model. Value is similar to lightness, as in HLS; however, a value of 1 represents a pure color when saturation is 1, while a lightness of 1 yields white no matter what the saturation. In both systems, 0 is black.

**Hub, The** — Module in LightWave that allows the Layout and Modeler modules to synchronise information. It uses the TCP/IP protocol to transfer information between modules.

**Hue** — The position of the color in the spectrum that describes the tone or tint of a color, such as red, yellow, or blue.

**HyperVoxel** — Voxels are volumetric rendering effects. HyperVoxels are voxels that are applied to nulls, points, or objects.

**IK** — Inverse Kinematics. The process of determining the motion of joints in a hierarchical 3D object given the desired start and end points, all the while obeying the laws of kinematics. Think of it like the strings on a marionette puppet.

**Image instance** — A copy or instance of a source image. Each time you use a source image, an instance of it is created. You can have as many instances of the same source as you need. You can then edit, crop, or even blur the instance without affecting the original source image.

**Image Map** — An image that is applied to an object's surface.

**Incandescence** — The emission of visible light by a hot object. In LightWave, this is the luminosity channel.

**Incremental Shading** — See Interpolative Shading.

**Indirect Illumination** — Light that bounces off one surface and illuminates another surface. This can only happen if the renderer supports radiosity. The LightWave renderer supports radiosity.

**Intelligentities** — Refers to LightWave's object format. The object format can contain morphs, multiple layers, and independent pivot points on a per layer basis.

**Intensity** — The strength at which the light source illuminates objects in the scene.

**Interference Checking** — The process of identifying if and where two or more pieces of geometry (usually solids) intersect. When moving parts are involved, a kinematics analysis is used to detect interferences.

**Internal attributes** — Properties of the camera such as depth of field and line-of-sight - compare with External Attributes.

**Interpolation** — Process used to estimate an unknown value between two or more known values. In animation, interpolation is the process used to calculate values at frames between two keyframes in a sequence.

**Interreflection** — When a reflective object reflects another reflective object. For example, if you place two mirrors in front of each other, the first one will display the second one, who, in turn, shows the first one. In real-life, there is virtually no upper limit of how many interreflections may occur, whereas in 3D rendering, one must set an upper limit to be able to render the scene. The default value for LightWave is 16, but it can be lowered to 0, if desired, or raised to 24 at a cost in increased rendering time.

**IR Transmitter** — A device that sends synchronisation signals to wireless shutter glasses.

**Isometric view** — Standard view in a 3D design where the top, front, and right side faces of a cube are equally inclined to the screen surface.



- Item** — An item in Layout refers to an object, bone, light, or camera.
- JPEG** (Joint Photographic Experts Group) — A widely accepted, international standard for compression of color images.
- JPS** — Stereoscopic JPEG file. JPS refers to a stereoscopic image file format that is based on JPEG compression. Used by DepthCharge & other stereoscopic imaging applications.
- Junkyard** — A special directory used by some studios to hold mechanical and non-organic pre-modelled parts.
- Keyframe** — (Key) A frame for which you define an animation channel(s) (e.g., position or rotation) for an item in Layout. Animations are composed of a beginning keyframe, an ending keyframe and usually some number of keyframes in between. See also **Tween**.
- Key-Light** — Dominant light source in a scene, normally created with a spotlight.
- Kinematics** — The properties of each 3D object that control its transformations. These transformation properties are used to modify the selected object's scaling (size), rotation (orientation), and translation (position) in X, Y, and Z in either local and global space. Although related, kinematics are not to be confused with inverse and forward kinematics for animation.
- Kit-Bashing** — An expression taken from model making. The practice of using model kits to give detailing to a larger project. This is still in use. It refers to the taking of models that you have already made, to use in the creation of another, perhaps even basically unrelated model.
- Lasso** — One way to perform a selection of points or polygons. This method involves drawing a loop that encircles all of the objects that need to be selected.
- Latent Surfaces** — Surfaces that are no longer visible after a Boolean or intersection operation because they lie inside or outside the solid.
- Lathe** — Creating a 3D object from a 2D shape by rotating it around an axis.
- Lattice** — Either a way of deforming object using a lattice or a way of creating outlined geometry.
- Layer** — A portion of a scene. Each layer consists of an object or multiple objects that can be edited separately from the rest of the objects in a scene. A layer is basically a building block for a scene and each layer contains separate blocks for a final model.
- Left-Handed Coordinate System** — Coordinate system where the positive part of the Z-axis goes away from the observer (from the screen).
- Lens** — Part of the camera determining the optical characteristics of the image, such as wide angle, fish eye, and depth of field.
- Lens Flare** — Optical light effect made up from a number of bright discs. If the rays from a light source reflect off the surface of a compound lens in a camera, it can generate star-like patterns on the image. Lens flares tend to be a cliché of bad CG imagery, probably because of their short rendering time and flashy appearance.
- LoD** — Level of Detail. This is a term which refers to varying the amount of detail in an object depending on the distance from the object to the camera. Example: A car for a close-up would need to have every little detail modelled into it. Chrome, bumpers, body seams, door handles, etc. But that same car, as seen from a helicopter flying over a highway, might be able to be a simple cube with an image map applied to it.
- Level-of-Detail Control** — The ability to vary the amount of details displayed in a graphics image to improve performance. For instance, at a distance, models can appear as simple 3D figures, but as users zoom in, a more detailed representation is presented.
- Light** — In LightWave, a light is generally used just like a light in real life. Lights illuminate a scene and allow you to see the objects within it. Different types of lights are distinguished: ambient light, diffuse light, point light, spotlight, etc. There are also different terms used to simulate the way material properties are illuminated: ambient component, diffuse component, specular component. Incident light at a surface = reflected + scattered + absorbed + transmitted. Light has a major impact on the photorealism of a rendered scene, but can be hard to recreate.
- Light Source** — There are several different sorts of light sources used in 3D graphics to simulate light: ambient, distant, linear, area, and spotlight. Special light effects can be recreated such as volumetric light and glow. With radiosity, an object with a high luminosity value can cast light as well.
- Lighting a Scene** — One of the ingredients of a nice rendering is realistic lighting. It is often good to use one single light source (the key light) doing most of the work, helped out by some additional, less intense lights (fill lights) which illuminate the background of the rendered object to create a smoother look. Try to avoid shadows with edges that are too crisp, since this is unusual in real life due to radiosity.
- Lighting Model** — This is a model that uses a mathematical formula to decide what will happen when light strikes an object's surface.
- Light-Map** — Luminance map generated (normally rendered) individually for each polygon and then blended with the texture map to give the impression of light and shadows falling onto the polygon without having to draw the effect on the texture itself. The advantage of separating the light-map from the texture map is that if you should want to create a new "mood" for a scene you can set up new lighting conditions for the scene, re-render the light-maps and apply them to the mesh again, without having to redraw all texture maps.
- Linear Patterning** — The repetitive placement of the active pattern cell along a line, line string, shape, arc, circle, ellipse, or curve element.



**Line-of-Sight (LoS)** — Has become quite important in modern real-time interactive simulators, especially for military purposes. To cut down on the polygon count and increase rendering performance, programmers are often forced to employ schemes to simplify terrain at large distances. This, however, has the unfortunate drawback of warping the terrain, something that may make a difference for long distance targeting purposes. Because of this, modern terrain rendering algorithms such as ROAM tend to not simplify along the primary LOS.

**Local Coordinate System** — As opposed to the world coordinate system, the Local Coordinate System is tied to a specific object. LCS are used, among other reasons, to simplify the representation of complex objects by using several, different LCSes as reference points for the object's vertices. It is also easier to transform the object if you for instance can rotate it around its own "center of gravity" instead of the origin of the World Coordinate System.

**Local Coordinates** — Every object has its own origin, which is subordinate to the world coordinate system (or other objects that are higher in the hierarchy). Local coordinates are useful for determining positions of subordinate objects.

**Local Illumination** — A mathematical model capable of creating imagery where only direct illumination is considered. Depending on the distance from the lightsource, etc, each surface in the model can be given a color and intensity. This does not include shadows, reflections and radiosity.

**Loop** — A continuous playback of an animation sequence.

**Low-Poly Modeling** — To model using as few polygons as possible, to speed up rendering and processing time. Common style for games, but as game processor engines get better, and computers faster, this is losing ground as an art form.

**LScript** — LightWave's built-in scripting language. Can be installed and used just like plugins.

**Lumel** — Short for LUMinance ELEment, the lumel is a pixel in a lightmap which constitutes the color level in a specific area of the texture map it is superimposed upon.

**Luminance** — The black and white information (brightness, sharpness, and contrast) encoded in a color. The amount of luminance contained in a color is directly proportional to the amount of light intensity.

**Luminance Map** — An image to control the luminance of a surface. Bright values in the image indicate more light intensity, dark values less.

**Luminosity** — Much like glow, luminosity is a measure of how much light a surface gives off before any light strikes it. This effect can be used to create an object that gives off its own light.

**Magnet** — This tool allows you to move points in an object as if he or she was using a magnet. It has an area of falloff where the strength of the magnet decreases gradually to 0 giving a soft selection effect.

**Map** — An attribute that can be added to an object's surface to give it a certain look. Projecting an image so that it covers the surface of an object or images that affect the way an object looks. There are a variety of different maps used to create specific effects: diffuse maps, bump maps, opacity maps, etc. Maps can be divided into bitmap-dependent texture maps and procedural maps. The latter categories can, in turn be divided into 2D maps and 3D maps.

**Mapping** — Process of making one image conform to the size, shape, and/or texture of another.

**Material** — Even if it is hidden beneath another texture map, there is an underlying material in any surfacing. The material is applied to the whole object, and can be made to look like wood, plastic, glass, metal etc, (hence the name), by modifying its properties. Also referred to as Surface.

**Material Properties** — The different properties of a material such as the ambient component, diffuse component and specular component in the reflection-illumination model.

**Matrix** — Matrices form the core of linear algebra and are important tools in most engineering disciplines. In essence a two-dimensional array of numbers, matrices are often used in transforms of different properties, such as rotation, scaling, translation, deformation, and much more.

**Memory Swapping** — The transferring of data back and forth between active RAM memory and disk. When this happens, it can considerably slow down computing tasks such as rendering.

**Mesh** — Object made up from a number of triangular faces. Also, slang used to refer to objects.

**Mesh Complexity** — Describes the amount of information (number of vertices, normals, triangles etc) used to create objects. More complex meshes need more memory and are slower to process

**Meta-primitive** — A Metaball, Metaedge or Metaface object.

**Metaform** — Option used with the Subdivide tools. It does not simply divide the individual polygons of an object, but rather renders the edges of the polygons to be smooth, making the object seem less faceted and cleaner.

**MIP-Mapping** — Using a pyramid structure of a predefined fixed amount of differently sized bitmaps (original size, original size/2, original size/4, etc) to speed up rendering time by using less detailed textures for distant objects (represented by only a few pixels on the screen), and the full-sized version of the bitmap when the objects are closer to the observer. This way, moiré-pattern can be avoided.

**Mirror** — Creates an exact mirror image of the selected object. This tool is very useful for any symmetrical object, including faces, cars, and airplanes. This tool literally cuts the Modeling time of this sort of object in half.





- Modal/Non-modal** — A modal panel must be closed before you can continue working with the rest of the application. A non-modal panel lets you shift the focus between it and another part of the application without having to close the panel—you can continue to work elsewhere in the current application while the panel is displayed. Modeler's Numeric Panel is non-modal because you can do other things while it is open. In contrast, Modeler's Display Options Panel is modal because you must close it before you can continue working.
- Modeling** — The process of creating, or recreating, an object inside your 3D software.
- Moiré Pattern** — Optical pattern created due to aliasing. Usually appears as a swirling pattern along a distant edge.
- Morgue** — A special directory, used by some studios, to hold already modelled organic body parts for other modellers to draw from. If you have modelled a good head, hands, ears, feet, etc. there is no reason to model them again.
- Motion Blur** — The blurring of objects that move while the camera shutter is open, creating the illusion of movement. Motion blur also prevents strobing caused by too-rapid movement.
- Motion Capture** — Method used to input live movements into a computer from an external source.
- Motion Channel** — Generally the same as Animation Channel, but refers only to position, rotation, and scale (i.e., not light intensity.).
- Motion Path** — The line an object follows while in motion.
- Multiplex** — The process of taking a right and left image and combining them with a multiplexing software tool or with a multiplexer to make one stereo 3D image.
- Multi-Texturing** — Applying two (or more) textures on the same face. For example, a polygon can have a texture map resembling a brick wall and then be multi-textured with a light-map to give the illusion of being lit.
- Natural Light** — Light that exists in nature, such as sunlight or moonlight, depends on the time of day, season and location on the Earth. The sunlight on a clear day has an RGB value of about R:250 G:255 B:175. For simulating overcast it might be a good idea to add the blue component, whereas a sunset could be a little more orange. As opposed to artificial light, the natural light has only one source (the sun) and can most effectively be recreated using a distant light.
- Node** — The basic graph element used to represent distinct items (vertices, faces, etc.). A signal coordinate in a grid, or finite element grid point used to describe the structure. A node will lie on each vertex of a finite element, and additional nodes may lie along element edges to define curved element topology.
- Non-Planar** — Generally refers to a polygon where all points do not reside in the same plane and can occur only with polygons using more than three points. Non-planar polygons can cause erratic rendering errors. As an example, a square piece of cardboard sitting upon a tabletop will become non-planar on all vertices when lifted by a corner. Inherent in manipulation and deformation of a model, non-planar "holes" can appear in the surface consistency of models. Solutions include "tripling" (halving the quads diagonally) or tessellating the polygons into triangles. As an example, a triangular piece of cardboard sitting upon a tabletop will remain planar on one vertex when lifted by any corner. Thus, when joined on their vertices, a group of triangles are more robust when deformed.
- Normal** — A polygon normal is the imaginary line projecting out perpendicular to a surface at any point indicating the direction of the polygon. A polygon surface normal is represented as dashed lines on selected polygons in Modeler. LightWave sees polygons or faces of an object only from the surface normal side. A single-sided polygon (like a piece of paper) with its normal facing away from the camera will be invisible to the camera from that viewpoint (unless the surface is using the Double Sided option). A vertex normal's direction is the average of the polygon normals it is connected to.
- NTSC** — National Television Standard Committee. The most common video standard in the United States and Japan. It has a frame-rate of roughly 30 fps. 60 times per second every other scan line is changed, resulting in smoother transitions. Its pixel resolution is 720x486 with a pixel aspect of .9
- Null** — Non-renderable helper-object used in modeling programs to simplify the manipulation of 3D-objects and texture mapping.
- NURBS** — Abbreviation for Non-Uniform Rational B-Splines.
- Nurnies** — American slang. See greeblies.
- Object** — A model or construction that when placed in a scene will render what it represents from the real world. An object is composed of points and faces. Points connected together to form a polygon define a face. Faces joined together form an object.
- Object Oriented Graphics** — Different from bitmap format, this image type consists of objects that have definite mathematical formulas behind them. These images always print at the maximum quality specified by the printer, unlike bitmapped images that always print at the same quality level. They can also be referred to as "vector graphics".
- Omni-Directional Light** — Same as a point light.
- Opacity** — The opposite of transparency.
- Opacity Map (or Transparency Map)** — Makes the surface more or less transparent depending on the pixel intensity (color value) of the opacity map where normally black is transparent and white is opaque.
- OpenGL** — A 3D graphics API that includes capabilities for 2D imaging. Basically, OpenGL is a set of instructions that can be used by a program to interpret images and display them on the screen. LightWave uses OpenGL for all its displays.



**Optical Light Effect** — If the observer (or camera) looks directly at a bright light source, it may appear to glow. If the light is refracted through a lens or even your own eyelashes (try squinting towards a spotlight!), the light will appear to form star-like patterns.

**Orbit** — To travel around a target - more commonly circular, but a comet's orbit can be elliptical.

**Origin** — The world Origin is the absolute center of the LightWave universe. A local Origin is the center of an object. Both are defined by the XYZ coordinates of 0, 0, 0.

**Orthogonal** — A view that displays a parallel projection along one of the major axes. In an orthogonal view, the camera is oriented to be perpendicular (orthogonal) to specific planes: the Top view faces the XZ plane, the Front view faces the XY plane, and the Right view faces the YZ plane. An orthogonal view eliminates the effect of distance from a viewpoint, which provides a useful means of locating points and objects in 3D space and is particularly helpful when modeling objects in wireframe mode. An orthogonal view is in contrast to a perspective view.

**Orthogonal Direction** — There are six different orthogonal directions in a three-dimensional space: up, down, back, forward, left and right.

**Orthographic Projection** — Viewing system where the projectors are parallel and therefore don't create a perspective with foreshortening.

**PAL** — Phase Alternating Line. The most common video standard in Europe. It has a frame-rate of 25 fps. It is interlaced, which means that 50 times per second every other scan line is changed, resulting in smoother transitions. The resolution is 720x576 pixels and the pixel aspect ratio is 1.0667.

**Pan** — To rotate the camera horizontally. As opposed to the orbit movement, pan rotates the camera around a single axis, as if it were mounted on a tripod.

**Panel** — In a 3D program, a screen that serves many functions such as informing you of errors, asking for user input, or informing you of the state a program is currently in. Otherwise known as a dialog, window or requester.

**Parabola** — A plane curve formed by the intersection of a right circular cone and a plane parallel to an element of the cone or by the locus of points equidistant from a fixed line and a fixed point not on the line.

**Parametrisation** — Technique for assigning values to the edit points as they are spaced along the length of a curve. Can be either uniform parametrisation or non-uniform parametrisation (chord length). The first edit point on the curve has the value 0.0 (regardless of whether it is uniform or non-uniform) and the following edit points are assigned greater values the closer they lie to the other end.

**Parameters** — Also generally known as properties, parameters are the "atomic" elements of a property set whose values determine the behavior of something. A parameter is one degree of freedom. You can set parameters in property editors.

**Parent** — An object that influences the motion of another object in a hierarchy, called the "child".

**Parenting** — The process of creating a hierarchical organization of objects in a scene. In parenting, an object (called the parent object) is "parented" to another object (called the child object). Parenting relationships can be nested to any degree, so that one or more objects are the children of another object, which is in turn the child of another.

**Particles** — 2-dimensional objects typically used in large quantities to create effects like rain and explosions.

**Passive Polarised 3D glasses** — 3D glasses made with polarising filters. Used in conjunction with a view screen that preserves polarised light.

**Penumbra** — A partial shadow, as in an eclipse, between regions of complete shadow and complete illumination, a fringe region of partial shadow around an umbra.

**Perspective** — A traditional art method of creating the illusion of three-dimensional form and distance on a two-dimensional surface. Perspective provides a three-dimensional view of the scene that indicates depth. In a perspective view, objects appear to converge toward a central vanishing point, and objects closer to the camera appear larger than those farther away. A perspective view is in contrast to an orthogonal view.

**Perspective Projection** — Simulating three-dimensionality by using foreshortening that triggers the human perception to interpret a two-dimensional image as if it was three-dimensional. An object is drawn smaller and smaller the further it is from the observer. This is achieved by using a center of projection to which all projectors converge, as opposed to where the projectors are parallel.

**Phong Shading** — The most frequently used interpolative shading technique used today. It uses a linear combination of three components - the ambient component, the diffuse component and the specular component. The placement of the highlight is less dependent on the underlying polygons as Gouraud shading since Phong shading interpolates the normals on a per-pixel basis instead of interpolating the light intensity based on the distance to the three vertices.

**Photorealism** — The process of generating computer images that mimic photographs.

**Pitch** — The amount that the camera or an object in the scene is tilted up or down. If you nod your head "yes", you are rotating your head in the pitch axis.

**Pivot Point** — A single point, usually in the geo-center of an object that is used for many functions. It is the point that is addressed to locate an object's position in 3D space. It is also the point around which all rotational moves are made and the reference point for transformations and scaling.

**Pixel** — Short for Picture Element, the smallest element of computer or video display.

**Plane** — Refers to a two-dimensional (i.e., flat and level) surface. Imagine a plane as a piece of glass that is infinitely large, but has no depth.

**Plugin** — A program that works with and extends the functionality of LightWave.





- Point** — A fundamental building element of an object in 3D space with an XYZ location. Point coordinates are the minimum information from which the geometry of an object can be calculated.
- Point Light** — Light source emitting light in all directions (omni-directionally) from a single point in space, think “light bulb”. It takes six shadow calculations (one in each orthogonal direction) to render shadows generated by a point light, which means that inserting multiple point lights into a scene might slow down rendering time considerably.
- Polygon** — Geometric shape in one or many planes. Polygonal Modeling consists of using many faces to create the shape. Since polygons in most cases are faceted simplifications of a much smoother shape, they are more or less inaccurate, as opposed to the more organic NURBS. The more the tessellation, the higher and the closer the accuracy compared to the desired shape.
- Poly-Line** — A geometric entity composed of one or more connected segments that are treated as a single entity.
- POV** — Abbreviation for Point of View.
- Primary Colors** — There are three primary colors of light: red, green and blue (RGB). Light colors are additive, which means that if these three colors are combined equally, the result is a white light. Black is thus the absence of light.
- Primitive** — Basic geometric shape used in Modeling. Some primitives consist of a combination of different primitives. Cone, box, sphere, tube, torus, and disc are common primitives.
- Procedural Map** — A map (often three-dimensional) generated mathematically using a set of user-customised variables instead of using an image. The procedural map does not need texture coordinates.
- Procedural Textures** — Mathematically generated textures (2D and 3D). Their advantage is that they are largely independent of the projection type.
- Projected shadow** — A shadow that falls from an object and projects on a surface.
- Projection Map** — A mapping procedure that allows you to apply the map to multiple objects as if they were one.
- Quad** — A polygon with four sides, short for quadrilateral.
- Quantise** — This tool causes points to snap to a specific (X, Y, Z) coordinate. This tool is generally used when a lot of precision is required.
- Quaternion** — Quaternions are mathematical objects consisting of a scalar and a vector which together form a four-dimensional vector space. Although having interesting uses in mathematics, their main use in computer graphics resides in their capability of easily representing 3D rotations. Although impossible to visualise, they suffer from no singularities like Euler angles, and are also easy to smoothly interpolate for keyframe animation (using a mathematical operation called SLERP for Spherical LinEar INTERPolation).
- Radiosity** — A more physically correct approach (developed in 1984 by Siegel and Howell) to simulate propagation of light in a virtual environment. It takes into account the fact that light bounces off a surface and creates diffused lighting on the surrounding objects. The scene is divided into a certain amount of triangles that are used to represent the original scene (which speeds up the time-consuming process), and then light interaction is calculated using these triangles. As far as visual quality is concerned, the more crucial the part of the scene the denser the triangles must be. This technique creates much more realistically lit environments, however it takes much longer to render due to the massive amount of calculations.
- Rail Clone** — This tool creates multiple copies of an object that are evenly spaced along one or more user-defined curves.
- Rail Extrude** — Used to extrude polygons along a specified line or combination of lines. This allows you to create a shape other than that created from a normal, linear extrude.
- Rapid Prototyping** — The process by which a computer model of a 3D object is turned into a real object. Rapid prototyping methods vary but often involve laying down strata of base material which is then bonded together using a substance like cyanoacrylate (superglue).
- Rasterisation** — The process of, on a per pixel basis, determining what value to assign to the pixels on the screen from a vector-based image.
- Ray Traced Shadow** — Shadow created by tracing the light rays from a light source. The ray traced shadows are more accurate than those created by shadow maps, but take more time to render and always have crisp edges.
- Ray Tracing** — An advanced rendering technique capable of calculating reflections, refractions and shadows. Ray Traced renderings take more time to generate, but have a more photorealistic quality than simple scanline rendering.
- Ray Tracing Depth (Ray Recursion Limit)** — Number of times the light bounces off a surface when ray tracing. Used to create reflections and/or refractions. For example, ray tracing two mirrors facing each other with the ray tracing depth set to 3 will allow the image of the reflected mirror to show the first mirror in it.
- Reflection** — Light that bounces off a surface. A mirror is highly reflective, whereas the reflection of a matte rubber surface is insignificant.
- Reflection Map** — Simulates reflections in a surface using the reflection map instead of actually ray tracing the reflected image. This speeds up rendering time, but can also be a give-away if the scene is animated.
- Reflection-Illumination Model** — Model used when creating two-dimensional images from three-dimensional meshes. To produce more realistic and convincing images, the reflection model imitates attributes of real-life objects.
- Refraction** — When light passes through a transparent material and into a denser, or less dense, medium the light rays are refracted and change direction. Each material has its own refraction and, depending on the density of the material, the refraction is more or less evident. Refractions are calculated similarly to reflections using ray tracing.



**Refraction Index** — A value describing the amount of refraction that takes place in a specific transparent material. For vacuum the refraction index is 1.0000, for air 1.0003, for glass approximately 1.5 and for water 1.3333.

**Refraction Map** — An image to control the level of refraction across a surface where dark values indicate a low refractive index and bright ones a high refractive index.

**Render** — To mathematically generate geometries, algorithms, reflections, etc. Our work would be meaningless without the ability to render. Creating a final image of a model that shows all of the surface properties which have been applied to an object. This process involves adding all colors; bump maps; shading; and other elements that add realism. In a normal 3D program, you can view the wireframe of the created image. When an image is rendered, the wireframe is covered with the specified colors and properties.

**Render Pass** — Division of a scene according to different aspects (such as highlight, mattes, or shadows) for the purposes of applying specific rendering options. Passes can then be composited during post-production. The default pass is the beauty pass, which includes all objects in the scene. Preset passes include matte, shadow, and highlight passes. You can also define your own passes to include any object you want to be affected by specific rendering properties. Render passes are further divided into partitions.

**Rendering Pipeline** — Description given to the process of creating the rendered images. Some studios have a process by which all the images go through. Some render in passes, one for the base, then the shadows, then the reflections, etc. This process is the pipeline.

**Resolution** — The number of picture elements in an image.

**Revolution** — A Modeling term defining a surface made by rotating a curve around the axis of another curve.

**RGB Color Model** — A color model that mixes the three primary colors to produce colors. To create yellow, red and green are mixed without any blue component. The higher the value of the red, green and blue, the clearer the color. Lower RGB values give darker colors, while higher RGB values give lighter colors.

**Rigging** — The process of making an object ready for animation. This does not have to be just characters; it is the same for all objects. Rigging involves creation and implementation of bones, hierarchies, clamps, weight maps and sliders.

**Right-Handed Coordinate System** — A coordinate system (frequently used in 3D-graphics applications) whose positive Z-axis emerges from the screen towards you, just like the one used in mathematics, as opposed to the left-handed coordinate system.

**Roll** — The amount that a camera is tilted to the left or right. Also known as the Bank Angle.

**Rotoscoping** — A technique in which video or film images are placed in the background of a scene, one frame at a time. You can use these reference images to create your own animation by tracing objects from the images or matching your objects with the images' motion. You can zoom and pan the scene while maintaining a perfect registration with the imported background.

**Row Interleaved** — A format to create 3D video or images in which each row or line of video alternates between the left eye and the right eye (from top to bottom).

**Rule-Based Design** — The definition of relationships between objects in the design. Another name used to describe Knowledge-Based Design.

**Scalar** — A quantity, such as mass, length, or speed, that is completely specified by its magnitude and has no direction, a one dimensional value.

**Scanner** — Device for reading images (from books, photos etc.) into the computer. This is useful for creating realistic textures. With a 3D scanner it is even possible to capture three-dimensional objects and convert them into models.

**Scene** — A Scene is a LightWave project defining the objects loaded and their motions, the number of lights and their values/motions, the resolution of the final image, special effects, Camera settings, and so on. This ASCII text file is generally saved from Layout.

**Scrub** — The process of manually dragging the frame advance control slider on the timeline to see or hear its effect on video/audio.

**S-Drill** — Refers to Solid Drill. Acts just as a drill would, using a 3D object as the drill bit. This tool can be used to take sections out of objects or perform other functions that a drill might perform.

**Seamless** — A seamless texture can be tiled without visible transitions where the bitmap begins and ends. This means that the upper part of the bitmap can be placed next to the lower part of the bitmap, or the right can be placed next to the left, forming a pattern that is apparently coherent.

**SECAM** — Séquentiel Couleur à Mémoire. The television broadcast standard for France, the former USSR, and various eastern European countries. Like PAL, SECAM is based on a 50 Hz power system, but it uses a different encoding process and displays 819 lines interlaced at 50 fields per second. SECAM is not compatible with NTSC or PAL, but conversion between the standards is possible.

**Secondary Animation** — Once the main movements of animation have been applied, this refers to the detail animation step. Hoses bouncing when a robot walks and flab wiggling when a heavysset character moves are examples of secondary animation.

**Sector** — Convex volume used to speed up rendering time.

**Self-Shadow** — Object property that allows one part of a complex object to cast a shadow onto another part of that same object (example: the branches of a tree casting shadows onto its trunk).



- Self-Illumination (or Luminosity)** — Allows non-homogeneous self-illumination of the surface. Some parts can be self-illuminated, some partially self-illuminated, and some not at all, based on the pixel intensity of the self-illumination map (normally black=left unchanged, white=self-illuminated).
- Session** — A session is a single use of an application. A session begins when you first start the application and ends when you exit.
- Shaded Mode** — Shaded mode generally refers to a viewport that has its Rendering Style (Display Options panel or viewport title bar) set to something other than wireframe. These modes show polygon surfaces with some level of shading.
- Shading** — Simulating that an object is lit by a light source.
- Shadow** — An area that is not or is only partially lit because of the interception of light by an opaque object between the area and the light source.
- Shadow Map** — Bitmap generated by the rendering engine during a pre-render pass of the lit scene. Generally a shadow map is less precise than a raytraced shadow, but takes less time to render. As opposed to a ray-traced shadow, a shadow map can create shadows with smooth edges. Furthermore, the shadow map is unable to show the color cast by a transparent object. The quality of the shadows in the rendered image depends on the size of the shadow map. The bigger the map the nicer the shadows. A shadow map that is too small might result in aliased or stairstepped edges. For example, a 256x256 shadow map (65k) is normally sufficient for resolutions of 320x200 and less. If an object is far away from the light source, the shadow map will have to be increased in order to maintain visual quality. If the final rendering is in high-resolution, the shadow map also needs to be hi-res.
- Skew** — Modifying an object by tilting it.
- Skin** — Creating three-dimensional object from two or more two-dimensional shapes and then extruding them along a path.
- Smoothing** — Technique that, when rendering or shading, smoothes out the edges between segments making objects appear smoother than their geometry really is.
- Soft Shadow** — does not have hard edges. Traditionally, ray traced shadows always have hard, crisp edges, whereas shadow-mapped are considered soft shadows. With global illumination methods, physically accurate, soft-edged shadows are achievable at a cost in rendering time.
- Space** — A set of elements or points satisfying specified geometric postulates: non-Euclidean space. The infinite extension of the three-dimensional region in which all matter exists.
- Specular** — This property determines how shiny (and sometimes wet) an object appears. It represents the highlight that the light creates when shining on an object.
- Specular Component** — Part of the reflection-illumination model. Specular surfaces are capable of reflecting light like a mirror.
- Specular Map** — Replaces the specular component of the reflection-illumination model, thus only visible in an object's surface's highlights.
- Specular Reflection** — The brightest area on a surface, reflecting surrounding light sources, creating highlights.
- Spherical Image Map** — One of the many methods of applying a texture to a surface. This method applies the texture to an object as it would apply the texture to a sphere. There are many other methods of texturing objects, such as Cubic and Planar image mapping.
- Spinning Light Trick** — Trick to create soft shadows with ray tracing. It involves parenting multiple lights to a null and spinning the null.
- Spline (Curves)** — Layout uses splines or curved paths between keys while moving items about. When Modeling, splines refer to open or closed curves.
- Spline Cage** — A spline cage is usually a three-dimensional object made up of connected spline curves.
- Spline Patching** — The process of adding polygons to fill in areas outlined by splines.
- Spot** — A small opaque circle placed in front of a light, usually to remove the specular hot spot from an object.
- Spotlight** — A lightsource emanating light in one direction only, in the shape of a cone.
- Staircasing (or Stairstepping)** — A graphical flaw caused by insufficient resolution. When rendering an object its contours might stand out too crisply from the background and the pixels might be obviously "zig-zagged", or look like stairs. To prevent this, pixels can be blended into their neighbours' colors by antialiasing.
- Stencil** — When using the drill tool, the stencil option adds the details of the drilling polygon to the polygon being drilled. This creates new geometry on a shape.
- Stereoscopic 3D** — Two separate photographs taken from slightly different angles that, when compiled, appear three-dimensional.
- Stretch Tool** — Allows you to change the size of an object along a particular axis.
- Subdivide** — Divides any selected polygons with three or four sides into smaller polygons. This makes an object appear smoother, but also makes the model more complex.
- Subdivision Surfaces** — Subdivision surfaces are a technique to create a smooth curved surface from a coarse polygon mesh. Several different subdivision schemes have been developed since Ed Catmull and Jim Clark first introduced the idea back in 1978. The most well known schemes are the triangle-based Loop scheme and Catmull & Clark's original scheme, which is based on quad polygons. Subdivision surfaces were introduced to the public in the Pixar movies, *Toy Story 2* and *Geri's Game*.
- SubPatch** — Refers to a Modeling mode wherein polygons become a cage that controls an underlying mesh of subdivision surfaces.



**Subtractive Opacity** — Type of opacity that subtracts the background color from the transparent object's material color.

**Super Pixel** — They are created in a supersampling image. Groups of the super pixels are filtered into the one single pixel that is displayed on the output display.

**Supersampling** — Generating images at a resolution  $n$  times  $n$  larger than the display resolution and then filtering the so-called super pixels into the smaller resolution image, creating smooth images with antialiasing.

**Surface** — Essentially, the surface is the skin of an object. The surface attributes can be changed using the Surface panel. Many features, such as the name and color attributes, affect the appearance of an object. A single object can have multiple surface names, each with its own independent attributes (e.g., color), and multiple objects can share the same surface name(s).

**Tangent** — A straight line that makes contact with a single point along a curve.

**Taper** — Modifying an object by progressively narrowing it along an axis.

**Target** — In aiming the camera, the target is the object that is selected for the camera to point toward. The target is kept in the center of the camera's view.

**TD** — Technical Director. A job in a studio that mainly concerns making rigs, and to help out the other departments wherever possible. They are the problem solvers.

**Tessellation** — Increasing the detail level of a polygonal 3D model by increasing its number of polygons, usually triangles. The more triangles, the smoother the shape and subsequently the larger the model. The tessellation can be performed by dividing one triangle into two (or more) smaller ones. By doing this the new, more faceted model can be modified without losing too much of its smoothness.

**Texture** — Normally texture describes the attributes of a surface, for example if it's coarse, smooth, wrinkled or rough, but it is also used with the meaning of texture map. There are textures made from bitmaps (texture map), and textures generated mathematically (procedural map). Textures specify how the surface of an object will look, and can be anything from simple, solid colors to complex images representing the surface of the object. The simplest example of a texture is placing a picture on a flat plane. The picture is the texture being applied to the plane.

**Texture Coordinates** — Coordinates used to describe how to map a texture map onto an object. There are different kinds of techniques to apply the texture: planar, cylindrical, spherical, cubic, front, and UV. Their names indicate how the texture is projected onto the object the mapping coordinates are applied to. Procedural maps do not need texture coordinates.

**Texture Map** — Map wrapped over the surface of an object. The texture map needs to be spaced correctly in U and V direction over the object.

**Texture Mapping** — The process of projecting a (usually) two-dimensional image onto a three-dimensional face such as a triangle or a quad, texture mapping is a relatively cheap way of adding tremendous detail to a scene without resorting to extremely detailed meshes that take an inordinate amount of memory and time to render.

**Tiling** — Repeatedly placing the same texture next to itself on the same surface, creating a pattern from one image. This is achieved by increasing the texture coordinates on a polygon to a value greater than 1. Normally, the entire bitmap is tiled from 0.0 to 1.0 in  $u$ - ( $=x$ ) and  $v$  ( $=y$ ).

**Timeline** — The slider below the Layout viewport representing time in animation.

**Transformation** — The act or an instance of transforming. The state of being transformed. A marked change, as in appearance or character, usually, /hopefully/ for the better.

**Truck** — To move the camera in the viewing plane.

**Twist** — Modifying a mesh by rotating its vertices non-uniformly along an axis.

**U-Direction** — Represents a grid line in one direction (normally that of the original curve) of a UV texture map.

**Umbra** — A dark area, especially the blackest part of a shadow from which all light is cut off. The completely dark portion of the shadow cast by the earth, moon, or other body during an eclipse.

**Unify** — This command creates single-sided polygons according to the properties of their surface normals. Basically, this tool transforms polygons that share points into a single polygon.

**Union** — One of the options in the Boolean tool. This option makes an object that is a combination of the two objects.

**UV-grid** — A grid system for identifying points on a surface. The U-direction and V-direction are for the surface, what the X-axis and Y-axis are for the coordinate system.

**V Map** — V Map is an abbreviation for vertex maps. V Map provide additional information associated with object points (vertices), like weight, UV and morph maps.

**V-Direction** — Represents a grid line in one direction (normally "up-down") on the surface of an object.

**Vector** — Entity with both magnitude and direction. A three-dimensional vector is written as:  $V=(v1, v2, v3)$  where each component is a scalar.

**Vertex** — (pl. vertices) Point at which the sides of a polygon intersect. In 3D graphics, vertex may also refer to a point at a specified location in three-dimensional space, and such a point is the smallest component in a 3D mesh.

**Vertex Count** — The number of vertices in a scene. Remember, the higher the mesh complexity the longer the rendering time.





**Vertex Normal** — Even though it is a single point in three dimensional space, its normal can be calculated based on the normal of the face they are describing. The three vertex normals of a single triangle without any neighboring triangles are set to be the same as the polygon's normal. For triangles surrounded by other triangles, the vertex normals are the average of the surrounding face normals.

**View Frustum** — Representing the field of view of the camera, the view frustum is a pyramid volume with the top sheared off. The top of the pyramid represents the viewport of the camera (usually the screen), and is often called the near (or hither) plane, while the bottom is called the far (or yon) plane.

**View Frustum Culling** — Removing faces that lie outside the observer's view. Only the face that is within the view frustum is kept for rendering — speeding up rendering time and helping to maintain a high framerate.

**Viewport** — Window area displaying orthogonal or perspective projection in a 3D application. The screen can either contain one big viewport or several smaller, tiled viewports. By simultaneously using several viewports displaying a three-dimensional object from different sides (e.g. top, front, left, perspective), modeling in a virtual 3D environment is made possible.

**VIPER** — Versatile Interactive Preview Render. A window that provides you with an interactive previewing system.

**Volume** — When selecting, a volume of an object is a 3D representation of the area to be edited. When editing, all of the parts of objects contained within this 3D selection can be edited without changing what lies outside of the selection.

**Volumetric Fog** — Fog that, opposed to ordinary fog, is restricted to fit within a containing volume.

**Volumetric Light** — Light simulating illumination of particles floating in mid-air, thereby making the light cone itself visible.

**Vortex** — A tool that rotates an object more in the center than in the outer edge. This tool can be easily related to a tornado, where the wind in the center moves faster than the wind in the outer part of the cone.

**Voxel** — Short for VOLUME ELEMENT, this term refers to a specific rendering technique common in medical visualisation as well as some interactive media. In essence, a voxel is a three-dimensional pixel, that is, a cube, with a specific color.

**Weights** — The strength of influence on a particular vertex of an assigned deformer, such as a bone. See V Maps.

**Weld** — This command takes the selected points and combines them into one point, a single point that is specified by the last point that is selected.

**WIP** — Short for Work In Progress.

**Wireframe** — A way of visualising geometry by drawing lines between its vertices but not shading the surfaces within.

**World Coordinate System** — The coordinate system, normally in three dimensions, used to describe the location in space of a specific point called vertex.

**X-Axis** — Usually is the axis that is left-right/side-side.

**Yaw** — To turn about the vertical axis, also known as heading.

**Y-Axis** — Usually is the axis that is up-down/top-bottom.

**Y-up** — Coordinate system with the Y-axis pointing upwards.

**Z-Axis** — Usually is the axis that is in-out/front-back.

**Z-Buffer** — Also called depth buffer, the z-buffer is a two-dimensional matrix of 16 or 32-bit integers with the same dimensions as the screen (or viewport). Whenever a polygon is drawn to the screen, the rasteriser checks the corresponding z-buffer value for each screen coordinate and skips drawing the current pixel if the z value is marked as being closer. This allows for some nice effects such as overlapping 3D models, and completely solves the rendering-order problem.

However, this comes at the price of slower performance and greater memory usage, two factors that have become more or less moot with the proliferation of modern 3D accelerators that tend to support z-buffers in hardware.

**Z-Up** — Coordinate system with the Z-axis pointing upwards.







# Index

## A

Add Displacement 156  
Additive Expression 91  
Alert Level 384  
Alternate Aspect Overlay 387

## B

Backdrop, Layout 346  
    Environments 347  
    Gradient 346  
Background Booster 348  
Bezier Spline 89  
BioVision Motion Capture Support 265  
Bone  
    Weight Map 238  
Bone Properties Panel 237  
Bones  
    Bone 207  
    Bone Connect 217  
    Bone Edit Mode 206  
    Bone Fuse 218  
    Bone Rest Position 216  
    Bones On/Off 207  
    Bone Split 216  
    Child Bone 207  
    Clear All 237  
    Copy Hierarchy 218  
    Draw Bones 207  
    Draw Child Bones 207  
    Enable Deform 207  
    Falloff type 237  
    Faster Bones 237  
    From Object 237  
    Joint Compensation 239  
    Joint Comp for Parent 239  
    Limited Range 238  
    Morphed Positions 237  
    Multiply Strength 238  
    Muscle Flexing 239  
    Parental Muscle Flexing 239  
    Rest Length 238  
    Rest Position 238  
    Rest Rotation 238  
    Strength 238  
    Working With The Interface. *See* Skelegons  
Bounding Box Threshold 385  
Bump Displacement 156

## C

Camera 72  
    Mask 147  
Camera Basics 18  
    Anti-aliasing 27  
    Antialiasing:Depth Buffer 29  
    Antialiasing:Soft filter 29  
    Antialiasing:Using Edge Detection 29

Aperture Height 25  
CameraSelector 18  
Camera Settings in a Viewport 25  
Depth of Field 34  
Frame Aspect Ratio 25  
limited region 26  
Masking out a region 27  
Memory 26  
Motion Blur Effects 31  
Multiple Cameras 18  
Pixel Aspect Ratio 24  
Resolution 23  
Stereoscopic rendering 33  
Camera View Background 385  
Character Custom Offset Type 79, 280  
Classic Scene Editor 128  
Color Format 381  
Color Picker Selection. *See* Color Selection  
Connecting LScripts to Expressions 100  
Content Directory 392, 394  
    Relative Links 394  
Convert Skelegons. *See* Bones  
Coordinate System 273  
Creating a Preview Animation 194  
Cyclist  
    (item motion modifier) 228

## D

Default Unit 382  
Deform\  
    Bend 174  
    Pole 174  
    Shear 175  
    Taper 175  
    Twist 176  
    Vortex 176  
Deform Displacement Plug-ins 174  
Delete Bone 224  
    Parent Child Bones 224  
Depth of Field Display 146  
Depth of Field Preview 195  
Displacement Maps 152  
Displacement Texture 177  
Display Options, Layout 389  
Distributed Rendering\  
    Introduction. *See* ScreamerNet  
Dual-handled Control Points 89  
Dynamic Object 72  
Dynamic Update 385

## E

Effector 177  
    (item motion modifier) 228  
Effector (Displacement) 148  
Effector Object 228  
Expression Builder 92  
    Joystick Control 98  
Expression (channel motion modifier) 177  
Expressions  
    Applying an Expression 95  
    Built-in Functions 101



Expression Builder 96  
 Using Expressions 94  
 Expression Syntax 91

## F

Filter Option Panel  
   Channels Tab 113  
   Items Tab 112  
 Fixed Near Clip Distance 384  
 Frame Rate Meter 148  
 Frame Slider  
   Label 381  
 Frames Per Foot 382  
 Frames Per Second 381  
 FX\_Link 226  
 FX Metalink  
   Morph 166  
 FX\_MetaLink 166  
 FX\_Motion 227

## G

General options, Layout 273  
 Gimbal Lock 274  
 graph Editor  
   Pre and Post Behaviors 86  
 Graph Editor 75  
   Adjusting the Curve Edit Window 79  
   Channel Bin 76  
   Channel Bin Pop-up Menu 84  
   Channels Pop-up 76  
   Collapse buttons 75  
   Curve Edit Window Pop-up Menu 85  
   Curves Tab 86  
   Curve Translation 78, 279  
   Editing Color Channels 85  
   Edit Mode 77  
   Frame Range 75  
   Incoming Curves 87  
   Keyboard Shortcuts 82  
   Key Pop-up Menu 85  
   Menu Layout 82  
   Modifier Tab 103  
   Panel Layout Adjustments 76  
   TCB Adjustments, interactive 89  
   Time Slider 76  
   Toolbar 79  
   Zooming and Panning 79  
 Grid, Layout 13  
   Effect on Positioning 13  
   Relative Camera/Light Sizes 13  
 Ground Fog. *See* Fog, normal

## H

Hermite Spline 89  
 HyperVoxels 352  
   Baking 360  
   Drawing 147  
   Gradient Input Parameters 364  
   HyperTexture Tab 363  
   Particles 173  
   ParticleStorm Color 360

Setting Management 357  
 Shading Tab\ Sprite Mode 362  
 Shading Tab\ Surface Mode 360  
 Shading Tab\ Volume Mode 360  
 Show Particles 359  
 Sprite Clip Frame Offset 363  
 Sprites 358  
 Sprite Texture Resolution 357  
 Stretching and Rotating 358  
 Transparent Surfaces 352  
 Volume Mode Advanced Sub-tab 361

## I

IKB Boost Tool  
   IKB Menu 240  
 IKB Calculate 250  
 IK Booster 147  
 IK Booster Link 249  
   IK Booster Link Modifier Options 250  
 IK Booster Track 244  
 IK Boost Tool  
   IK Booster and the Dope Track 242  
   Motions 241  
 Image Filters. *See* Image Editor  
 Image World 347  
 Inertia 178  
 Input Device 380  
 Item Shape 149

## J

JointMorph 173  
 Joint Morph Plus 173

## K

Keyboard Shortcuts 379  
   assign a command 379  
   Finding Assignments 379  
   Mapping Sets 379  
   unassign a command 379  
 Keyframer 266  
 Keyframe Shifting 245  
 Keyframing. *See* Graph Editor  
   Automatically Creating 381  
   Creating 258  
   Creating and Modifying Keys Automatically 258  
   Deleting. *See* Graphic Editor

## L

Layout  
   Exporting Items 61  
   Load Commands 58  
 Layout Edit Drop Down Menu 378  
 Layout Grid Tools 397  
 Layout Items Tab 70  
   Load Scene 70  
 Layout Selection Tools 398  
   All Drop Down Menu 398  
   Order Drop Down Menu 398  
   Related Drop Down Menu 398  
 Layout Transport Controls 195



- Keyboard Shortcuts 195
- Lazy Points 181
- Lens Flares
  - OpenGL 387
- Level-of-Detail Mesh Refinement 149
- Level-of-Detail Object Replacement 143
- Libraries 91
- Light Properties 38
  - Basic Light Attributes 38
  - Global Illumination: Ambient Light 38
  - Lens Flare: Anamorphic Distort 51
  - Lens Flare: Central Ring 51
  - Lens Flare: Fade and Dissolve Options 50
  - Lens Flare: Glow Options 51
  - Lens Flare: Lens Flare Options 49
  - Lens Flare: Lens Flare Viewport Preview 50
  - Lens Flares 49
  - Lens Flare: Streaks 51
  - Lens Reflections 52
- Light Color and Intensity 46
- Light Exclusion 40
- Lighting Introduction 38
- Lights Display Size 39
- Light Types 41
  - Light Types: Distant Light 41
  - Light Types: Intensity Falloff 41
  - Light Types: Linear and Area Lights 43
  - Light Types: Point Light 41
  - Light Types: Spotlight 42
  - Light Types: Spotlight: Viewport Display 43
  - Light Types: Spotlight: Virtual Projector 42
- Saving Lights 41
- Shadow: Cache Shadow Map 47
- Shadow: Lightening Shadows 48
- Shadows 47
  - Shadow: Shadow Color 48
  - Shadow: Shadow Map Area 48
  - Shadow: Shadow Mapping 47
  - Shadow: Shadow Map Size 48
  - Shadow: Shadow Type 47
  - Shadow: Transparent Objects 47
- Volumetric Lights 53
  - Volumetric Lights: Light Textures 54
  - Volumetric Lights: Shadows within Lights 55
- Lights. *See* \
- Lightwave
  - LightWave Layout Interface 10
  - Virtual World 4
- LightWave Macro Recorder 137
- Limits 236
  - Minimum Joint Angles 236
- LScript 131, 231
  - LS Compiler 132
  - LScript Commander 132
  - LScript/RT 132
- LScript Expressions 93
- Luxigons 55
- LW\_ColrPikr 380
- LW\_HVRealFlow\_import\
  - Legacy Tool 177
- LW\_HyperVoxel\_Particles\
  - Legacy Tool 177

## M

- Manipulating Selections 124
- Math Morph 182
- MathMotion 233
- MD\_MetaPlug 182
- MD\_MetaPlug\_Morph 182
- MD\_Scan 182
- Measurement Unit System 382
- Metamotion
  - Metaform Shaper and MetaformTarget- Legacy Tools 183
- Meta-Primitive Objects 146
  - Display and Render Levels 146
- MipMap 390
- Mirror Hierarchy 220
- Mirror Tool 73
- MM (Motion Mixer) Channel Driver 106
- Mocap\_BVH\_Setup 60
- Modify
  - Transform 272
- Modify Tab 268
  - Translate 268
- Morph Gizmo Render
  - Legacy Tool 183
- Morph Mixer. *See* Endomorphs
- Morph List. *See* Endomorphs
- Morph Mixer Options 184
- Morph Slider Area 185
- Morph Targets. *See* Endomorphs
  - Multiple Target/Single Envelope 152
- Motion Baker 231
- Motion Blur 195
- Motion Capture Preview 150
- Motion Designer
  - Motion files, saving 261
- Motion Instancing 78, 280
- Motion Menu 282
- Motion Mixer 277
  - Actor Maps 283
  - Baking 283
  - Blending Motions with Transitions 280
  - Channel Editor 284
  - Editing Motions 283
  - Motion Properties 279
  - Offset Editor 284
  - Pre and Post Behavior Options 279
  - X Channels 285
- Motion Paths
  - Editing directly 259
- Motions
  - Motion Options Panel 224, 277, 283
- Move Keys 268
- Moving an item 270

## N

- NoisyChannel 106
- NormalDisplacement 178
- Normal Fog. *See* Volumetrics
- Null Object 71



## O

- Object Comments 148
- Object Displacement
  - Morph Mixer 183
- Object Properties 142
  - Deform Tab 152
  - Dynamics Tab 191
  - Edges Tab 189
  - Geometry Tab 142
  - Geometry Tab:Add Custom Object 142
  - Geometry Tab:Object Replacement 143
  - Lights Tab 191
  - Lights Tab:Exclusions 191
  - Panel Introduction 142
  - Render Tab 185
  - Render Tab:Alpha Channel 186
  - Render Tab:Clip Mapping 185
  - Render Tab:Distance Dissolve 186
  - Render Tab:Matte Object 186
  - Render Tab:Object Dissolve 186
  - Render Tab:Unaffected By Fog 187
  - Render Tab:Unseen By Rays 187
- ObjectSequence 143
- Object Shadow Options 188
- Objects, Layout
  - Properties 146
- ObjList 143
- Oscillator
  - (item motion modifier) 231
- Oscillator, item motion modifier 105
- Overlay Color 389

## P

- Parenter 235
- Pivot Point 268
  - Moving the Pivot Point 268
  - Rotate Pivot 271
- Plugins 133
  - Add Plug-ins 133
  - Edit Plug-ins 133
  - Flush Plug-ins 133
  - Last Plug-in 133
  - Master Plug-ins 134
- Print Assistant 24
- Protractor 150
- Proxy Pick 137

## R

- Range Finder 150
- Relative XZ Custom Offset Type 79, 280
- Render Options
  - Enable Viper 197
- Replacing Scene Items 74
- Resetting tool 274
- RIG
  - Export RIG 223
  - Import RIG 221
- Rotate an item 271
- Ruler 150

## S

- Saslite 183
- Sasquatch Lite 156
  - SasLite Options Dialog Box 158
- Scene Editor 107
  - Dope Sheet 122
  - General Rules 110
  - Hierarchy 109
  - Item List 107
  - Keyboard Shortcuts 125
  - Playback Controls 125
  - Selection 111
  - Selection Sets 113
  - Sorting 110
  - Type and Color 108
- Scene Editor Options 125
  - Dope Sheet Options 126
  - General Options 125
  - Properties Options 126
- Scenes
  - Navigating 257
  - Playing 257
- Schematic View 14, 388
  - View Tools 395
- Serpent 179
- SetDrivenKey 106
- Setup Tab 206
- Shadows
  - HyperVoxels 188
- Shockwave 3D 65
- ShowCurve 151
- Show Field Chart 387
- Show Safe Areas 387
- Simple Constraints 234
  - Adding Constraints 234
  - Deleting Constraints 235
- Skelegons. *See* Setup Tab
  - Bone Twist 213
  - Modify 209
  - Single Bone 212
  - Tip Mover 212
- Sky Baker Panel 373
- Skygen 374,
- SkyTracer 366
- SkyTracer2 366
  - Atmosphere Panel 367
  - Clouds Panel 368
  - Suns Panel 372
  - Texture Editor Panel 371
- Sliders 151, 274
  - configure your sliders 275
  - Working with OpenGL Sliders 275
- Sock Monkey 180
  - Basic Setup 180
- SockMonkey 150
- Speedometer 150
- Spline Control 183, 275
  - Spline Control Options 276
- Spread Sheet Property Panel 116
- Spread Sheet Scene Manager 137
  - Workspaces 137



- Steamy Particles
  - Legacy Tool 183
- SubPatch 144
  - Display and Render Levels 145
  - Using SubPatch 144
- Surface Filter Options Panel 115
  - Channels Tab 115
  - Surfaces Tab 115
- Surface List Display 114
- Surface List View 113

## T

- TCB Spline 87
- TCB Tool 251
- TextureChannel 106
- Textured Environment 347
- TextureMotion 233
- TimeWarp Curves 78, 280
- Toolbar, Layout 384
- Tools
  - IKB Boost Tool 239
- Track Area 281
- Trail
  - Particle Trail 183

## U

- UnParent Bone 218
- Unseen By Camera 187
- Utilities Tab 131
  - Commands 131

## V

- View Layout 397
- View Mode 396
- Viewport, Layout
  - Controllers. *See* \
- View Tab 396
  - Viewports 396
  - Viewports:Fit All 396
  - Viewports:Fit Selected 396
  - Viewport:Zoom In and Zoom Out in Layout 396
- VIPER 197
  - HyperVoxels 354
  - Other Preview modes 198
  - Preview Options 198
  - Surface Preview Mode 197
  - Window and Presets 55
- Visual Browser 380
- Volumetric Lights
  - Background 348
- Volumetrics. *See* Fog, GroundFog, and HyperVoxels
  - Particles 349
  - Plug-ins 350
- VRML97 Custom Object 151
- VRML Implementation 63

## W

- Weight Curve 78, 279
- Weight Normalization 238
- Workspace 125
- World Coordinates 228

## Z

- Z-Buffer 357







