LightWave 3D 7

NewTek
5131 Beckwith
San Antonio, TX 78249

Manual version: 1.1

Software License and Limited Warranty

PLEASE READ CAREFULLY BEFORE INSTALLING THIS SOFTWARE. BY INSTALLING 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 FOR A FULL REFUND.

1. Grant of License

The enclosed computer program(s) (the "Software") is licensed, not sold, to you by 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 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, or in an electronic format via BBS, ftp, e-mail, 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, 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 "© 2001 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 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 AS TO THE SOFTWARE AND ACCOMPANYING WRITTEN MATERIALS AND LIMITED TO 90 DAYS AS TO THE MEDIA. 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. If you are a US Government end-user, this License of the Software conveys only "RESTRICTED RIGHTS," and its use, disclosure, and duplication are subject to Federal Acquisition Regulations, 52.227-7013 (c)(1)(ii). (See the US Government Restricted provision below.)

7. Trademarks

LightWave 3D, LightWave, Video Toaster and 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) (l) (ii) of the Rights in Technical Data and Computer Software clause at 52.227-7013. Manufacturer: NewTek, 5131 Beckwith, 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.

Credits and Acknowledgements

Lead Programmers

Allen Hastings
Stuart Ferguson

Senior Programmers

Arnie Cachelin
Matt Craig
Gregory Duquesne
Jamie Finch
Daisuke Ino
Ryan Mapes
Ernie Wright

Additional Programming

Joe Angell
Christian Aubert
Neil Barnes
Robert Gougher
James Jones
Marvin Landis
Mike Reed
Jon Tindall
Steve Worley

Build and Integrations Management

Jason Craig

Product Design and Testing

Kenneth Woodruff
Rich Hurrey
Aristomenis "Meni" Tsirbas

Installer

Kris Debolt

Documentation

Douglas J. Nakakihara
Stephanie Barton
Brian Marshall

Product Management

Art Howe

Product Marketing

Rob "Bubba" Hoffmann
Kriss Schreiner
David Tracy

Special Thanks to:

Tim Jenison, Brad Peebler, Jason Linhart, Robin Hastings, Sandi Spires, John Gross, Paul Davies, Michael Sherak, Taron, Terrence Walker, Christian Aubert, Sébastien Hudon, Frédéric Hébert, Michel Langlois, Lee Stranahan, John Teska, Danny Braet, Fori Owurowa, Shaan Pruden, David Herrington, Paul Debevec, Apple Computers, Hollie Wendt, James Jones, DStorm, Yoshiaki Tazaki, Andy Frerking, David Warner, the students of 3D Exchange, and, of course, all the Beta Team members. A very special thank you to Jacqueline Nakakihara, Taylor Nakakihara, Lexy, the Palm, the Minstrel, VB6, the Net, Bruce Winter and Misterhouse, Josie and the PCs, CZ Jones, J Alba, dougworld.com, Taco Taco, ICQ and many more—you know who you are. Also, you can't learn English by reading a dictionary.

All rendering speed tests rated in cowmarks (based on the average unladen cow). No cows were permanently harmed in the creation of this product—with the exception of one contumacious individual—although I'm sure the Twist and Pole Evenly tools hurt a bit. Cow sequences, featuring a herd of thousands (Texas longhorns), choreographed by someone who should know better than to morph a cow into a sphere. Any similarities between the cow object and someone you know is a tragic coincidence, we would hope.

TABLE OF CONTENTS

Chapter 1: Introduction

Chapter 2: Conventions

Chapter 3: Common Interface Items

Chapter 6: Layout: General Functions

Chapter 7: Objects in Layout

Chapter 8: Keyframing

Chapter 9: Object Properties

Chapter 11: Item Motion Options

Chapter 12: The Scene Editor

Chapter 13: Camera Basics

Chapter 14: Backdrops and Image Processing

Chapter 15: Volumetrics

Chapter 16: Rendering Options

Chapter 17: Particle FX

Chapter 18: Motion Designer

Chapter 19: Distributed Rendering

Chapter 20: LightWave 3D Modeling

Chapter 21: Points and Polygons

Chapter 22: Creating Geometry

Chapter 23: Modifying Geometry

Chapter 24: Multiplying Geometry

Chapter 26: Detailing Geometry

Chapter 27: Display Menu

Chapter 28: Vertex Maps

Chapter 29: Organic Modeling Tools

Chapter 30: Image Editor

Chapter 31: Surface Editor

Chapter 32: Shadow and Light

Appendix

Index

chapter 1
Introduction

Chapter 1: Introduction

Congratulations on your acquisition of LightWave 3D, a truly powerful piece of software. This program represents over 12 years of research and development from a team of insanely dedicated programmers, artists and visionaries.

About the Manual

The opening chapters of this manual give you an overview of LightWave 3D, and cover the functions that are common to both the animation and modeling aspects of LightWave. Following that, there are chapters that cover how to move your objects, how to set lighting and render options, and other information on how to create your animations. The final set of chapters go over how to build your 3-D objects, apply surface textures, add lighting to your scene, and more.

LightWave 3D is generally divided into two separate environments: LightWave Layout and LightWave Modeler. (It is a common convention to refer to the animation and rendering toolset as *LightWave*, or just *Layout*, and to refer to the object creation toolset as *Modeler*.)

For Layout, the buttons on the toolbar are often shortcuts to controls that can also be accessed from one of Layout's many panels. The manual generally takes a panel and topic-based approach to explaining Layout, since many controls for certain tasks are grouped together on panels.

Modeler is a little different. All of its tools are accessed from the toolbar, and panels are used to execute more focused tasks. As such, the Modeler portion of the manual is generally organized by the order of tools in the toolbar.

LightWave Overview

With LightWave, you are the producer, art director, cinematographer and director, all rolled into one. Your actors consist of objects that you build in Modeler or import from another program. You place lights to illuminate them, assign surface colors and values to *clothe* them, and then choreograph their movement to animate them. Once you have your actors in place, a camera records all of their movements, creating images of their movements that you can compile into animations.

LightWave can create animations and still images for virtually any use, from tiny 16 x 16 pixel Web graphics, all the way up to a resolution of 16,000 x 16,000 pixels, and anywhere in between. A multitude of preset resolutions are also provided, including industry standard D1 and D2 (NTSC and PAL) video resolutions.

You create animations by generating a single frame at a time, then recording each frame to some type of playback device, like videotape or a hard-disk playback device, or even film. Frames can also be saved in computer animation format, like AVI.

HARDWARE LOCK INSTALLATION

If you have a USB hardware lock, you can install it with your machine running. If you have a Mac ADB hardware lock (from a prior version), install it with your machine turned off.

Parallel Port Lock (PC)

1. Turn off your computer.
2. Remove any peripherals from the parallel port on which you want to install the hardware lock. If possible, install the hardware lock on an unused parallel port.
3. Plug the hardware lock into the parallel port.
4. Plug any removed peripherals into the hardware lock.
5. Restart your system.

UPGRADING FROM A PRIOR VERSION

If you already have a version of LightWave 3D on your computer, you may follow the same installation instructions as new users. The installation will not delete any objects, scenes, images, etc. unless a file on your system has the same name as a file installed by the installer in the same subdirectory, in which case it will be overwritten.

WARNING
We strongly recommend you backup your system before you install LightWave 3D.

SOFTWARE INSTALLATION

To install LightWave 3D on your system, insert the LightWave 3D CD-ROM in the appropriate drive. On the PC, run SETUP.EXE and on the Mac, run the LightWave installer program.

Follow the instructions for installation as presented. You will be given options for what to install. A full installation includes optional “content” data, such as objects, scenes, and images that you can install according to your interests and available hard drive space—you can delete these files later if you need to. Note that content data is not required to run LightWave 3D; however, some files will be required for many of the exercises and tutorials in the manual.

When the installation is complete, reboot your computer.

REGISTERING YOUR SOFTWARE

Initially, LightWave will only run a certain number of days until you register the software. Run LightWave to get your hardware lock serial number—a dialog will appear telling you this information. Write this number down.

A registration folder should have been installed in the LightWave start-up menu group. There will be links to a form that must be completed and sent to NewTek. You must provide your hardware lock serial number.

Registration is important since it allows you to receive technical support, upgrades, special offers, new product information, and your permanent license key.

Contact Information for Registration

1-800-TOASTER (862-7837)
International: 1-210-370-8000
Fax: 1-210-370-8030

Hours (Central time): Mon-Thur: 9:00 AM to 6:00 PM, Friday: 9:00 AM to 5:30 PM

On-line registration: www.newtek.com
E-mail: cs@newtek.com

Mail:

NewTek Inc.
Attn: Customer Service
5131 Beckwith Blvd.
San Antonio, TX 78249

Licensing Your Software

Once you obtain your licensing code number, edit the LICENSE.KEY file using a text editor (e.g., Notepad (PC), Simple Text (MacOS 9.x), Text Edit (MacOS X), etc.) and add your code. LICENSE.KEY should be located in your PROGRAMS\LIGHTWAVE_SUPPORT directory. You can have multiple code numbers in the file, but each one needs to be on its own line.

NOTE
Make sure you save LICENSE.KEY as a plain text file, not RTF or other file format.

RUNNING THE PROGRAM

LightWave 3D is started the same way as any other program on the particular platform on which it's installed. Start LightWave (to start Layout) or Modeler by double-clicking their respective icons, or by clicking their icons and selecting Open from the menu. Each program can be started independently; however, you may also access them by clicking the appropriate button on their respective interfaces.

OPTIMIZING RAM USAGE

If your computer is accessing virtual memory frequently during rendering, you may find a substantial increase in performance by installing more RAM on your machine or reducing the amount of RAM LightWave needs to store the render information.

One of the best ways to minimize the need for RAM is to reduce the color of the images used for texture mapping effects. Except for those used as a Surface Color, texture maps usually need to be only 256-level gray-scale images.

Other ways to reduce RAM usage are to render your animation in multiple passes or decrease **Segment Memory Limit** on Layout's Camera Properties panel.

LIGHTWAVE 3D RESOURCES

Internet Resources

A LightWave-specific newsgroup and mailing list are maintained on the Internet. Here you can find new users asking questions about using LightWave and experts answering them. Also, many topics related to computer animation are discussed in these groups, such as the performance of various accelerators, CPU speeds, animation recording devices, and many more. The newsgroup's Internet address is: comp.graphics.apps.lightwave.

Internet mailing list information can be found at www.tv3d.com.

NewTek Web and FTP Sites

In addition to information about NewTek products, upgrades, and the latest releases of LightWave software, our World Wide Web sites (www.LightWave3D.com and www.newtek.com) have tutorials, LightWave images and animations, technical support FAQs, tech support e-mail link, and links to related sites.

NewTek maintains an FTP site (ftp.newtek.com) on the Internet. Here you can find objects, scene files, images, and other items of interest to LightWave users.

Community

A unique attribute of LightWave 3D is the *LightWave community*. It is rare to find such a fun-loving bunch of hooligans that are so talented and yet so willing to share their secrets, ideas, and creations. Make certain you also take a look into the available resources such as local users groups, training videos (www.DesktopImages.com), and NewTek specific magazines (www.keyframemag.com and www.newtekpro.com).

Technical Support

The best source for help with installing or configuring software or hardware is the retailer from whom you purchased your NewTek product. While we have made every effort to keep your software and hardware trouble-free and easy to use, you may occasionally need help right from the source. If you have problems with NewTek supplied hardware or LightWave doesn’t seem to be functioning as it should, please contact technical support in one of the following ways:

- By email: tech@newtek.com
- By fax: (210) 370-8030
- By telephone: (210) 341-8444. Technicians are available to answer questions from 8:00 a.m. to 8:00 p.m. Central Time, Monday through Thursday, and Friday from 8:00 a.m. to 5:30 p.m. Central Time.

Please supply in your communication or have the following information handy when calling:

- Your computer’s operating system and version
- The version of LightWave you are using
- The amount of RAM in your computer
- Any relevant specifics about your system (display card type, memory managers, accelerator type, etc.)
- Your product serial number

NOTE
Your product must be registered before you can receive support.

chapter 2
Conventions

Chapter 2: Conventions

TYPOGRAPHIC CONVENTIONS

The following conventions will be used throughout this manual.

Directory Structure

Except for a few system files, all of the software will be installed in a single subdirectory on the drive you specified during the installation process. Therefore, subdirectories referred to in this documentation are located in this main subdirectory. (For example, if the discussion refers to the Images drawer, the actual path might be C:\LIGHTWAVE\IMAGES.)

Typefaces

ALL CAPS	Computer keys, directories, device names (e.g., ENTER, C:\LIGHTWAVE\OBJECTS, CTRL+P, etc.).
Bold	Names of interface menus, fields, buttons, etc. are set in bold type.

Keystroke Combinations

KEY1 + KEY2	Simultaneous keystrokes. Hold the first key and press the second key.

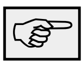
Mouse Operations

LMB	Left mouse button
MMB	Middle mouse button (if applicable)
RMB	Right mouse button
Selecting	Single-clicking an element with the LMB so that it becomes active or selected.
Deselecting	Single-clicking an element with the LMB so that it becomes inactive or unselected.
Activating	Selecting an option by clicking on its toggle button.

Deactivating	Unselecting an option by clicking on its toggle button.
Clicking	Placing your mouse pointer over something and then pressing a mouse button. This nearly always means the LMB.
Right-clicking	Clicking an element with your RMB.
Double-clicking	Rapidly clicking an element twice.
Dragging	Selecting an element with your mouse pointer and continuing to hold the mouse button down as you move your mouse. This nearly always means with the LMB.

Attenti-cons

The warning symbol will highlight a discussion that warns the user about something. You should pay special attention to text marked with this symbol.



The note symbol will highlight a discussion that is particularly noteworthy.

The hint symbol will highlight tips and suggestions that are usually of a time-saving nature.

KEY LIGHTWAVE TERMS & CONCEPTS

The following are some common terms and concepts used with LightWave that you should familiarize yourself with:

Alpha matte/image	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 grayscale or black-and-white images, but the brightness values can also be extracted from a color image.
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 *envelopeable* elements like light intensity. See also *motion channel*.
Axis	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.
Bone	For any object, you can define a skeletal system composed of bones. By moving bones around, you can change the shape of an object.

Bounding Box	A Bounding Box is a six-sided box conforming to the outer dimensions of an object. It is commonly used as a quickly drawn stand-in for a more complex object.
Camera	As in real life, a camera records events. Looking through LightWave's camera shows you the view as it will be generated.
Cattiwompus	Weird. Mixed up. Unusual. Distorted.
Channel	See *Animation Channel*.
Child	See *Parenting*.
Control Mesh	A Control Mesh is a cage of points used to shape SubPatches.
Default unit	The Default unit is the unit of measure (e.g., 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.
Endomorph	An EndoMorph is an object containing one or more morph maps.
Envelope	An envelope is a way of setting a particular value that usually changes over time using a graphical input mode.
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 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, a .5 percent flatness means that no point will be outside the ideal plane of the polygon by more than 5 millimeters. (1 x .005)
Frames	LightWave works with frames. A frame is one image out of many that define an animation. There are approximately 30 frames per second in NTSC video, 25 frames per second in PAL video, and 24 frames per second in film.

GCore — Short for Geometry Core, the *engine* that sits underneath all animation and modeling tools.

Geometry — Geometry refers to the positional layout of points and polygons for an object.

High Dynamic Range Image — HDRI is an image with a very large intensity difference between the brightest and darkest pixels. In typical 8/24-bit images, the maximum possible intensity range is 255 times brighter than the darkest gray 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.

Hub, the — The Hub is a *message board* that LightWave modules use to synchronize information.

HyperVoxel — HyperVoxels are volumetric rendering effects.

Item — An item in Layout refers to an object, bone, light, or camera.

Intelligentities — Intelligentities refer to LightWave's object format.

Keyframe — (Also known as just a *key*.) A keyframe is 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*.

Light — A light in LightWave is generally used just like a light in real life. Lights illuminate a scene and allow you to see the objects within it.

LScript — This is LightWave's built-in scripting language. LScripts can be installed and used just like plug-ins.

Meta-primitive — A Metaball, Metaedge or Metaface object.

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.
Motion Channel	Motion Channel is generally the same as *Animation Channel*, but refers only to position, rotation, and scale. (i.e., not light intensity.)
Non-planar	Non-planar refers generally 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.
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.
Null Object	A null object is an object that contains no geometry and will not show up in a rendered image. Nulls are useful for tracking, grouping (parenting), and using with inverse kinematics functions. They are also commonly used to control plug-ins.
NURBS	NURBS are Non-uniform Rational B-Splines.
Object	An object is composed of points and faces. Points connected together to form a polygon define a face. Faces joined together form an object.
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	Orthogonal usually refers to a display that has only two axes at right angles from each other in the same flat plane.
Parenting	Parenting refers to LightWave's ability to set hierarchical associations between items in a Scene. Generally, the parent item will have some

	level of influence on its *child*, whether it is position, rotation, size, etc. The child can also be a parent to another item.
Particles	Usually refers to animated points used for sparks, fire, and even liquid. LightWave's particle engine is called Particle FX.
Phong Shading	Phong Shading is a computer process whereby faceted object faces are shaded and rendered with a smoothed over appearance. LightWave uses a Phong shading algorithm in its rendering calculations.
Pivot Point	The position in an object that acts as the center of rotation and position reference.
Pixel	A pixel is the smallest unit of measurement in an image and is used to describe the image's width and height.
Plane	A plane refers to a two-dimensional (i.e., flat and level) surface. You might want to think of a plane as a piece of glass that is infinitely large, but has no depth.
Plug-in	A plug-in is a program that works with and extends the functionality of LightWave.
POV	POV refers to Point of view.
Radiosity	Radiosity refers to the reflection of light off of diffuse surfaces.
Ray Tracing	Ray tracing is a process whereby a "ray" from a light source is followed as it interacts with objects in a scene and travels into the camera. Ray tracing allows for such realistic events as shadows, refraction, and reflection.
Render	Rendering is the computer's process of calculating and generating an image based on the values you have selected for the different options in LightWave.
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	Scrubbing usually refers to the action of dragging a slider and seeing/listening to its effect on video/audio.

Session	A session is a single use of an application. A session starts when you first boot 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 titlebar) set to something other than wireframe. These modes show polygon surfaces with some level of shading.
Spline (Curves)	LightWave 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	Spline patching is the process of adding polygons to fill in areas outlined by splines.
Subdivided	Subdivided generally refers to the result of increasing the number of polygons by dividing existing ones. The overall shape of the subdivided polygons may or may not change depending on the method of subdivision.
SubPatch	SubPatch is a modeling mode where polygons become a cage that controls an underlying mesh of NURBS.
Surface	Essentially, the surface is the skin 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 tangent is a straight line that makes contact with a single point along a curve.
Texture	A texture is an attribute of a defined surface, or an entire object, in the case of displacement and clip maps.
Tween	Tween is the internal process of calculating the animation channel values between keys.
Vertex	A vertex is the point at which the sides of a polygon intersect.
V (Vertex) Maps	V Maps is an abbreviation for vertex maps. V Maps provide additional information associated with object points (vertices), like weight, UV and morph maps.
VIPER	Versatile Interactive Preview Render window that provides the user with an interactive previewing system.

Volumetric Lights	Volumetric Lights offer a special light property that allows you to see the beam of light.
Weights	See *V Maps*.
Wacky	Crazy. Zany. Mixed-up. Cool.

WORKING WITH THE INTERFACE

In no time, you'll find yourself mastering LightWave's intuitive interface. LightWave uses very few picture-icons that may confuse you. Instead, most functions are listed on the interface panels in plain text. Listed below are a few other conventions involved with the LightWave interface:

Button	A button refers to an area on the screen that you click on with your mouse to cause some function to occur. Generally, only a single click is required. There are also special types of buttons, like toggle, pop-up menu, and envelope buttons. Some buttons become highlighted, indicating a chosen or active status.
Contextual Pop-ups	This refers to *context-sensitive* pop-up menus. Such menus appear when you SHIFT+CTRL+LMB and are aware of the area the mouse pointer is over.
Dialog	See *Requester*
Drag Button	A drag button is similar in effect to a slider, except the button does not move. To use, just click on it and hold the mouse button, then drag the mouse. Depending on the parameter being adjusted, the direction and mouse button used will have different specific effects.
Envelope Button	An envelope button is a small button marked with an E. Selecting this button will display the Graph Editor where you can create an envelope for the setting. A highlighted envelope button indicates a value has an envelope applied. To turn off (i.e., remove) an envelope, hold the SHIFT key and click on the Envelope button.
Ghosted Item	LightWave will *ghost out* parameters that are not available to you. This is usually the result of a certain option not being activated. Selecting a ghosted item will display a message informing you why it cannot be used.
Information Field/Display	Information Fields/Displays are text displays

found throughout the different panels. These displays cannot be changed directly and simply provide information and feedback.

Input Field — Input fields are areas on the screen where you can enter data.

Mini-Slider — A mini-slider is a button with two arrows on it, pointing right and left. Clicking on one and holding the LMB, then dragging to the right or left will raise or lower the value of the parameter next to it. In many cases, however, the slider does not encompass the entire spectrum of possible values.

Panels — Panels are any one of the windows that open when you click on a button in LightWave. Many panels have additional tabbed sections that are selected when you click on a tab.

Pop-up Menu — Pop-up menu buttons have a downward facing arrow on their right edge. To use, click on it and hold the LMB. The menu will pop up and as you move your pointer over the menu, each item will become highlighted. When the desired selection is highlighted, release the LMB.

If you decide not to select an item, simply move the pointer off the menu and release the LMB.

Certain pop-up menus contain lists of objects, images, and lights, which are normally listed in the order they were loaded or created.

Requester — A requester is also known as a dialog box. These appear on the screen for operations like file loading and saving. This also refers to smaller windows that appear requesting the user input data into various fields.

Reset Area — The reset area is a non-active open area (e.g., not a button) on the Modeler toolbar that acts as a reset button, much like you might use the ESC key on other applications.

Scrollbar — See Slider

Shaded Display — See *Solid-Shaded Display.*

Slider — A slider allows you to modify a setting by dragging the slider's button along the bar. Alternatively, you can click to the right or left of the button or use the arrow buttons at either end to incrementally change the setting value. (Also known as a scrollbar.)

Solid Shaded Display	Solid shaded display refers to a non-wireframe display mode where some level of surface texture detail is visible.
Texture Button	A texture button is a small button marked with a T. Selecting this button will present a texture panel allowing you to define a texture for the chosen parameter. A highlighted texture button indicates a texture in use for the given parameter. To turn off (i.e., remove) a texture, hold the SHIFT key and click on the Texture button.
Toggle Button	A toggle button is a small button that becomes highlighted with a check mark when clicked. This indicates the adjacent feature is active.

chapter 3

Common Interface Items

Chapter 3: Common Interface Items

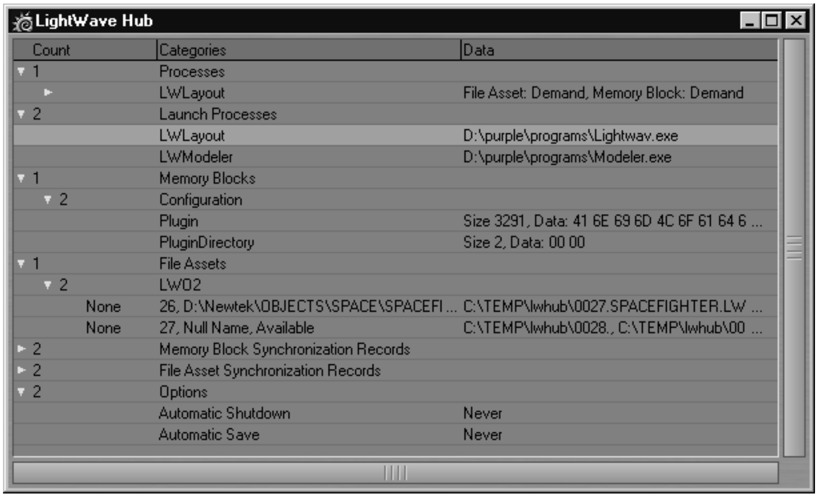
This chapter covers common interface items as well as panels and dialogs that are shared between modules, panels, plug-ins, and so on.

THE HUB

The Hub is essentially a *message board* that LightWave modules use to synchronize information. It contains entries like *synced* object filenames and configuration memory blocks. When the same object is loaded into both Layout and Modeler, changes made to the object are automatically synchronized. If the object appears in Layout, but not Modeler, you can quickly load it into Modeler by selecting it from the object pop-up menu—initially, the name will be ghosted.

NOTE
If you have modified an object in Modeler (without saving) and then load the object file into Layout, the modified version will appear—since it is synced—not the data from the object file.

The Hub runs automatically when you run LightWave. Basically, it is a background process and you shouldn't need to directly interact with it. However, you can bring up its interface by clicking its icon (when running).



The Hub interface window

If you expand the Launch Processes menu, you should see applications that the HUB is aware of (e.g., LWLayout, LWModeler). You can launch the application by double-clicking it. If you single-click (to select it) and then press the L key, you will also launch the application. If you press the DELETE key instead, you will delete the application from the launch list. (This has no effect on the program files on your hard drive.)

NOTE
The Launch Processes menu will be empty initially. However, once you run Layout or Modeler, entries will be added.

Under Windows, the Hub icon will appear in the System tray. You can right-click on it to access a small menu:

Open	Open Hub window
Close	Close Hub window
Launch	Launch applications that the Hub knows about
Properties	Set options to quit the Hub after you exit LightWave
Exit	Close the Hub program

Properties

You can set the **Automatic Shutdown** to various time intervals, after which the Hub will shutdown when there is no activity. Note that even when Layout and Modeler are idle, there is still some minimal communication activity which will keep the Hub from shutting down.

When Layout or Modeler are running, the Hub will periodically request that the applications write their recent changes to temporary files. The frequency of this **Automatic Save** is set by the Hub option. The temporary files are located in a LWHUB directory inside your normal system temporary directory. The files have names that include the base name of the object with a numeric prefix. So, if you are editing an object called MYOBJECT.LWO and you crash after an editing session, you can still find an *auto-saved* version of your edits in the temporary directory with a name like 0012.MYOBJECT.LWO.

NOTE
If you would like to run Layout and/or Modeler independently, append -0 to the command (e.g., Lightwav.exe -0. That's a zero, by the way.) You will have to load and save objects manually.

THE IMAGE VIEWER

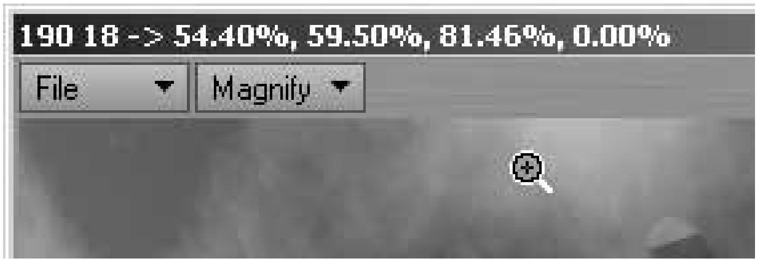
The **Image Viewer** is used throughout LightWave (Render Display, Image Editor, etc.) to show an image using colors up to the capabilities of your computer's display. Once open, you do not need to close the Image Viewer window. In fact, if you do, any unsaved images may be lost.

From Render Display (Render Options panel)

The Image Viewer can sometimes hold multiple images, depending on how it is used by LightWave. In such a case, you can select those images using the **Layer** pop-up menu.

You can choose to see the regular image or the alpha image, if applicable, using the pop-up menu in the upper-right corner.

If you drag your mouse pointer over the image, information about the pixel beneath the pointer will appear in the titlebar. The first set of values is the X and Y position. This is followed by RGBA color information where 100% equals 255 for non-floating-point images. For floating-point images, 100% equals 1.



Titlebar pixel information

You can adjust the size of the window by dragging the lower-right corner. Anytime the entire image is not fully visible in the viewer window, hold the ALT key to scroll around.

The File Menu

Using the **File** pop-up menu, you can save the current image to a file. Once you select the format type, a file dialog will appear. Make sure you add the appropriate filename extension, if necessary (e.g., PICTURE.TGA). The commands to clear layers are accessed from this menu.

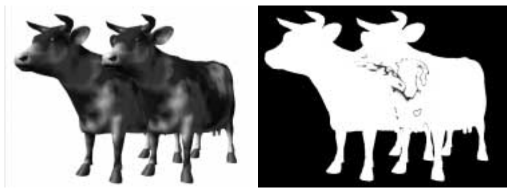
Image Control Panel

A **Image Controls** menu item also appears on the **File** menu.

Image Controls

On the Controls dialog, you can select a foreground image by clicking in the **FG** column. You can also blend a foreground and background image in real-time using the **Blend Mode** pop-up menu and then dragging the **Blend** slider.

Left to right: Horizontal, Vertical and Dissolve blend



Left to right: Alpha and Difference blend

Most of the modes are self-explanatory. The **Difference** mode shows the *difference* between the foreground and background pixels (i.e., foreground minus background). If the **Blend** slider is all the way to the left, the actual difference is shown. Increasing the **Blend** *scales up* the difference to make it more noticeable. If the foreground and background pixels are very close, the result looks black.

NOTE
The **Blend** slider will have no effect when using the **Alpha** blend mode.

If you are displaying a high dynamic range image, you can access special parameters if you activate the **Exposure** option. This allows you to adjust HDR data much like the HDRExpose image filter, but in real-time.

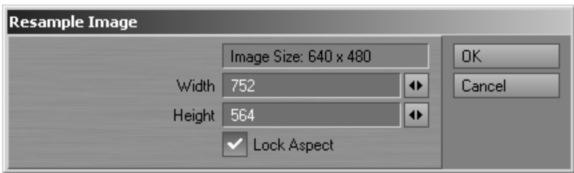
The **White Point** is the input intensity considered to be the hottest white. Anything above this will be the same white in the output. The **Black Point**, expressed as a percentage of the nominal black point (1/255), is the darkest non-black pixel level in the input image that will be preserved. Anything darker will come out black. This control is overridden by the **Auto** option, which uses the image data to compute a black point.

NOTE
When the Image Viewer is the active window, pressing the P key will display/hide the Control panel.

File Saving

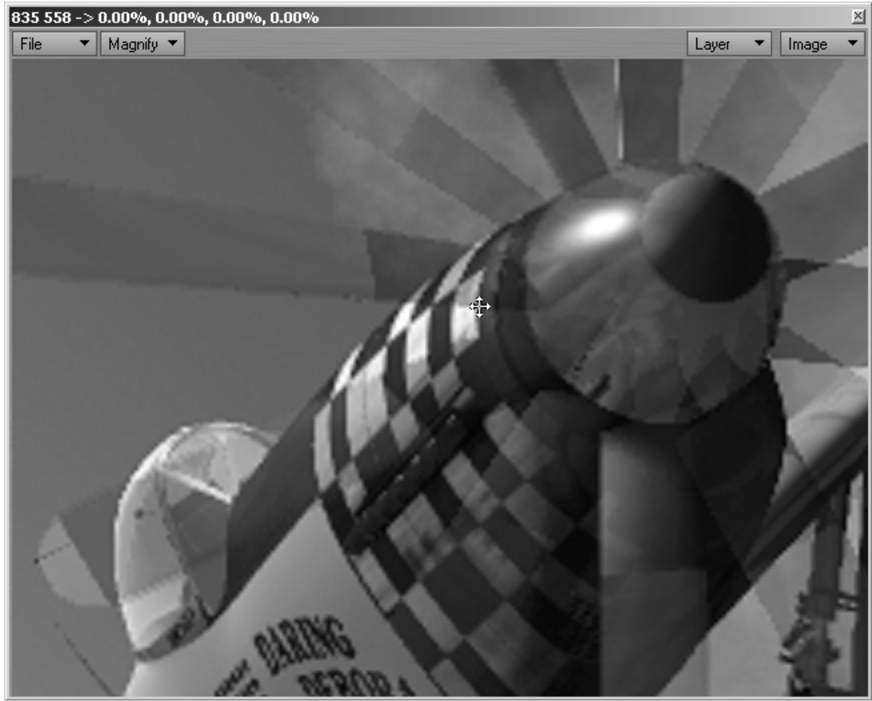
If you are viewing a high-dynamic-range image, and you use the previously described exposure controls to adjust the appearance of the 24-bit display, choosing a file format from the **Save Exposed** sub-menu saves the adjusted image as a 24/32-bit image. The normal **saveRGBA** sub-menu preserves all the color information without any adjustments. The full precision of the image (i.e., 24/32-bit, floating-point) is saved (up to the capabilities of the file format).

The **Save Resampled** submenu allows you to save a 24/32-bit scaled version of the image. You can set the **Width** and **Height** independently or if **Lock Aspect** is checked, changing either setting automatically adjusts the other to maintain the same image aspect ratio.



The Magnify Menu

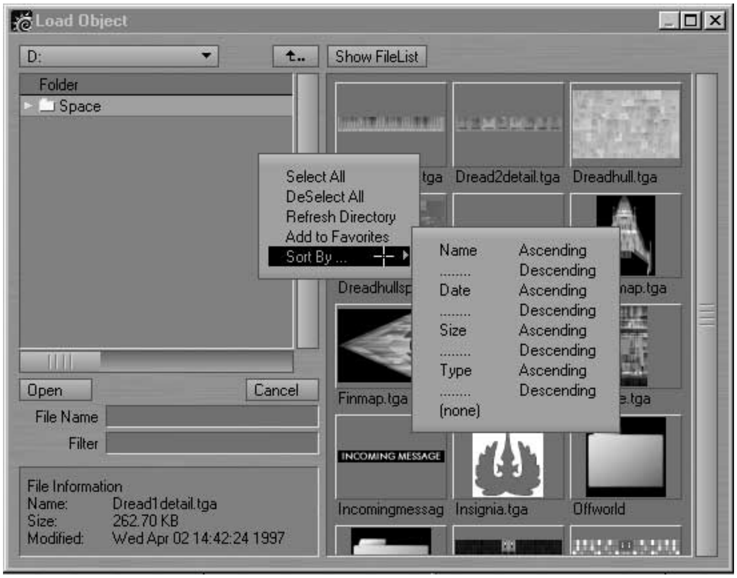
The **Magnify** menu has several settings that will affect the magnification level of the image. You may also press the plus (+) or minus (-) key to increase or decrease the level of magnification. If the entire image is not visible in the window, holding the ALT key down as you drag on the window will scroll the image around.



Magnified image

THE VISUAL BROWSER

You can make the LightWave visual browser the default file dialog for Layout and/or Modeler by choosing **VBFileRequester** on the **File Dialog** pop-up menu. This appears on the Interface tab of Modeler's Display Options panel (**Display** > Viewport: **View Options**) and Layout's General Options tab of the Preferences panel (**Display** > Options: **General Options**).



The Visual Browser

Most of its features should be self-explanatory. The button above the list window will toggle between **Show Icons** and **Show Filelist**, allowing you to switch between an icon or list view. Right-click on the list window or an item in the directory window to display the options pop-up menu. The **Add to Favorites** option will add the current directory to the drive select pop-up button in the top-left corner.

VIPER: THE INTERACTIVE PREVIEW WINDOW

The *VIPER* system (Versatile Interactive Preview Render) is LightWave's sophisticated internal preview engine. VIPER lets you preview HyperVoxels, volumetric lights, and surface settings with incredible speed and accuracy. It is accessed using the **VIPER** button on Layout's toolbar or from some plug-in panels.

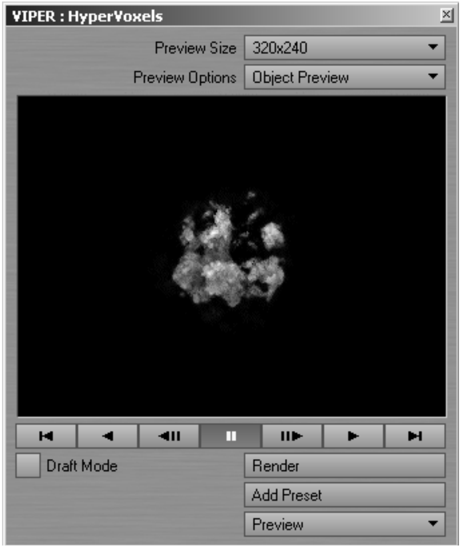
Basically, the VIPER window updates are nearly instantaneous as you make changes to settings. This provides valuable feedback to the animator. When used on surface previews, VIPER uses the internal render data buffers to make changes to what are typically render-intensive elements in the scene.

VIPER

The **Draft Mode** option will use a lower resolution for faster updates. Click the **Render** button to force a refresh. You can press the ESC key to abort the render. Use the **Preview Options** pop-up menu to access display options—these will vary depending on what feature is used in the window. You can select from various window sizes using the **Preview Size** pop-up menu.

For certain features, like surfaces, volumetric lights and HyperVoxels, clicking **Add Preset** (or double-clicking the image) will add it to the Preset Shelf, discussed next.

The **Preview** pop-up menu operates in a manner similar to its cousin on the main interface. You can use this function to preview things like animated procedural textures, volumetric light textures, HyperVoxels, and so on. Note that objects will not move like they would with a standard preview animation and pressing the ESC key will abort the preview creation process. (See Chapter 8 for more information on making preview animations.) The playback controls will appear when a preview animation is available. Choose **Free Preview** from the **Preview** pop-up menu and the controls will disappear.



Playing a VIPER preview

Surface Previews

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 during a render. LightWave also generates, alpha (transparency), z-buffer (depth), luminosity, diffuse, specular, mirror, 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.

NOTE
Because VIPER requires rendering data, it is available for surface samples only when the Surface Editor is launched from Layout.

VIPER uses information from the last rendered image when displaying certain things. This includes surface previews and when **Use Z-Buffer in Preview** is active on the HyperVoxels panel. However, this information is normally dumped after a render. To preserve it, the **Enable VIPER** option (Render Options) needs to be activated before you render a frame.

NOTE
If needed, you will be asked if you want LightWave to activate the **Enable VIPER** option for you. However, when rendering, you should disable this option, since it increases rendering time and memory consumption.

When Used With Surface Previews

If you click on the VIPER image, the visible surface will become the current surface in the Surface Editor.

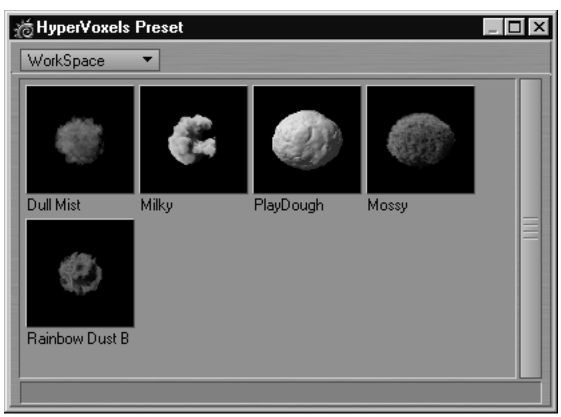
Since VIPER does not do a full-scene evaluation, there are some things that are not accounted for, like vertex maps (weight, UV, etc.) on SubPatch objects, ray-traced effects (reflections, refractions, shadows), shadow maps, fog, double-sided polygons, radiosity, light falloff, etc. As such, it is not a replacement for the render preview window. Also, VIPER will not be affected by moving geometry.

Non-Surface Preview Uses

When using VIPER to preview non-surface preview items, like hypervoxels and volumetric lights, generally it does not use the internal render buffers. Thus, there is no need to activate the **Enable VIPER** option nor render a frame.

THE PRESET SHELF

The Preset Shelf is a sizable floating window that holds a list of previews along with all of the associated settings. It can be used for settings with surfaces, volumetric lights, hypervoxels, etc. It is accessed using the **Presets** button on the toolbar or from some plug-in panels. You add to the shelf by double-clicking on the VIPER window or preview window, if one exists on the panel (e.g., Surface Editor). The shelf survives from session to session.



The shelf can be resized by dragging an edge of the window

The window is context sensitive, so if you are working on surfaces, surface presets are displayed, if you are working on hypervoxels, those presets are shown, and so on.

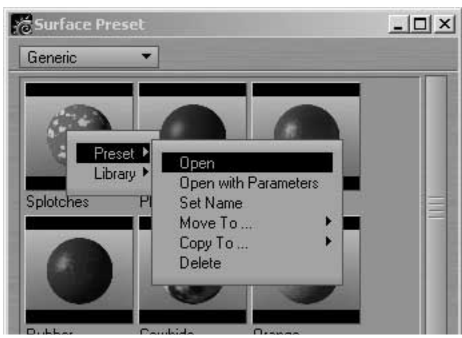
Each editor has a default *library* named **Workspace**—a library is a grouping of presets for a particular editor. You can create a custom library by right-clicking over a preview and choosing **Library** > **Create**. For example, using the Surface Editor, you might make a library of *Wood* presets, a library of *Stone*, etc. The **Library** > **Discard Presets** menu clears all of the presets from the displayed library.

To apply one of the presets:

1 Double-click on a preview or select **Preset** > **Open** from the RMB pop-up menu.

2 Click the **Yes** button on the confirmation dialog.

NOTE
Preset > **Open w/Parameters** works with certain preset groups, like hypervoxels, and allows you to pick which part of the settings you want to use.



Right-click menu

Choose **Set Name** to open a dialog box where you can set a **Name** to be used for the sample, it appears underneath the sample image. You can add a longer string of text in the **Description** field, which appears at the bottom of the panel when your mousepointer is over a preview.

There are also options to move/copy presets between libraries. The **Delete** option removes the sample from the shelf.

LScript

LScript is a high-level wrapper for the LightWave plug-in 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 plug-in API, making plug-in development faster.

Because LScript has its roots in the C language, the transition between scripting and native-language (binary) plug-in 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 plug-in development.

Nearly all of the LightWave plug-in architectures have scripting capabilities through LScript.

LScripts can be installed in the same way plug-ins 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. Scripts written on one platform should work directly and immediately on any other platform supported by LightWave. This differs from traditional plug-in development in that each platform must have its own compiler, and each plug-in must be compiled and maintained on that platform. LScripts are platform independent.

See the on-line documentation for more information.

LightWave Panels and Dialogs

LightWave's panels and requesters are generally non-modal and can be left open even while you interact, say, with the Layout window and its controls.

Math in Input Fields

Numeric input fields support basic math operations (+, -, *, /, and ^) and algebraic equations. You can even mix units. Here is an example: (12ft+14m)^2+(3/5). The result, 312.3908 m, would appear after you press the Enter key. Note that the result is shown using the default unit; here, it is meters. Also, since we didn't specify the unit of measure for the 3 and 5, the default is used.

ENTER/TAB Keys with Input Fields

When your enter values into dialogs that have multiple input fields, the TAB and ENTER keys have special functions to save you time. Pressing the TAB key stores the entered/current value and automatically advances you to the next input field. SHIFT + TAB moves to the previous field. The ENTER key also stores value, but dismisses dialog, if appropriate.

Yes and No

Whenever a requester asks you a **Yes** or **No** question, pressing the ENTER key is the same as selecting Yes, while pressing the ESC key is the same as selecting No. Likewise, an Error, Warning or Reminder requester with a choice of **Continue** or **Cancel** can use ESC for Cancel and ENTER for Continue. For those requesters with only a **Continue**, pressing either ESC or ENTER will close the panel.

Color Selection

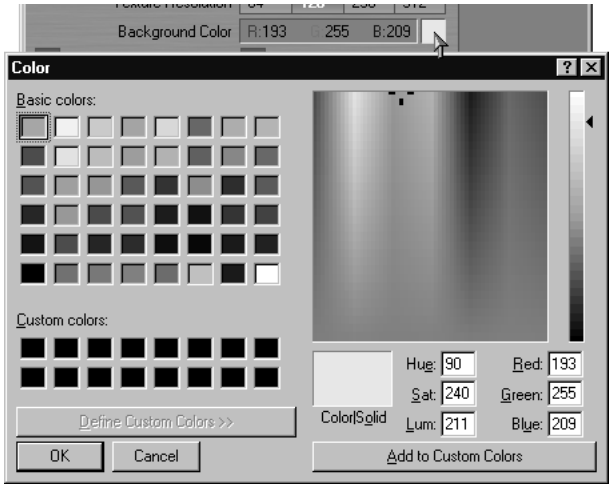
Where you must select a certain color (e.g., surface color), there will be a three-number color component display and a color swatch box. You can also drag your mouse on each color component to change its value. By default, the standard **RGB** color space is displayed; however, if you right-click on the display, you may also use the Hue, Saturation, and Value (**HSV**) color space, if that is more familiar to you. HSV is based on the artist concepts of tint, shade, and tone. There are 16.7 million possible color combinations.

NOTE
Unlike **RGB**, **HSV** color components are not independent. For example, if Value is 0 (i.e., brightness is 0), you will not be able to change Hue (color) nor Saturation. Similarly, if Saturation is 0, you will not be able to change Hue. Moreover, reducing one component may cause another component to decrease.

NOTE
LightWave will attempt to display the selected color to the best of your display's capabilities. This will generally result in an approximation of the color if you are using less than a 24-bit display mode.

Color Pickers

Clicking the color swatch box will open a color selection requester that is standard for your specific platform or the built-in color picker.



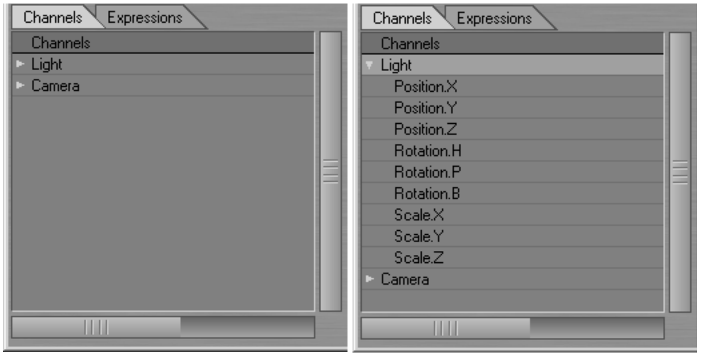
Windows color picker

The LW_ColrPikr color picker is a color selector with more features, such as a wavelength and temperature-guided layout of color. For Layout, select this on the General Options tab of the Preferences panel (**Layout** > **Options** > **General Options**). For Modeler, go to the Interface tab of the Display Options panel (**Modeler** > **Options** > **General Options**).

LightWave color picker

Standard List Windows

Windows that contain lists of items (surfaces, scene items, menu items, etc.) all have a slider bar to scroll through the list. If the list is hierarchical in nature, there will be arrowheads to the left of item names. If the arrowhead is facing to the right, then they are subordinate items to this *parent* item. Clicking the arrowhead will make it face down and also reveal the subordinate items.



Left: List is collapsed for both entries. *Right*: Light entry is uncollapsed

If selection is appropriate, you can click on an item to select it. CTRL + click will select non-contiguous items and SHIFT + Click will select a range of items.

List windows that are hierarchical in nature can be reorganized by dragging items around in the list. Where appropriate, multiple selection is supported.

Reorganizing Lists

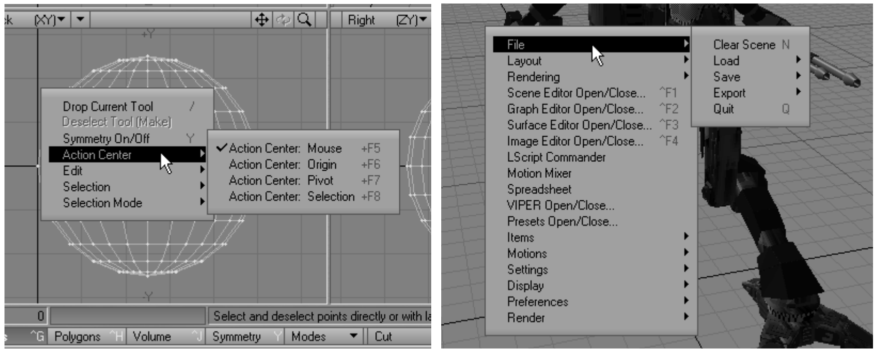
In some list windows (e.g., menu configuration), you can reorganize entries by just dragging them in the window. Place your pointer over the entry and drag it up or down. You will see a thin line appear as you drag. This is the insert point. Releasing the mouse button will drop the entry here.

Now, if your pointer is at a position where the dropped item could be, say, a sub-item of an entry, the same level as the entry, or even the same level as the next entry, you will see the line change its length as you drag. The position of the left end of the line determines the level of the drop.

Left: Same level drop. *Right*: Sub-level drop

Context Pop-up Menus

Both Layout and Modeler have special “context” pop-up menus that you display by holding CTRL + SHIFT and clicking your LMB, MMB, or RMB with your mousepointer over a viewport—there are different menus for each mousebutton. These can be customized as discussed later in this manual.



Left: Modeler LMB menu. *Right*: Layout LMB menu

chapter 4
Plug-ins

Chapter 4: Plug-ins

Plug-ins can greatly increase the available features in LightWave 3D. A single plug-in file can contain multiple commands. Some plug-in commands are tools, and as such, they can be active or inactive just like a standard built-in tool. All of the plug-ins that come with LightWave should already be installed and available. However, you may sometimes need to re-add one because of an update or just to add a new third-party plug-in.

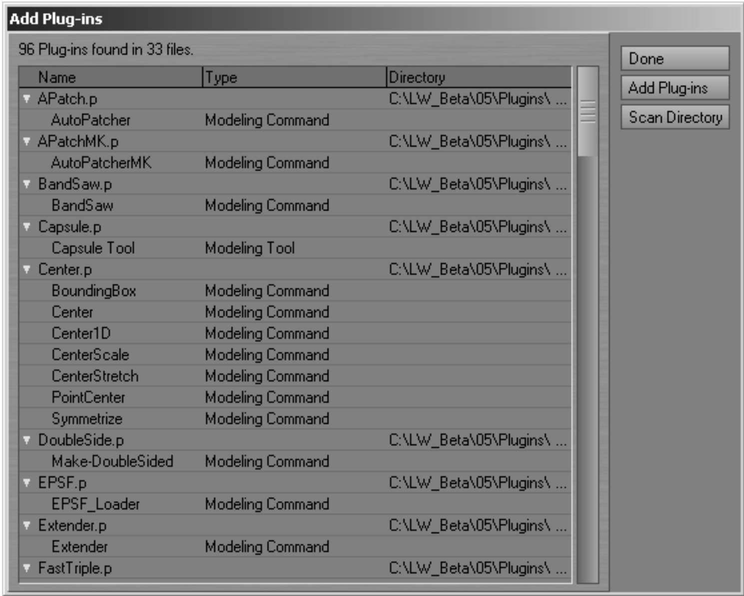
Generally, plug-in files are stored in various subdirectories in the LIGHTWAVE6_5\PROGRAMS\PLUGINS subdirectory. Some plug-ins work only in Layout or Modeler, while others work in both. You can even install an LScript as a plug-in! It becomes a command that can be used like any other command and assigned to a menu or keyboard shortcut.

ADD PLUG-INS COMMAND

To add plug-ins from Modeler, choose **Modeler** > **Plug-ins** > **Add Plug-ins**. From Layout, choose **Layout** > **Plug-ins** > **Add Plug-ins**. (Hold the SHIFT key to select a range. Hold CTRL to select multiple non-contiguous files.) Generally, it doesn't matter from which application you add the plug-ins.

In Layout, an informational dialog will appear telling you how many plug-in (commands) were added. In Modeler, the Add Plug-ins panel appears after you select the .p file(s).

The number of individual plug-ins found will appear in an informational dialog. Then, the added plug-ins, their type and related .p filenames are listed in the window.



Add Plug-ins Panel

NOTE
Re-adding a plug-in that is already added will not do any harm.

To add additional plug-ins from the Add Plug-ins panel:

Click the **Add Plug-ins** button and select one or more .p plug-in files from a directory.

NOTE
A single plug-in 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 plug-in. This is normal.

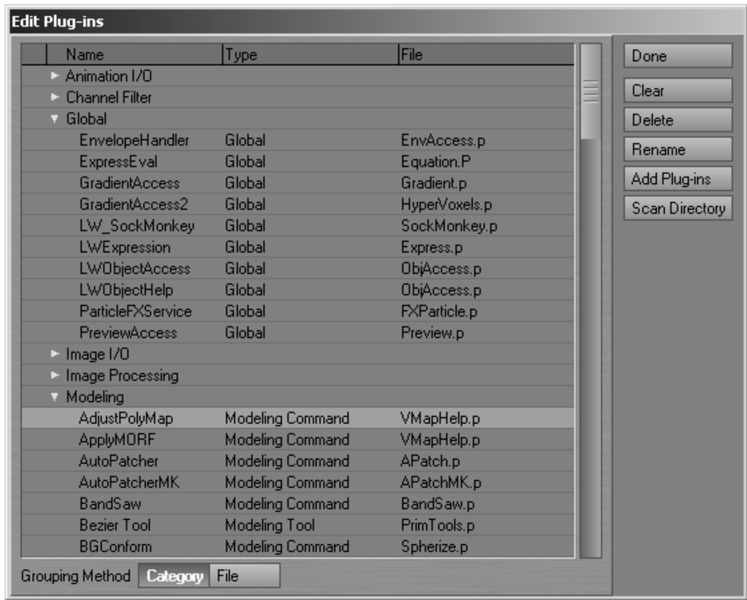
To add a directory of plug-ins:

You can quickly add all of the plug-ins in a directory (including subdirectories) using the **Scan Directory** button. (It may take a few seconds to scan all of your plug-in files. Be patient!)

NOTE
You can get immediate access to the Scan Directory feature on the Edit Plug-ins panel, discussed next.

EDIT PLUG-INS COMMAND (MODELER)

Use the Edit Plug-ins command to review and delete plug-ins that have been added. In Modeler, choose **Modeler** > **Plug-ins** > **Edit Plug-ins**. In Layout, choose **Layout** > **Plug-ins** > **Edit Plug-ins**.



Category Grouping Method

To delete a plug-in command:

You can delete a plug-in command by selecting it and clicking the **Delete** button. (Other commands from the plug-in will continue to exist.) Click **Clear** to delete all plug-ins listed. These operations have no effect on the actual files stored on your hard drives.

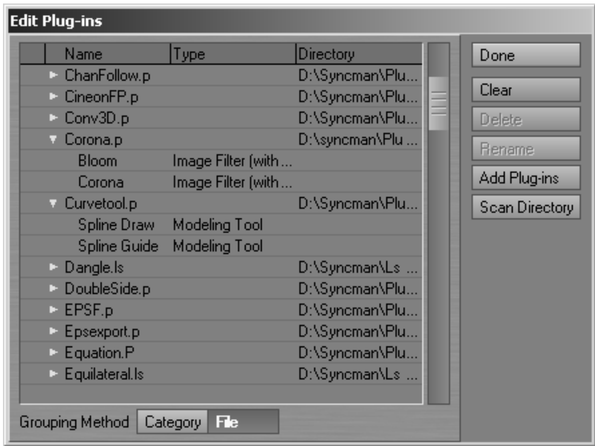
To rename a plug-in command:

You can rename a plug-in command's name by selecting it, clicking the **Rename** button and entering a new name.

NOTE
You can also add plug-ins on this panel. The **Add Plug-ins** and **Scan Directory** buttons operate the same as they do for the Add Plug-ins panel, discussed earlier.

File Grouping Method

If desired, you can list the plug-in commands by their .p plug-in filenames. Just select the **File** Grouping Method button at the bottom of the Edit Plug-ins panel.



File Grouping Method

Plug-ins Configuration File

Plug-ins configuration information is centralized and shared between applications. The file is named LWEXT3.CFG. Adding plug-ins from Layout or Modeler automatically makes them available to all LightWave applications.

Legacy Plug-ins

Some plug-ins are included in the LEGACY_PLUGINS folder for transitional purposes. These should be used only when needed for the support of older scene files. Generally, the functionality of these tools has been integrated into the new system and, as such, the LightWave native tools should be used for any new content creation.

chapter 5

Customizing Your Interface

Chapter 5: Customizing Your Interface

WARNING
We strongly suggest that you keep the default menu organization 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.

NOTE
Some custom menu configurations have been included on the CD. Try them. You might like 'em!

LightWave menus are customizable. You can add, remove, group, and reorganize commands.

Choose **Layout** > **Interface** > **Edit Menu Layout** or **Modeler** > **Interface** > **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) Menus window, several main sections have sub-items, which are indented to show the hierarchical relationships. **Top Menu 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. **Bottom Edge** is for Modeler only and relates to the controls along the bottom edge of the screen. **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.

Layout Configure Menus panel

In the Menus 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 .

FIRST-LEVEL MENU ITEMS

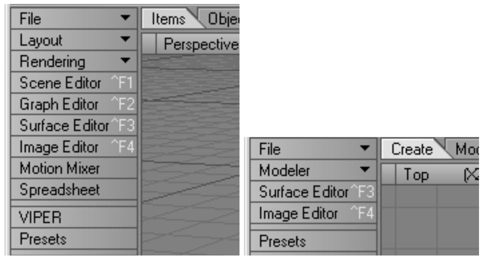
Menu items on the first-level must always be a group. Commands and sub-groups may exist on the second menu level. On the toolbar, a Main Menu group's name will appear as a heading above the grouped command buttons. If the group name is blank, no heading will appear. On the next menu level, commands or sub-groups can also be used. If a sub-group is used, it will appear as a pop-up menu on the interface.

The hierarchy would look like this:

- Section Type (e.g., Main Menu)
  - Group (Group name appears on row of horizontal tabs on the main interface)
    - Command/Sub-group (If a group, its name, if any, appears as a header above the next group of commands)
      - Command/Sub-group (If a group, its name appears on pop-up button)

TOP MENU GROUP

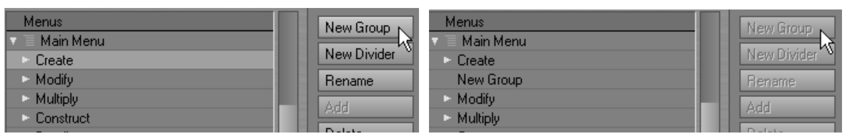
Note that top items of the toolbar are constant no matter which menu tab is selected. These are contained in the Top Menu Group. This gives you immediate access to basic file operations, global settings, major editors, and so on.



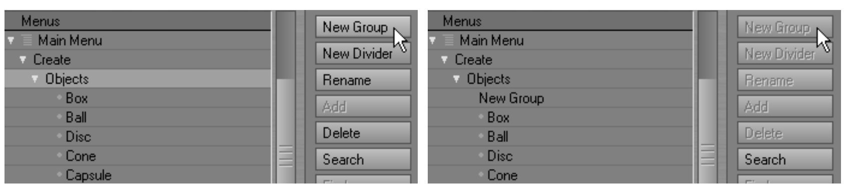
Constant menus for Layout (left) and Modeler (right)

ADDING NEW GROUPS

If you select a collapsed group when you click the **New Group** button, it will be added at the same level as the group. If you select an uncollapsed group when you click the **New Group** button, it will be added inside the group.



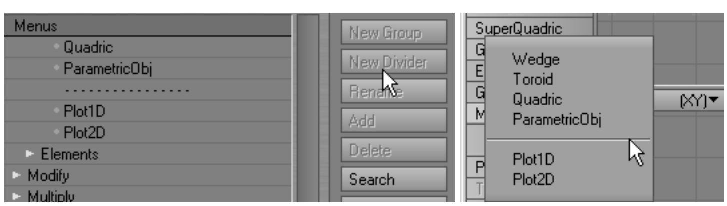
Group added to collapsed group



Group added to uncollapsed group

DIVIDER LINE

Clicking the **New Divider** button will add a dashed line below the selected menu item. This is displayed as a dividing line between menu items when it appears in a pop-up menu group.



Left: Divider line added. *Right*: Divider line visible in pop-up menu

RENAMING MENU ITEMS

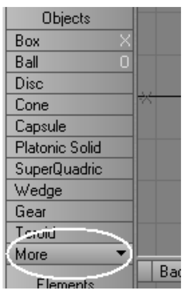
You can rename any menu item (including commands) by selecting it with your mouse and then clicking the **Rename** button.

NOTE

The command name is not always what is displayed on a button. If the command uses a shorter name, the button may use the exact or a similar name. However, the button name can be very different in some cases. Of course, you can always rename the displayed name.

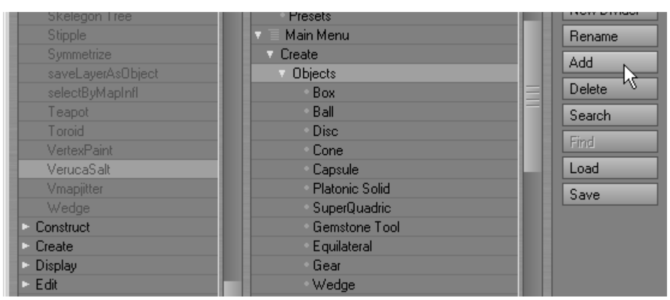
AUTOMATIC MORE BUTTONS

If an interface (e.g., Layout or Modeler) is sized too small to fit all toolbar buttons, a **More** pop-up button will appear at the end of menu groups and contain the hidden buttons. Note that sometimes the menu names can be different depending on whether they are listed on the toolbar or listed in a pop-up menu.

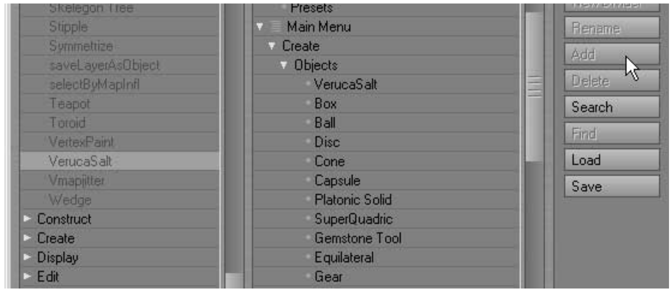


ADDING COMMANDS

To add a command, select it in the (left) Command window and select the target position in the (right) Menus window. Then click the **Add** button. The command will be added under the selected target—inside the group if a group was selected. Alternatively, you may drag the command to the Menus window.



Command and menu location selected



Command added to menu

Commands that are already added to a menu appear ghosted in the Command window. That just indicates that the command already appears somewhere in your menus. However, the same command can still be added more than one time. Since Modeler plug-in commands are automatically added to the Middle Mouse Button Menu, their commands will always be ghosted.

REORGANIZING MENUS

See Chapter 3, “Reorganizing Lists” for information on how to reorganize entries.

HINT
To place the group/command at a specific point in a group, make sure the group is uncollapsed before you begin dragging.

DEFAULT MENU LOCATIONS FOR PLUG-IN COMMANDS

The plug-ins that come with LightWave have default (Layout or Modeler) menu locations for their commands. However, when you add plug-ins, the commands will be placed in their default menu locations only if you are using the default menu. (You are using the default menu, if you have never customized the menu configuration or have chosen **Default** from the **Presets** pop-up menu.)

Modeler places commands without default menu locations in the **Construct** > Utility: **Additional** pop-up menu. For Layout, you must manually assign such commands to menus.

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.

Since a command can be assigned to more than one menu location, successive **Find** clicks will show the next assignment, if more than one exists.

Clicking **Search** allows you to type in a search phrase to find a command. Note that this will only find the first instance and it is case sensitive.

HINT
If you know the keyboard shortcut, but want to find the corresponding menu location, use the Find function on the *Configure Keys* panel to determine the command name. Then, use the Find function on the *Configure Menus* panel to see where it is assigned.

DELETING MENU ITEMS

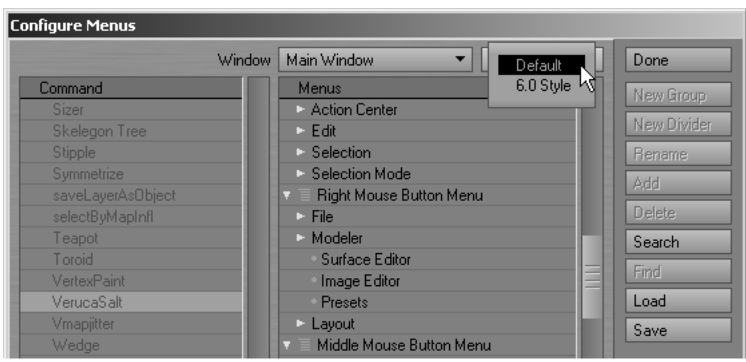
To delete a command/group, select it and click the **Delete** button.

WINDOW POP-UP

The **Window** pop-up menu lets you select different menus to edit (e.g., Graph Editor), if available.

MAINTAINING MENU SETS

The **Load** and **Save** buttons let you retrieve and store *menu sets* that you develop. To restore the default menu configuration, choose **Default** from the **Presets** pop-up menu. If you'd like to have a menu set similar to LightWave 6.0, choose **6.0 Style**. (For Layout, you can even choose **5.6 Style**.)

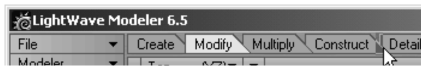


Menu presets

NOTE
In Modeler, you can use **Modeler** > **Preferences** > **Revert to Startup Preferences** to restore your initial settings; however, any changes since you first ran the application will not be reflected.

ARRANGING MENU TABS

You can reorganize the main menu tabs by simply dragging them left or right on the main interface. Before you release the mouse button you will see an insertion point market appear.



KEYBOARD SHORTCUTS

Keyboard shortcuts are displayed on the right side of buttons, if applicable. Keystrokes that need the SHIFT key are displayed in uppercase if they are letters. (e.g., A) Unshifted keystrokes are displayed in lowercase. (e.g., a) For example, M is SHIFT + M and & would be SHIFT + 7. This manual, however, always explicitly indicates when the SHIFT key should be used.

WARNING
You need to be aware of the state of your CAPS LOCK. If this is active, it can affect upper and lowercase keyboard shortcuts.

Panel-Specific Shortcuts

Some panels have their own special keyboard shortcuts. To use them, you must have that panel active. (Click on it, if it isn't already active.) If the panel is not active and that shortcut has another meaning for LightWave in general, the general function will be run. Moreover, if the panel is active, but doesn't use the keyboard shortcut, it will be passed to the main interface.

CUSTOMIZING KEYBOARD SHORTCUTS

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 **Layout** / **Modeler** > **Interface** > **Edit Keyboard Shortcuts**.

Modeler Configure Keys panel

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:

1 Select a command in the left window.

2 Select the target key in the right window. (You can hit the desired key or keystroke combo to quickly select the key.)

3 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:

1 Select the target key in the right window.

2 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. Note that this will only find the first instance and it is case sensitive.

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.

NOTE
In Modeler, you can use **Modeler** > **Preferences** > **Revert to Startup Preferences** to restore your initial settings; however, any changes since you first ran the application will not be reflected.

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.

GENERIC PLUG-INS

Generic Layout plug-ins (**Scene** > Utilities: **Generics**) will appear in the Plug-in commands group. Thus, they can be added to menus or mapped to keyboard shortcuts.

MIDDLE MOUSE BUTTON MENU

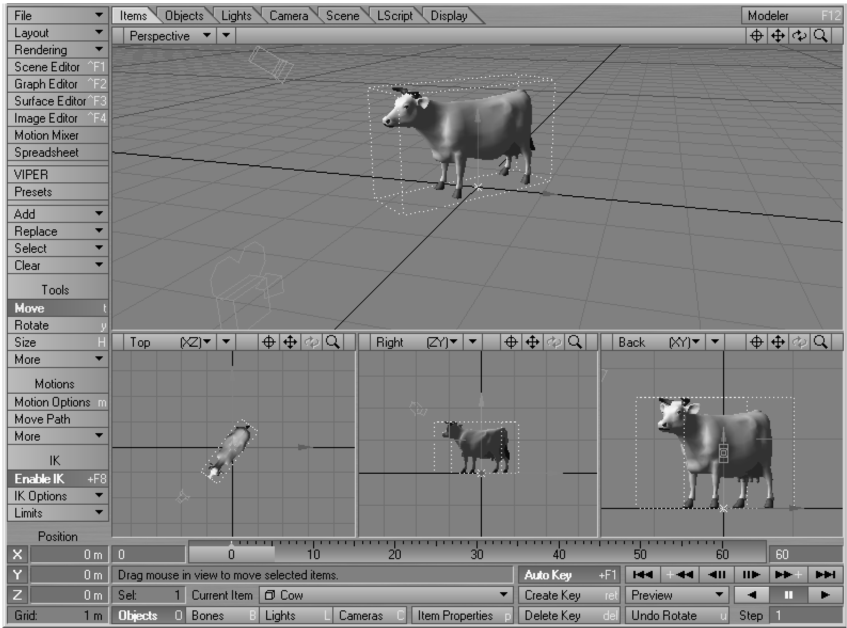
The context menu for the middle mouse button exactly mirrors the group structure for commands, as found in the menu and key editing dialogs—including any plug-in commands added to the system. If you know where a command is located in the master groupings, you can find it in the middle button menu.

chapter 6
Layout: General Functions

Chapter 6: Layout: General Functions

LightWave's Layout has been designed to provide the most efficient interface possible for your 3-D 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 the user. 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.

Default single viewport layout



1 Top, 3 Bottom viewport layout

THE TOOLBAR MENUS

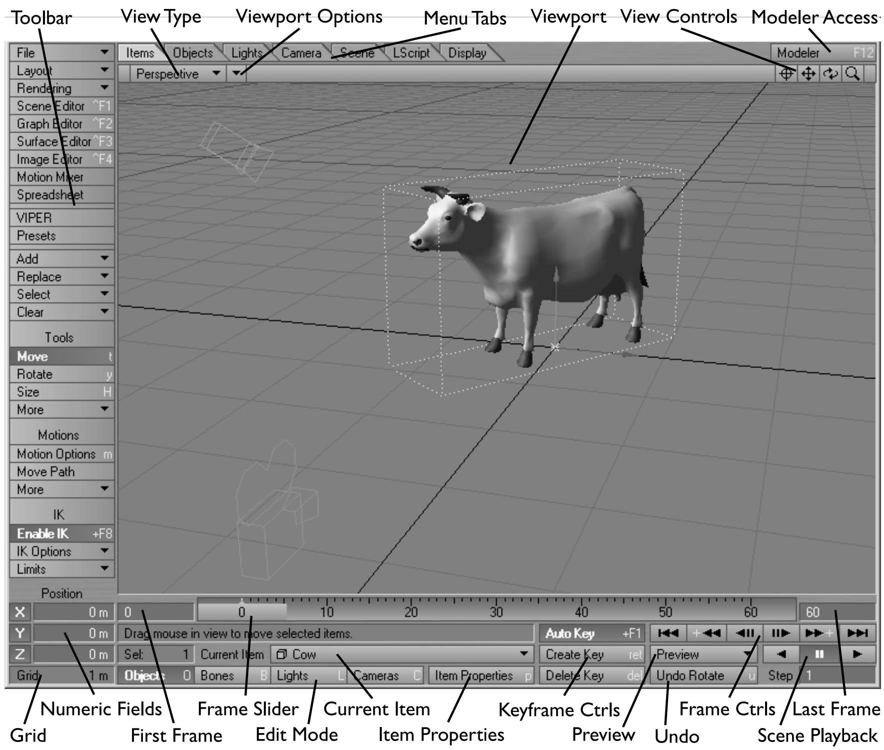
The vertical menu along the left is the *toolbar*. The toolbar contains a set of buttons, and each button can bring up control panels, tools, activate settings, and so on. The buttons can also be pop-up menus. The active tab along the top of the interface determines which set of buttons is visible on the toolbar.

Many of the buttons are shortcuts to controls that can also be accessed from one of Layout's many panels.

NOTE
An option on the Preference panel's General Options tab (**Display** > Options: **General Options**) lets you place the vertical menu on the right or left side of the interface, or hide it altogether.

You can select several predefined menus by clicking on the tabs above the viewports.

NOTE
You have absolute control over these menus. See Chapter 5 for information on how to customize your menus.



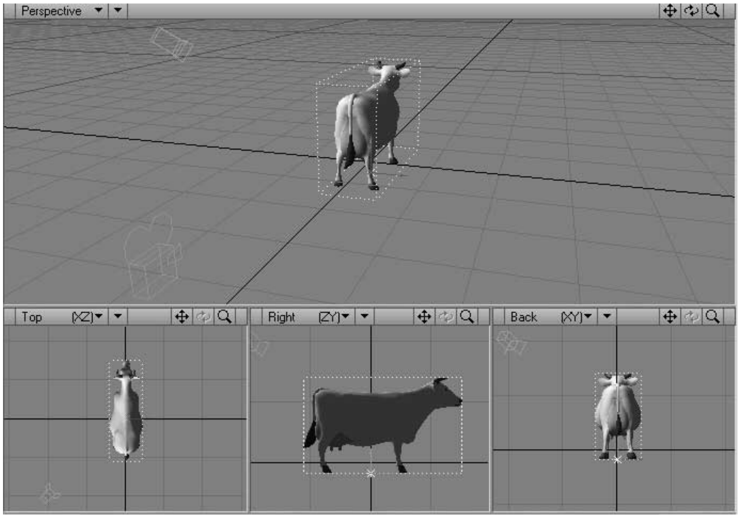
Major Layout interface functions

MODELER ACCESS

If the Hub is running (and connected), a **Modeler** button will appear in the upper-right corner of the Layout interface.

LIGHTWAVE'S VIRTUAL WORLD

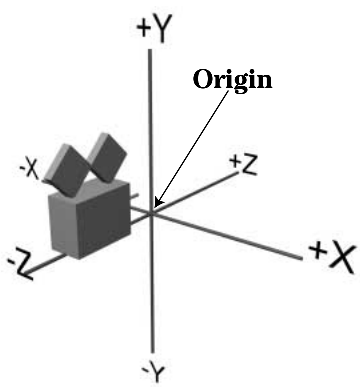
LightWave's world is defined using three axes: X, Y, and Z. Here we've loaded the ANIMALS\COW.LWO object and haven't rotated it. At its default position, from front to back, the cow's body is aligned along the Z axis and is facing the the positive Z direction. The X axis runs left to right, with right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



Viewing from different angles

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.)

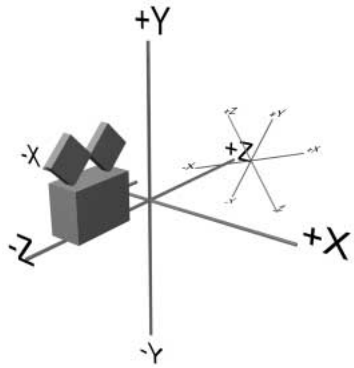


The Origin and world axes

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.



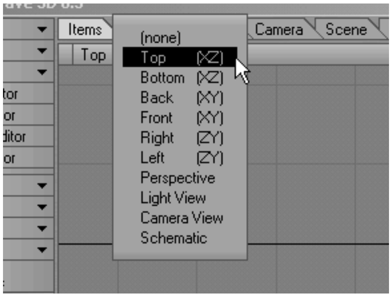
Local axes vs. world axes

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. Get it? Now sit down at your PC before someone sees you.

YOUR POINT OF VIEW

By default, Layout uses a single viewport. Later on in this chapter, you'll learn how to use up to four simultaneous viewports! Anyway, 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 3-D space on a 2-D display (i.e., your monitor) can be difficult at times, so you will switch between nearly all of these as you edit your scene.

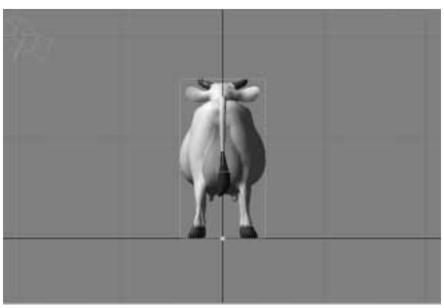
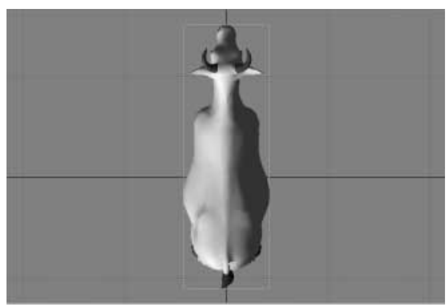


View Type pop-up menu on titlebar

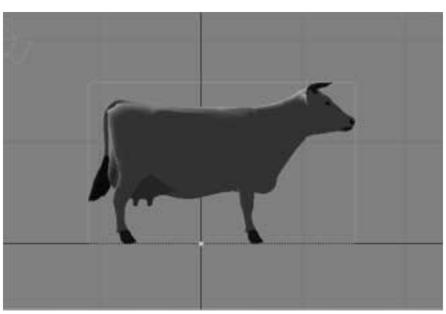
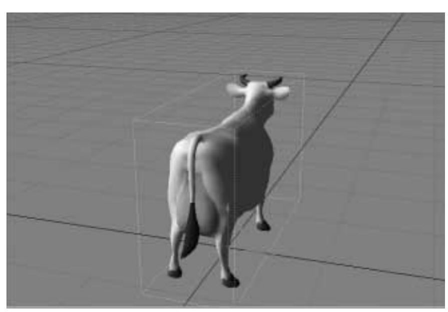
It is sometimes easier to work in just two dimensions at a time. The options with the axis notations (e.g., **Top (XZ)**) are the "orthogonal" views, which let you move items in only two dimensions (horizontally or vertically), along the XY, XZ, or ZY axes. The **(none)** setting blanks out the viewport. 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.

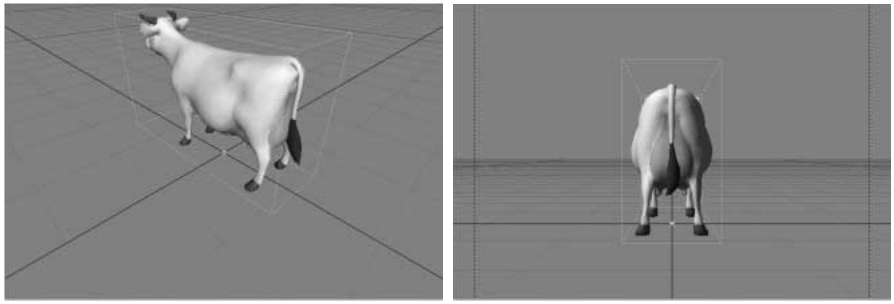



Left: Back. *Right*: Top

Left: Right. *Right*: Perspective

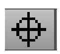
There are also *pseudo-physical* POVs. When setting up a light, you'll often want to *look through it* to see exactly what it points at. In such a case, you'll use the Light view to look through the current light. You'll always want to see your scene from the Camera view at some point since that is the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



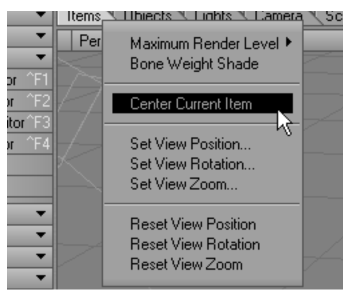
Left: Light View. *Right*: Camera View

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 obviously have no effect when you use the Light or Camera views, 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.

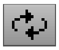
Move 

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

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.

Keyboard shortcut: Shift + Alt

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

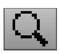
Rotate 

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

NOTE
Alt is the same as Shift + Alt (i.e., Move) when used in an orthogonal view.

Zoom 

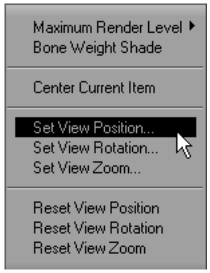
All views: Zooms in and out when you drag left and right. (You can also use the < and > keys.)

Keyboard shortcut: Ctrl + Alt

Taking Aim

Both the orthogonal and perspective views are based on a single *aimpoint*. In other words, you are always looking at the same point in 3D space no matter which of the views you use (except Light and Camera view, of course.) It is also the center of the view rotation. That's why, if you move around in the Back view, it also affects the other views, like Perspective.

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.



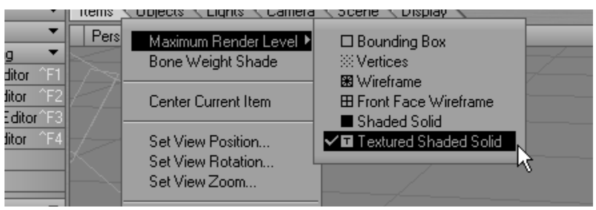
NOTE
If you are using multiple viewports, each has its own independent set of position, rotation, and zoom values.

Resetting Views

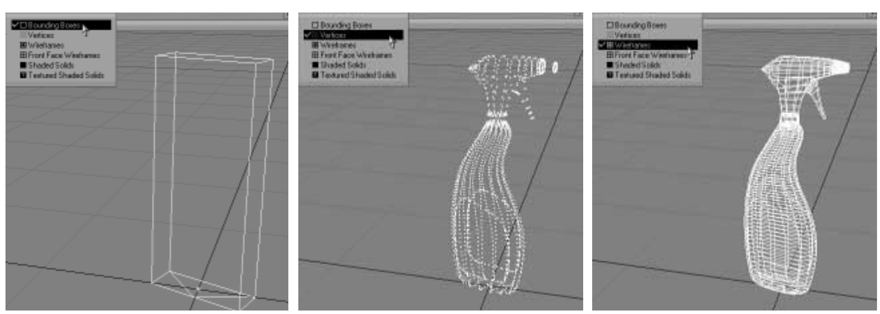
Also on the pop-up menu, above, are options to reset a view's position (Move), rotation, and zoom to default values.

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, discussed in Chapter 12.



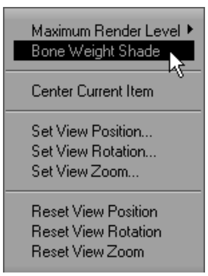
Maximum Render Level menu


Left to right: Bounding Box, Vertices, and Wireframe

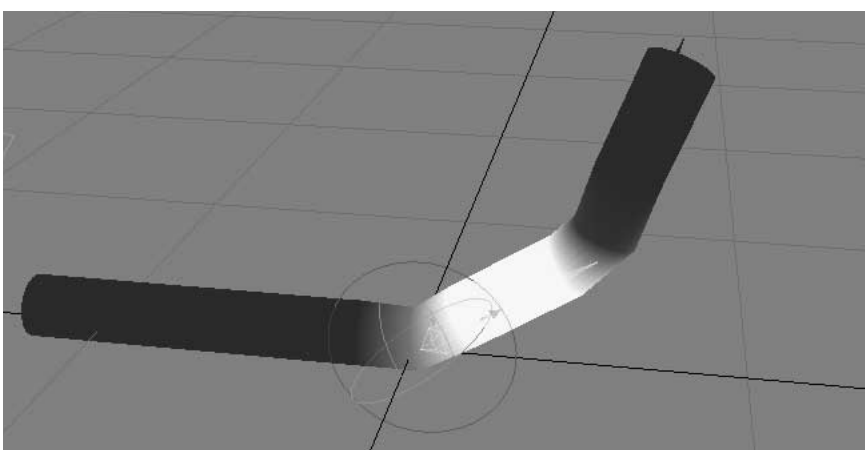
Left to right: Front Face Wireframe, Shaded Solid, Textured Shaded Solid

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.



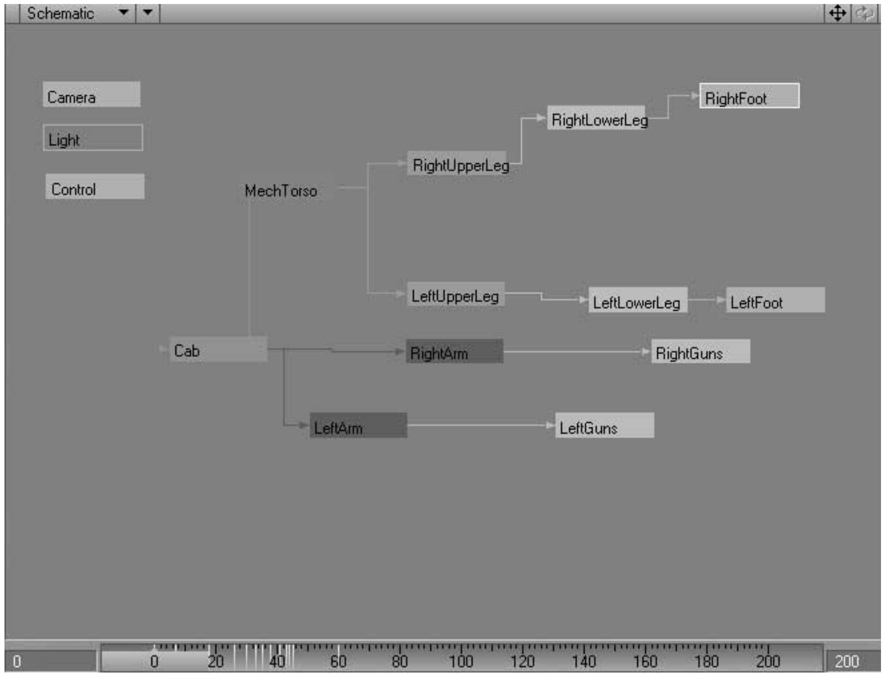
Bone Weight Shade menu item



Bone Weight Shade

THE 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.



The Schematic view

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 plug-in, discussed later.

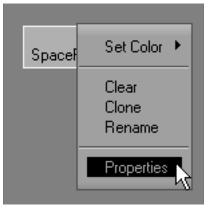
Navigation is consistent with other 2-D viewports (e.g., ALT/ALT+SHIFT =move, CTRL+ALT=zoom, **Center Current Item**, etc.). Press the A key to fill all items into view or G key to center.

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.

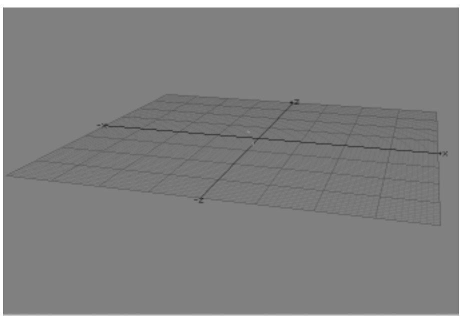
Other Schematic View Options

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.



THE GRID

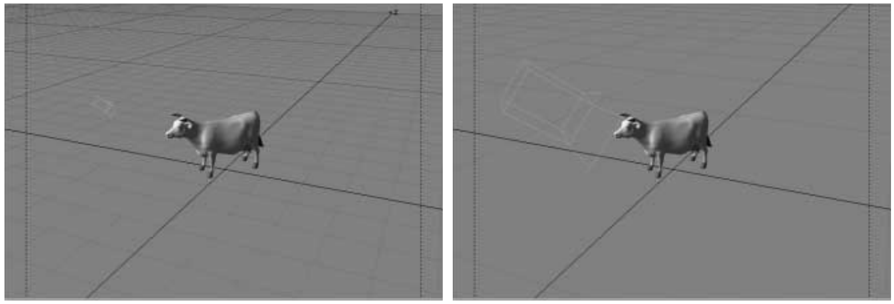
A grid of squares—cleverly known as the *grid*—is visible in any of the orthogonal views, as well as the Perspective view. The grid serves as a visual reference when you move items around, but it will never render in a final image (no matter how much you wish it to). The grid lines are darker every tenth square for visual reference. The Origin is located at the center of the grid.



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 (**Display** > Options: **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. If you have very large grid squares, you will also have very large lights/camera compared to objects, and vice versa.



Left: 1m grid. *Right*: 5m grid

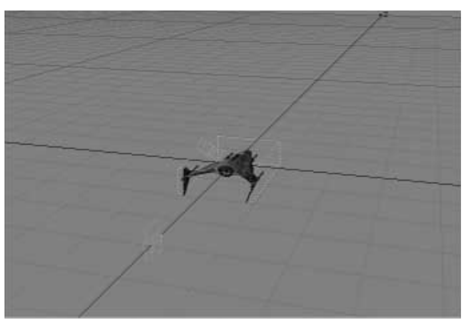
NOTE
The size of the lights/cameras do not affect their functionality.

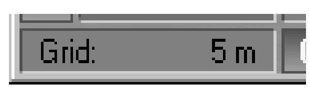
The Grid Square Size Effect on Positioning

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.

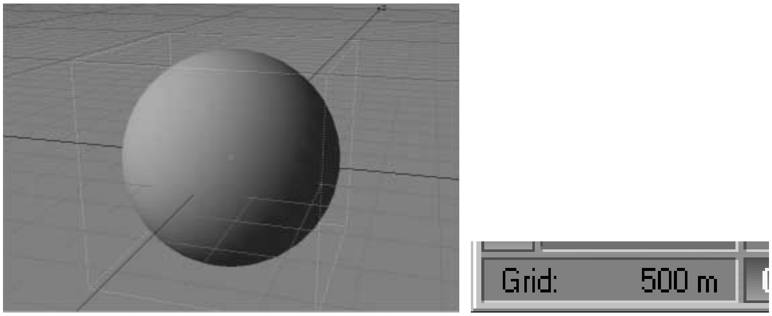
Grid Square Size Auto-Adjustment

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.





5m grid



500m grid

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** <u>to the same value</u>. Then, load the larger objects.

Content Directory

Before we get into how you create a LightWave scene, you need to understand an important concept called the *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/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.

Let's say your weird carnival-ride-operator-friend had this screaming *rendering monster* that you could use. Well, you just copy all of the files in the STRETCHPRINCESS directory to your zippy removable disk and trudge over to your friend's place next to the fun house. All you do is insert the disk, set the Content Directory to his zippy removable disk, seamlessly load the scene, and render the scene. Now that's portability!

C:\MyProjects\StretchPrincess\Objects
C:\MyProjects\StretchPrincess\Images
C:\MyProjects\StretchPrincess\Scenes

Z:\Objects
Z:\Images
Z:\Scenes

Content Directory

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 (**Display** > Options: **General Options**) and click the **Content Directory** button. 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 utilize 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.

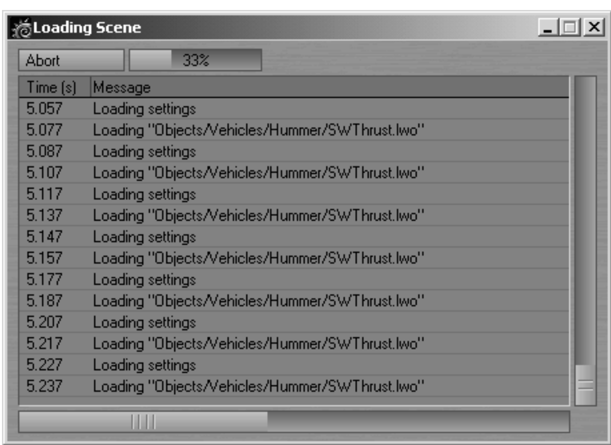
NOTE
The generic plug-in Content Manager, discussed later in the chapter, can be used to *collect* a scene's supporting files and ensure correct compliance with your content directory.

Scene File Management

To load an existing scene:

1 Make sure the Content Directory is set properly.

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.



To save a scene:

1 Make sure your Content Directory is properly set.

2 Choose **File** > **Save** > **Save Scene** to save a scene. **Save Scene As** lets you save the current scene using a different name.

NOTE
Remember that this operation does not save any loaded object files. That must be done as a separate step.

Saving a Copy

Choosing **File** > **Save** > **Save Scene Copy** will let you save a copy of the current scene using a different name, without affecting the current scene.

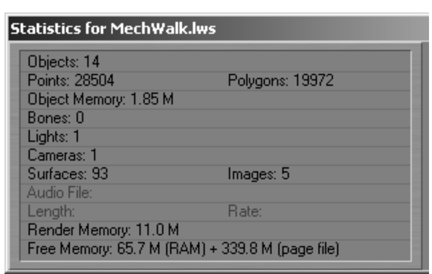
RECIPE FOR A SCENE

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

Scene Statistics

Pressing the w key will display the Scene Statistics panel, which provides various information about the current scene.



Scene Stats

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.

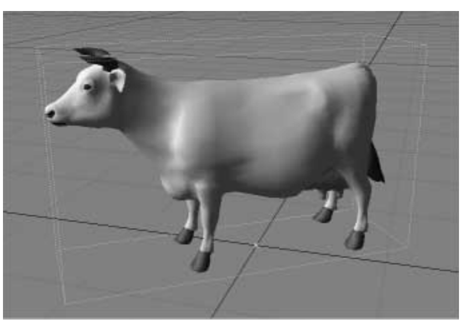
To select an item:

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). (See Chapter 12 for information on locking items.)
- Use the ItemPicker master plug-in, discussed later.

NOTE
You can use your UP and DOWN ARROWKEYS 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.



Cow is selected

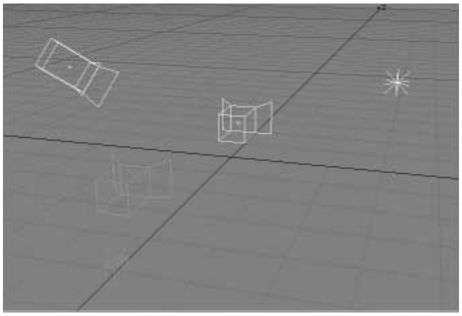
NOTE
You can select a bone by clicking near its midpoint, rather than its pivot point, making it possible to pick different bones that branch from the same point in a hierarchy.

Selecting Multiple Items

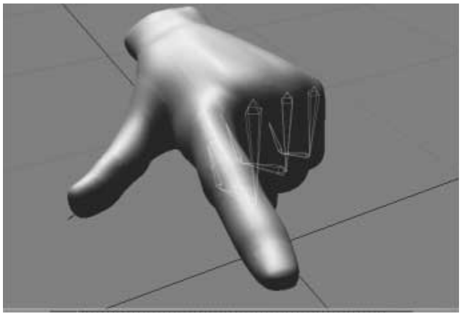
You can select multiple items of the same type, like all objects or all lights, and perform edits on them simultaneously.

To select multiple items:

Holding the SHIFT (or CTRL) key will allow you to select multiple items of the same type.

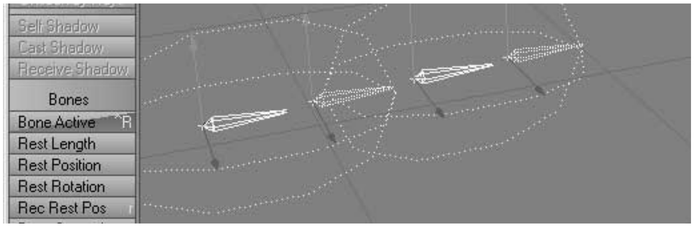


If you select multiple items, many operations will be applied to all of them. This can save a lot of time under the right circumstances. Such operations include Move, Rotate, and Size, as well as certain item properties, like Unseen by Rays, Unseen by Camera, Self Shadow, Cast Shadow, Receive Shadow, Bone Active, Affect Diffuse, Affect Specular, Affect Caustics, Affect OpenGL, and so on.



Of course, these aren't the fingers that were bent when this was shown to me.

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.



Bone Active button showing mixed selections

Selecting by Name

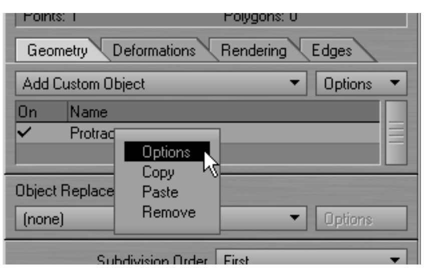
Pressing the apostrophe (‘) key launches a special selector dialog. Simply type-in a few characters that uniquely identify the desired item and click **OK**. You can select any type of item.

Unselecting Items

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

LAYOUT LIST ITEM POP-UP MENU

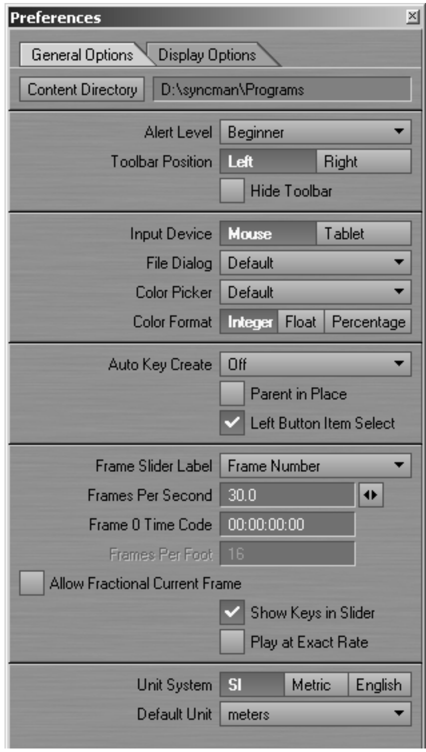
For lists in Layout like surface shaders, image filters, etc., a special pop-up menu appears when you right-click on an item in the list. Here you can bring up its options panel, as well as remove the item from the list. (Note that choosing **Options** is the same as double-clicking the item.) The **Copy** option copies the selected item along with its settings to a memory buffer. The **Paste** option will add the item to the list. This memory buffer is independent for each list window.



Layout List pop-up menu

GENERAL OPTIONS

Choose **Display** > **General Options** to bring up the General Options tab on the Preferences panel.



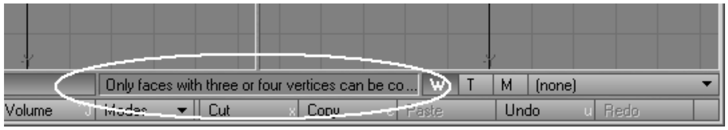
General Options tab

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 the user needs 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

NOTE
Modeler has its own **Alert Level** setting on the Interface tab of the Display Options panel.



Modeler information line

Content Directory

Use the **Content Directory** button to set this previously discussed and very important setting.

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 god, you can activate the **Hide Toolbar** option, which makes the toolbar invisible.

NOTE
To redisplay the toolbar, press the letter o key to bring up the General Options tab of the Preferences panel.

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 Selection

The **File Dialog** and **Color Picker** pop-up menus let you use custom LightWave dialogs for file loading/saving and picking colors. (See Chapter 3 for more information.) Selecting **Default** will use your standard system dialogs.

The **Color Format** setting determines the scale used where the color selector appears. **Integer** uses values 000 to 255, **Float** uses .00 to 1.00, and **Percentage** uses 0% to 100%.

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 composing 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

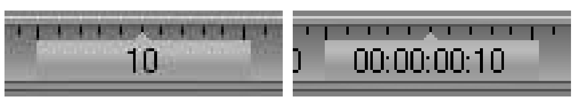
When **Parent in Place** is active, an item will maintain a constant position, size, and orientation when it is parented or unparented. (See Chapter 11, "Parent in Place.")

Left Mouse Button Item Select

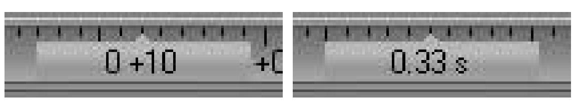
When **Left Mouse Button Item Select** is active, you may click on an item with the LMB to select it in a viewport. (Those of you with three-button mouses can always use the MMB.)

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**.



Left: Frame. *Right*: SMPTE Time code



Left: Film Key Code. *Right*: 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 plug-in architecture. If you had based a moving texture velocity so that it appeared to repeat every 30 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 (30 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 (**Rendering** > **Render Options**).

Frames Per Foot

When using **Film Key Code**, you can specify the **Frames Per Foot**.

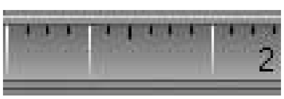
Allow Fractional Current Frame

When **Allow Fractional Current Frame** (formerly Allow Fractional Frames) is turned on, the frame slider can be placed at non-integral frame values, either by adjusting the frame slider, jumping to the previous or next key, or typing into the Go to Frame requester.

NOTE
Keyframes are based on time so they're always allowed to lie on fractional frames.

Show Keys in Slider

When **Show Keys in Slider** is active, a white 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’Unites.”) 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.

GENERIC PLUG-INS

Generic plug-ins are miscellaneous type plug-ins that more or less stand on their own. These can range from small utility modules to full-blown applications. They are selected on the **Scene** > Utilities: **Generics** menu.

Generic plug-ins have the unique ability (among Layout plug-in commands) to be added to menus and keyboard shortcuts (they appear in the Plug-in command group when configuring menus and keyboard shortcuts). See Chapter 5 for more information.

Comments

The Comments panel lets you add several lines of comments to an item in the Scene. The comment text is saved with the scene file.

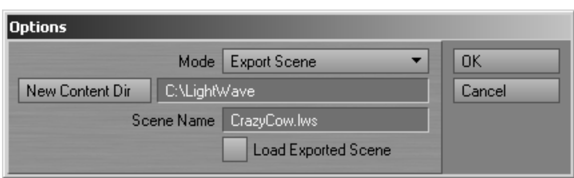
Comments panel

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 (for those rare occasions, right?).

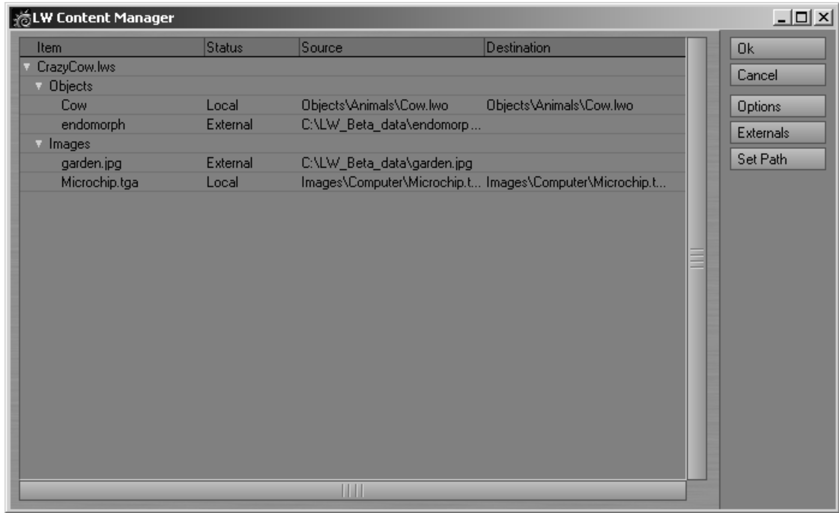
Export Scene Mode

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 the user. You can change the destination of local files, if you desire.

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.

FX_...

See Chapter 17 for information on the FX_Browser, FX_Linker, and FX_Property commands.

ImageLister

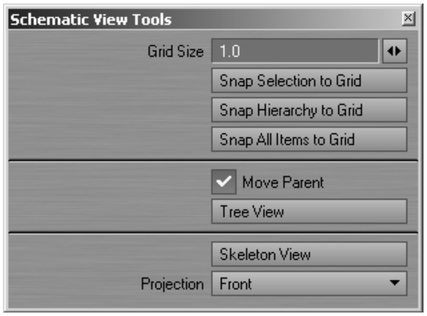
ImageLister outputs a text file with a list of the currently loaded images.

MD_Controller

See Chapter 18 for information on MD_Controller.

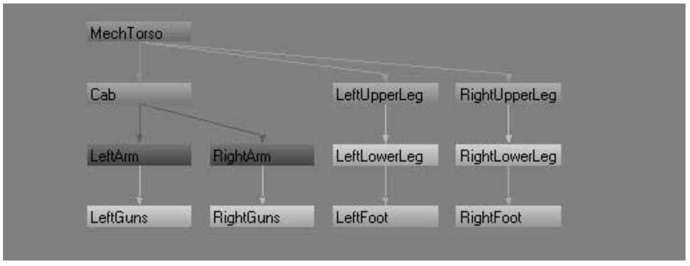
Schematic View Tools

This panel contains some tools to help you organize the scene items when using the Schematic View.



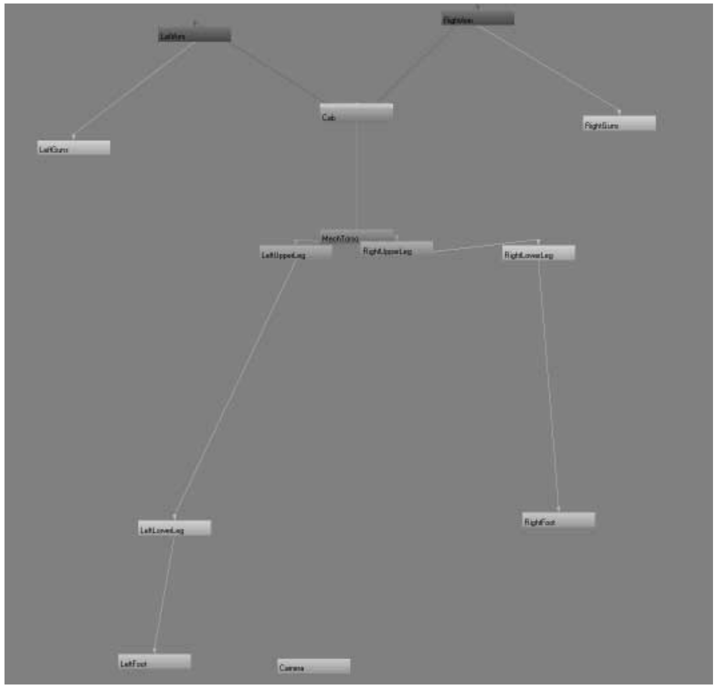
Schematic View Tools panel

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.



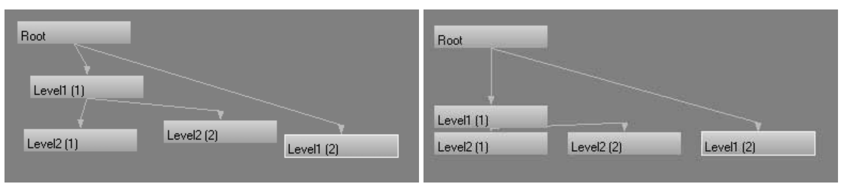
After using Tree View

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.



After using Skeleton View with Front Projection on a walking robot

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.



Before (left) and after (right) using Snap All Items to Grid

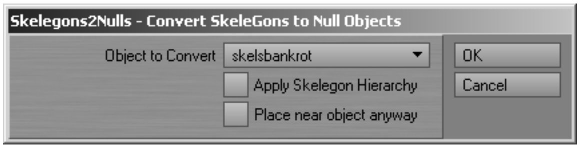
SelectGroup

SelectGroup selects all child items of currently selected scene items.

Skelegons To Nulls

Skelegons2Nulls converts skelegons from the object selected on the **Object to Convert** pop-up menu into null objects. It can optionally generate a hierarchy based on the skelegon relationships if **Apply Skelegon Hierarchy** is active. If **Apply Skelegon Hierarchy** is inactive, the nulls will be positioned relative to the Origin.

If you activate **Place near object** (**Apply Skelegon Hierarchy** must be inactive), the nulls are created in place. That is, wherever the object is located in 3D space. The object should also be positioned off the Origin and un-rotated for this option to make a difference.



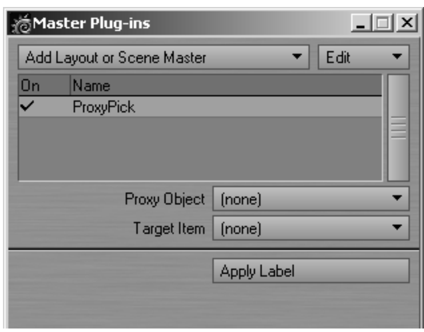
Use this plug-in if you want to use SockMonkey instead of bones, based on your skelegon set-up.

Other Generics

Sasquatch Lite, Shockwave3D Export, Spreadsheet, and Export Scene as VRML97 are discussed at the end of this chapter.

SCENE MASTER PLUG-INS

Choose **Display** > Utilities: **Master Plug-ins** to display the Master Plug-ins panel. Use the **Add Layout or Scene Master** pop-up button to add global-type plug-ins.



Master Plug-ins panel. The ProxyPick plug-in's options are visible on the lower portion of the panel

NOTE
In some cases, plug-ins will be listed on the pop-up menu that are not meant for direct use, but are used internally.

ItemPicker

ItemPicker 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.

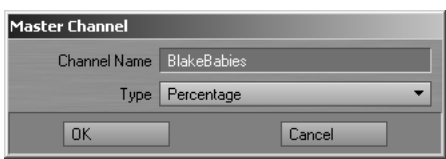
ItemPicker interface

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. See Chapter 12.

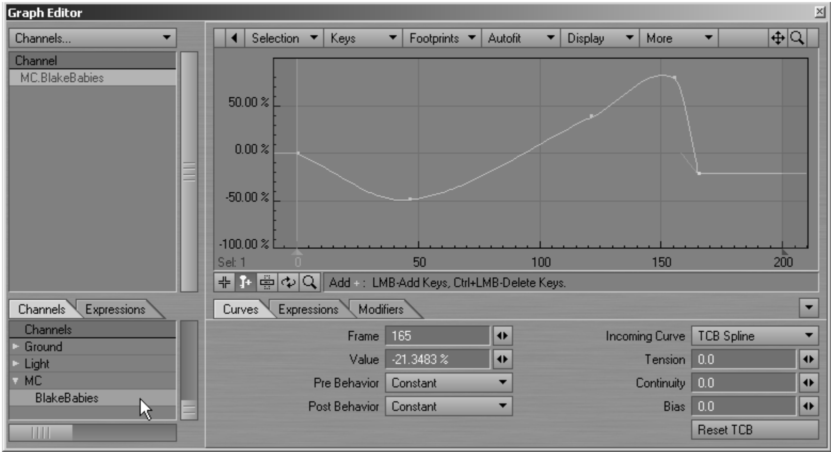
MasterChannel

The MasterChannel 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.



MasterChannel dialog

To create the master channel, add the plug-in 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 plug-in more than once to create multiple master channels.



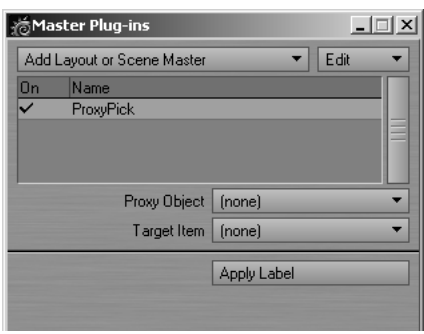
Custom channel added to MC group

ProxyPick

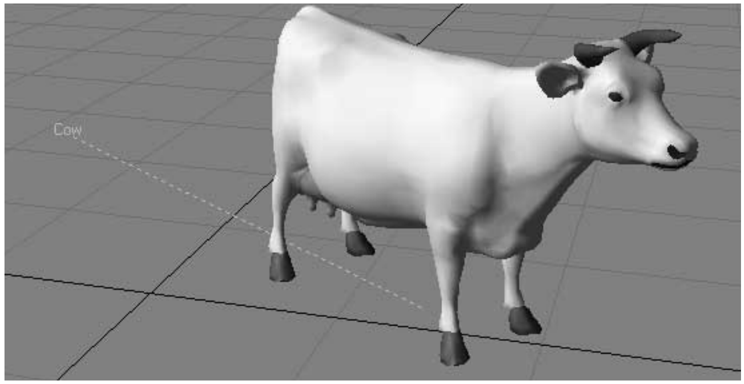
ProxyPick translates the *selection* of one object, known as the "proxy", into the selection of another, known as the "target." This is useful for picking small but crucial items out of complex, crowded scenes.

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.

Remember to disable this plug-in when you want to actually select the proxy rather than the target, as it is not yet telepathic.



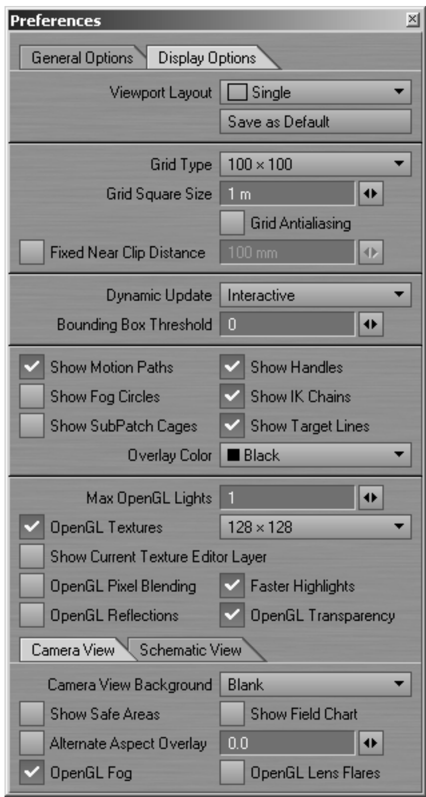
ProxyPick



ProxyPick using null as proxy and cow as target

GLOBAL DISPLAY OPTIONS

The Display Options tab on the Preferences panel (**Display** > Options: **Display Options**) controls the settings for display options that are global in nature.

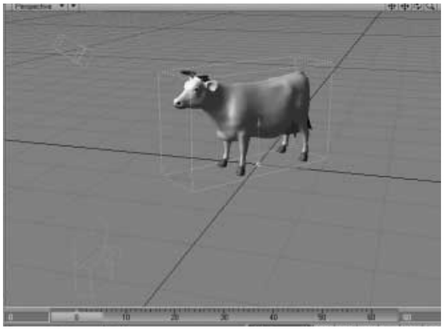


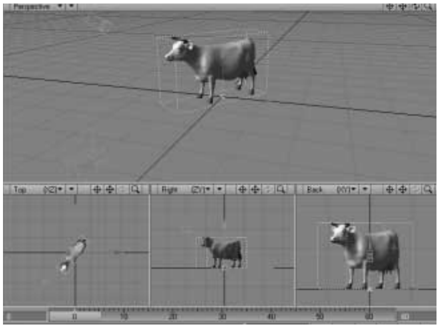
Display Options tab

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

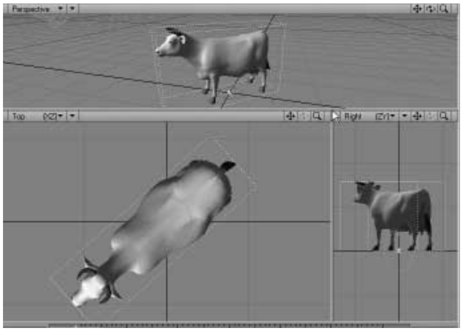
You can display more than one viewport by changing the **Viewport Layout** pop-up menu. All of the normal viewport options can be set independently and all viewports will update simultaneously.





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 customize the relative sizes of multiple viewports by dragging the borders between them.



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. 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.

NOTE
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.

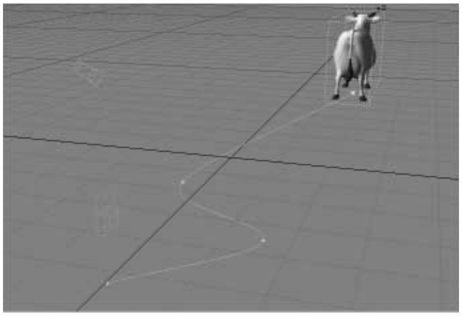
Bounding Box Threshold

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.

Display Characteristic Settings

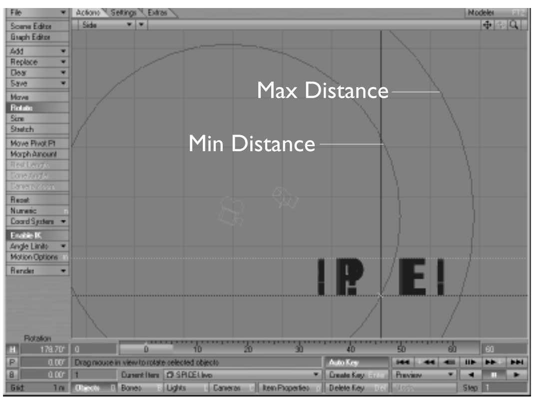
The **Show Motion Paths** option toggles the visibility of the motion path and keyframes for the current active item in Layout.

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

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.

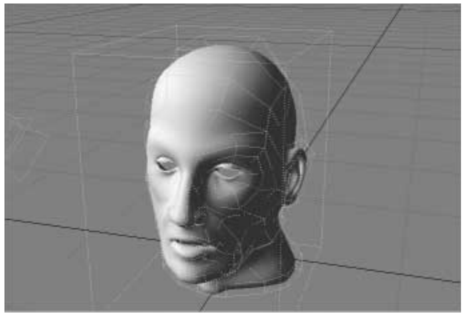


Fog circles

If you activate fog and the **Show Fog Circles** option, but do not see the indicator, check for the following factors:

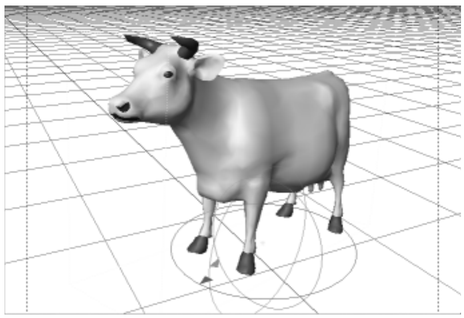
- Verify that **Fog Type** on the **Volumetrics** tab of the Effects panel is *not* set to **Off**.
- Verify that you are using an orthogonal view.
- Verify that you should be able to see the fog circles. Are you too close or too far from the camera for the circles to be visible? Use the grid as a guide.

When using SubPatch objects, you may want to see the SubPatch cage. Activate **Show SubPatch Cages**, if this is the case.



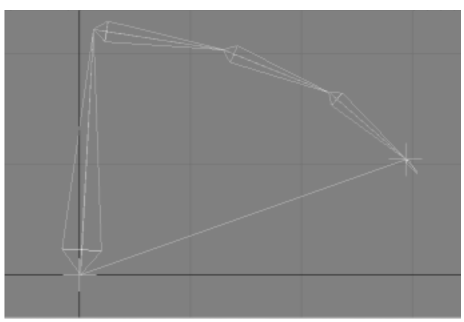
SubPatch cages

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. (See Chapter 7 for more information.)



Rotation handles

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.



IK chains

The **Show Target Lines** option turns on/off the line that connects to targeted items.

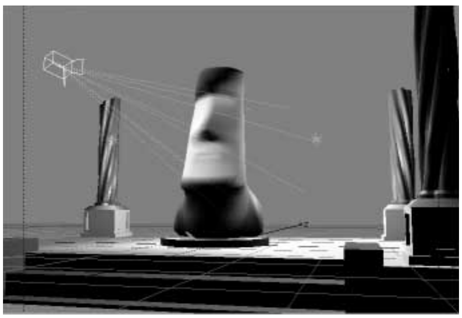
Overlay Color

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

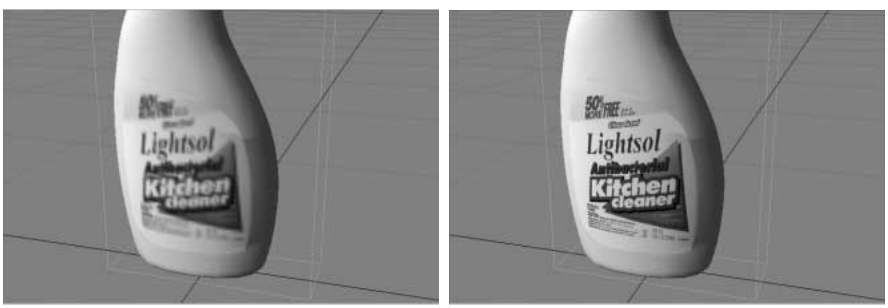
The following options affect the shaded OpenGL display characteristics.

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.



OpenGL light

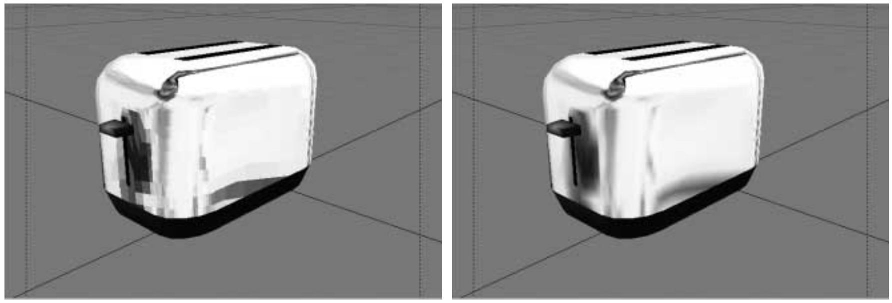
Activate **OpenGL Textures** to show image-mapped textures (not procedural) in the viewports. Use the size pop-up menu to select the pixel resolution (e.g., 128 x 128). Lower settings will update faster and use fewer system resources.



Left: 64 x 64. *Right* 512 x 512

Usually the first Color or Diffuse image map layer is shown in the Texture mode viewports. Enable **Show Current 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.



OpenGL Pixel Blending. *Left*: Off. *Right*: On

The **OpenGL Reflections** option lets you see the effects of image-mapped reflections (not ray-traced) in a Layout window.

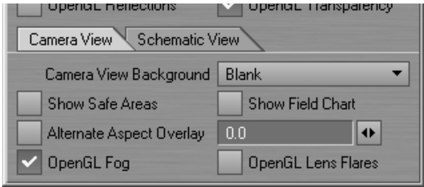
OpenGL reflection

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 **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.

Camera View Tab



Camera View tab

Camera View Background

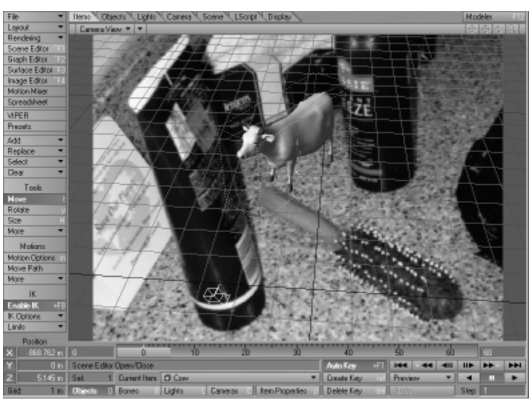
LightWave lets you choose how you want the Layout interface background to appear when using the Camera view (only).

Blank is the default choice and will show nothing but the normal blank gray 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 (**Scene** > Effects: **Compositing**) 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.



Background image

LightWave also allows you to use a preview animation as a background, if use the **Preview** option. This requires that you to have a preview animation loaded into memory of course. This can be one that was just generated or one that was loaded.

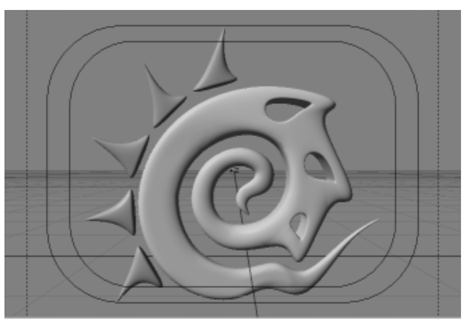
NOTE
Creating and loading a preview animation is discussed in Chapter 8.

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.

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.

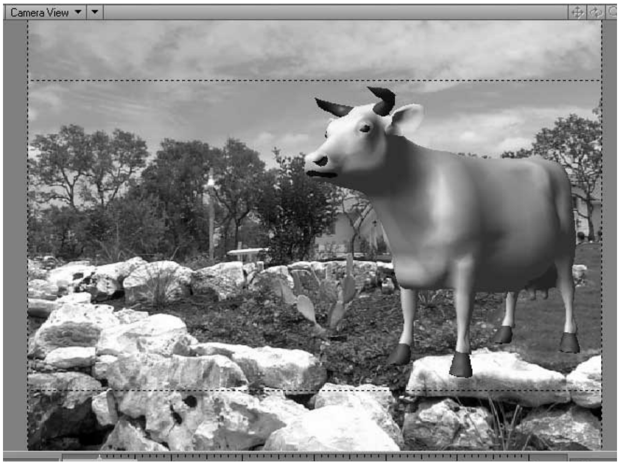


Safe areas

NOTE
These are reference guidelines and are not absolute positions.

Alternate Aspect Overlay

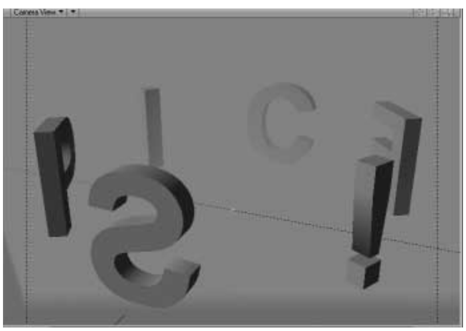
The **Alternate Aspect Overlay** setting lets you overlay the borders of an alternate frame aspect ratio, within the normal frame in camera viewports. This *information-only* display mode can be used to simultaneously compose shots for both film and television, for example. The default alternate aspect ratio is 1.85.



Alternate Aspect Overlay

OpenGL Fog

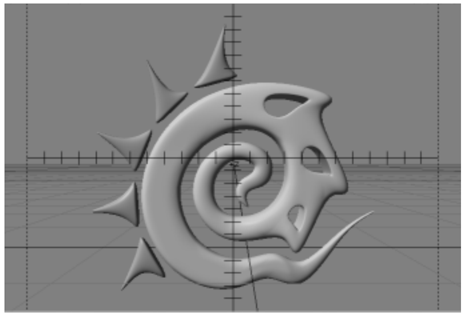
If you have the **OpenGL Fog** option active, you can see an approximation of your fog settings from the Camera View as well.



OpenGL fog

Show Field Chart

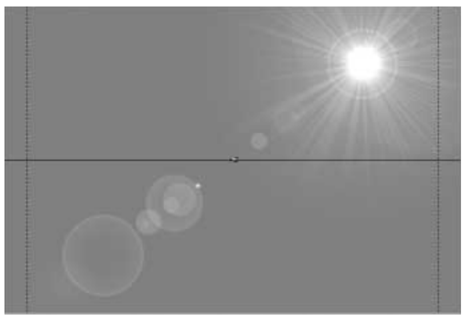
The **Show Field Chart** option turns on a crosshair chart similar to a 12-field chart as used in the film industry. It divides the camera view into quadrants, with each quadrant having 12 segments.



Field chart

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.

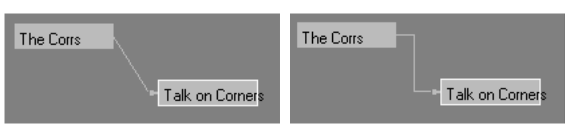


OpenGL lens flares

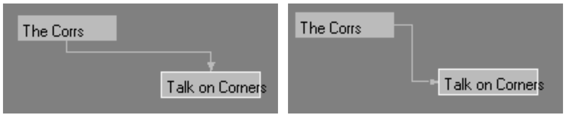
Schematic View Tabs

Several display options for the Schematic view appear on the Schematic View tab.

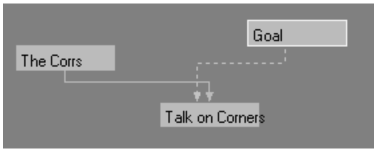
Schematic View tab



Link Style. Left: Straight. Right: Right Angle



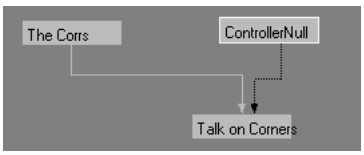
Link Direction. Left:Vertical. Right: Horizontal



Show Goal Links active

If the **Drag Descendants** option is active, moving a parent will move all its children as well.

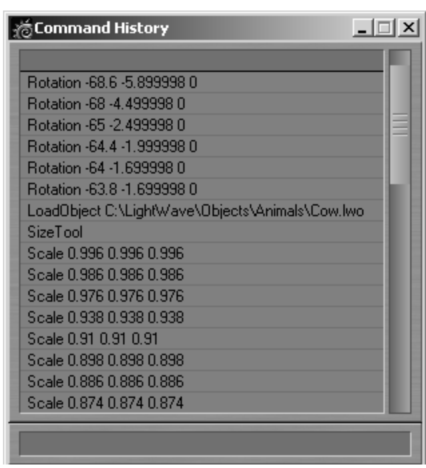
If the **Show Plug-in Links** option is active, motion dependencies from motion and channel modifier plug-ins are shown as dotted black lines.



Here, Cyclist has been added to Talk on Corner. The Cycle Controller for the plug-in is ControllerNull. Thus, Talk on Corner's motion is dependent on ControllerNull.

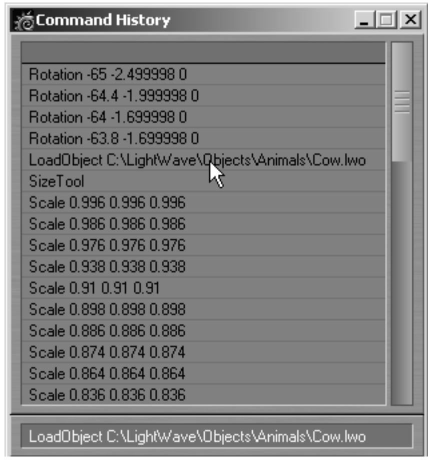
Layout Commands

The Layout interface is built on a *command system*. Buttons, keyboard shortcuts, plug-ins, 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 **Layout** > **Commands** > **Command History**.



Command History window

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.



Click an entry to select it

Alternatively, you can choose **Layout** > **Commands** > **Command Input** and enter a command into the dialog that appears.

You can save a list of commands to a file by choosing **Layout** > **Commands** > **Save Command List**.

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.

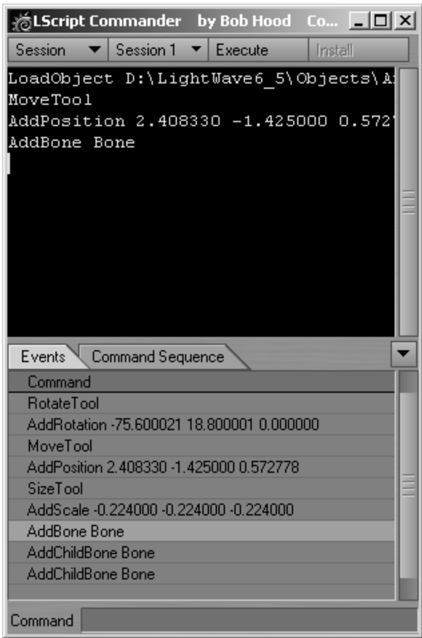
To execute commands in a programmatic fashion, use LScript. The documentation for LScript is included on the LightWave CD.

LSCRIPTS MENU TAB

The LScript menu tab contains tools for creating and running LScripts. as well as some actual LScripts that you can use in Layout. See the on-line LScript documentation for more information.

LScript Commander

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 LScript Command panel

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 that the currently selected item belongs to.

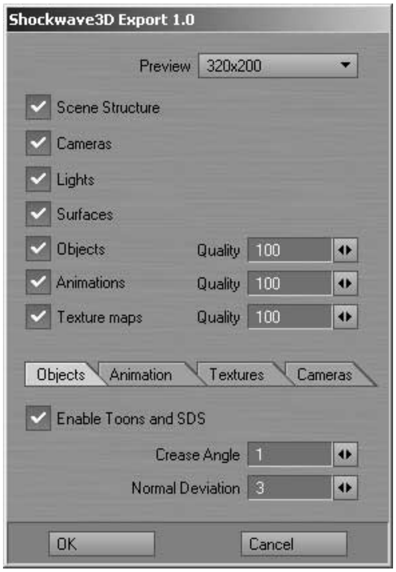
Select Children

This LScript will select all of the children (and their children) of the currently selected item.

SHOCKWAVE3D EXPORTER

The Shockwave3D Exporter (Scene menu tab) allows the user to export Layout's current scene as a Macromedia® 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 Macromedia® Director® as a Cast Member. With this exporter, the user can combine the content creation toolset of LightWave with the interactive functionality of Macromedia Director, creating an integrated solution for distributing multimedia content.

NOTE
Although the exported file can be previewed in LightWave, you must have Macromedia Director in order to create files for distribution. More information on Shockwave and Director can be found at Macromedia's website: http://www.macromedia.com/software/director/. Also, be sure to check out NewTek's Shockwave3D discussion in the NewTek discussion forum at http://www.newtek.com/discuss.



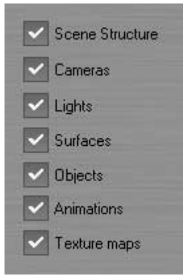
The Shockwave3D Exporter Interface.

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 labeled 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 familiarize 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.



Selecting which items to be exported.

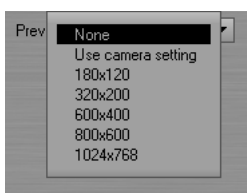
Quality Controls

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.

Adjusting the quality levels.

Preview Options

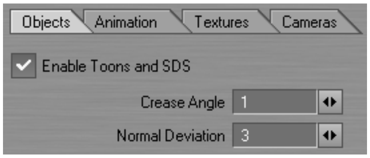
On the Microsoft Windows platform, the Shockwave3D Exporter has the option of playing a Shockwave-enabled preview window. You can simply choose the resolution of the preview window, and on export, a separate window will open displaying the exported file. This representation of the exported scene is how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



Selecting the Preview window size.

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's) objects. These features are automatically enabled and their parameters can be controlled in Macromedia Director. Unfortunately, object morphing is not supported in Shockwave3D.



The options tab for Objects

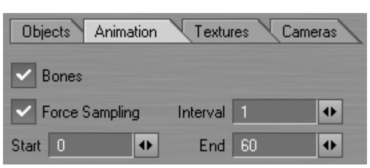
By activating the **Enable Toons and SDS** (cel shading and subdivision surfaces) feature, the objects are exported with the neighboring mesh information needed by these features in Director. Since this option increases the size of the exported file, deactivate **Enable Toons and SDS** if these features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

The **Crease Angle** determines how much of an angle is allowed between neighboring normals before polygons are joined together. This parameter is used as a geometric smoothing angle when objects are being optimized. **Normal Deviation** sets a limit that normals can deviate from when the exporter creates any LOD's. 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 optimization 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.



The options tab for Animation.

The user 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

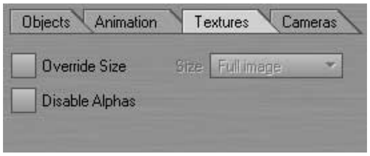
Animating bones is one of the more advanced features of the Shockwave3D Exporter. However, one thing to keep in mind is that Shockwave uses a different bones system than LightWave. The major difference between the two systems is how points get bone influences assigned to them. The animator should make 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

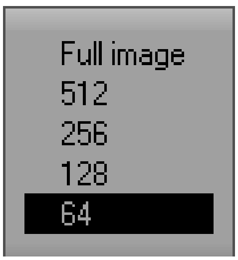
Surfacing is converted from LightWave to Shockwave as closely as possible. This includes, Color, Luminosity, Diffuse, Specularity, Glossiness, Reflection and Transparency. However, because the rendering methods (rendering vs. OpenGL) used in these two applications are setup differently, some tweaking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



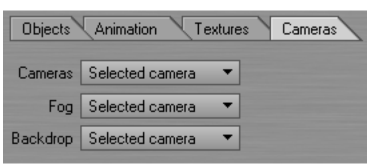
The options tab for Textures.



The image resolution values.

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 options tab for Cameras.

The Selected Camera values.

The **Camera** pop-up menu lets the user 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 the user 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.

VRML97 Exporter

The VRML97 Exporter plug-in (**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

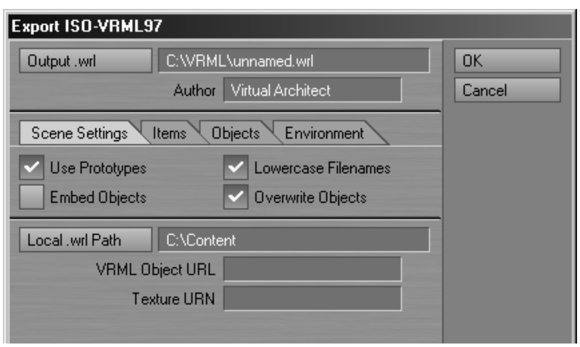
- Keyframed hierarchical animation
- Light intensity envelopes, including ambient
- Non-linear fog
- Color image texture mapping using projection or UV maps
- Solid, non-linear gradient and image backgrounds
- Support for SkyTracer warp image environments
- Particle animation with single-point-polygon 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

- 3-D Sounds
- Level-of-detail object replacement animation
- Object instancing
- Vertex color and lighting support
- Multiple custom viewpoints
- Custom VRML nodes
- Touch activated behaviors
- Viewer proximity activated behaviors
- Object visibility activated behaviors
- 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.

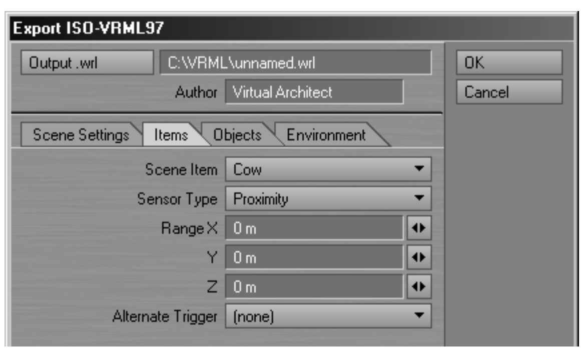
Use **Embed Objects** to include the geometry for all meshes in the main VRML97 World file. This may be more convenient, but for complex worlds, or reusing objects, it is less efficient. Using external object files allows the main world to load faster, and display bounding boxes while the objects are 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 .wrl Path is the file path on your machine where external VRML objects will be found and/or saved. This will default to the current LightWave Content 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.

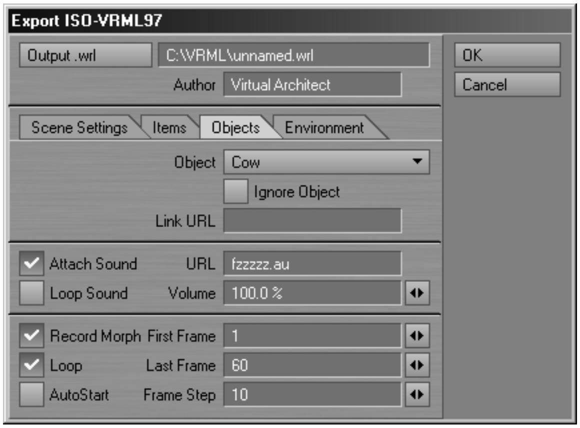


On the **Scene Item** pop-up menu, select a scene element to which you want to apply the settings on this tab.

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.

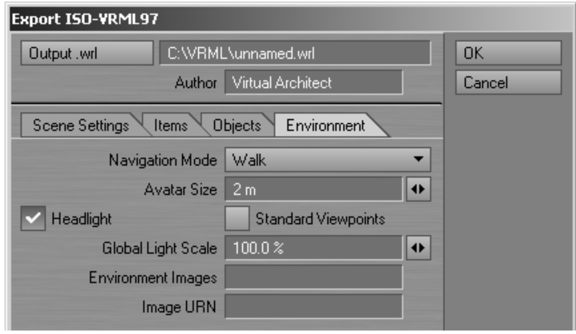
Use **Attach Sound** to add a sound effect, triggered with the selected object's animation. Enter the URL for the audio file triggered in the **URL** field. You can also set the volume and whether the sound should be looped once it has started.

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 filenaming 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 LightWave 3D'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 Behavior and Post Behavior set for the channels in the LightWave motion has a critical influence on the VRML behavior of an object. If the Pre behavior 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 behavior 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

Currently, double-sided surfaces are not supported in VRML97. Thus LightWave 3D 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 minimize 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 modernized, 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 2-D pixel positions in an image. They describe how the image lies on the 3-D 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 very 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 3-D 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.

PointSet objects 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

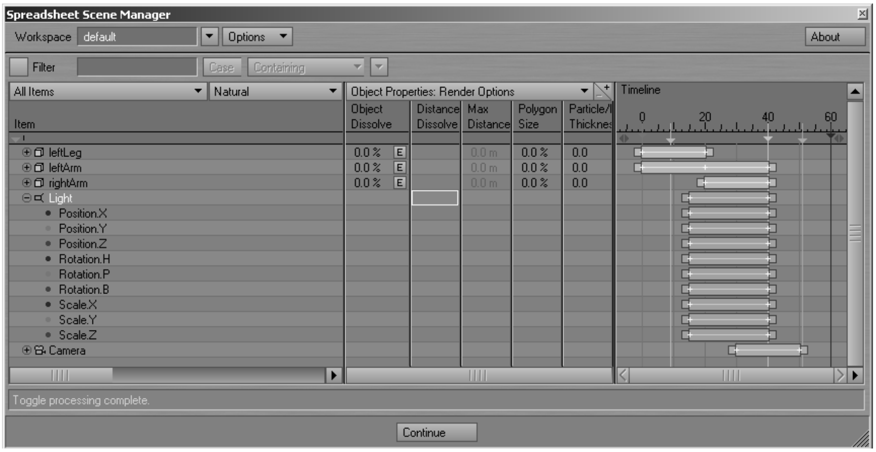
Many of the VRML attributes set in the exporter UI are stored in the Lightwave scene file as comments. These comments can be viewed and edited on an item-by-item basis with the Comments (Layout Generic) plug-in. These Comments should be formatted as <Tag>=<value> where the Tag is one of the following:

Item	**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.

SPREADSHEET SCENE MANAGER

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 (**Scene** > Utilities: **Generics** > **Spreadsheet**) organizes these properties and lets you edit a wide range of values quickly and easily.



The Spreadsheet Scene Manager interface.

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.

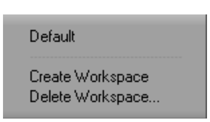
Workspaces

You can use the Spreadsheet to customize 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 plug-in.

Customizing the Workspace.

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.



The Workspace drop-list.

Creating a workspace:

1 Customize the Workspace to fit your display needs.

2 Apply a Name to the Workspace by entering text in the **Workspace** name field.

3 Select the **Create Workspace** command in the drop-list menu.

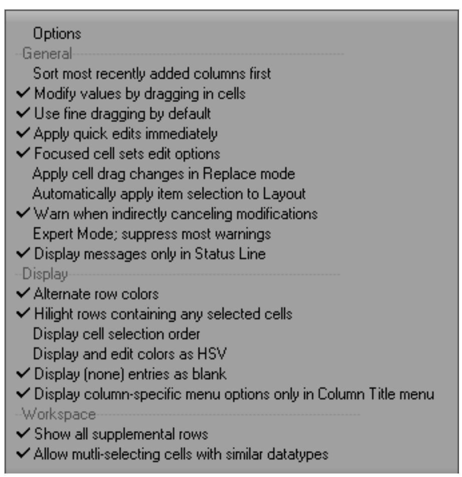
Once a Workspace is created, the name of the Workspace appears in the top portion of this drop-list menu. To revert to the settings saved in this Workspace, simply select its name from the drop-list. This will load the settings for the selected Workspace and reconfigure the interface to match the saved 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 customizing 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.



The Options drop-list menu.

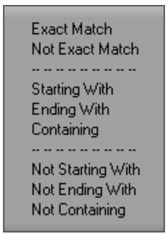
You can use this check-list (shown above), or you can access the tabbed interface by selecting the first item in the list, labeled **Options**.

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.



The Filter's parameters drop-list

To use a Filter:

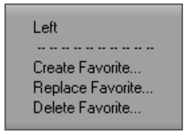
1. Create a scene with three nulls named leftLeg, leftArm, and rightArm.
2. Select **Scene** > **Generics** > **Spreadsheet**.
3. Activate the **Filter** option.
4. Enter the word "Left" in the text field.
5. Select the **Containing** filter parameter.

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.



The "left" Favorite 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 minimized 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.

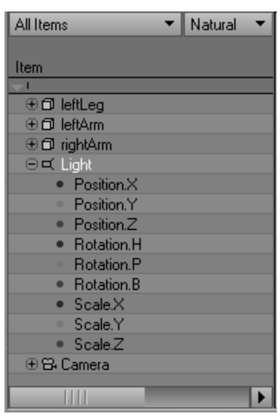
RMB: Opens the cell's Options menu.

In the Items List:

Double-click: Selects the object in Layout

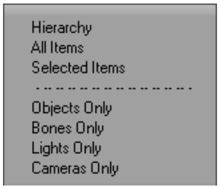
The Item List

Items in a scene are displayed in the Spreadsheet Scene Manager's **Item** list window.



The Items list portion of the Panel.

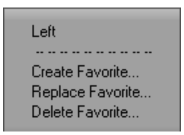
Items in this list are determined by any active filters and by the current Display Mode. Use the plus (+) and minus (-) icons to open and close an item's channel lists.



The Display Mode drop-list.

By selecting a Display Mode you specify the items for the Spreadsheet to 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.

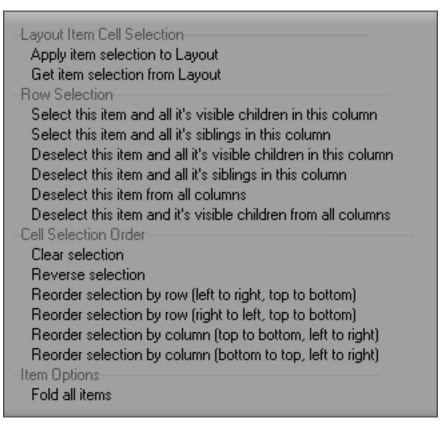
The **Sort Order** mode determines how the items are actually listed in the window.



The Sort Order drop-list.

The **Natural** and **Reverse Natural** modes list the items in either the order or the reverse order that they were loaded or created in the scene. **Alphabetical** and **Reverse Alphabetical** will display the items in a fashion that is easy to locate individual items.

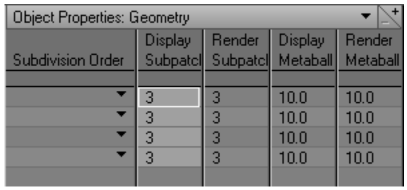
In the **Selected Items** display mode, you have two additional sorting modes: **Selected** and **Reverse Selected**. These modes refer to how the items were selected in Layout.



The Item Selection options menu.

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 organized using rows and columns, much like a spreadsheet used in business applications. That is why this plug-in is called the Spreadsheet Scene Manager.



The Property Editor.

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.

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.

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.

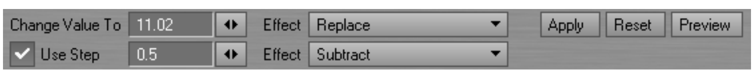
2 Using the LMB, edit a selected cell. Once you let go of the mouse button, the values for all selected cells will adjust accordingly.

The second method is to use **Edit** controls.

1 Select a cell as you did above.

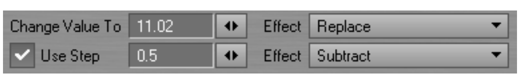
2 Use the **Edit** controls located at the bottom of the interface to adjust the values in the cells. These controls reflect the type of value you are editing, so they vary from cell to cell. The image below shows numerical data (integers, percents, distances, angles, time and floats) but it gives you an idea of how the editing controls work.

Controls for numerical data:



The Edit controls used 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 available Effect options.

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.

Envelopes can be edited in cells using the E button.

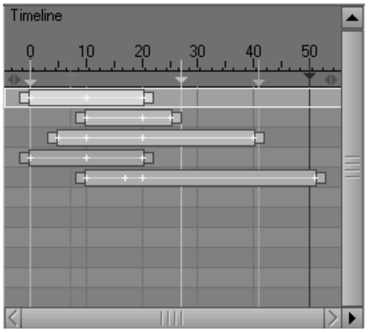
You can add envelopes by group selecting the cells and pressing the **E** button. You can also use the SHIFT+E combination to remove envelopes from properties.

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 Timeline Editor.

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.

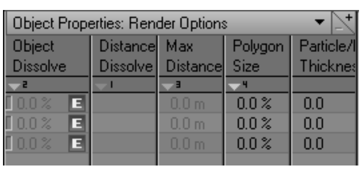
You can adjust the **Start Time** and **End Time** by dragging the black arrows beneath the frame hash marks.

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.



The flags lets you determine the new sort order.

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 7
Objects in Layout

Chapter 7: Objects in Layout

LOADING AN OBJECT INTO LAYOUT

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

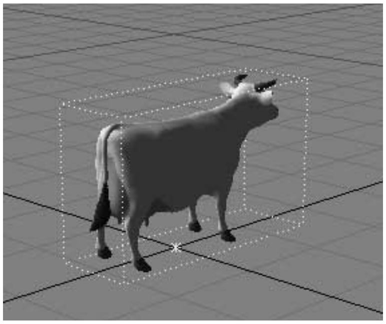
To load an object file into a scene:

1. Make sure your Content Directory is properly set. This will help with any images that are referenced in the object file.
2. Select **Items** > **Add** > **Objects** > **Load Object** (or **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. (See Chapter 20, “Multi-layer Object Standard.”)

As discussed in the prior chapter, 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.

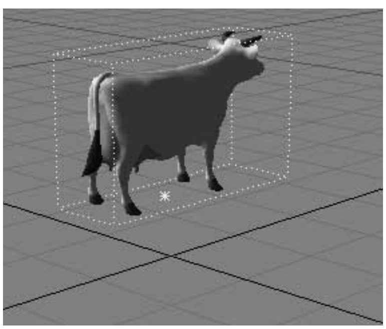


Object loaded at Origin

When you move an object, you are repositioning its local Origin relative to the world Origin. The local Origin provides a single point of reference for an object's position in 3-D space. The object's geometry is then applied relatively to the local Origin's new position.

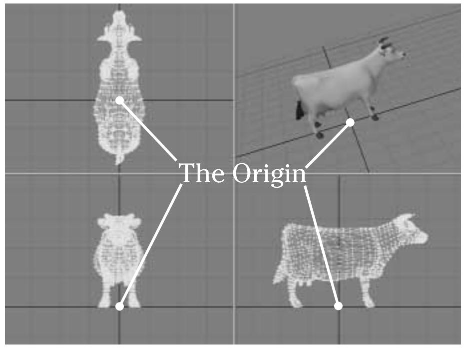
NOTE

Actually, an object's (numerical) position is first determined by its pivot point, discussed later, which can provide an offset amount to the local Origin's position. However, the pivot point is usually located at the local Origin, by default.



Object moved off the Origin

An object's local Origin is established when it is modeled and is most easily seen when the object is viewed in Modeler—the local Origin is at the point where all axes meet (i.e., equal to 0).



The location of the origin

Loading from Modeler

For information on how to load an object directly from Modeler, see Chapter 20, "Layout Communication."

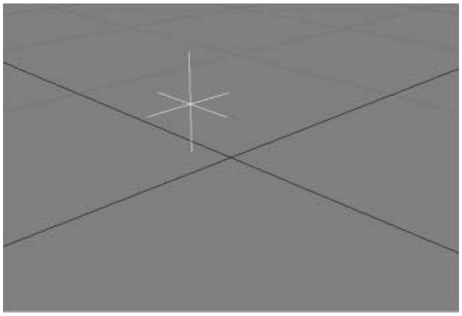
Loading a Single Layer

Choose **Items** > **Add** > **Objects** > **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.

THE NULL OBJECT: THE STAR OF OBJECTS

Null objects are special objects that can be used for many different purposes in LightWave. Generally, they are used any time you need a 3-D point that can be referenced. Nulls can act as a spot for lights or the camera to watch, or 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.

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.



A null object

To add a null object:

1 Select **Items** > **Add** > **Objects** > **Add Null**.

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

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.).

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.

SAVING AN 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.

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!

Saving a 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.

Object vs. Scene File

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. Loading objects from a scene is accomplished using **File** > **Load** > **Load Items From Scene**. Note that this will load all of the objects (and optionally the lights) in the scene.

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.

For example, say we had a flag object that was really just 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.

Saving a Transformed 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.

CLONING ITEMS

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

REPLACING, RENAMING, AND CLEARING ITEMS

The command in the **Items** > **Replace** menu lets you replace the current object with 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.

You can rename items, like cameras, bones, and lights, but not objects, using **Items** > **Replace** > **Rename Current Item**.

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

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 Object Layer** to replace an object with a single layer from a multi-layered object. A dialog will appear to let you specify the layer.

TOOLS AND COMMANDS

In Layout, tools operate in a manner similar to Modeler tools. Once they are selected, you need to select another tool to stop using it. However, unlike Modeler, one tool is always selected. You are familiar with the Move and Rotate tools, but there are many others like the Adjust Limited Region tool. You can use the SPACEBAR to cycle through the main tools.

NOTE
You can see a list of tools in the Tool group when setting up shortcuts and menus.

MOVING AN ITEM

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.

To move an item in a scene:

1 Select the item(s).

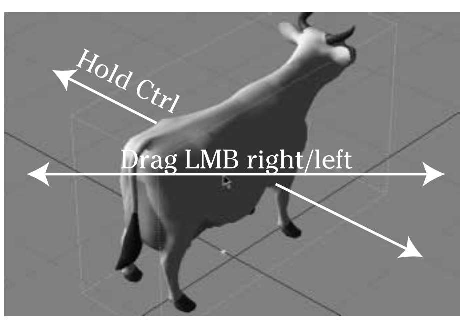
2 Click **Items** > Tools: **Move**.

3 Drag your mouse as described previously.

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 is you use (or how much you rotate the item).

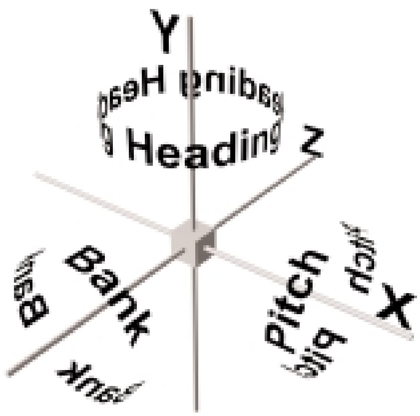
Movement	(Ctrl) Move
Left/Right LMB	X axis
Up/Down LMB	Z axis
Left/Right RMB	Y axis



ROTATING AN ITEM

When you adjust rotation, in contrast to adjusting position, the action is relative to the global axes around the item’s pivot point (discussed later). By default, the pivot point is at the item’s local Origin.

Rotation uses a similar three-coordinate system: **H**eading, **P**itch, and **B**ank. (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 (and sticking your tongue out. Nyaa nyaa!).



Rotation coordinate names and related axes

NOTE
Parenting (see Chapter 11) can dramatically impact the results of rotation.

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

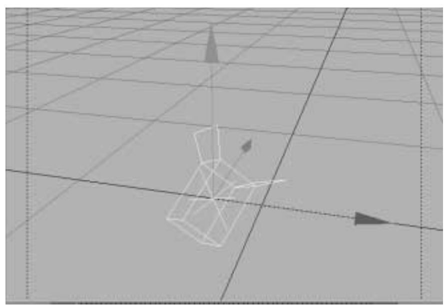
To rotate an item in a scene:

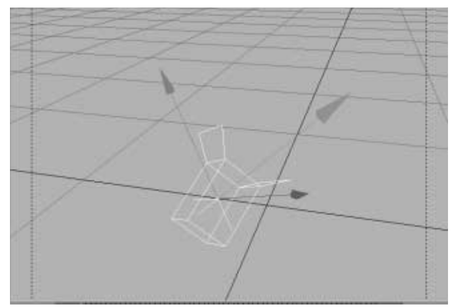
1. Select the item.
2. Click **Items** > Tools: **Rotate**.
3. Drag your mouse as described previously.

COORDINATE SYSTEM

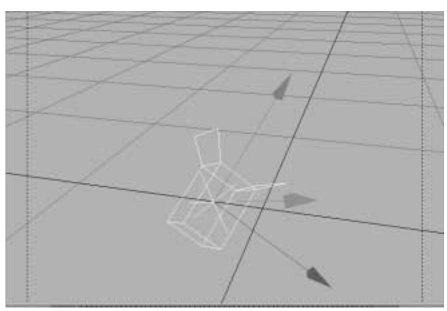
The **Coordinate System** setting (**Items** > Motions: **Coord System**) affects the Move, Rotate, and Move Pivot Pt tools. **World** allows easy movement based on the world axes, even for items deep within a hierarchy that contains rotated parents. **Parent** is for movement based on the axes of an item's parent(s). If an item has no parent then this setting is equivalent to **World**. **Local** is for movement based on an item's own axes (such as dollying a rotated camera along its view direction). For unrotated items it is equivalent to **Parent**. **Local**, which can still be temporarily activated by holding CTRL, as discussed previously.

Below, we have rotated a null and then parented a spotlight 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.





Left: World. *Right*: 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 spotlight’s rotation.

NOTE

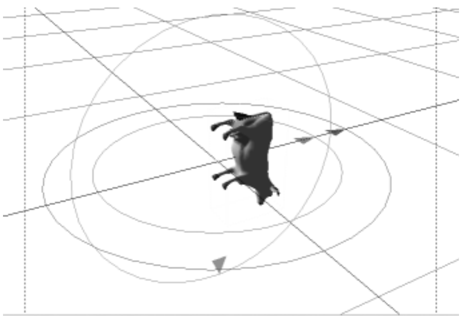
The position and rotation coordinate system settings are independent. To change the system, select either the Move or Rotate tool first.

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.



Two rotation handles lined up resulting in Gimbal lock

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.

SCALING OBJECTS

You can scale an object (but not lights or cameras) using the Size and Stretch functions. The difference between the two is that Size scales your object proportionately along all axes, while Stretch lets you scale each axis independently. Both operations happen around its pivot point (discussed later).

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. 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

To size and stretch an object:

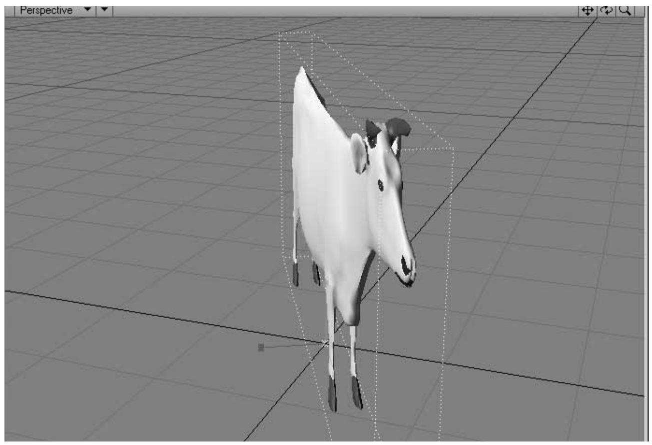
1 Select the item(s).

2 Click **Items** > Tools: **Size** or **Stretch**.

3 Drag your mouse as described previously.

SQUASHING OBJECTS

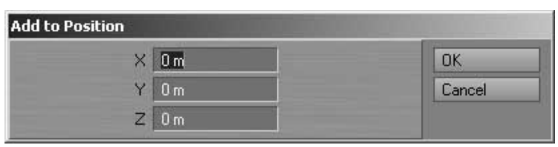
The Squash tool is similar to the Stretch tool; however, when one of the scale channels is modified, the other two channels are automatically adjusted to preserve the object's volume.



After using the Squash tool

ADDITIVE CHANGES

You can numerically add to Position, Rotation, and Scale values by using the functions in the **Items** > Tools: **Additive** menu group. They work like editing the input fields in the lower-left corner of Layout, but add to the existing values instead of replacing them.



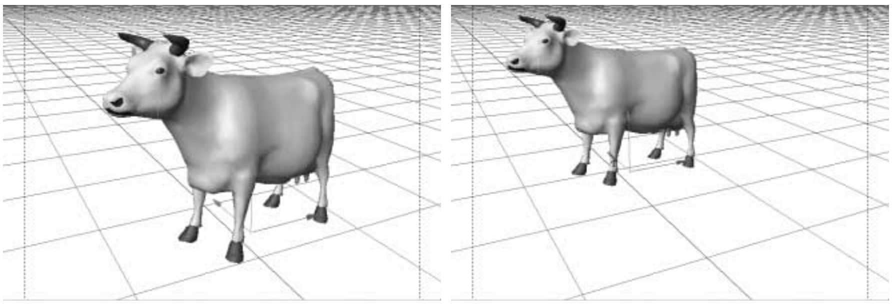
Add to Position dialog

ITEM HANDLES

If you have **Show Handles** active on the Display Options tab of the Preferences panel (**Display** > Options: **Display Options**), special handles will be available when the Move, Rotate, or Stretch tool is active. The handles can be dragged (anywhere along their axis lines) to edit items along the chosen axis. This can be a quicker alternative to using standard mouse dragging; moreover, movement along all three axes can be performed with just one mouse button and no qualifier keys.

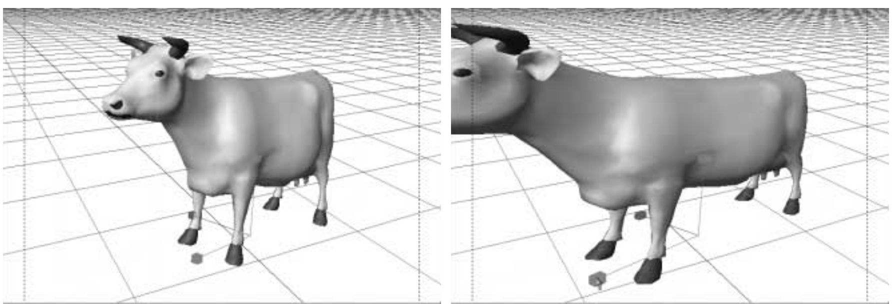
NOTE
Handles are drawn at a constant size relative to the viewports they appear in, regardless of the item's scale or distance, the viewport's zoom factor, or the grid square size.

When moving, three arrows will appear that point toward the positive side of the X (red), Y (green), and Z (blue) axes.



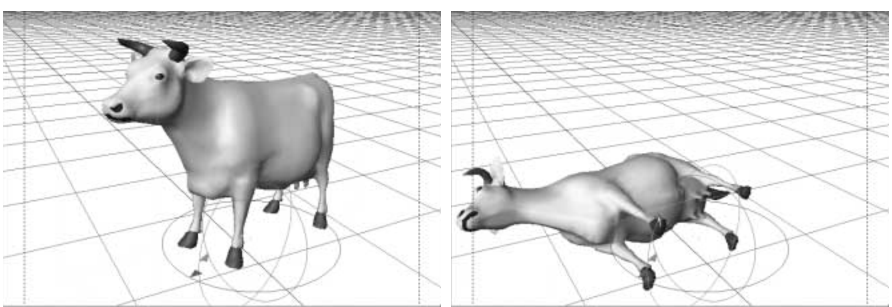
Move handles

Stretch handles (for objects) are similar but appear as cubes instead of arrowheads.



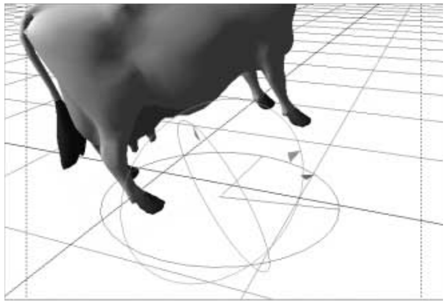
Stretch handles

When you rotate an item, heading (red), pitch (green), and bank (blue) rings will appear. You can manipulate the rotation discs by dragging anywhere on their circumference. The arrowheads—only visible using the Parent coordinate system—provide a reference for the related rotation value.

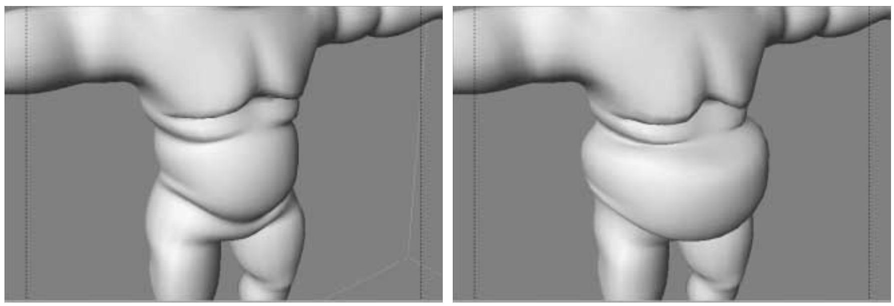


Rotate handles

Below, you can see how the rotation limit values on the Motion Options panel (**Items** > Motions: **Motion Options**), if any, are also visible on the discs (like a piece of pie) in the Parent coordinate system.



Pie area indicates rotation limits



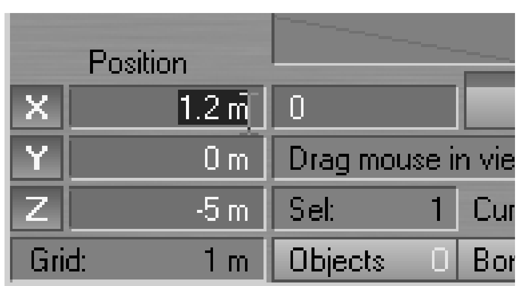
Love handles

NOTE

If **Show Handles** is active on the Display Options tab of the Preferences panel and you click on an item to move it, you may have to move your pointer away from the object, before you start dragging.

NUMERICAL ADJUSTMENTS

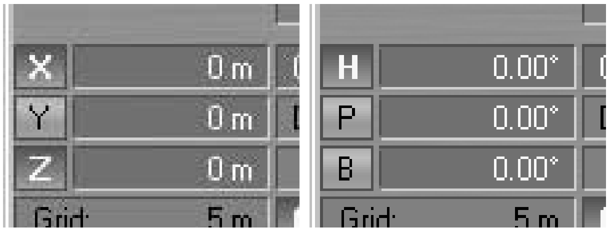
You can also change position and rotation settings numerically by entering the desired values in the XYZ or HPB fields, located in the lower left corner. The function of these fields change depending on the editing being done.



Numeric position fields displayed

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 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.



Left: Y inactive. *Right*: pitch and bank inactive

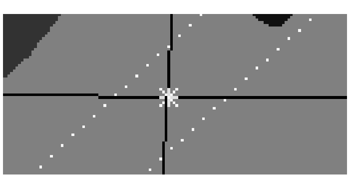
NOTE
You can also confine changes by using any of the orthogonal views (i.e., Front, Top, and Side). In these views, you can generally make changes only along the display's horizontal and vertical axes using your mouse.

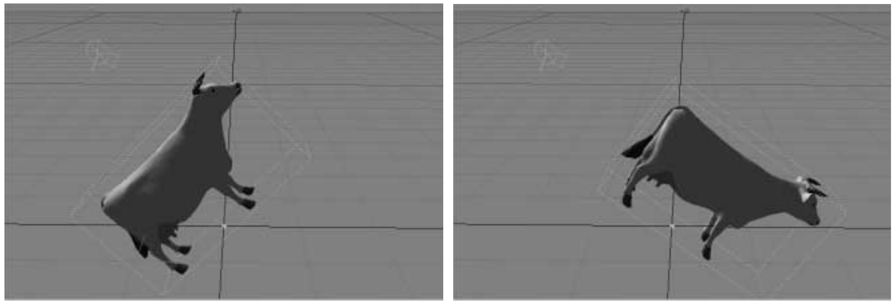
Pivot Point

WARNING
Understanding pivot points is fundamental to understanding LightWave.

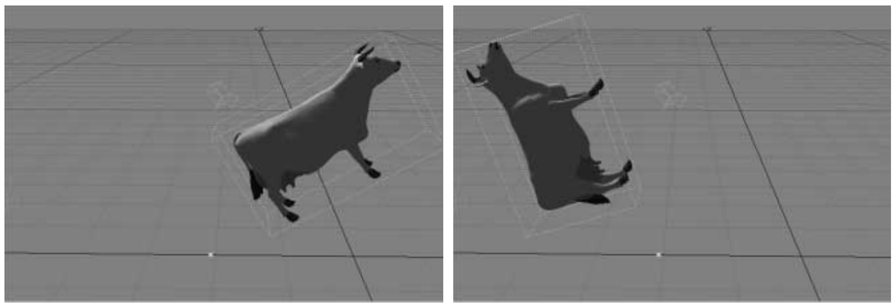
The *pivot point* is a point of reference used for all objects and does not correspond to any point used in an object's geometry. The pivot point is the center of position, rotation, and scaling. By default, it is located at the object's local Origin. The pivot point, a small yellow star, becomes visible when you select an object.

NOTE
You cannot animate the pivot point.


The pivot point



Rotation with pivot point at default location.



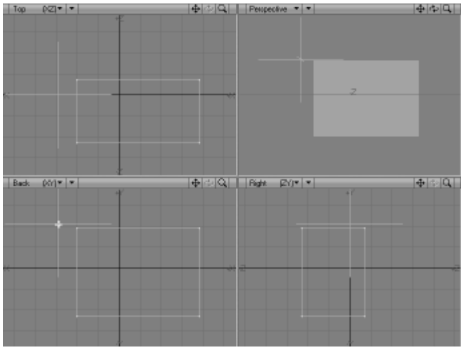
Rotation with pivot point moved behind the cow. Note the difference in the center of rotation.

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 should 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 **Items** > Tools: **Pivot** > **Move Pivot Point Tool** to activate the **Pivot Point** tool and move the crosshairs to a new position in any viewport. It is just like moving a point.


Moving the pivot point in Modeler

To move the pivot point in Layout:

1 Select the object.

2 Select **Items** > Tools: **Pivot** > **Move Pivot Point Tool**

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

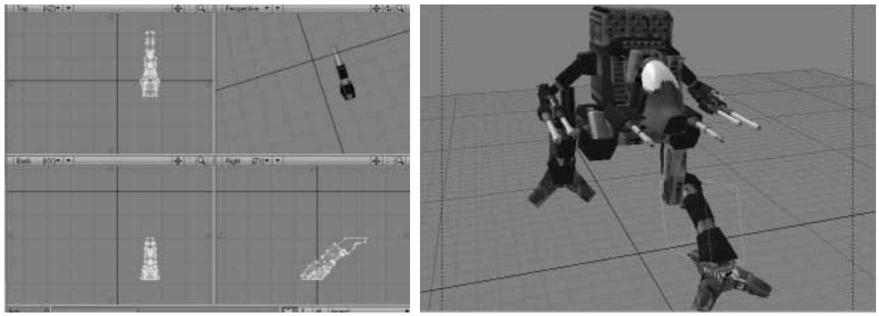
Setting the pivot point in Modeler saves its position in the object file. Setting it in Layout only saves the data in the scene file. As such, it is usually best to set your pivot point position in Modeler.

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 modeled 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 modeled 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.



Left: Left ankle object. Right: Pivot point moved so rotation is at the knee

Rotating the Pivot Point

You can also rotate the pivot point. This 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, as discussed later. You may want to stick with normal item rotation for other rotational needs.

To rotate the pivot point in Layout:

1 Select the object.

2 Select **Items** > Tools: **Pivot** > **Rotate Pivot Point Tool**.

3 Rotate the pivot point as you would any item in Layout.

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 **Items** > Tools: **Pivot** > **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.

RESETTING POSITION, ROTATION, AND SIZE

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 **Items** > Tools: **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.

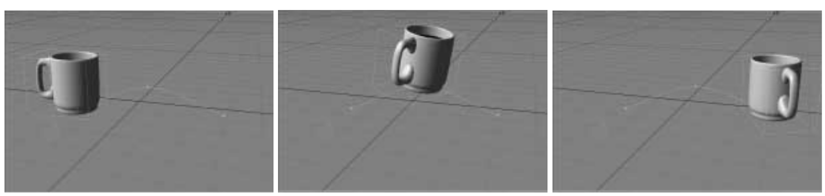
chapter 8

Keyframing

Chapter 8: Keyframing

In order for the items (cameras, objects, lights and bones) to move about, you must tell them where they must be located at any given frame during the course of an animation. You do this by creating key *poses*, known as *keyframes* (or simply *keys*), at strategic points in time throughout an animation. Once keyframes are created, LightWave automatically calculates the in-between (*tweens*) frames in smooth (or abrupt) movements.

For instance, you could create an animation where a coffee cup jumps up and down and spins around. This could be done easily by creating one keyframe at the point where the coffee cup begins to jump up; moving and/or rotating the coffee cup to the top of its leap and creating another key; then, creating a third key after you move it back to where it lands.



A cup at different keyframes

All items in a LightWave scene can have independent keyframes. The camera can do a slow *truck* in on a scene while the lights move up and down and the objects fly about. The keyframes for each of these items are independent of the keyframes for every other item. Moreover, you can even independently control each individual animation channel (e.g., XYZ positions and HPB rotations). Thus, an item's X position can change while the Y and Z positions remain constant, or Y and Z positions may move using totally different keys.

Generally, LightWave works in frames. A certain number of frames make up animations when played back on video (30 fps for NTSC and 25 fps for PAL) or film (24 frames per second). Of course, a single frame will make a still image such as those used for print.

By now, you should be fairly comfortable with positioning, rotating, and scaling objects in Layout. However, all of your efforts tweaking your objects just so will be wasted if you forget to keyframe the object once you are done—freezing the pose of the item at a point in time.

THE FRAME SLIDER

The Frame Slider, located beneath the viewport area can be dragged to change the current frame or shuttle through your scene. The input field to the left of the slider 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.

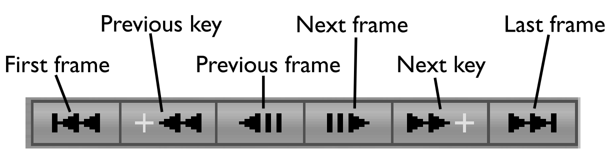
The frame slider

If you hold the ALT key while you drag, you can *pan* the current range of frames. The first and last frames will change, but the total number of frames between the first and last will remain constant.

NOTE
For additional information on frame slider options, see Chapter 6.

NAVIGATING THE SCENE

Besides dragging the frame slider to navigate through your scene, you can also use the transport controls.



Keyboard Shortcuts

There are also some keyboard shortcuts that can be used:

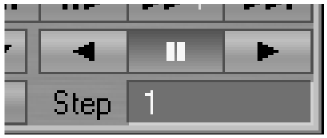
- Previous frame (LEFT ARROW key)
- Next frame (RIGHT ARROW key)
- Previous keyframe (SHIFT + LEFT ARROW key)
- Next keyframe (SHIFT + RIGHT ARROW key)

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 Step input field

NOTE
Also see the **Play at Exact Rate** option on the General Options tab of the Preferences panel discussed in Chapter 6.

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 activate.) How cool is that?!

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 (if you want to call it that).

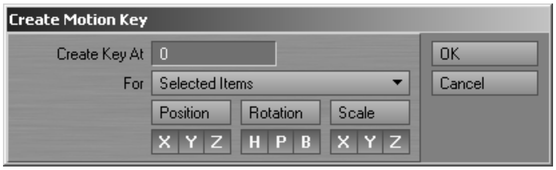
NOTE
Frame 0 is the default starting point, but you can create keys at frames at less than 0, if you need to.

To create a keyframe:

1 Select the item(s) and pose (move, rotate, size, etc.) it.

2 Click the **Create Key** button or press the ENTER key.

3 The **Create Motion Key** dialog will appear.



Create Motion Key dialog

By default the **Create Key At** field will contain the current frame. You can change the frame number to create the key—<u>using the animation channel values for the current frame</u>—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). (See Chapter 6 for more information.)

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. This will affect only existing keys. You can use the Undo function if you make a mistake, but remember that it is only a single-level undo.

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 (**Items** > Motions: **Move Path**).

DELETING KEYFRAMES

You delete keyframes in much the same way your create them.

To delete a keyframe:

1 Select the item. Normally, you will also go to the keyframe you wish to delete.

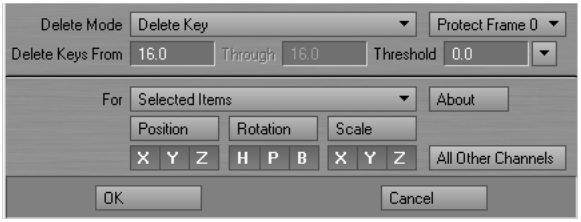
2 Click the **Delete Key** button or press the DEL key.

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 dialog

Delete Motion Key Generic Plug-in

Use the Delete Motion Key generic plug-in (**Scene** > Utilities: **Generics**) to delete keyframes, clear motions, delete ranges of keys, delete keys within a threshold, and more. Delete Motion Key can be used to completely replace the built-in Delete Key dialog. In fact, you may want to remap the DEL key to this plug-in.



Delete Motion Key dialog

To use Delete Motion Key:

1. Select the item(s) you want to delete keyframes from. Normally, you will also go to the keyframe you wish to delete.
2. Run Delete Motion Key
3. Choose a **Delete Mode**.
4. Enter a range of keys in the **Delete Keys From** and **Through** input fields.
5. Enter a **Threshold**.
6. Choose a **For** mode.
7. Enable or disable channels
8. Click **OK**.

Note that many of the above steps are optional. For example, to simply delete a keyframe at the current time (using the previous For mode), run Delete Motion Key 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 very 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 plug-ins (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 mean that any keyframe within 0.1 frames will be deleted. For example, if the you are trying to delete keyframe 20 with a Threshold of 0.1, all keys between the range of 19.9 through 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 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. Yikes!

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**. Likewise, you may 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.

CREATING A PREVIEW ANIMATION

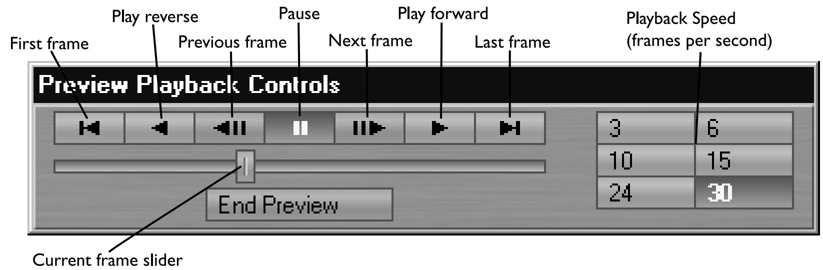
In addition to playing a scene using the transport controls, you can also create special preview animations. In most cases, these will give you a more accurate preview of your final animation.

To Create a Preview Animation:

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.
2. Make sure the viewport's **Maximum Render Level** is set as desired.
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.
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.

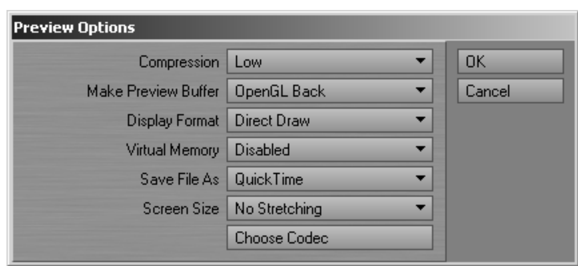
Here's a rundown of the Preview Playback Controls:



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 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 its a different size than what was used when the preview was made.

UNDO/REDO CHANGES

The undo function will undo the last motion change (Move, Rotate, Size, or Stretch tool). The button description will change, letting you know what will be undone (e.g., **Undo Move**) or redone (e.g., **Redo Rotate**).

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 ARROW key and then the LEFT ARROW 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.

NOTE
Undo is not the same as using **Reset**, discussed previously.

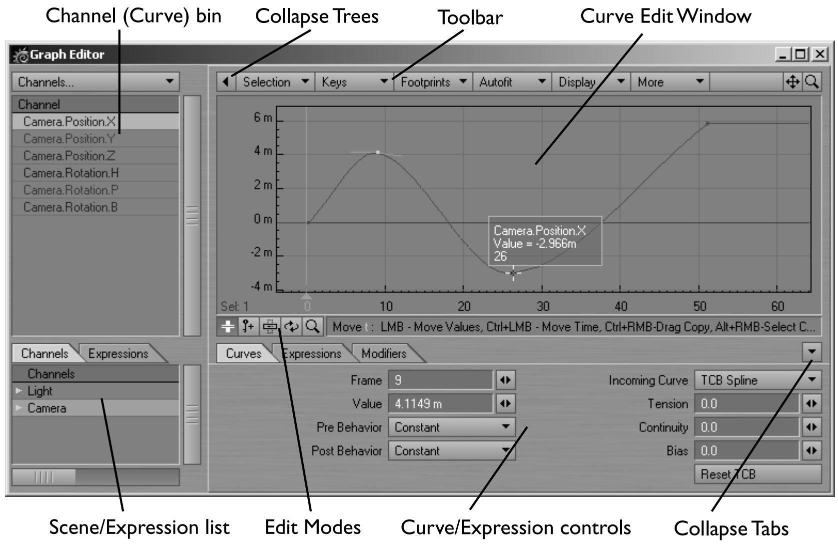
THE GRAPH EDITOR

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.

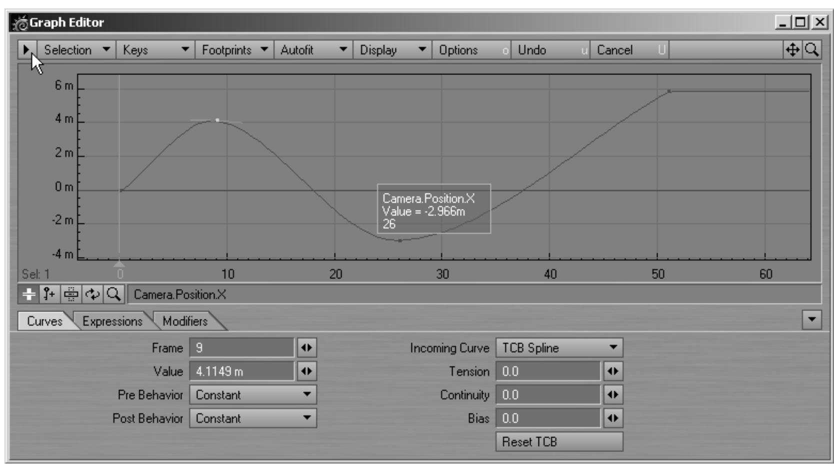
NOTE
The Graph Editor also controls all envelopes for options like light intensity, color, camera zoom, etc.

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 .

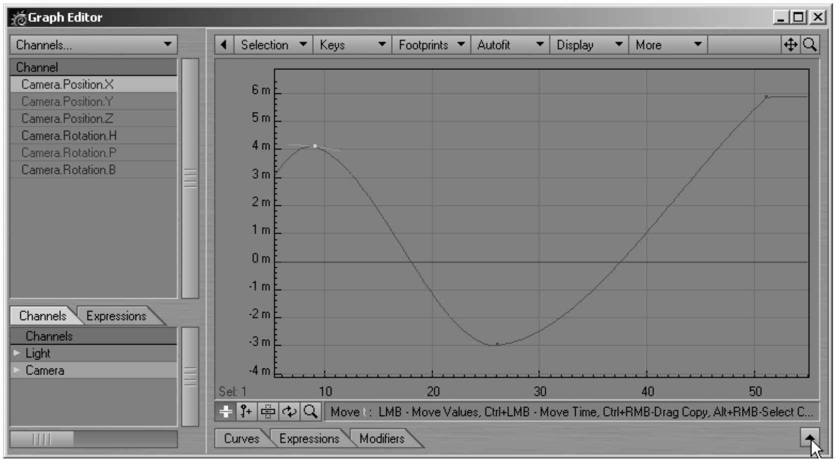
The Graph Editor has four major interface areas: the Channel (Curve) bin, the Curve Edit Window, the Curve/Expressions controls, and the Scene/Expressions list.



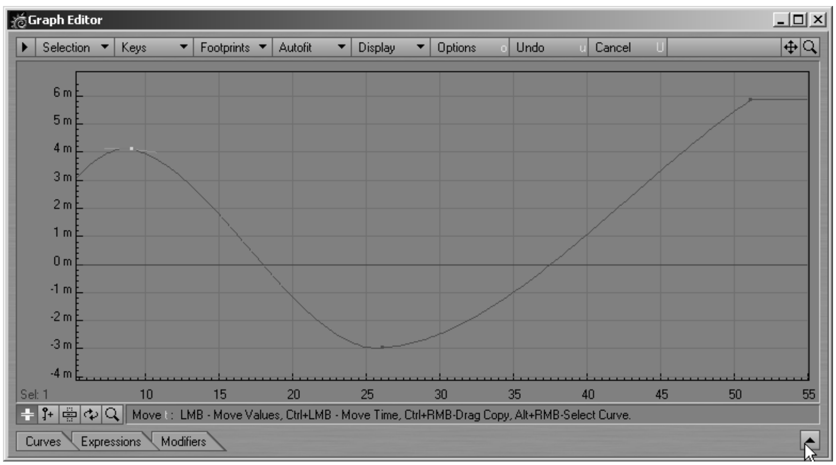
You can click the Collapse buttons to hide the left side and bottom areas of the panel. This will increase the screen real estate used by the Curve Edit window. When the left side is collapsed, the primary selected curve is shown in the information field.



Collapse Trees active



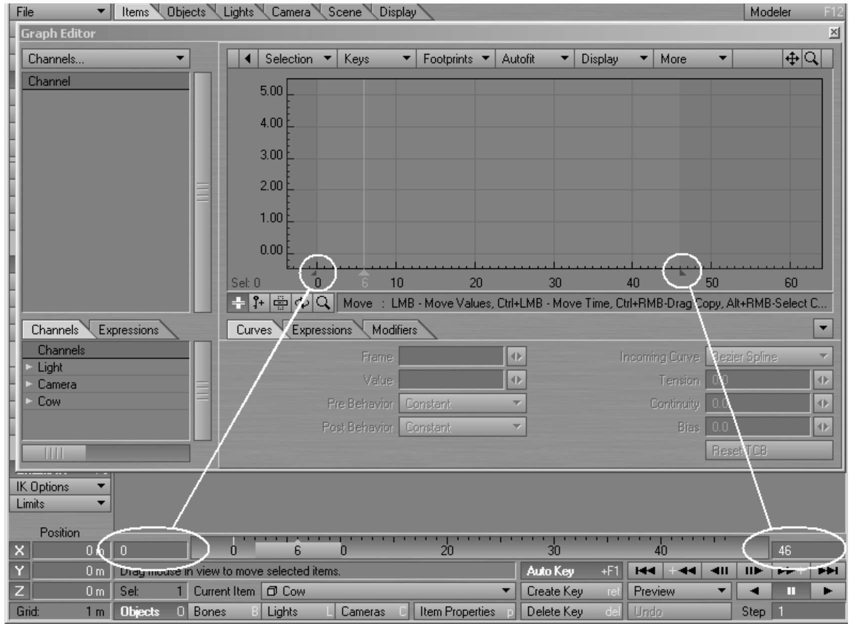
Collapse Tabs active



Collapse Trees and Collapse Tabs both active

Frame Range

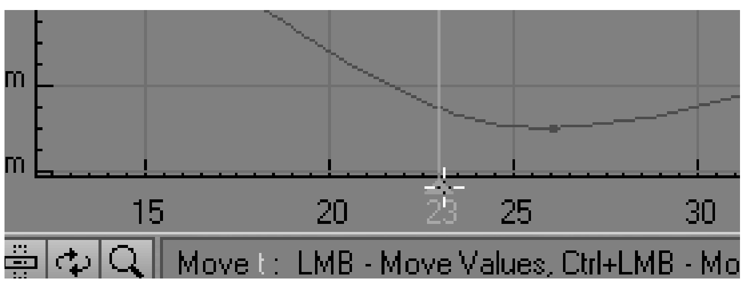
If frames *outside* the range of frames in your scene (i.e., less than your first frame or greater than your last frame) are visible in the Curve Edit Window, those areas will be slightly darkened. There will also be a small handle at the very bottom of each border. You can drag these to interactively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

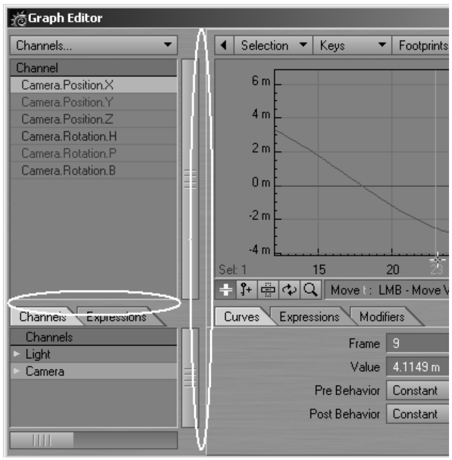


The time slider

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).



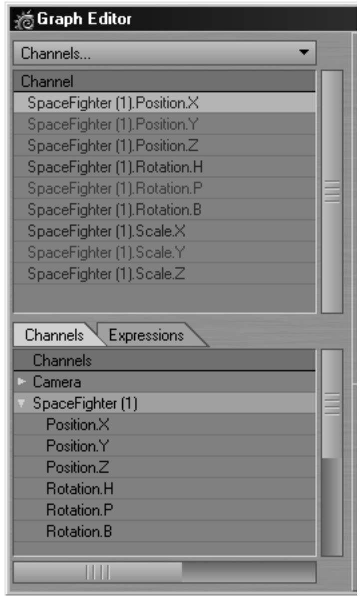
Draggable borders

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 the user which curves are available for display and edit. Deleting curves from this window <u>will not</u> 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:

1. Locate the desired item in the Scene list.
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).

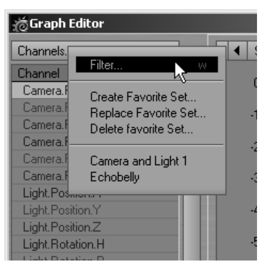


Adding channels to the bin

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.



Channel pop-up menu

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:

1 Make sure the desired curves appear in the Channel Bin.

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

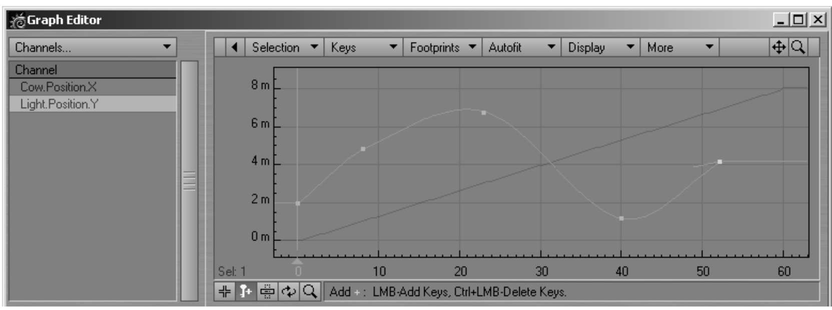
3 Enter an appropriate **Name** in the input field that appears and click **OK**.

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.

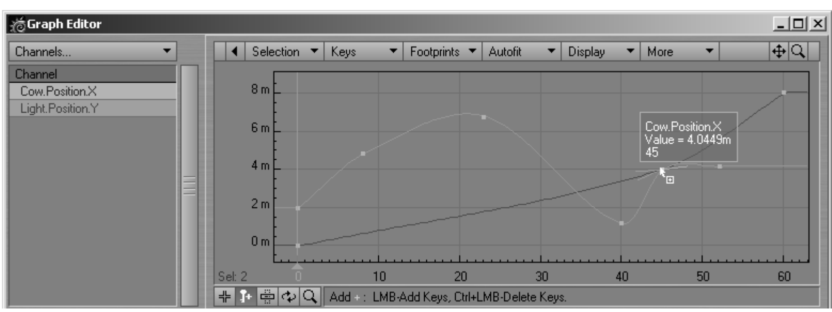
Editing Curves

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 *wacky* 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!



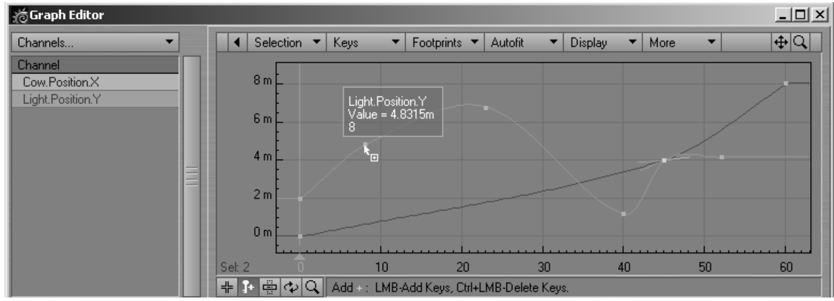
Editing the Cow.Position.X and Light.Position.Y channels

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.



Creating identical keys on different curves

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.

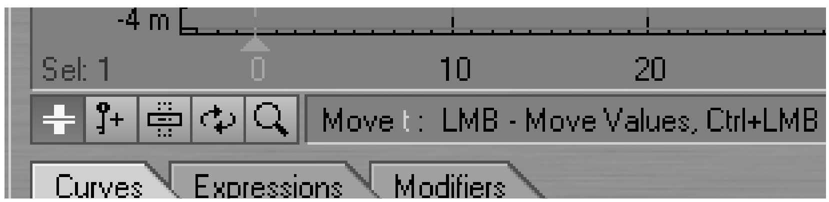


A key's data label

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.



Mode buttons

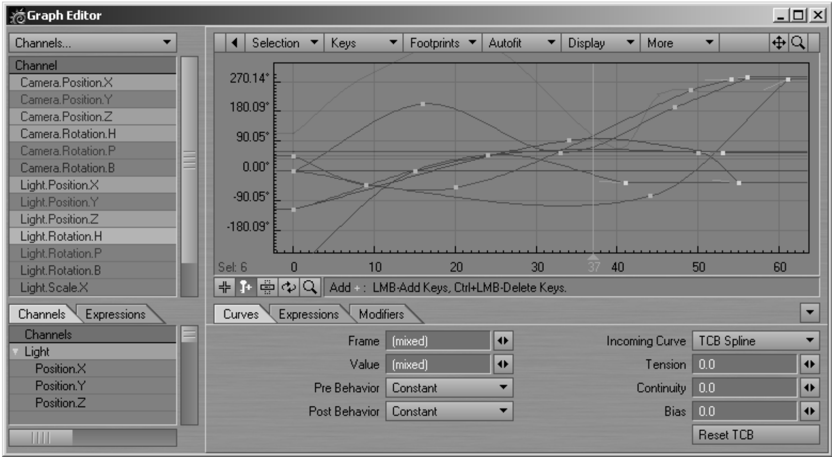
To select curves for editing:

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 arrow 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.



Selecting curves

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.

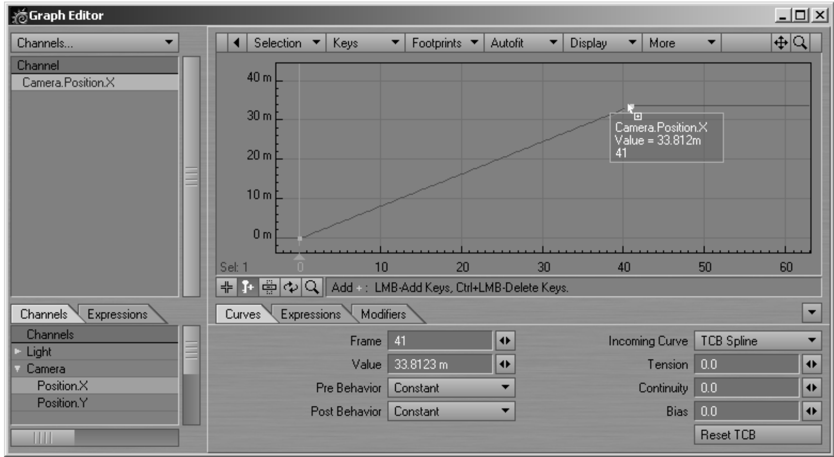
If only a single key is selected, you can use your keyboard left and right arrowkeys to cycle the selection of keys along curves.

To add keys to curves:

1 Select the channel curve(s).

2 Click the Add mode button.

3 Click on the graph at the desired frame (horizontal) and value (vertical) position. Before you release the mousebutton, you can drag up and down to adjust the value. Hold the CTRL key to adjust the frame.



Adding key to curve

To delete keys from curves:

1 Select the channel curve(s) and then the key(s).

2 Press the DELETE key.

In the Add mode, you can delete keys with CTRL + LMB.

To change key value/time:

1 Select the key(s) and click the Move mode button.

2 Drag with your LMB to change the value. Hold the CTRL key to change time.

To scale selected keys:

1 Select the key(s) and click the Stretch mode button.

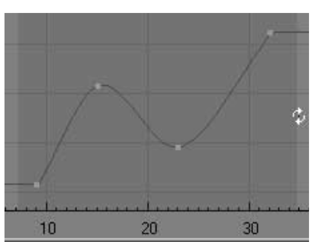
2 Place your mouse pointer at the position you want to use as the center of the scaling.

3 Drag using your LMB to scale value. Hold the CTRL key to scale time.

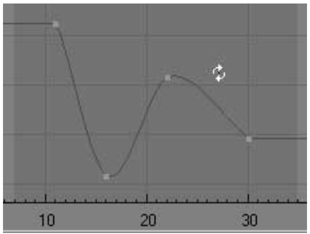
To roll a range of keys:

1 Select the Roll mode.

2 Drag out the time range using your RMB.



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.



Copying Keys

You may copy keys using your mouse or keyboard.

To copy selected keys with a mouse:

1 Select the keys for editing. Any mode except Roll can be used. Drag using CTRL+RMB.

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.

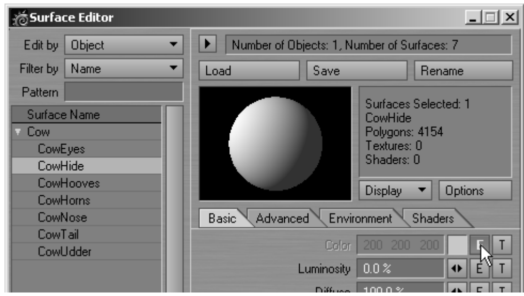
2b When you copy multiple curves, color-coded insertion marks will appear. With your mousepointer not directly over a curve, release to paste. Selected keys may be on different curves, but they can be copied only onto the same curve.

NOTE
Existing keys will be shifted over if you paste in more than one key (unless the Insert Overwrites Keys option is enabled).

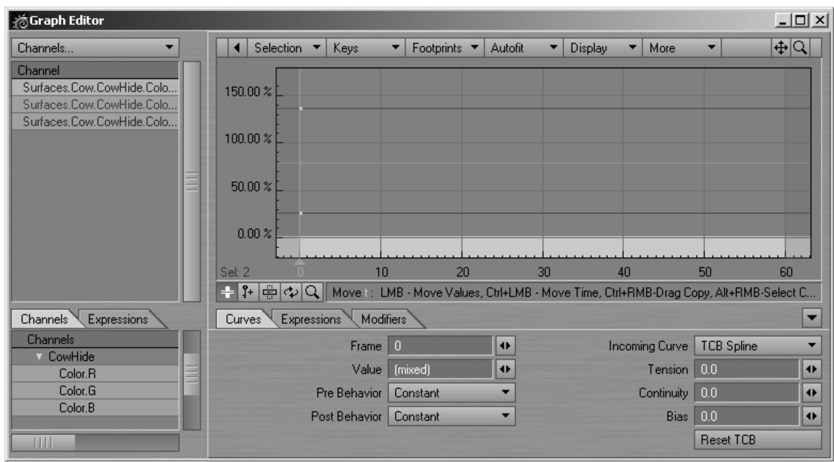
See also the Copy Time Slice command (**Keys** > **Copy Time Slice**), discussed later.

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.

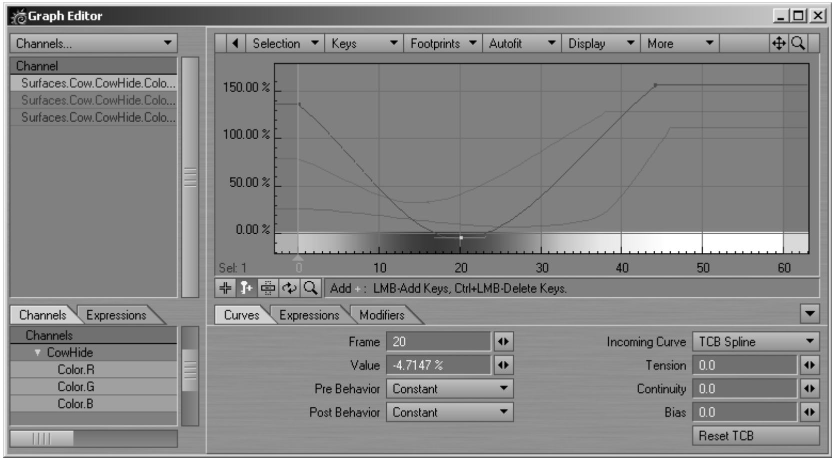


Clicking the Color Envelope button



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.



Color bar is a mixture of all three color curves

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.

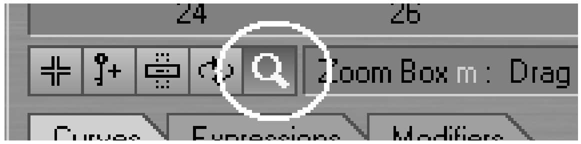
Adjusting the Curve Edit Window

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.

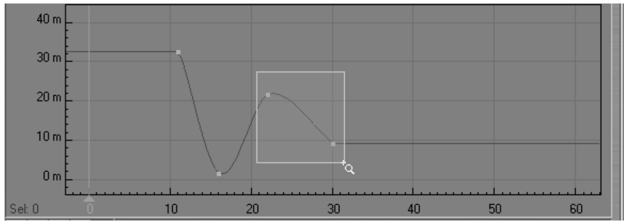
The drag and zoom drag button

Zooming and Panning

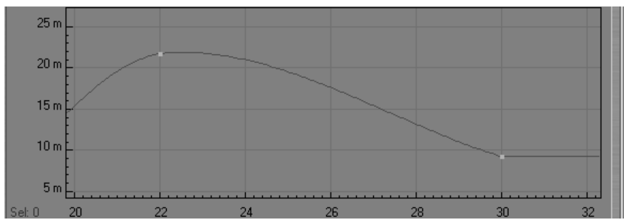
The Zoom Box tool button is the right-most Mode button. Click it and you can drag out a box on the graph and zoom in on that area.



The Zoom Box tool button



Dragging out the zoom box



After using the Zoom Box tool

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

This command will *add* the channels for the currently selected item(s) in Layout to the Channel Bin.

Get Layout Selected

This command will replace the contents of the Channel Bin with the channels for the currently selected item(s) in Layout.

Clear Unselected Channels

This command will remove any unselected curves from the Channel Bin.

Clear Channel Bin

This command will empty the Channel Bin.

Remove Channel from Bin

This command will remove all selected curves from the Channel Bin.

Invert Channel Section

This command will invert the selection state of curves in the Channel Bin.

Select All Curves in Bin

This command selects all of the curves in the Channel Bin.

Reset Bin Selection

This command leaves only the first curve in the Channel Bin selected.

Filter Curves

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

Use this to filter out everything but Position channels from the Channel Bin.

Filter Rotation Channels

Use this to filter out everything but Rotation channels from the Channel Bin.

Filter Scale Channels

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

This allows you to create a key, Layout-style. A dialog will appear where you can enter the **Frame** and **Value**.

Delete Selected Keys

This command will delete any selected keys.

Lock Selected Keys

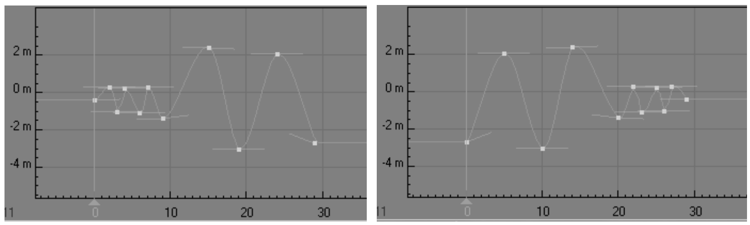
This locks the selected keys so they are uneditable. Locked keys are gray.

Unlock Selected Keys

Unlocks selected keys.

Invert Selected Keys

This flips the order of selected keys in time.



Before and after inverting keys

Snap Keys to Frames

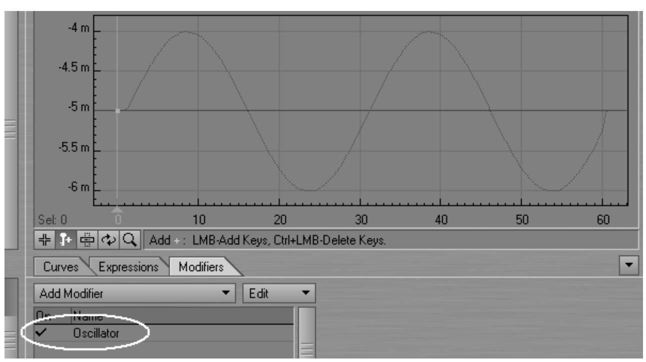
This command causes every selected key that falls on a fractional frame to *snap* to the nearest whole frame.

Set Key Values

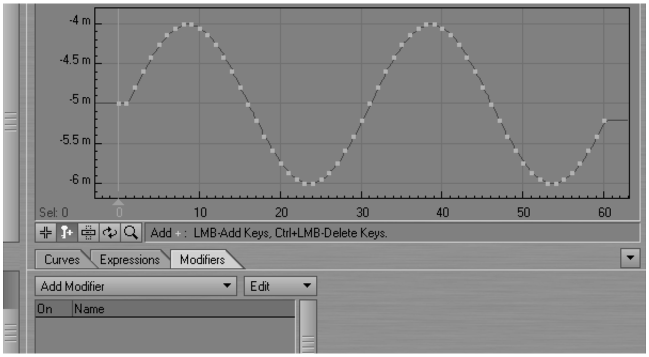
This command will bring up a dialog where you can enter a new **Value** for the selected keys.

Bake Selected Curves

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.



Single-key curve being modified by the Oscillator modifier



After Bake Selected Curves command

Copy Time Slice

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

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

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

This creates a key on the curve that matches the footprint value at the current time. Essentially, its like doing a **Copy Footprint Time Slice** operation and immediate paste.

Copy Selected Keys

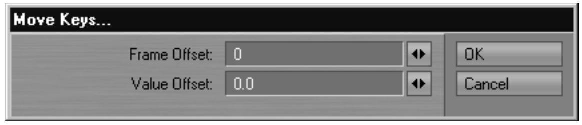
This copies selected keys to a memory buffer. To paste the buffer, you must have your mouse pointer over a curve and use the **Paste Keys** command on the Curve Edit Window pop-up menu (CTRL+SHIFT+LMB), discussed later.

Add to Key Bin

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.

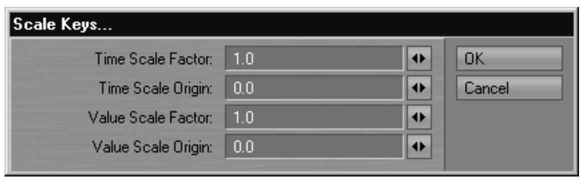
Numeric Move

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

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

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

Shifts the values of the selected keys to the right without affecting their time. This command works only with contiguously selected keys.

Reduce Keys

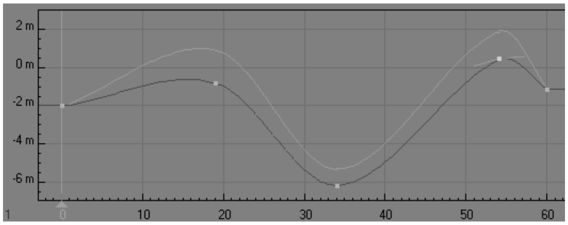
The Reduce functions let you remove consecutive keys that are within a certain threshold value of one another. The threshold is set by selecting **Set Key Reduction Threshold**.

There are two modes: **Reduce Keys** and **Reduce Keys (Recursive)**. 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** creates the imprint that will be visible as a shade of the real curve's color (once you make a change). **Pickup Footprint** erases the footprint for the current curve(s). **Backtrack Footprint** will restore the curve to its *footprinted* state.



Lighter curve is footprint

Toolbar Autofit Menu

This group of commands affect the range of frames and values shown in the Curve Edit Window.

Autofit

You can automatically scale the graph display to show all of the selected curves with this command.

Autofit Selected

You can automatically scale the graph display to show all of the selected keys with this command.

Autofit By Type

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

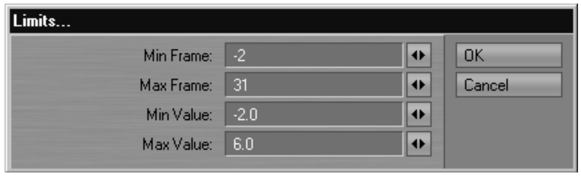
This works like Autofit By Type, but only scales the graph vertically, retaining the current frame range.

Toolbar Display Menu

The Display menu contains commands that affect the graph display, as well as global options.

Numeric Limits

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.



Numeric Limits dialog

Go To Frame

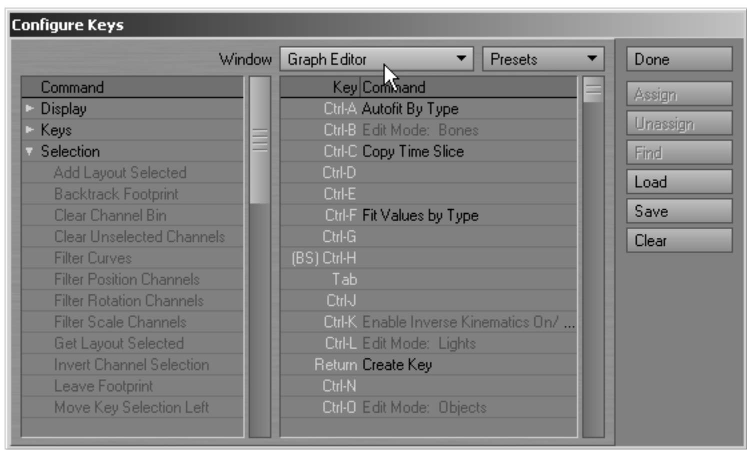
This action sets the current frame to the entered value. It also centers the Curve Edit Window around that frame.

Reset Graph

This simply resets the graph to default frame and value ranges.

Edit Keyboard Shortcuts

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**.) See Chapter 5 for information on how to set keyboard shortcuts.

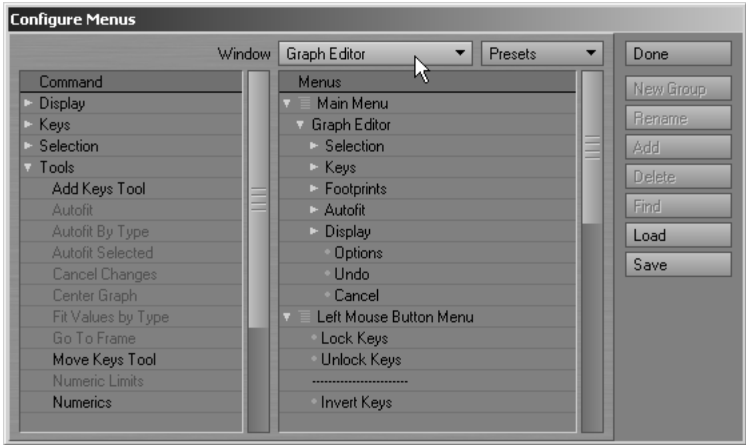


Graph Editor Keyboard Shortcuts

Edit Menu Layout

The Graph Editor has its own set of menus. These can be customized 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 (now famous) 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.



Edit Graph Editor menus

Insert Overwrites Keys

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

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

This option will add an extra amount of outside space when you use autofit commands**.**

Allow Fractional Keyframes

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

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

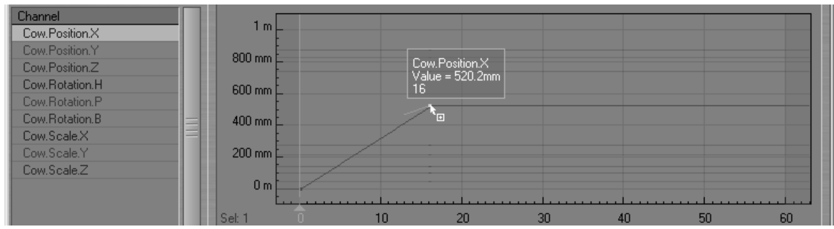
This option will scroll the Curve Edit window to keep the frame slider centered. This is useful for watching the curves scroll by as Layout is playing.

Allow Passthrough Keys

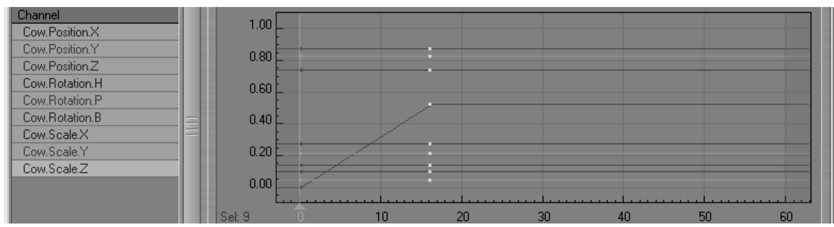
This option lets you drag keys *through* each other in time. Normally, you stop at a neighboring key.

Lock Motion Keys in Time

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.)



Here we have added a single key to Camera.Position.X



With all motion channels selected, you can see that keys were added to all of the siblings, locking them in place

Move No Keys Sel

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

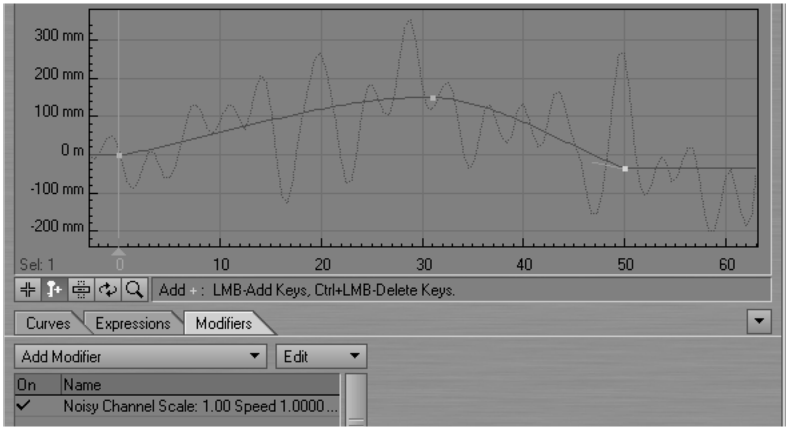
This is a mode that will automatically bring the channels for the currently selected Layout item into the Graph Editor.

Fit Values when Selected

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 plug-ins) visible as a dotted line.



Modified curve shown as 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

This collapses or shows the *Tabs* and *Trees* areas of the Graph Editor.

Collapse/Show Tabs

This is the same as clicking the Collapse Tabs button.

Collapse/Show Trees

This is the same as clicking the Collapse Trees button.

Other Commands

Graph Editor Options

This brings up the Graph Editor Options panel. Note that most of these options can be set from the Display menu.

Undo Last Action

Use this command for a single level undo/redo of the last edit.

HINT
Also see the discussion on the Footprints feature.

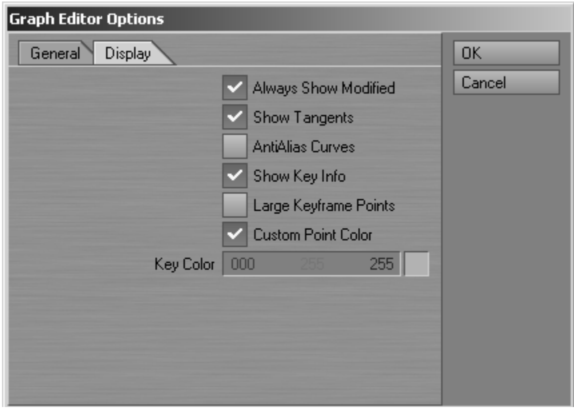
Cancel Changes

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).

Graph Editor Options

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.

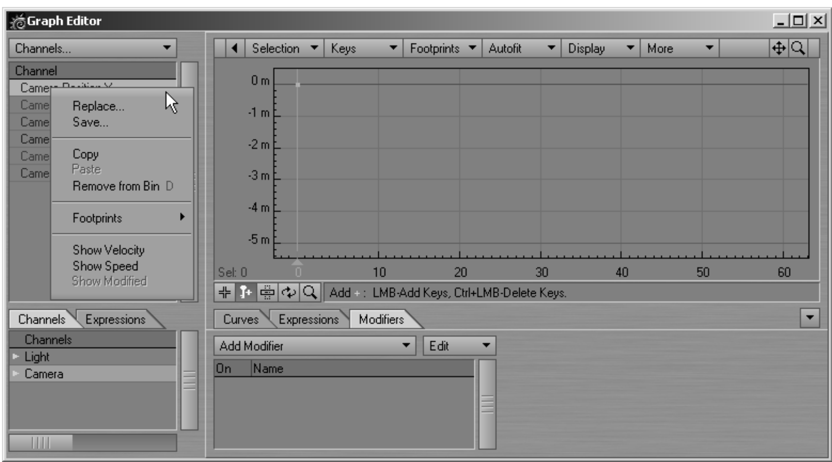
Graph Editor Options panel, General tab



Graph Editor Options panel, Display tab

Channel Bin Pop-up Menu

The Channel Bin has a pop-up menu that appears when you right-click a curve:



Channel Bin Pop-up Menu

Replace Use to load a curve from a file.

Save Saves a single curve to a file.

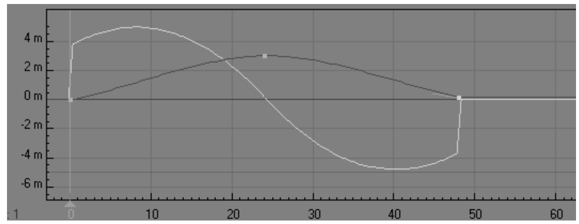
Copy Copies the curve under your mouse pointer when you open the menu to the curve memory buffer.

Paste Pastes the curve stored in the memory buffer. Your mousepointer must be over the curve you wish to replace before you opening the menu.

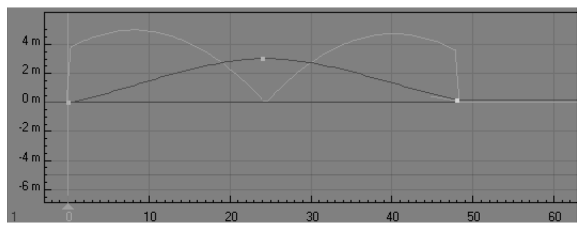
Remove from Bin Removes the selected curve(s) from the Channel Bin.

Footprints See previous discussion on Footprints.

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.



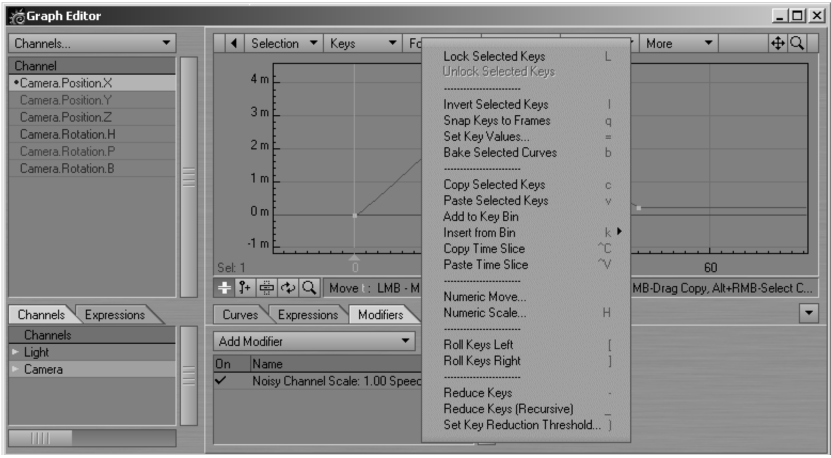
Show Speed Adds a non-editable background curve representing the speed of the current curve. Speed is defined as the magnitude of the velocity vector. This means the Speed curve represents the time rate of change of all three position, rotation, or scale curves.



Show Modified If the **Show Modifiers** option (Display menu) is inactive and your curve is being modified, like from a modifier plug-in, 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.



Curve Edit Window Pop-up Menu

Most of the commands have been discussed earlier; however, some only appear on this menu because they require your mousepointer 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).

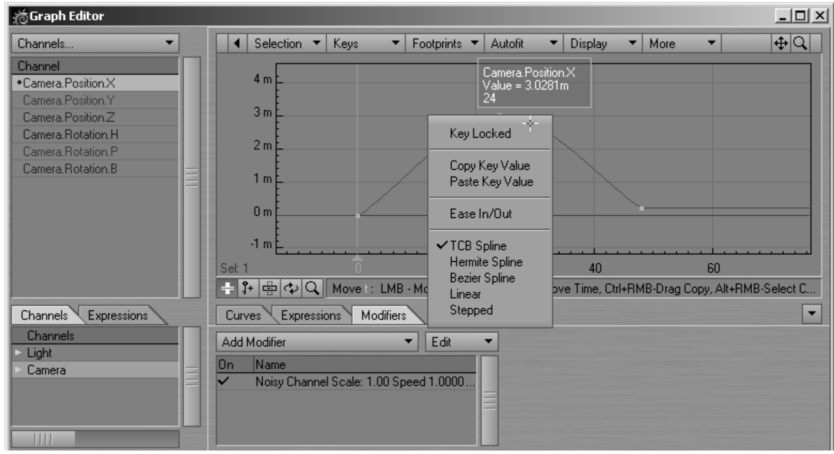
Add to Key Bin Creates named set of keys that you can insert into a curve later.

Insert Works like **Paste Keys**, but gets data from 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 Pop-up Menu

Key Locked Locks/unlocks the key so it is uneditable; locked keys are gray

Copy Key Value Copies the key value to memory buffer

Paste Key Value Pastes the value stored in memory to the key

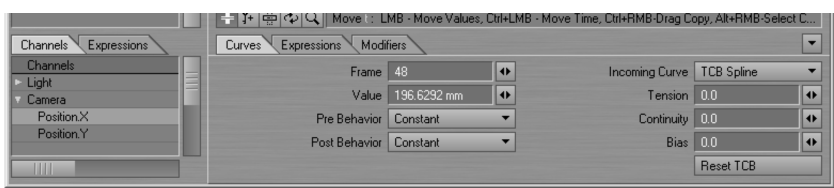
Ease In/Out Sets the **Incoming Curve** for the key to TCB Spline and sets the **Tension** to 1.

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 customizable menus will appear when you Ctrl+Shift+RMB and Ctrl+Shift+MMB.

Curve Controls

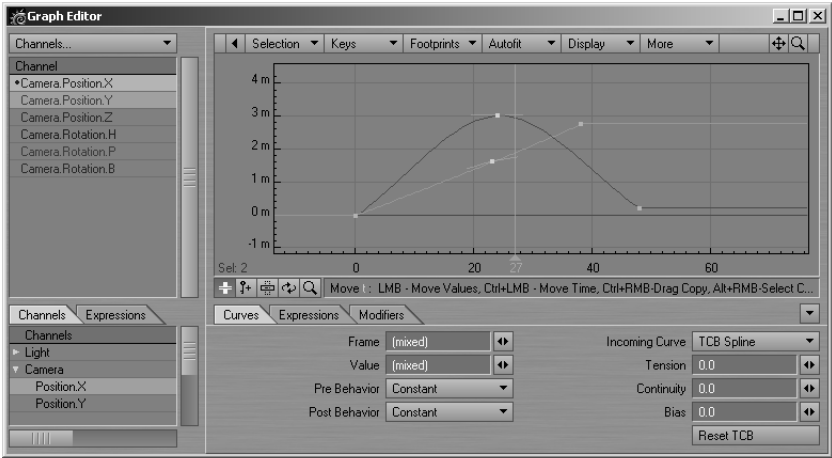
The **Curves** tab in the Curve Controls area contains specific values for a selected key(s). The **Frame** field contains the frame number and the **Value** field holds the related value.



Curves tab

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.

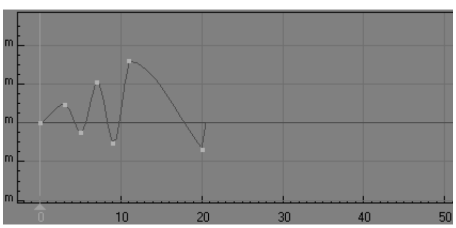


Mixed values

Pre and Post Behaviors

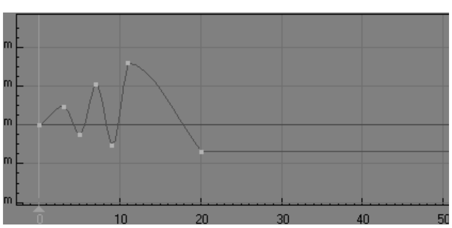
The **Pre Behavior** setting determines what happens before the first keyframe. The **Post Behavior** determines what happens after the last keyframe. The available settings are:

With **Reset**, the motion value is reset to zero.



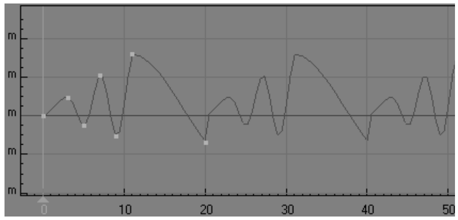
Post Behavior set to Reset

With **Constant**, the values beyond the ends are constant, that is, equal to the first or last keyframe value.



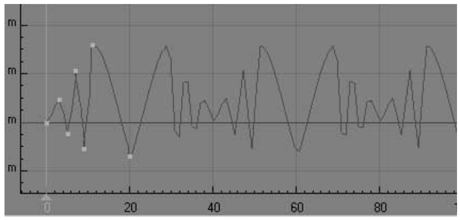
Post Behavior set to Constant

With **Repeat**, the motion repeats from the first to last keyframe.



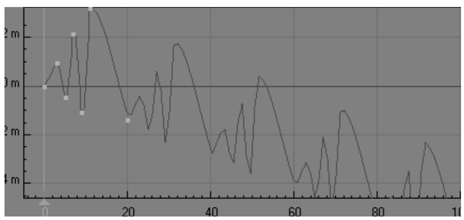
Post Behavior set to Repeat

With **Oscillate**, the motion is mirrored over and over.



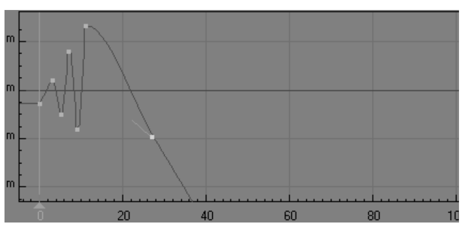
Post Behavior set to Oscillate

With **Offset Repeat**, the motion repeats, but it is offset by the difference between the first and last keyframe values.



Post Behavior set to Offset Repeat

With **Linear**, the curve receives a linear angle consistent with the angle at the start or end points.



Post Behavior set to Linear

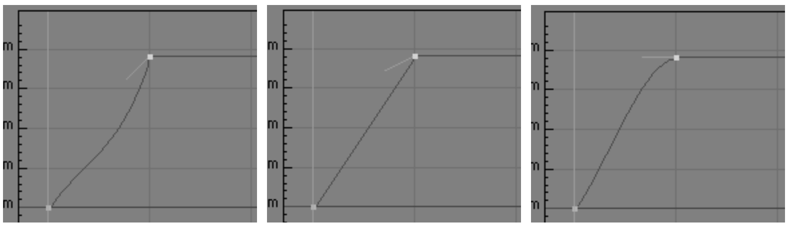
Incoming Curves

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*).



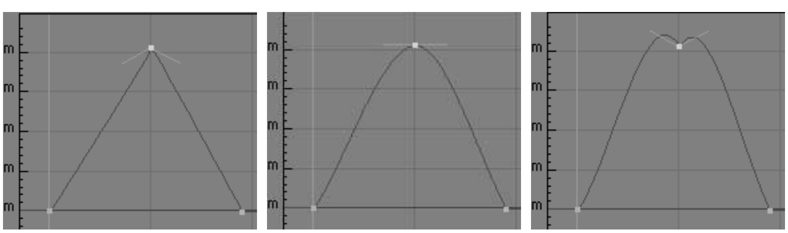
Left to Right: Tension -1, 0 and 1

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 have the logo start slowly, while a negative value would have 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.

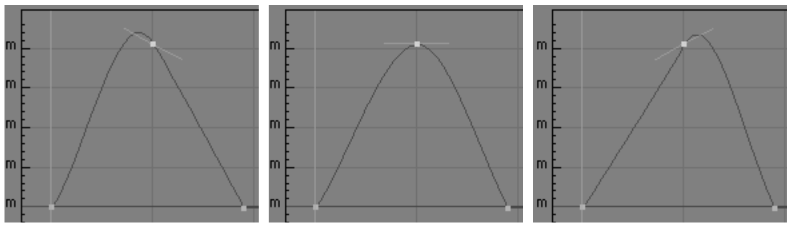


Left to Right: Continuity -1, 0 and 1

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 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.



Left to Right: Bias -1, 0 and 1

Interactive TCB Adjustments

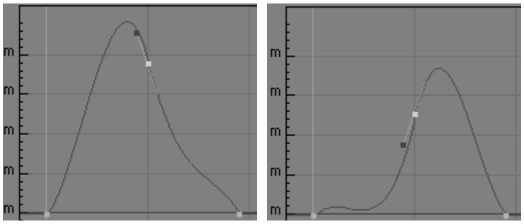
You can interactively adjust TCB keys with your mouse. Simply press F1 for tension, F2 for continuity, or F3 for bias and then drag your mouse left to decrease or right to increase the value. You will see a small indicator in the lower-left corner of the graph.

This only works in the Move edit mode and for the first mouse drag. The tool stops when you release the mouse button.

Interactive TCB adjustment display

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.



Hermite spline incoming curve

NOTE
Note that TCB Splines generally limit you to more realistic results.

Bezier Spline

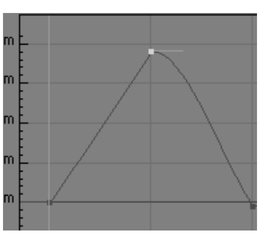
Splines using **Bezier Spline** operate like bezier curves do in many industry-standard paint and illustration packages. When you create a key, you must drag before releasing your mouse button to edit the handles. (*See note below.*) Otherwise, bezier keys have handles that coincide with the key. For 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 FACTS
Bezier curves were developed by Pierre Bezier for designing Renault automobile bodies.

Linear

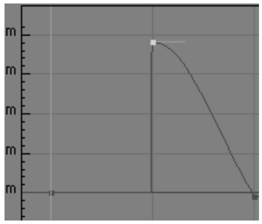
Linear removes the gradual, smooth nature of a spline curve change and replaces it with more direct, sudden change. **Linear** affects the changes between the current keyframe to the previous keyframe only. By turning **Linear** on or off at different keyframes, graphs may contain both gradual and sudden changes.



Linear incoming curve

Stepped Transition

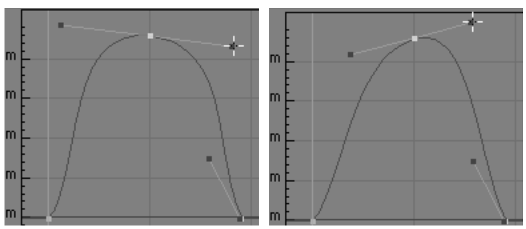
Stepped transition holds the preceding keyframe value and then abruptly jumps to the next keyframe value at that frame.



Stepped transition incoming curve

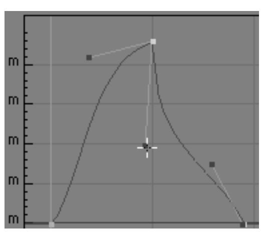
Dual-handled Control Points

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.



Unified tangents

You can break the tangents, that is, make them operate separately by holding the ALT key and then dragging either side.



Broken tangents

To reunite the tangents, just double-click on either control handle. The opposite side will line back up.

Integrated Expressions

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:

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.

2 Enter a name for your expression in the **Name** field.

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.

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:

1 Do the edit in the Graph Editor;

2 Turn off the expression in the Graph Editor. Perform the edit and then turn it back on; or

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:

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.

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:

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.

2 Select the channel(s) in the Channel Bin.

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:

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.

2 Click the **Get Channels** button. The contents of the Channel Bin will be replaced with the channels attached to the current expression.

NOTE
The name of an expression on the **Expressions...** pop-up menu will list the number of channels attached to it in parentheses.

To remove a channel from an expression:

1 Select the channel(s) in the Channel Bin.

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 Used**. 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:

1 Add two null objects named *Control* and *Action* to an empty scene.

2 Animate the Control null on its Y axis.

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.

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.

5 The expression [CONTROL.POSITION.Y,TIME - 1] would lock Action to Control, but give it a one second delay.

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.

NOTE
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

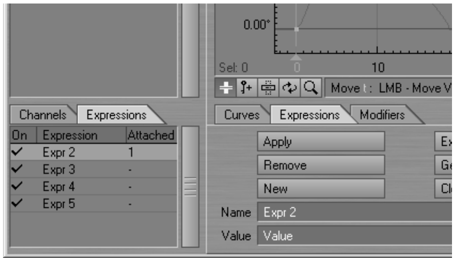
Here is an expression that performs the same feat, but using vectors:

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.



Expressions tab

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 also will 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.

Graph Editor Exercises

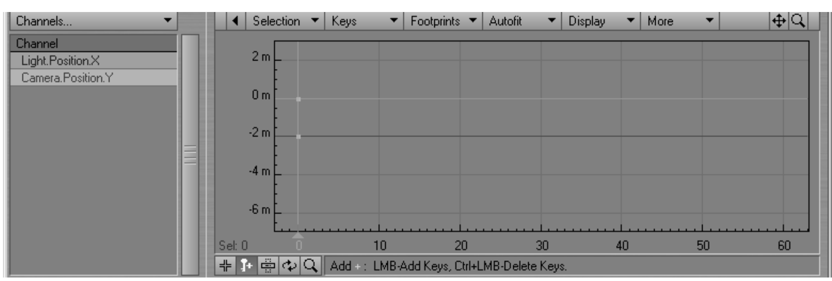
Here are some exercises to help you learn the Graph Editor:

Exercise: Creating Curves

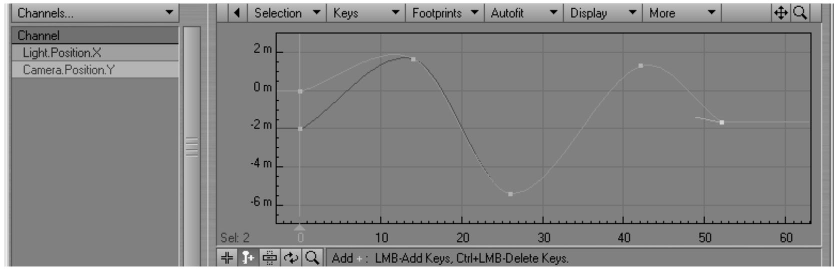
1 Clear the scene and click the **Graph Editor** button.

2 Locate the Light's Position.X and double-click on it to move it into the Channel Bin by itself. Hold the SHIFT key and double-click on the Camera's Position.Y curve to add it to the Channel Bin.

3 Hold the SHIFT key and click on both curves to select them in the Channel Bin. You will see they are both active in the Curve Edit Window.

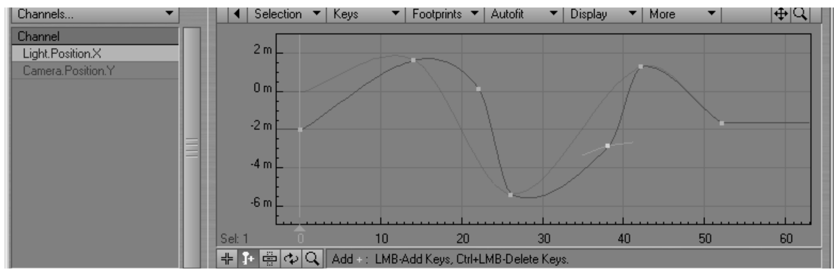


4 Choose the Add mode and randomly create two more keys to the right of the first key frame. You will notice that the new keys are created for both curves so that they match as closely as possible.



5 With both curves selected you can also select key frames from both curves and edit them. This can be very useful for making relative changes to multiple curves simultaneously.

6 Click on Light.Position.X, which will automatically deselect Camera.Position.Y. Create more keys for the Light.Position.X. Now, you can see how to create portions of a curve that match and use background curves as reference.

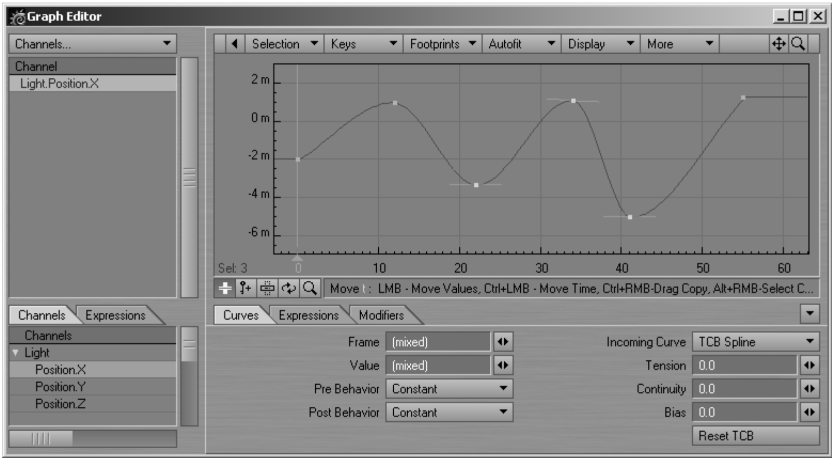


Exercise: Setting Values

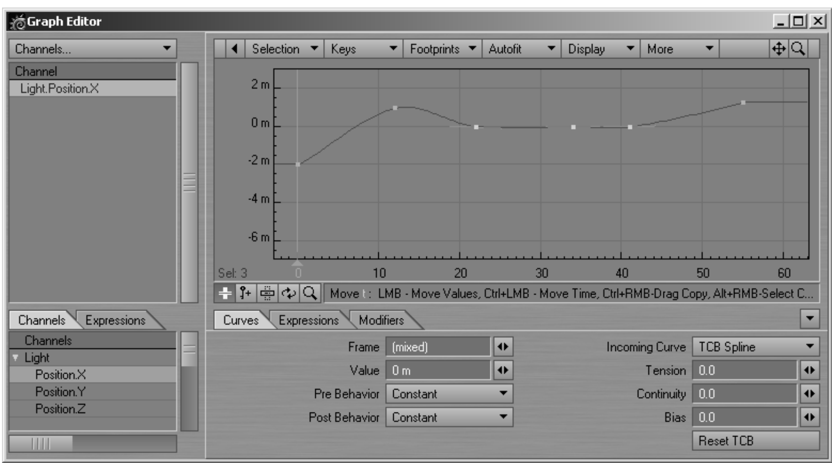
1 Clear the scene and click the **Graph Editor** button.

2 Add Light.Position.X to the bin and create four or five random key frames. Change to the Move mode and unselect all keys by clicking in the graph border.

3 Use the bounding box (RMB) to select two or more of the keys. Notice that the **Frame** and **Value** fields have changed to *(mixed)* because you have selected multiple keys that also have different values.



4 Double-click in the **Value** field and enter 0. You will see that all selected keys are now set to a value of zero.



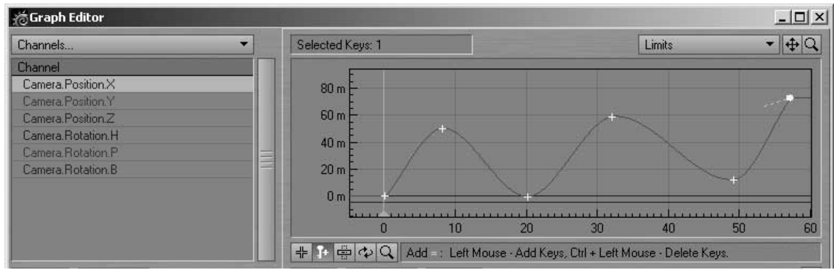
5 Experiment with multiple curves selected. You can set **Frame** or **Value** to match across curves.

Exercise: Footprints

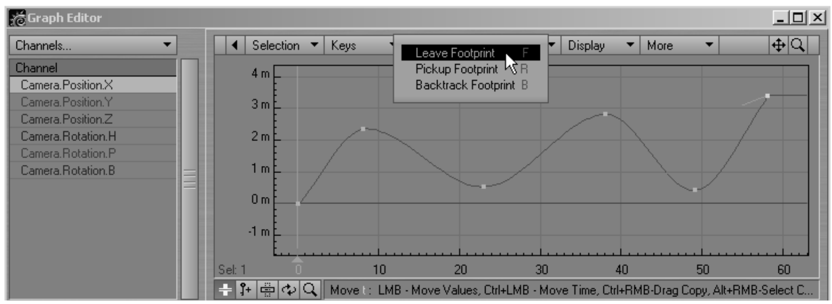
1 Clear the scene and click the **Graph Editor** button.

2 Choose all Camera channels by double-clicking on the Camera in the Scene List.

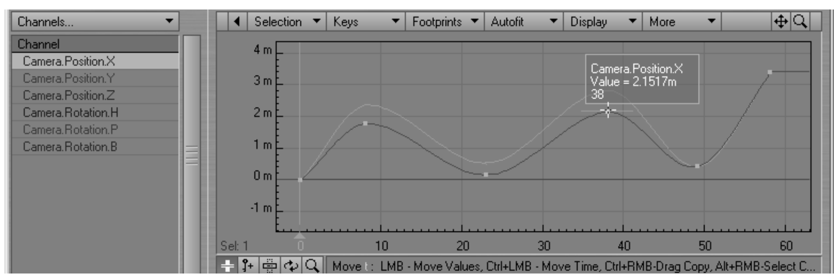
3 Choose the Camera.Position.X and make about five random keys.



4 Since we plan to edit this curve, let's save its current state. Select **Footprints** > **Leave Footprint** from the toolbar.



5 In the Move mode, edit some points on the graph; you will see a new curve appear underneath that is lighter than the selected curve. This is the footprint of the original curve.



6 Select **Footprint** > **Backtrack Footprint** to restore the original curve. If you liked your changes, you would select **Pickup Footprint**, which would delete the footprint altogether, or you could use **Leave Footprint**, which would update the footprint to the new curve.

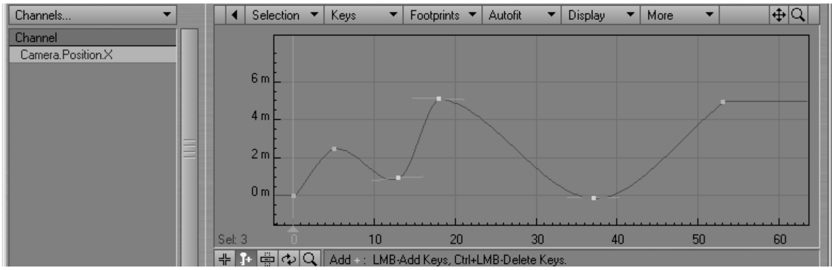
Exercise: Locking Keys

Now that you know how to create a fail-safe for a curve, let's take a look at another way to protect curves or keys from editing.

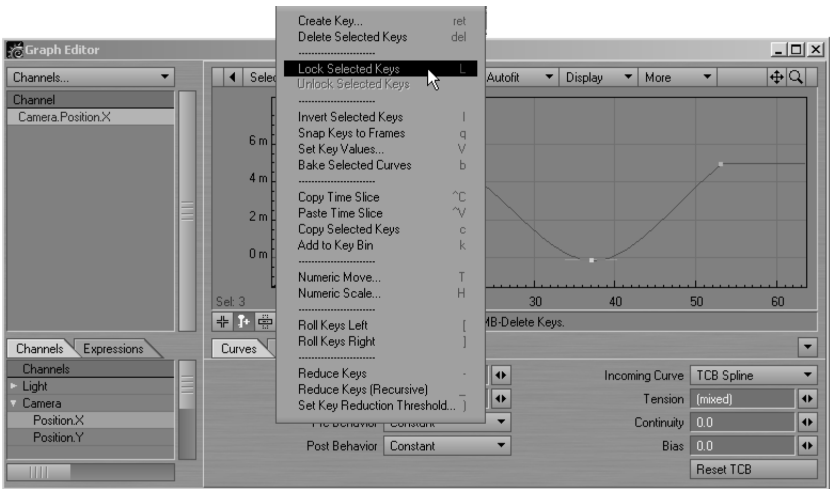
1 Clear the scene and click the **Graph Editor** button.

2 Add Camera.Position.X to the bin and make about five random keys.

3 Unselect all keys and then using the bounding box, select three keys in the middle.

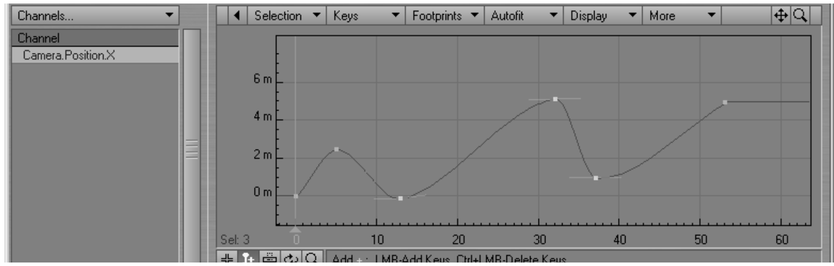


4 From the toolbar, choose **Keys** > **Lock Selected Keys**. The selected keys will now appear gray. Any attempt to edit these keys will be futile. This is a great way to protect keys from being accidentally bumped. It also lets you experiment with keys around an area that should be locked.



5 In the Move mode, select all keys. Now choose **Keys** > **Unlock Selected Keys**.

6 Now try **Keys** > **Invert Selected Keys** from the toolbar. This will reverse the order of all of the selected keys. You can invert them again to get them back into place (or press the U key to undo the edit).

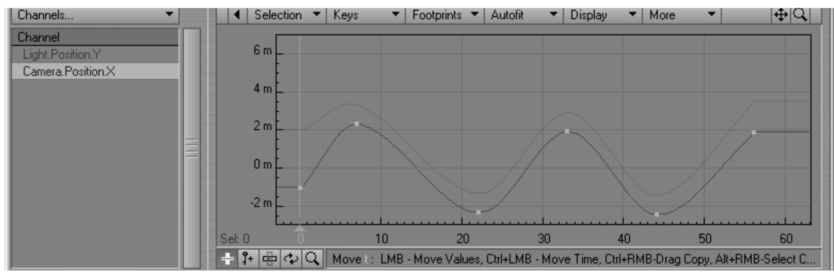


Exercise: The Channel Bin

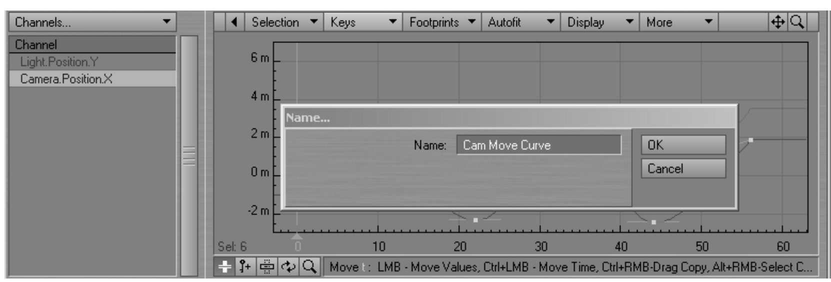
1 Clear the scene and click the **Graph Editor** button. Place the Camera.Position.X and the Light.Position.Y into the Channel Bin.

2 Select both curves and create five keys.

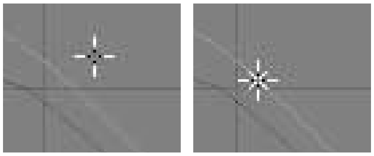
3 Select just Camera.Position.X and move the keys around randomly so you can see both curves.



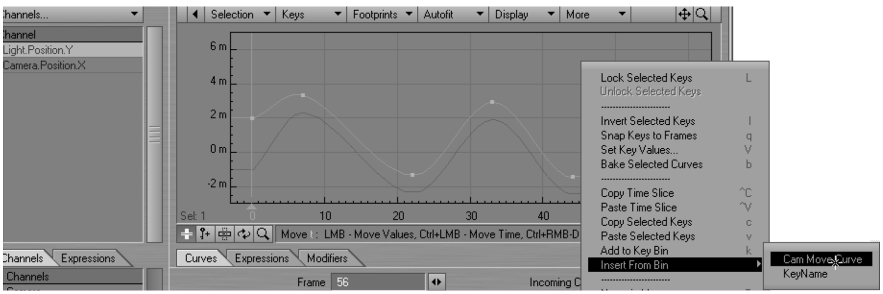
4 Select all keys for just Camera.Position.X and choose **Keys** > **Add to Key Bin**. Create a bin named CAM MOVE CURVE.



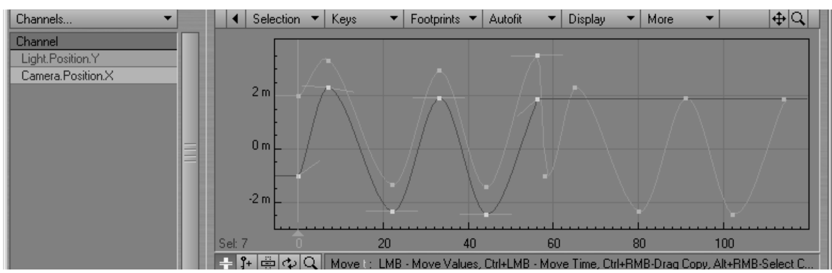
5 Select the Light.Position.Y curve. Move your mousepointer over the curve and notice the change in the pointer. The little X indicates this is an insertion point. Your cursor must be over a curve to insert the keys, since there may be multiple curves active.



6 Place your pointer at the end of the Light.Position.Y, open the Curve Edit Window pop-up (CTRL+SHIFT+LMB) and choose **Insert** > **Cam Move Curve**.

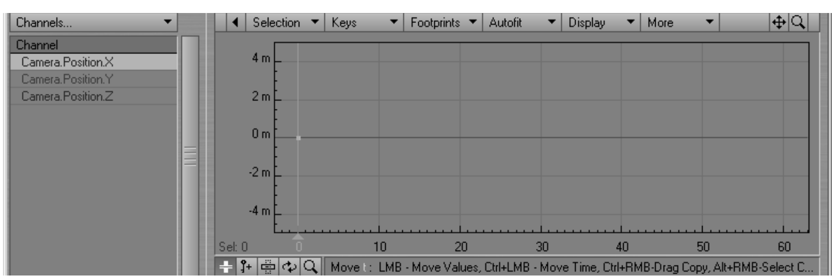


You have now appended the Light.Position.Y motion curve with the keys from the Camera.Position.X curve! Press the A key to "autofit" the two curves into the display.



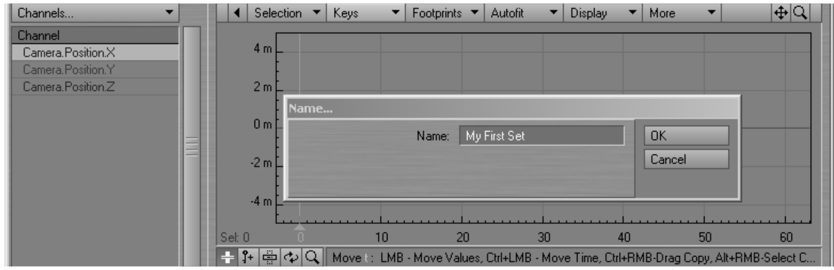
Exercise: Favorites

1 Clear the scene and click the **Graph Editor** button. Place some of the individual Camera curves into the Channel Bin.



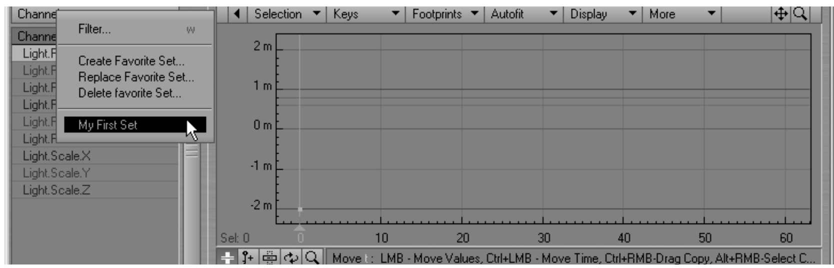
2 Choose **Create Set** from the **Channels** pop-up menu just above the Channel Bin.

3 Type the name of your set, perhaps MY FIRST SET.



4 Now, in the Scene List, double-click on the Light. This should replace the contents of the Channel Bin with only the Light curves.

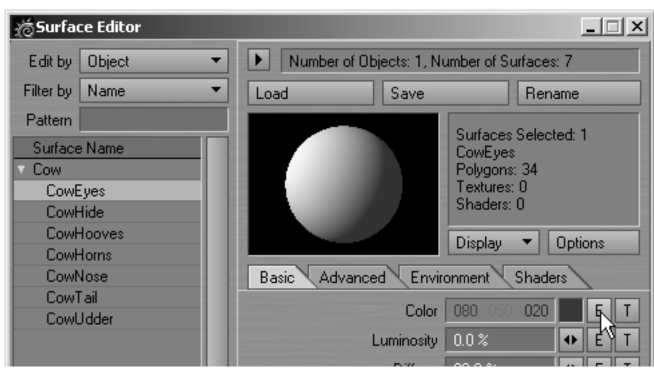
5 Select MY FIRST SET from the **Channels** pop-up. Your previous Camera curves will now appear back in the Channel Bin.



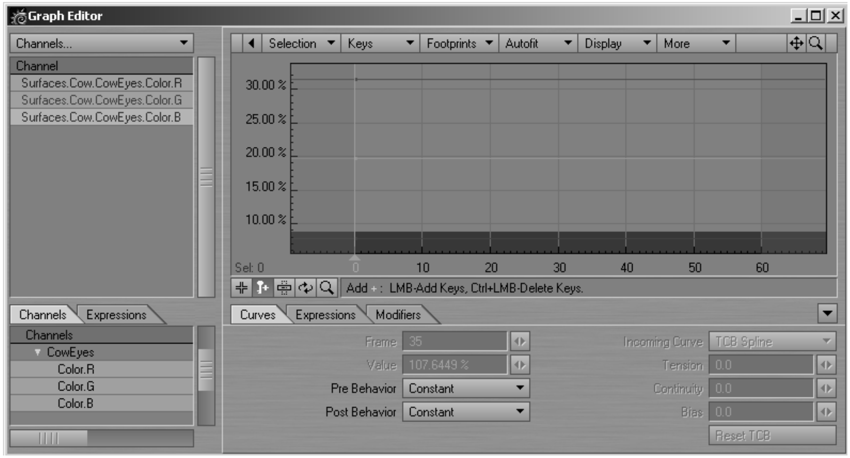
Exercise: Color Channels

1 Clear the scene and add the COW.LWO object.

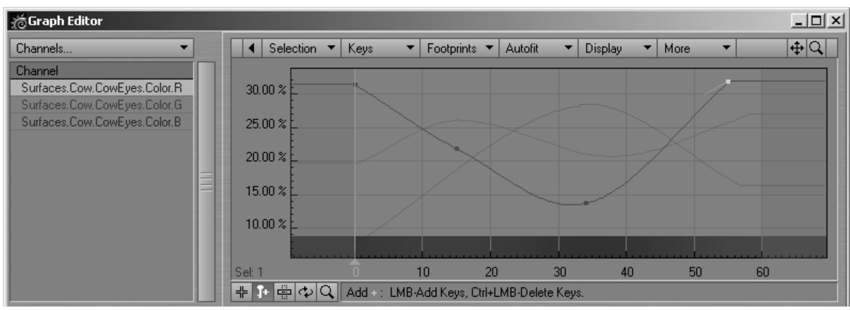
2 Open the Surface Editor, select the first surface, CowEyes, and click the **E** button next to the Color attribute.



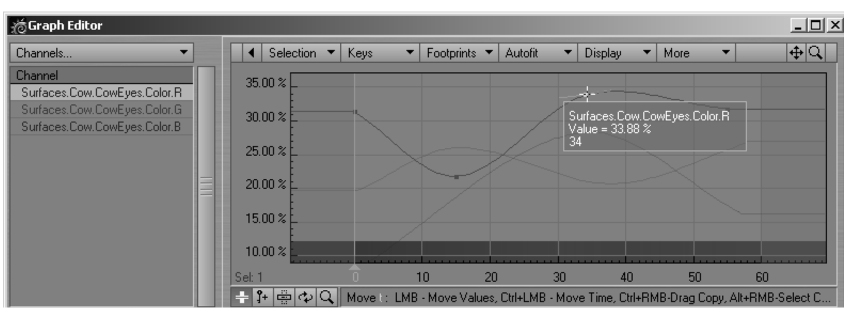
3 The Graph Editor will appear with the surface's three color channels in the Channel Bin.



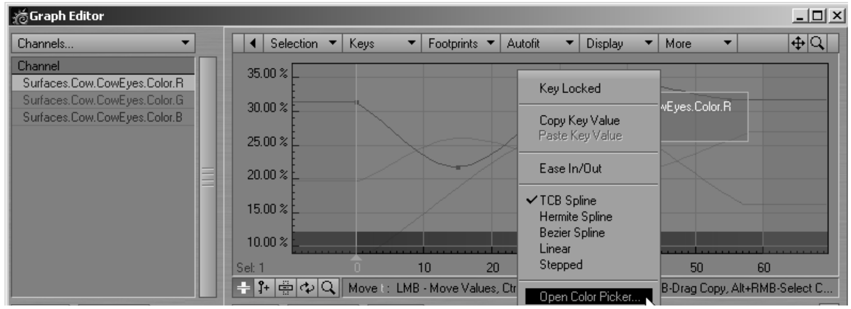
4 Create three random keys for each color channel. Note how the color bar appears when a color channel is selected, but doesn't change when different color channels are selected. This is because it represents all three color channels whether or not all three are selected.



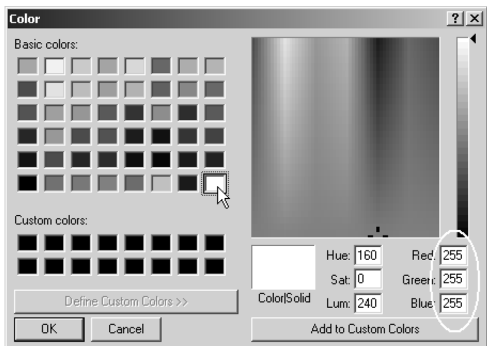
5 In the Move mode, try dragging one of the lower value keys up and note the change in the color bar. It is reflecting the overall color change based on the change in this one channel.



6 Now, select just the red channel and right-click on one of the keys. Choose **Open Color Picker** from the pop-up menu.

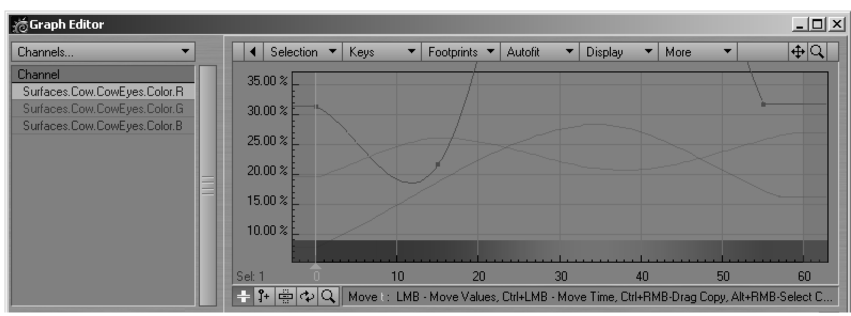


7 Your color requester will appear. Select pure white (RGB 255, 255, 255) and click OK.

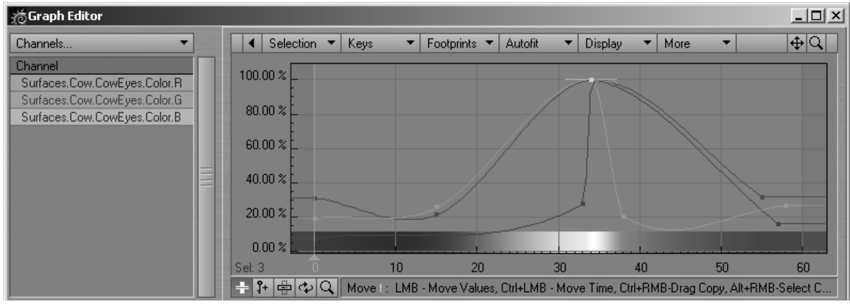


This is the standard Windows color requester. Yours may differ.

8 Notice that only the red channel was affected and the curve flies off the graph.



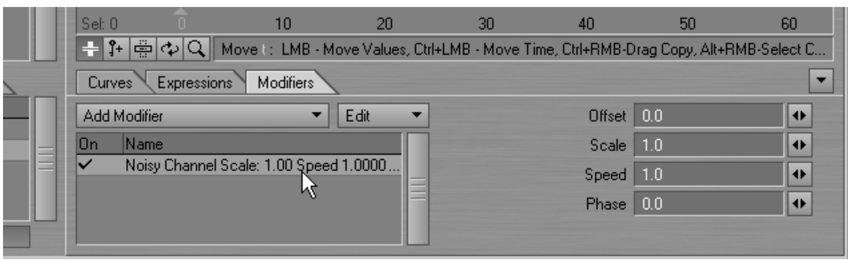
9 Undo the operation by pressing the U key. Select all three channels and select the white color again. Now all three channels are affected and collectively create the color white. Note that keys were automatically created for the other curves. (A quick way to get grayscales is to just select all channels and create keys.)



All three channels set using white.

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 (see Chapter 11).



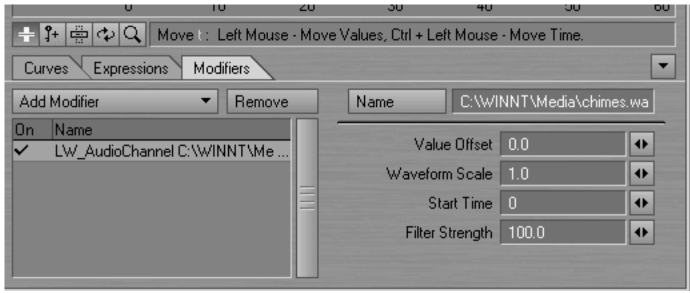
To use a channel motion modifier, select the target curve in the Graph Editor's Channel Bin. Then select the modifier from the **Add Modifier** pop-up menu on the Modifier tab. Once added, click on its name in the list to access its settings, if any. Channels with modifiers will have a small dot to the 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 plug-in discussed in Chapter 6. This lets you create custom user-defined channels.

AudioChannel

The AudioChannel modifier modifies a curve based on an audio file.



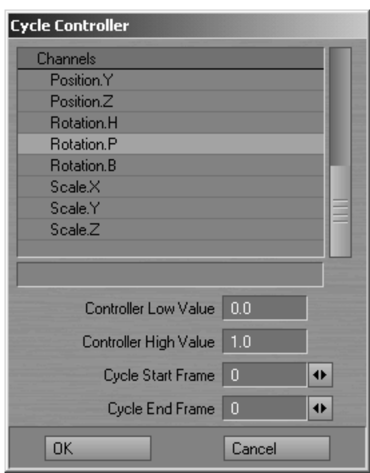
Click the **Name** button to select the file you want to use. The **Value Offset** lets you move the entire motion up or down. (The units are the ones used on the graph.) **Waveform Scale** is a multiplier. Thus, a value of 1 will have no effect. A value of 2 will double the values of the effect and .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

See "SetDrivenKey."

Cycler

This is a channel-oriented version of the Cyclist item motion modifier discussed in Chapter 11.



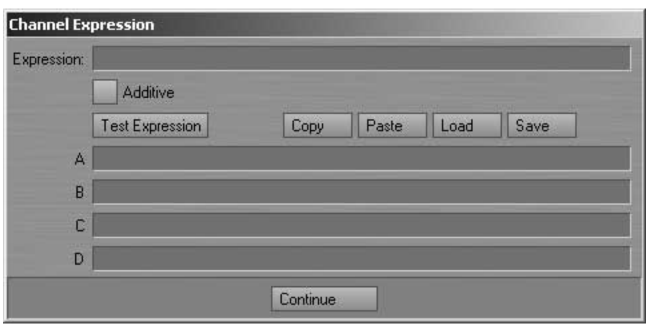
Select the controlling channel in the list window. The **Controller Low Value** and **Controller High Value** settings define how much change is required to equal one full animation cycle. The unit of measure for this parameter depends on the selected control channel.

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.

NOTE

Many motion modifiers can be duplicated using appropriate expressions. Since we at NewTek don't want to force you to hire a TD, or learn calculus, we have wrapped some of the most common and useful expressions into preset plug-ins—optimized for their purposes—like Cyclist, Oscillator, Gravity, and so on.



The Channel Expressions panel

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, and then you may **Load** it in to 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*diameter it moves. This expression (pitch = pi*Z/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.

WARNING
Using Expressions requires a strong background in mathematics and computer programming. Transferring such skills in sufficient quantities to your brain is well beyond the scope of this user documentation. Check our web site and LScript documentation for more information.

Object References

An objects is referenced by its name (e.g., myBox, Null (1), Object:Layer1, etc.). Bones are addressed as an item of their parent object in expressions, which allows bones with the same name in different objects. Here is an example: Null2.Bone1.wpos(Time).x.

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.

<table>
<tr><th>Data Type</th><th>Command Syntax</th></tr>
<tr><td colspan="2">Built-in Functions</td></tr>
<tr><td>double</td><td>sqrt(double)</td></tr>
<tr><td>double</td><td>exp(double)</td></tr>
<tr><td>double</td><td>log(double)</td></tr>
<tr><td>double</td><td>sin(double)</td></tr>
<tr><td>double</td><td>cos(double)</td></tr>
<tr><td>double</td><td>tan(double)</td></tr>
<tr><td>double</td><td>asin(double)</td></tr>
<tr><td>double</td><td>acos(double)</td></tr>
<tr><td>double</td><td>atan(double)</td></tr>
<tr><td>int</td><td>random(int,int)</td></tr>
<tr><td>double</td><td>vmag(vector | double,double,double)</td></tr>
<tr><td>(int | double | vector)</td><td>abs(int | double | vector)</td></tr>
<tr><td>double</td><td>ceil(double | vector)</td></tr>
<tr><td>double</td><td>floor(double)</td></tr>
<tr><td>double</td><td>cosh(double)</td></tr>
<tr><td>double</td><td>sinh(double)</td></tr>
<tr><td>double</td><td>tanh(double)</td></tr>
<tr><td>double</td><td>mod(double | vector,double | vector)</td></tr>
<tr><td>double</td><td>pow(double,double)</td></tr>
<tr><td>double</td><td>rad(double | vector)</td></tr>
<tr><td>double</td><td>deg(double | vector)</td></tr>
</table>

(double | int | vector)	max(double | int | vector,double | int | vector)
(double | int | vector)	min(double | int | vector,double | int | vector)
double	range(double,double,double)
double	selector(double,double,double,double)
double	step(double,double,double)
double	round(double,int)
double	frac(double)
double	fac(double)
double	cot(double)
double	sec(double)
double	csc(double)
vector	cross3d(vector,vector)
double	dot3d(vector,vector)
double	cross2d(double,double,double,double)
double	dot2d(double,double,double,double)
vector	normalize(vector)
vector	center(vector,vector)
vector	extent(vector,vector)
string	parse(string,string)
double	number(string)
vector	vector(string)
string	string(double + int + string + vector)
int	integer(double | string)
string	strleft(string,int)
string	strright(string,int)
string	strsub(string,int,int)
string	strupper(string)
string	strlower(string)
double	randu([double | int])
string	hex(int[,int[,int | "true"]])
string	octal(int[,int[,int | "true"]])
double	angle(vector,vector,int)

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	scale(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	aspectratio
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)
double	h (selects the first element of a multiple-data type)
double	p (selects the second element of a multiple-data type)
double	b (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	rbg (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>

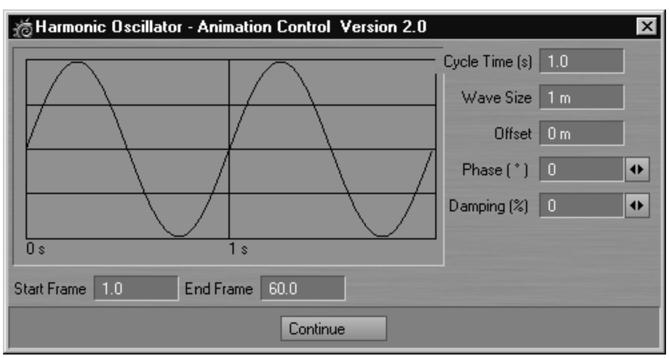
BackLight.color(frame / Scene.fps).rgb returns RGB vector value for color *BackLight* at a user-defined *frame* converted to a time index using the Scene object's *fps* setting. The *frame* variable is returned to the caller and can have its value explicitly set before each evaluation of the expression.

*2 * "1 2 3".asVec.y* returns 4

((Scene.usingLR ? (Scene.lr.right - Scene.lr.left) : Scene.width) / 2).asInt finds the horizontal center of the frame.

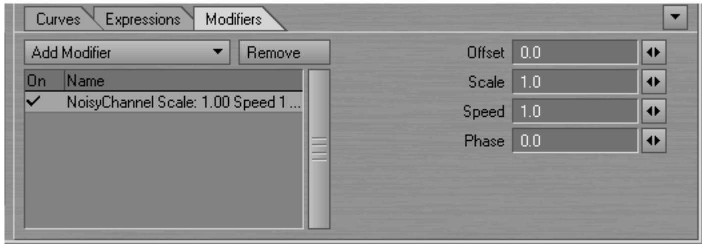
Oscillator

This is a channel-oriented version of the Oscillator item motion modifier discussed in Chapter 11. The settings are identical, except the channel is determined by which channel the modifier is added to in the Graph Editor. The effect is always additive.

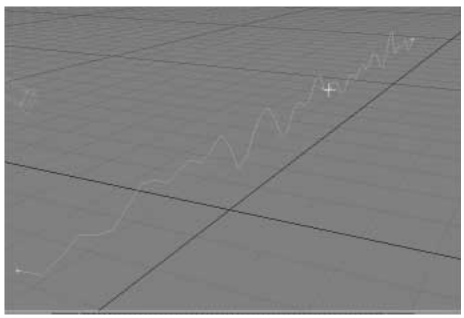


NoisyChannel

This modifier randomizes a channel.



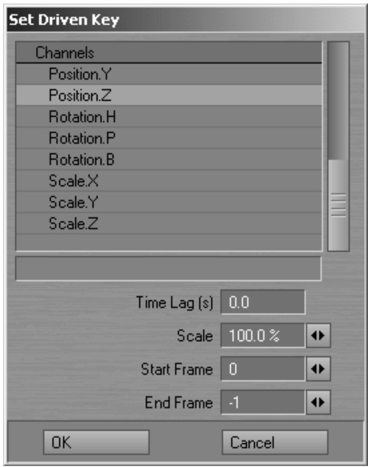
The **Offset** lets you move the entire motion up or down. (The units are the ones used on the graph.) The **Scale** parameter multiplies the noise amount added in to the channel, so a factor of 1 will have a noise effect, a factor of 0 will have no effect, 2 will double the effect, .5 would halve it, 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)

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 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.

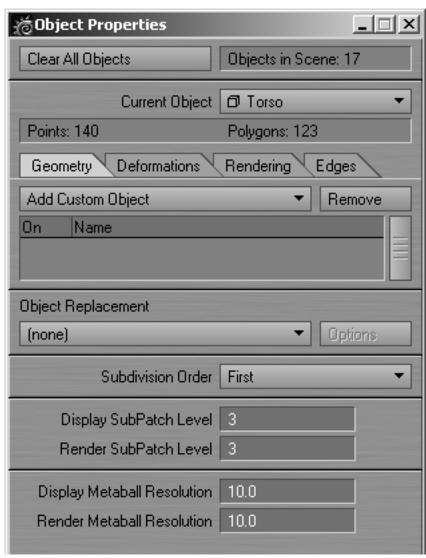
TextureChannel

This is a channel-oriented version of the TextureMotion item motion modifier discussed in Chapter 11. 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.

chapter 9
Object Properties

Chapter 9: Object Properties

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.



The Object Properties panel

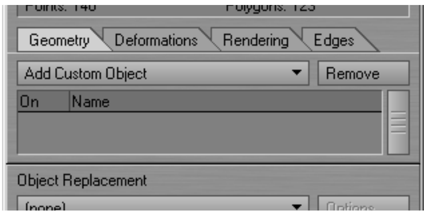
NOTE

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.

CUSTOM OBJECTS

Use the **Custom Object** pop-up menu on the Geometry tab of the Object Properties panel to apply a Custom Object plug-in 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.

Many custom objects are used internally by other plug-ins, so you may see some that are not listed below. ParticleFX, for example, uses custom objects to create its emitters, winds, and so on.



Custom Object pop-up menu

Camera Mask

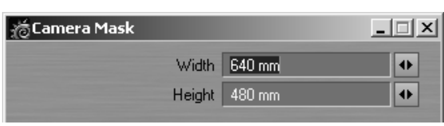
It is sometimes necessary to use a textured object for the background instead of using a **Background Image** (Effects panel, Compositing tab) for effects like casting shadows onto the background or when you want to move the background around. The Camera Mask custom object can be used to compute the exact Z distance needed to fill your camera view.

To use Camera Mask:

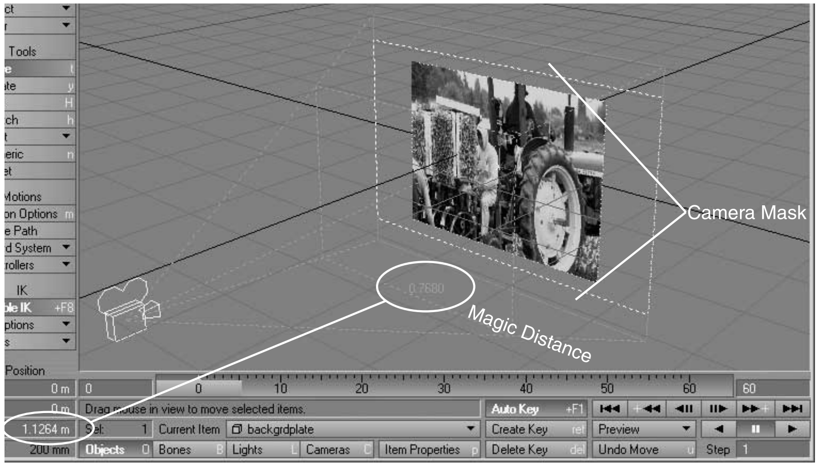
1 Model a flat Z-facing rectangle object with the proper aspect ratio. For example, 640 mm x 480 mm for a 4:3 aspect ratio. Apply your surface texture.

2 Load the object into Layout and parent it to the camera.

3 Add the **Camera Mask** custom object plug-in 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.

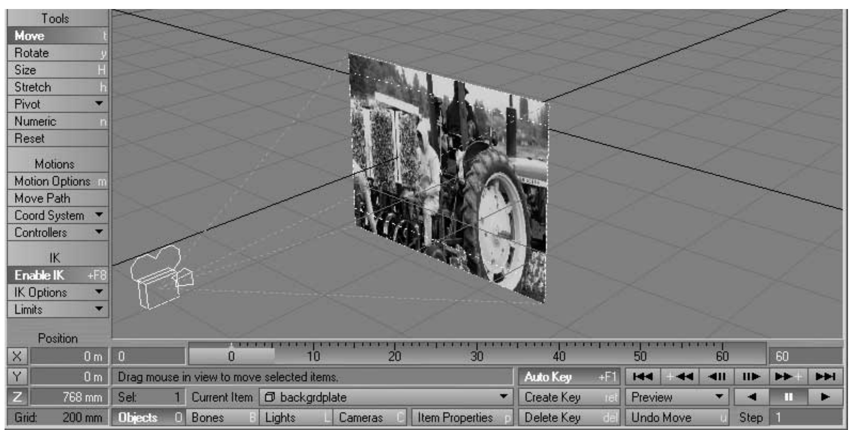


4 Now, adjust the Z position of the object. You will see a representation of the camera view extend from the object with your textured rectangle attached to the end. If you have a **Camera Mask** set on the Camera Properties panel, it will be visible.



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: That you may see OpenGL display errors when the mask is solid.)

5 Adjust the object's Z Position to the magic distance. The object will now fill the camera view.



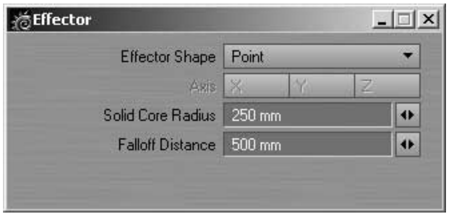
You might also use a background plane with parts cut out, in conjunction with a Background Image, so you can position things between the plane and Background Image.

A cross (+) also appears, which marks the camera's **Focal Distance** setting from the Camera Properties panel (see Chapter 13, "Depth of Field").

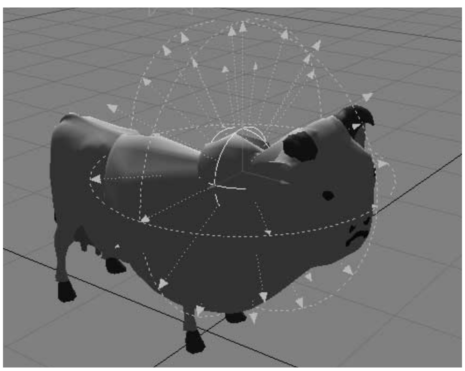
Effector

The Effector custom object can be used with the Effector Displacement plug-in, discussed later in this chapter. This custom object is designed to give you better visual feedback in Layout about your effector. Note that it does *not* directly communicate with the related Displacement plug-in, so all settings must be set manually.

It has two modes to match the shape of the effector: **Point** and **Plane**.

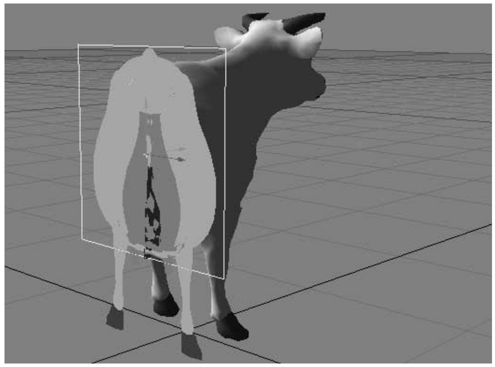


When **Effector Shape** is set to **Point**, set the **Solid Core Radius** and **Falloff Distance** to match those settings on the Displacement plug-in's panel. The arrows and dotted-lines indicate the falloff area. The solid-line ball in the center is the solid core.



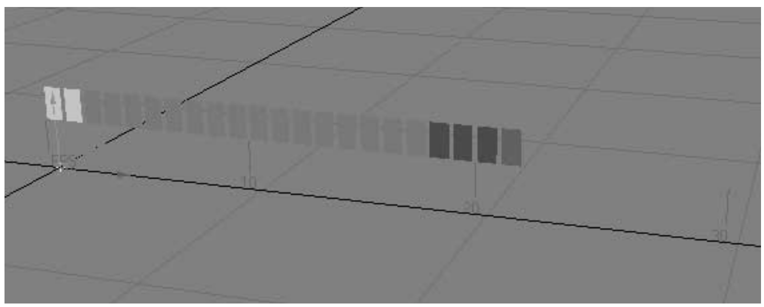
The Effector custom object in Point mode used with the Effector Displacement plug-in

When **Effector Shape** is set to **Plane**, the effector will look like a four-sided plane. The **Axis** settings will become available, which you can set accordingly. The other settings have no effect in this mode.


The Effector custom object in Plane mode used with the Effector Displacement plug-in

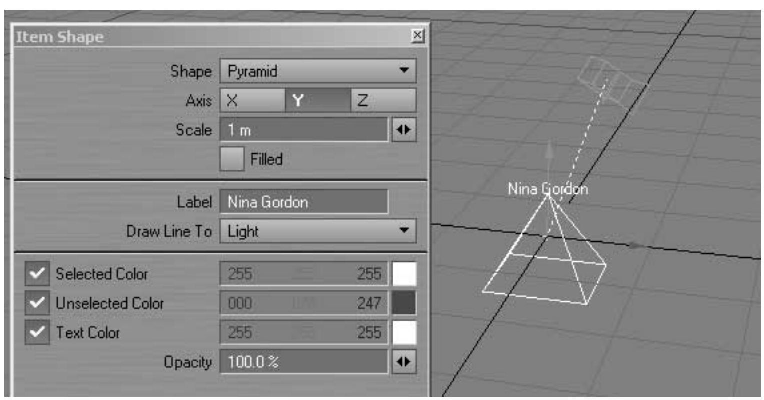
Frame Rate Meter

Adding **Frame Rate Meter** to a null object will display frames-per-second information when you play a scene or drag the frame slider.



Item Shape

Item Shape lets you add a custom object where you can specify its shape and look. The settings should be fairly obvious.

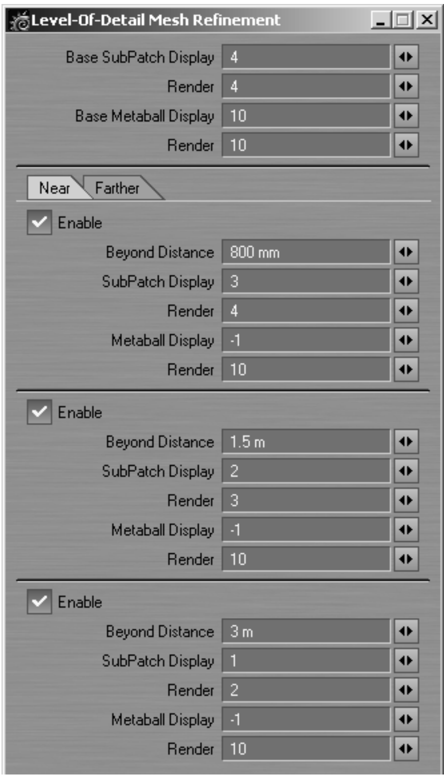


Level-of-Detail Mesh Refinement

Adding the **Level-Of-Detail Mesh Refinement** custom object plug-in 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.

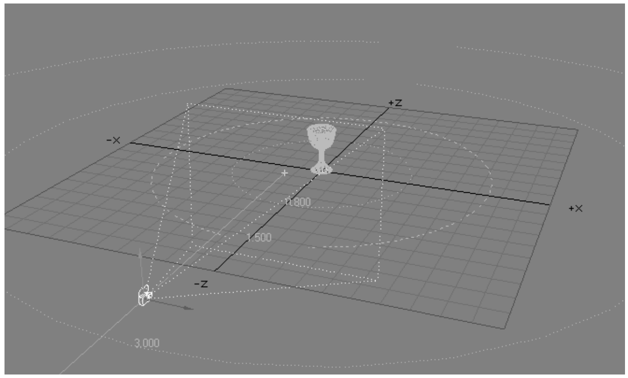
From top to bottom, the entry groups must be in near-to-far order, so the furthest group is last. When the distance from the camera to the object is within each range, the corresponding display/render settings are used.

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.



Level-of-Detail Mesh Refinement

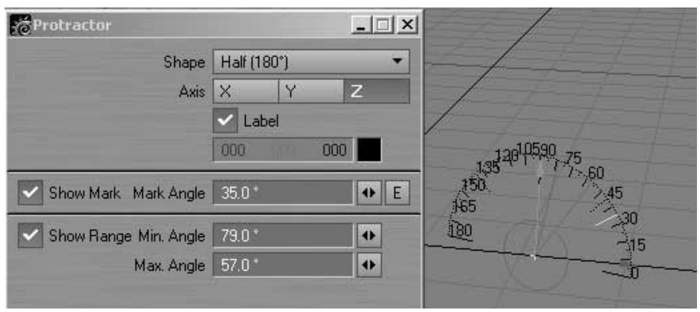
In your viewport, concentric rings are displayed around the object showing the defined distance ranges.



Concentric rings

Protractor

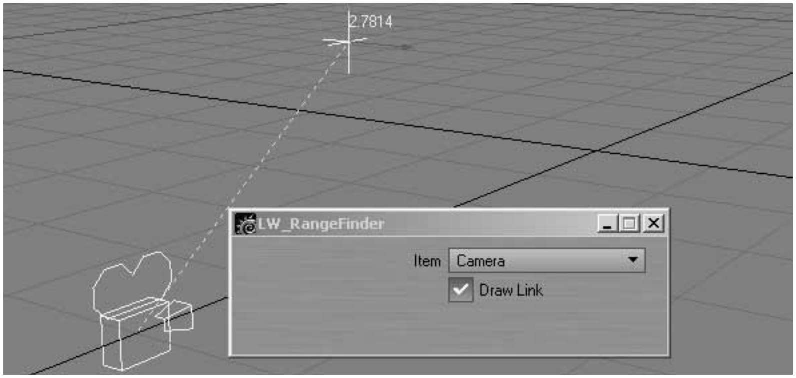
Protractor sets up a custom object you can use to measure angles. With the **Shape** setting you can make it a **Full** or **Half** circle. The **Label** option places numerical labels at set intervals. Use the **Show Mark** option to place a tick mark at a set angle. Use **Show Range** to highlight a set range.



Protractor settings

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.

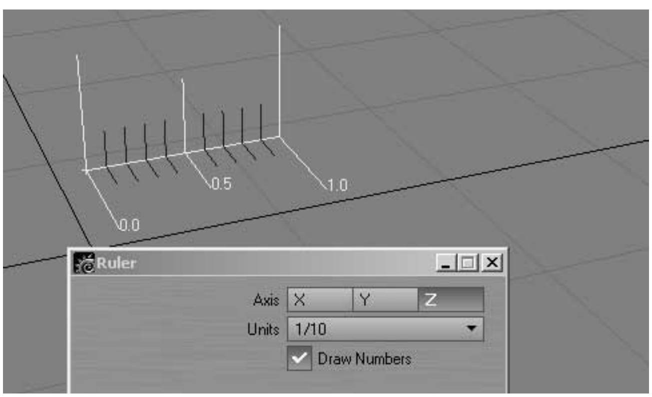


Range Finder in action

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.

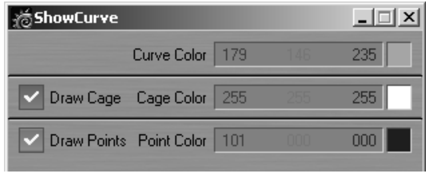
Ruler

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.



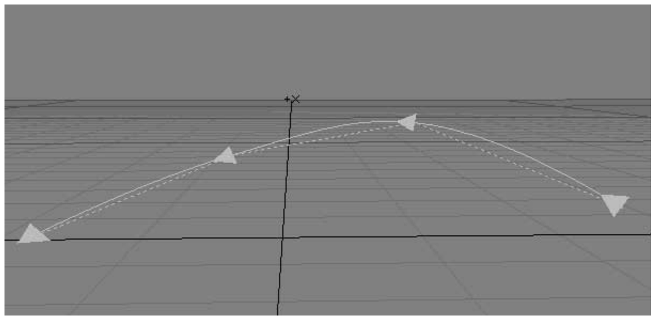
ShowCurve

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 adjust 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.



The ShowCurve options panel

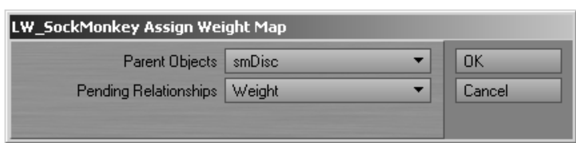
ShowCurve was designed to be used with the CurveConform displacement plug-in, discussed later in this chapter.



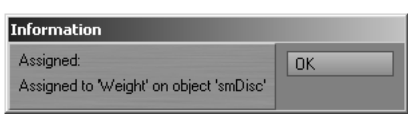
A curve with ShowCurve applied

SockMonkey

The **SockMonkey** custom object plug-in draws bounding boxes for links created in the main SockMonkey displacement plug-in, discussed later. However, if you use the **Auto-add Control Item** button on the main interface, this custom object plug-in 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 plug-in 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. Instead, you will get the following dialog:

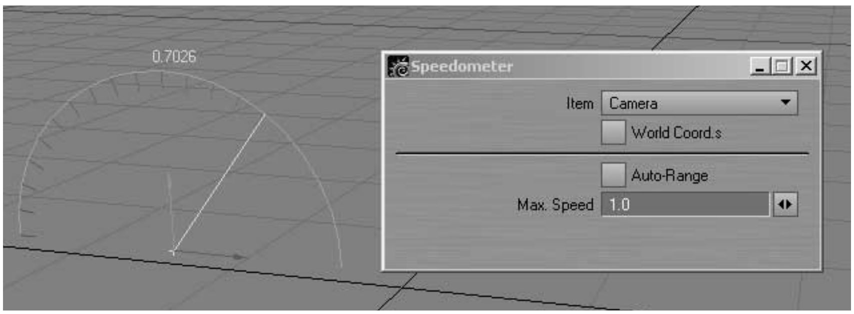


If not assigned to a link, adding this plug-in does nothing.

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.)

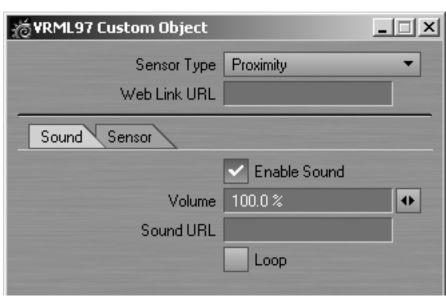
The **Max Speed** setting determines the value when the pointer is *pegged* all the way to the right. Select **Auto-Range** to have the plug-in determine the maximum. Note that the pointer can go past the maximum if the speed exceeds it.

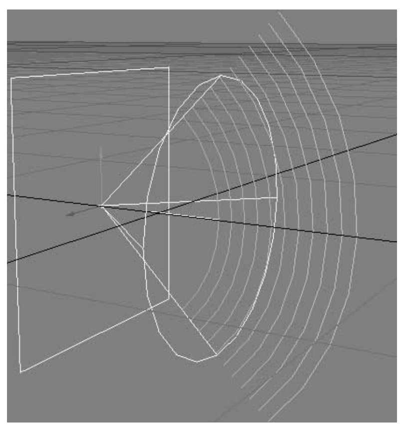


Speedometer

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 REPLACEMENT

You can use special plug-ins to replace objects during the course of an animation.

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!

Level-of-Detail Object Replacement panel

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 plug-in 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 don't need to be similar copies of the same thing. You can replace the cow with a chrome teapot if you want to (and you probably do by now).

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** > **Replace With Object File** except that it happens during the animation. ObjectSequence is to objects, what an image sequence is to images.

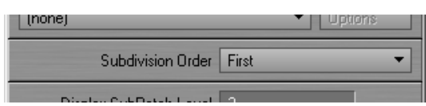
To perform an object replacement, you must have multiple objects with names that differ only by a three-digit number. For example, if you want to change between a series of box objects, you would name the first object box000.lwo. If you want that object replaced by a second object at frame 10, name the 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 plug-in. 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.

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 bend the heck out of the object, you may get unwanted *pinching* in your object. In this case, **Last** 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 the morphing occurs (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.

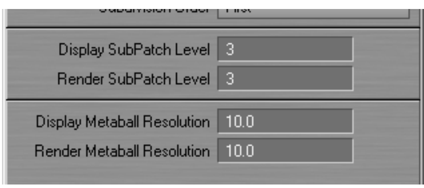
If you use a displacement map on a SubPatch object, you probably want the subdivision to occur before the displacement (using **First**, etc.). This will give the displacement more points to displace.

If you need to choose one of the *in-between* settings, here is the order in which LightWave performs object deformations:

1 Morphing
2 *Before-bones plug-in* displacements
3 Bones
4 Object coordinate plug-in and built-in displacements
5 Motion (scale, rotate, move)
6 World coordinate plug-in and built-in displacements

SubPatch Display and Render Levels

SubPatch objects may be used in Layout the same 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.

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.

NOTE
For an object to be a SubPatch object, it must be saved with the SubPatch mode active. See Chapter 29 for more information on SubPatch objects.

NOTE
See also the Level_Of_Detail Custom Object plug-in discussed earlier in the chapter.

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. There is also a companion setting, **Render Metaball Resolution**, that controls the resolution used for rendering, which can be different.

NOTE
See also the Level-Of-Detail Mesh Refinement custom object plug-in discussed earlier in the chapter.

MORPH TARGETS

Metamorphosing, or *Morphing*, causes a 3-D 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.

Deformations tab selected

The number of points in each object must be equal. You cannot morph successfully between objects that have a different number of points. Point order also needs to be the same.

NOTE
Using Endomorph objects, discussed later, 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 morph to D, and so on.

NOTE
Multi Target/Single Env with surface morphing will reflect only a surface change during the first morph.

Animating Endomorphs

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*. (See Chapter 28 for information on creating endomorph objects.)

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 MorphMixer Displacement plug-in.

To animate an endomorph:

1 In Layout, add the **MorphMixer** displacement plug-in to your endomorph object on the Object Properties panel's Deformations tab.

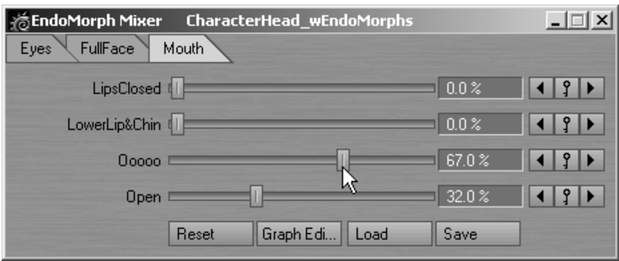
2 Open the plug-in's option panel (double-click it in the list).

3 Now, move your Layout frame slider to the frame where you want to pose your object and set the morph maps as desired on the MorphMixer panel.

NOTE
If you do not have Dynamic Update set to Interactive (Display Options tab of the Preferences panel), you can get Layout to update as you drag by holding the ALT key while you drag a MorphMixer slider.

MorphMixer automatically creates keys at the current frame for any morph channel you adjust. It is like using the normal **Auto Key Create** function (General Options tab of the Preferences panel). If a key exists for a morph channel at the current frame, a key icon will appear to the right of the slider. You can also use the arrow buttons to jump from key to key—this also changes the current frame.

You can create keys for *all* morphs on the current tab—at the current frame—by holding CTRL+SHIFT down while dragging any slider.



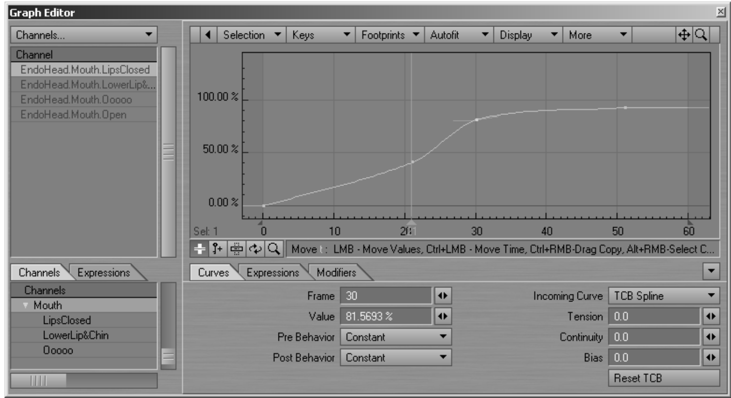
The **Reset** button creates keys at 0% for all channels in the group.

4 Repeat the previous step for all of the keys you want to create.

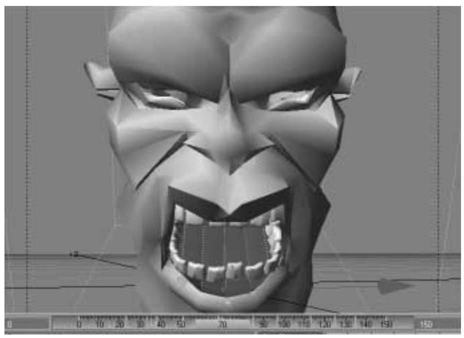
5 Since the MorphMixer plug-in adds the morph maps as animation channels, just like XYZ position, you can also use the Graph Editor to manipulate values. If you click the **Graph Editor** button on the MorphMixer panel, it will appear with the group's channels already in the curve bin.

The maximum number of morph targets in a group is 32. For larger sets of morph targets, create additional groups. Use the **Load** and **Save** functions to retrieve and save MorphMixer data from and to files on your hard drive.

NOTE
If you use the Graph Editor, you must still add the MorphMixer displacement plug-in to make the animation channels available, although you do not have to open the MorphMixer options panel.



If you are lip syncing to audio, you can load an audio file (Scene Editor panel) as a reference for setting your keys.

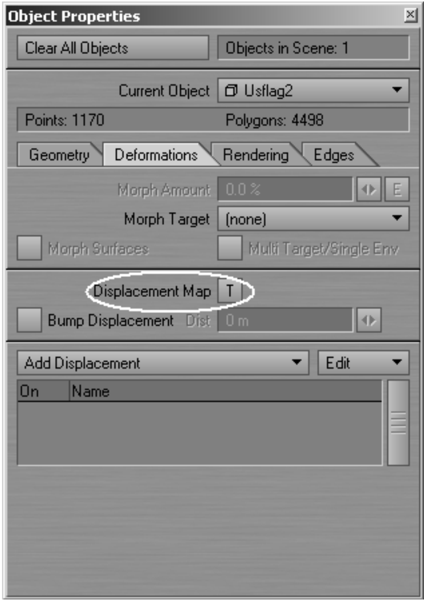


Saving Morph Mix

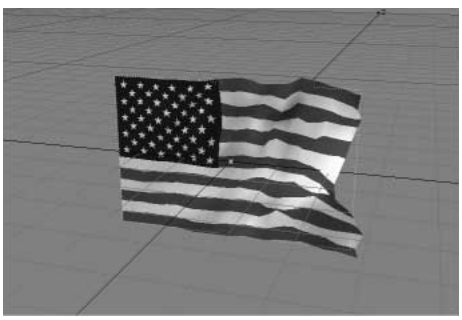
You can save the current "mix" of your morphs back into the endomorph by choosing **File** > **Save** > **Save Endomorph**. The morph will appear on a new "Miscellaneous" tab on MorphMixer. (Note that you may need to remove and then re-add MorphMixer.) Don't forget to save your object if you want to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

DISPLACEMENT MAPS

Displacement maps are similar to surface maps, which add color and texture to object surfaces. (In fact, you can refer to Chapter 31 for descriptions of the various displacement map settings.) 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.



Displacement Map button



Displacement map on flag object

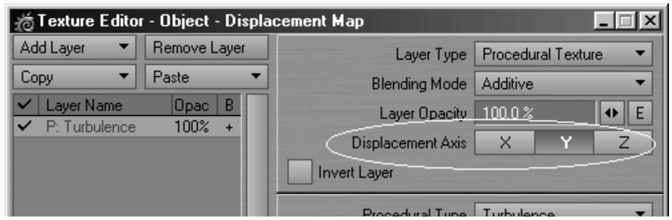
NOTE
Remember that displacement mapping is saved in the scene file not in the object file. If you want to load an object with its displacement map information, choose **File** > **Load** > **Load Items From Scene**.

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.

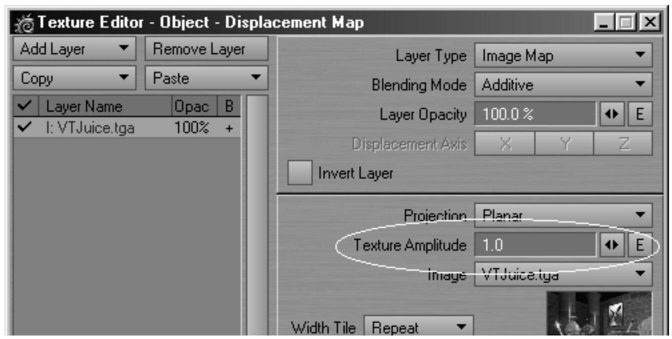
Differences from Surface Textures

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 3-D textures (i.e., procedural and gradients) have an option that lets you select the **Displacement Axis**.



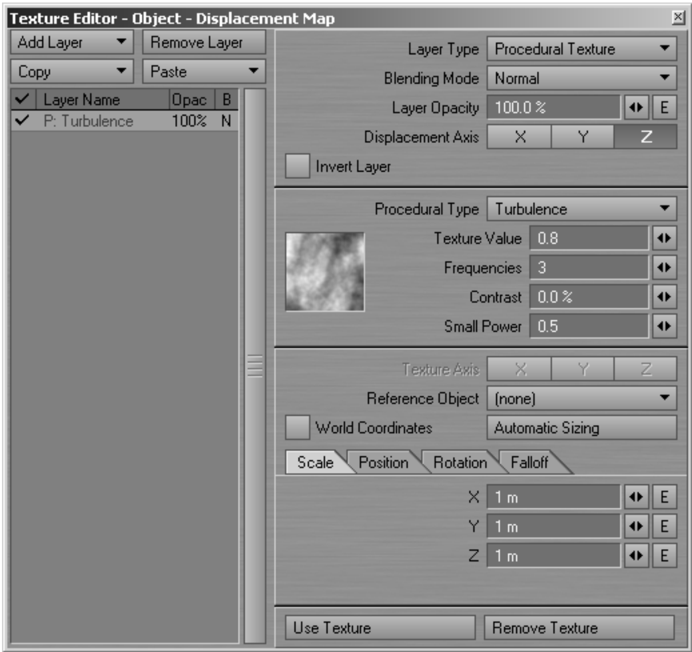
You can also *factor* the effect you would normally get from an image map displacement using the **Texture Amplitude** setting. Use a value less than 1 to lessen the effect or more than 1 to increase it.



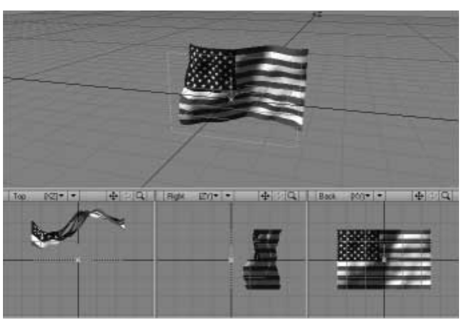
Exercise: displacement map

1. Load the USFLAG2.LWO object (TUTORIALS folder) into Layout.
2. Press P to open the Object Properties panel.

3 Click the **Displacement Map T**exture 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.



4 Study the effect of the displacement map on the object.

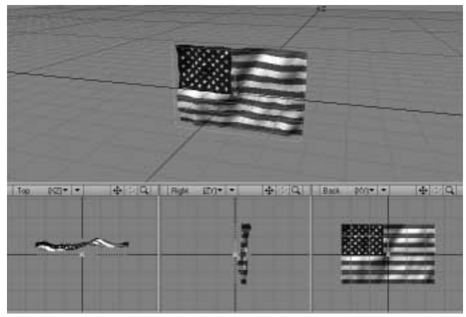


Interesting, but a little too exaggerated. Essentially, the size and amount of the displacement is out of proportion for our object. It's like dropping a boulder into a wading pool. What we really need is a small stone.

5 Go back to the Texture editor and change the **Texture Value** to .2. This will reduce the general amount of displacement.

6 Now, our object is only one meter wide. So to get smaller ripples, change the texture's **Scale** to 400mm for all axes.

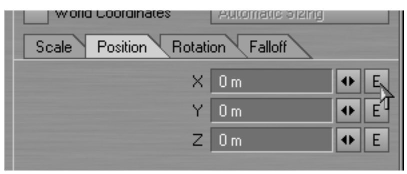
7 Click the Play button in the lower right corner of the main interface. The display updates automatically and we get better feedback as we animate the texture. (Note: the flag won't move yet.)



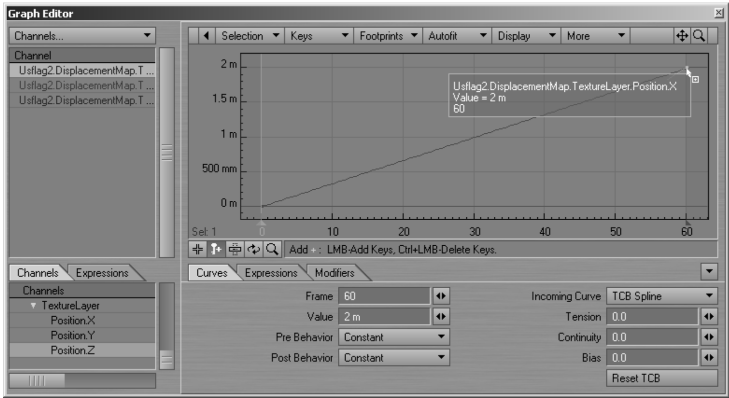
NOTE

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.

8 You can animate the wave by animating the texture's **Position**, which is the center of the texture. Click on the **E**nvelope button for the X Position.



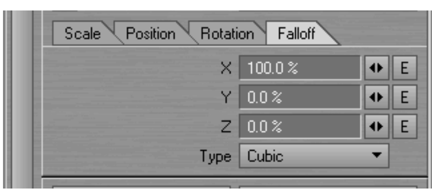
9 The Graph Editor will appear. Select only the **Position X** for the texture and add a key at frame 60 (the last frame in the scene) with a value of about 2m. (You may need to adjust the graph zoom a little. Just drag on the magnifying glass icon.)



If your scene is still playing, you should see your flag waving. We are moving the texture two meters over 60 frames.

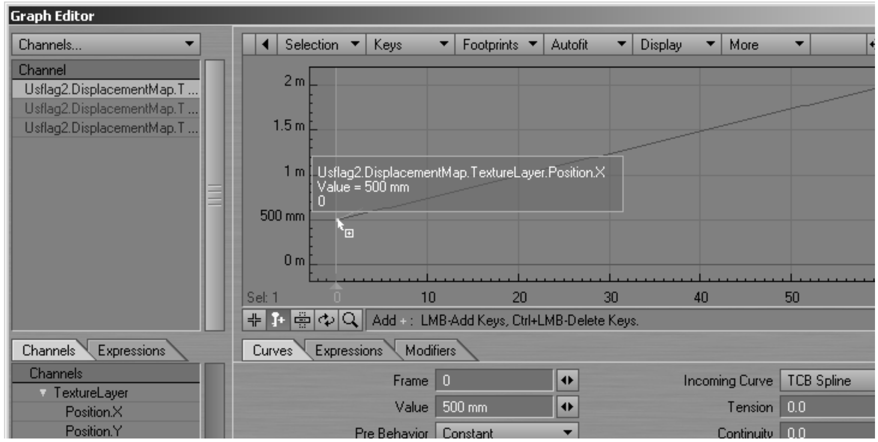
10 Now of course, if this were a real flag, it would be attached to a pole and the left side would not wave as much as the right. We can simulate this by moving the center of the texture and applying falloff to the effect.

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.

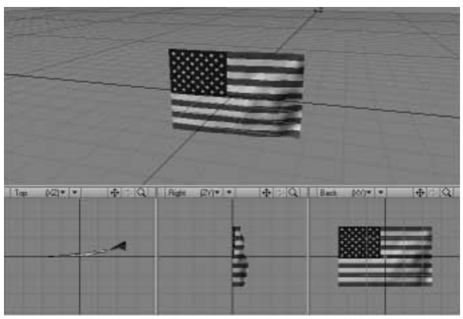


11 The flag is one meter wide, but its local Origin is at its center—you could figure this out by looking at it in Modeler or where it loads by default in Layout. Thus, the right edge is at 500mm. This should be the starting point for the texture's X Position.

Back in the Graph Editor, switch to the Move mode and drag the first key (at frame 0) up, but not past 500mm.

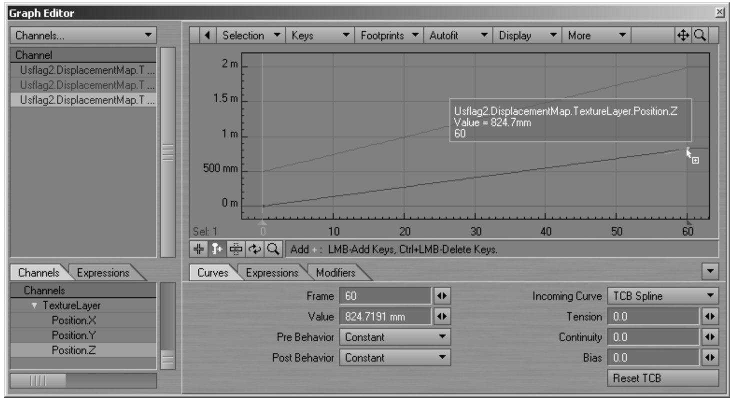


As you drag, look at the layout viewport. You should see the texture move to the right (along the X axis). Since you have falloff, as you move the key, the falloff will begin at a different point—eventually the right edge of the flag—and fall off completely at the left.



NOTE
When you animate the Position values, the Falloff center is based on the Position at frame 0.

12 To add a little more variance to the wave, you could add keys to the Z position of the displacement map.

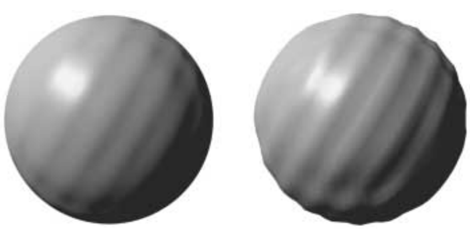


Displacement Mapped Objects

Generally, when you construct objects in Modeler, you try to minimize 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 *fakes* the change with shading.



Left: Ripple Bump map. *Right*: Ripple Displacement map.

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.

Bump Displacement

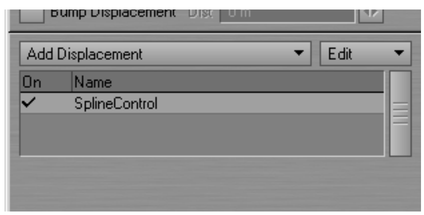
The **Bump Displacement** option uses the bump texture on a polygon and vertex, and applies it as a displacement texture. The direction of the displacement is set by the vertex normal, and the amount of displacement is set by the **Dist** value.

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.

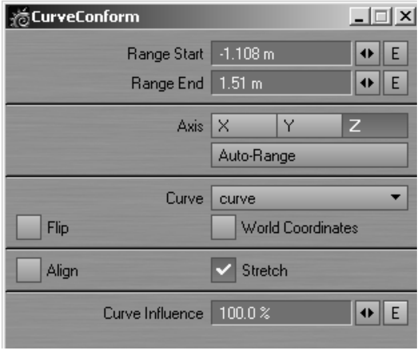
DISPLACEMENT PLUG-INS

In addition to the MorphMixer Displacement plug-in, several other plug-ins let you deform the object in various ways.



CurveConform

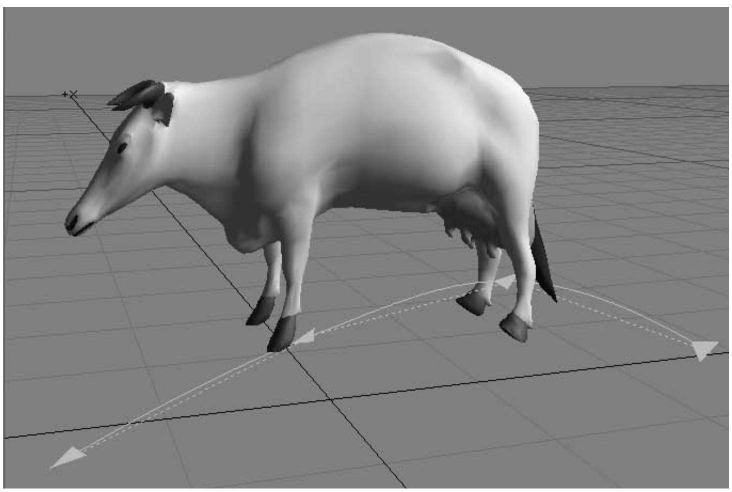
CurveConform uses a curve object to deform an object's mesh. In order to determine what part of the curve applies to what part of the mesh, you must define an axis and a range of distances along that axis. The axis is the direction in the mesh which will be transformed to lie along the curve. The **Range Start** and **Range End** values define where the beginning and end of the curve match up with the selected **Axis**.



The CurveConform options panel

When CurveConform 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.



CurveConform deforms the cow. Stretch option is active. The ShowCurve custom object plug-in has been applied to the curve object to make it visible.

The **Flip** option reverses the influence direction of the curve. **World Coordinates** leaves the curve's position fixed in 3-D space, deforming only that part of the mesh that moves into its range.

Stretch alters the mapping of the range to the curve's arclength (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.

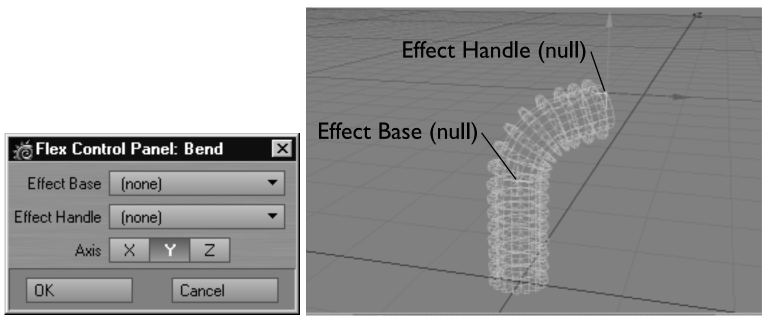
The **Curve Influence** percentage blends the deformed shape with the original un-deformed shape

Deform Displacement Plug-ins

The Deform Displacement plug-ins let you deform objects much like Modeler's Flex and Deform tools. Each needs two null objects to operate.

Deform:Bend

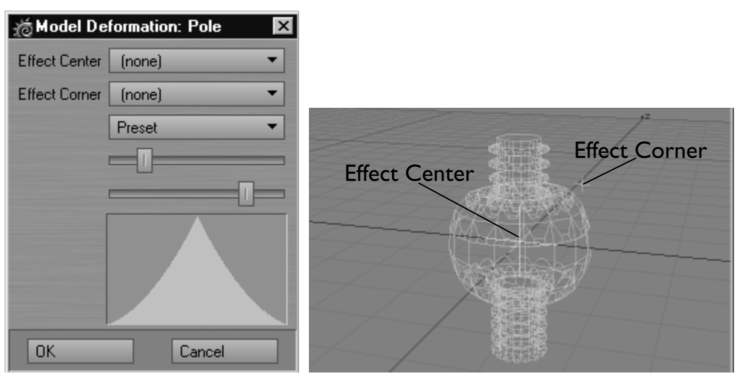
The **Effect Base** object defines the start of the bending point along the selected **Axis**. The Effect Handle object determines the direction of the bending.



Deform: Bend

Deform:Pole

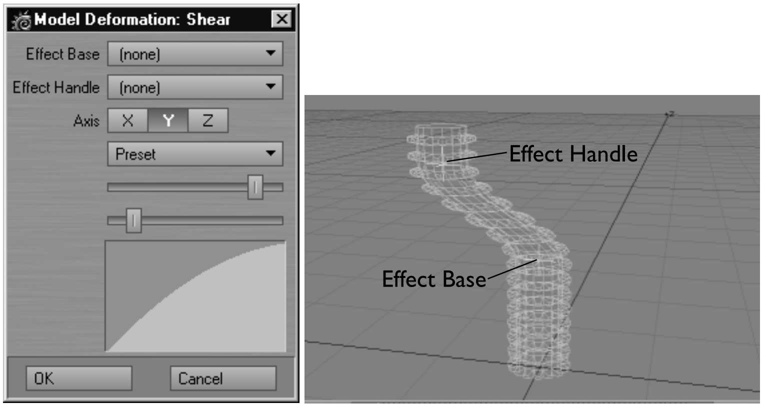
The **Effect Center** is the center of the effect and scaling it will distort the geometry. The **Effect Corner** defines the corner of the influenced area. Use the sliders to shape the influence area.



Deform: Pole

Deform:Shear

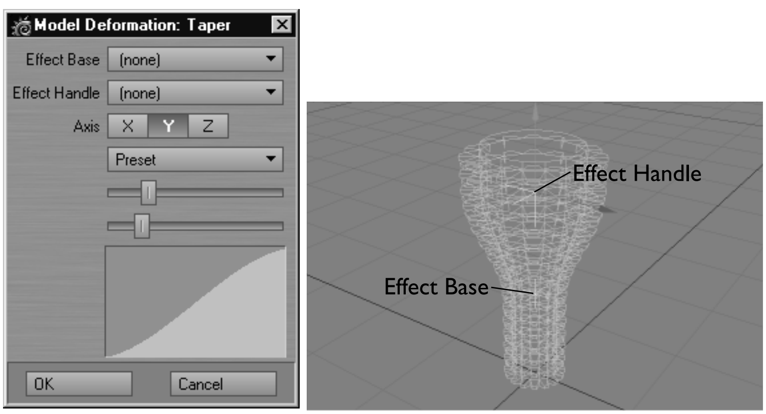
The setup for Deform:Shear is nearly the same as Deform:Bend. You can also control how the effect is applied from the base to the handle by using the two tension sliders. The top slider controls the beginning and the bottom one controls the end. Use the **Preset** pop-up to select from some common settings.



Deform: Shear

Deform:Taper

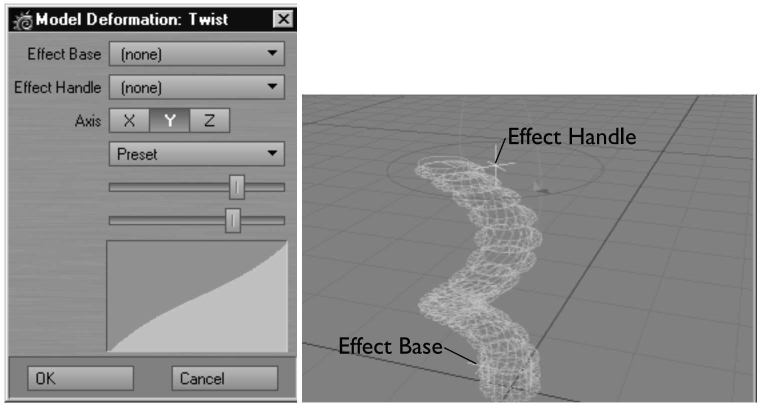
Deform:Taper works just like Deform:Shear, except that you size the handle instead of moving it.



Deform: Taper

Deform:Twist

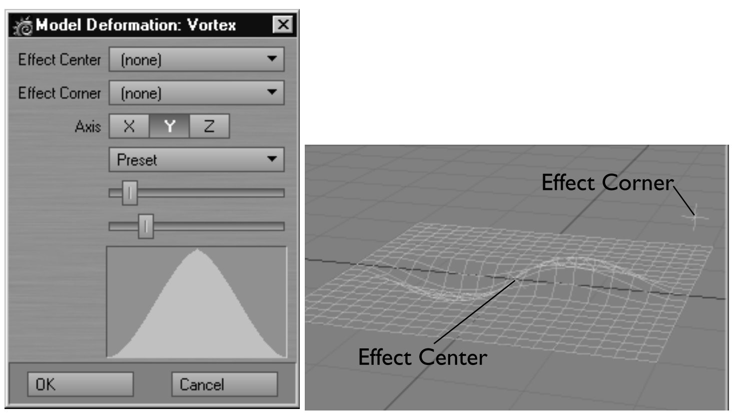
Deform:Twist works just like Deform:Shear, except that you rotate the handle instead of moving it. The center of the twisting is defined by the base object.



Deform:Twist

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.



Deform:Vortex

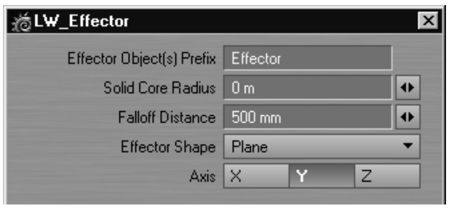
HINT
Add Deform:Vortex three times using a different **Axis** setting on each to have full rotational influence.

DisplacementTexture

Use Displacement Texture to use a texture as a displacement map. Ah, so you say you can already do this with the Displacement Map function. Well, smarty-pants-brain-boy, using the plug-in lets you order the deformation when you use multiple displacement plug-ins. For example, Motion Designer totally overrides any effect the Displacement Map function has. With Displacement Texture, you could place it after Motion Designer.

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.

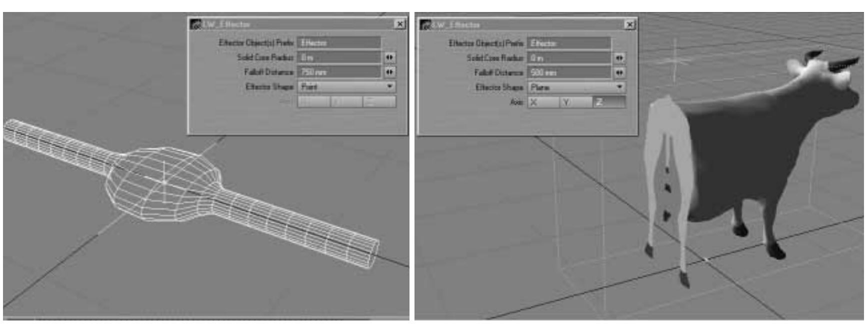


The **Effector Object(s) Prefix** is a name prefix with the default of Effector, and any object that begins with this name will be an effector. This lets you have more than one effector based simply on their name.

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.

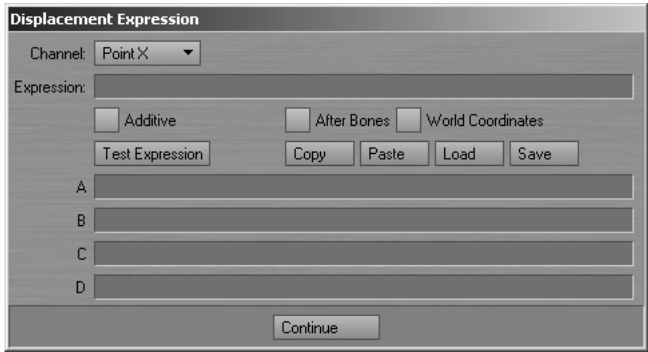
The impact of the effector object is set and animated by keyframing its XYZ **Size** channels. Positive values repel and negative values attract.



Effector in action

Expression

Expression is a sister plug-in to the Expression channel modifier (see Chapter 8). You use it in the same way.

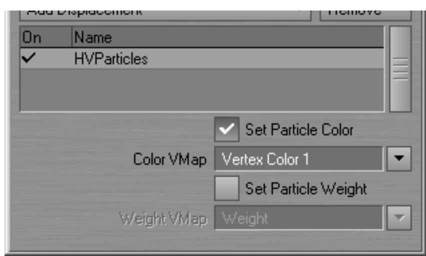


You can set independent expressions for the XYZ point displacements using the Channel pop-up menu.

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.

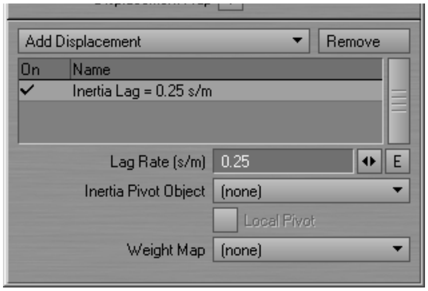
HyperVoxelsParticles

The HyperVoxelsParticles Displacement plug-in lets you set the base color for a hypervoxels particle to the color of the vertex color map, if one exists.

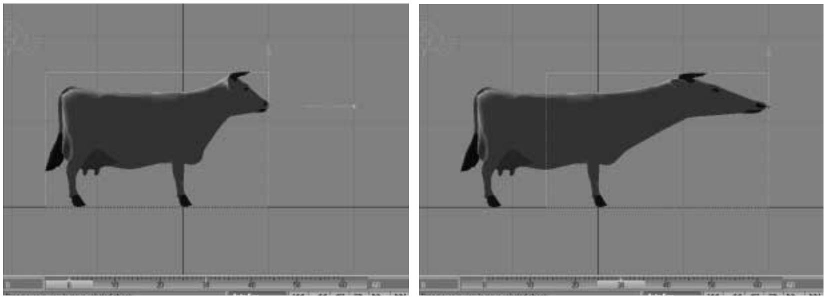


Inertia

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 (30 frames) behind.

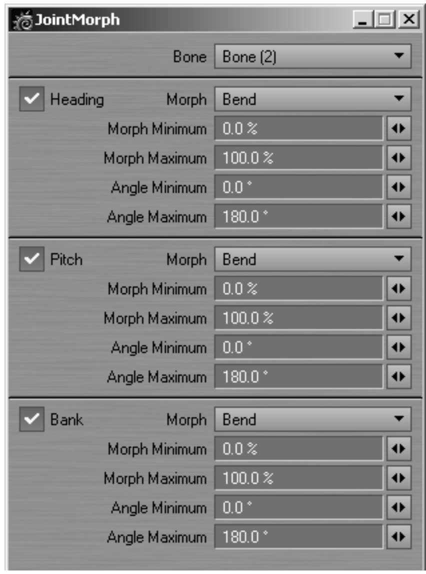


Normally, the object's movement triggers the effect; however, you can point at a different object instead by using **Inertia Pivot Object**. Then, the object doesn't even need to move to get the effect. **Inertia Pivot Object** replaces the item's pivot point, and becomes the center from which the inertia 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.

Joint Morph

JointMorph *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 plug-in 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.



Joint Morph panel

To use, just choose a **Bone** from the endomorph. This will be the bone whose rotation will affect morphing. Next, activate which rotations should influence which morph maps. They can be different, so changing the heading would morph to a different morph map than changing the pitch.

The **Angle Minimum** and **Angle Maximum** angles correspond to the **Morph Minimum** and **Morph Maximum** morph percentages. So when the bone heading is at the **Angle Minimum**, the morph will be applied at the **Morph Minimum** percentage. As the angle increases to the **Angle Maximum**, the morph increases to the **Morph Maximum**. There is no limit applied to the morphing, so going beyond the range of angles results in morph percentages beyond the range of morphs.

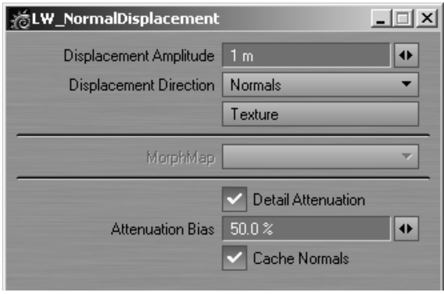
Morph Mixer

For information on Morph Mixer, see “Animating Endomorphs” earlier in the chapter.

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 the user (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.

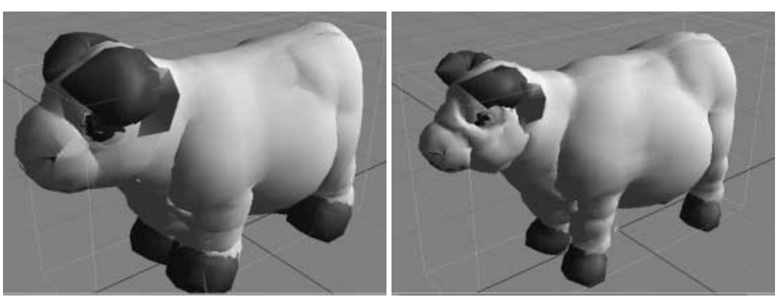


The **Displacement Amplitude** sets the amount of displacement. **Displacement Direction** lets you choose whether the direction of the displacement is along the vertex **Normals** or using **Morph Maps**. The **Texture** button brings up the Texture Editor, where you can add a texture (like in a regular 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).



Cache Normals. Left: Off. Right: On (Hmmm Cow becomes a sheep!)

NOTE
Play your scene while tweaking the settings to get real-time feedback.

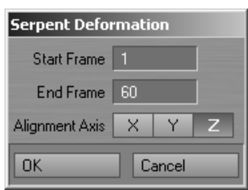
Remember that textures are three-dimensional and generally have different values for any point in 3-D space (assuming your texture is something dynamic, like Turbulence). The value of the texture at each vertex acts as a multiplier against the **Displacement Amplitude** value. So if a texture was at 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.

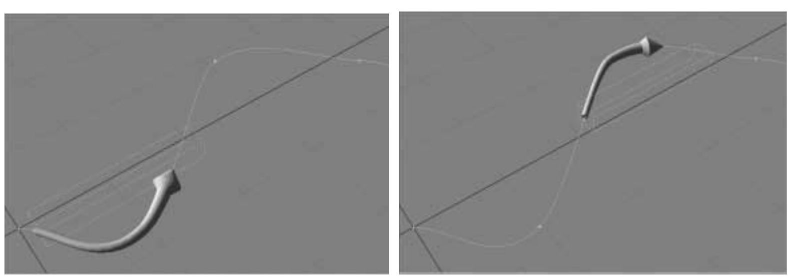
Because each vertex gets its own texture value, when using morph maps, the amount of morphing is usually different for each vertex. If the texture is animated, you will see the amount of morphing change over time.

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 plug-in that deforms an object so that it *hugs* its path. The object is expected to travel along some axis of alignment.





Serpent in action

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 maximize your efforts:

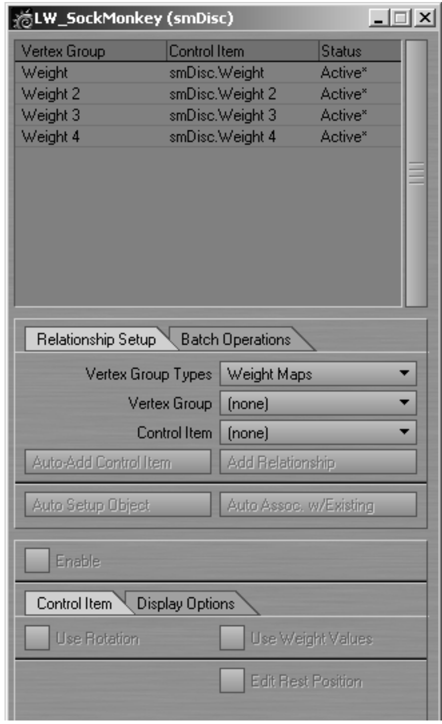
- Set up the motion path (i.e. keyframes) before applying the plug-in to the object.
- Apply the plug-in to a low-polygon-count stand-in object and refine your Scene.
- Use **Items** > **Replace** > **Replace With Object File** to replace the stand-in object with the finished version.

NOTE
Serpent does not work well with Align To Path controllers (Motion Options panel) activated for the object.

SockMonkey

SockMonkey is an animation plug-in that uses scene items—typically null objects—to deform an object's mesh. You can deform an object like it is a puppet.

SockMonkey 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.



SockMonkey panel

Basic SockMonkey Setup:

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.

2 Load the object into Layout and add the SockMonkey displacement plug.

3 Open the SockMonkey options panel. On the Relationship Setup tab, set **Vertex Group Types** to the type you used in step 1.

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.

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.

WARNING
Excessive use of this plug-in has caused certain side-effects in laboratory animals, including dry-mouth and bloating.

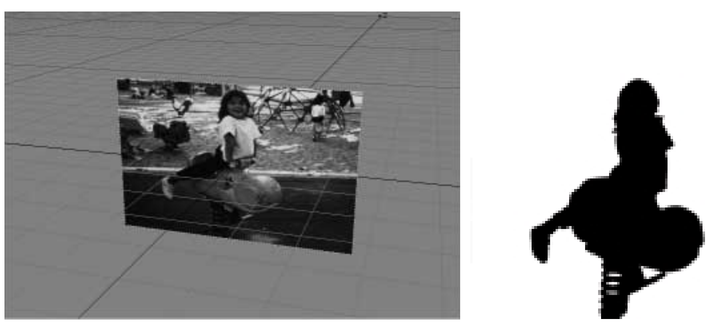
Courtesy of Ashbury Designs. (www.temecula-usa.com/Adesigns)

DISABLING DEFORMATIONS

The Enable Deformations On/Off command (Motion group) can be used to temporarily disable deformations (i.e., morphing, bones, displacement plug-ins, and displacement maps) in order to improve interactivity when editing other aspects of a scene. Assign this to a menu or keyboard shortcut for easy access, if you need it.

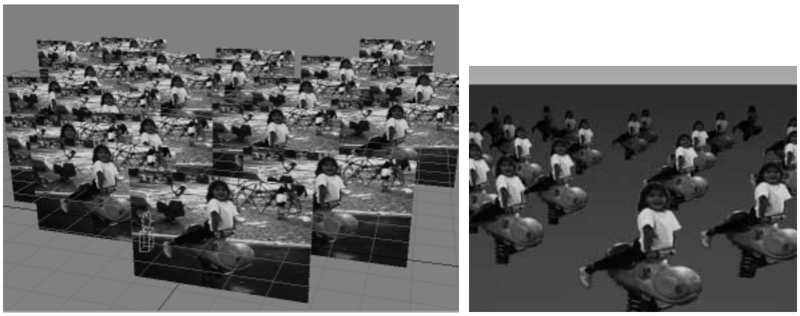
CLIP MAPPING

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 in Layout. *Right*: Clip-map image

Rendered image



Left: Object cloned 19 times. *Right*: Army of kids!

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**.

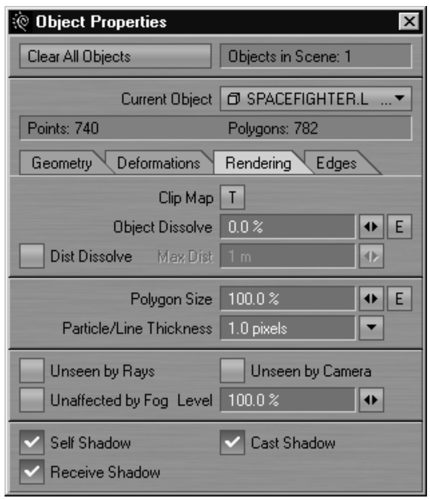
HINT

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.

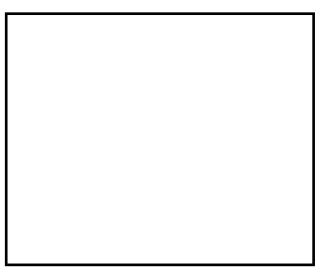
OBJECT DISSOLVE

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.



Don't confuse **Object Dissolve** with surface transparency. Transparency is applied on a surface-by-surface basis. Moreover, a 100-percent transparent surface can still have specular highlights. A 100-percent dissolved object just doesn't exist. However, in some instances you may want to use **Object Dissolve** as a global transparency effect.



The object above is shown at 100% Object Dissolve. (It's also a polar bear eating an ice cream cone in a snow storm.)

NOTE
Wireframe objects that are 100-percent dissolved will be visible in the Layout window only if they are selected.

NOTE
You can deactivate an object on the Scene Editor which is effectively the same as 100% Dissolve.

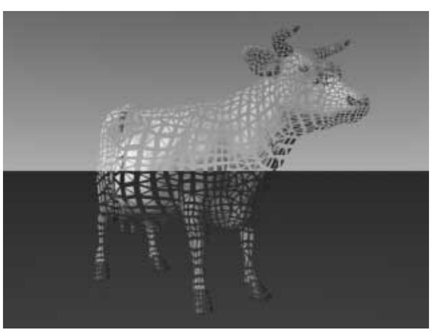
DISTANCE DISSOLVE

Activating **Dist**ance **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 generally 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.

SIZING OBJECT POLYGONS

Entering a percentage value for **Polygon Size** lets you shrink or expand an object's polygons. They will expand or shrink from the center of each polygon, creating gaps between them or overlapping. This feature is often used with enveloping to simulate objects exploding or collapsing in on themselves.

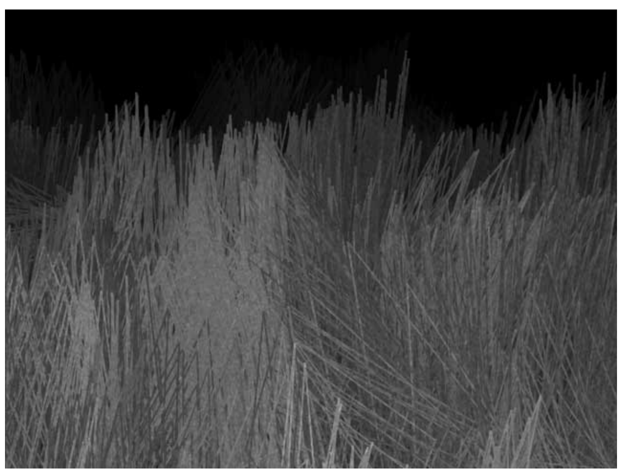


HINT
For explosions, you might want to further cut up your object with the Knife tool. Also try enveloping object sizing so that the polygons move quickly at first and then slow down as they dissolve. Adding some carefully timed lens flares would be a nice touch, too.

Particle and Line Objects

Particles are objects that are composed of single-point polygons. Lines are objects composed of two points connected into line segments (line segments may be a string of more than two points, as long as each segment is a two-point polygon). These are special objects that are always rendered at a defined size no matter how close they are to the camera.

Particles are most often used for star fields, but they can also be used to simulate smoke, jet exhaust, and sparkles, particularly when used in conjunction with some special surface shaders. Lines can be used for items like brush bristles, a field of wheat, and hair.



The **Particle/Line Thickness** value tells LightWave how big to make the particle or line in pixels. There is a pop-up menu to select from some pre-defined sizes.

Size and Aspect Scaling

The size and aspect of particles and lines are scaled automatically by the **Resolution Multiplier** (Camera Properties). Also accurate width compensation is made for any **Pixel Aspect Ratio** (Camera Properties) other than 1.0.

NOTE
The **Particle/Line Thickness** setting has no effect on objects that are not particles or lines.

HINT
To create starfields, use some of the provided starfield objects, which are really just single-point polygons arranged in the shape of a huge hollow ball. You can even use the same object a few times, each with a different rotation and **Particle/Line Thickness** value. Place the camera in the center of the point balls.

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

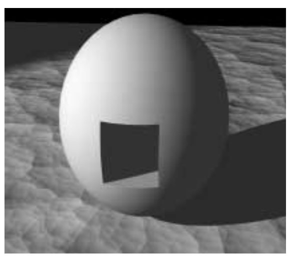
An alternative to making an object 100-percent dissolved is to activate **Unseen by Camera**. This makes the object invisible to the camera when you render; however, you will still be able to see it and work with it in the Layout window.

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.

Invisible cow casts shadow

OBJECT SHADOW OPTIONS

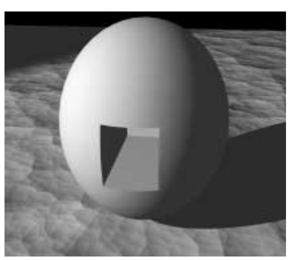
If a light casts shadows, any objects illuminated by it will generally cast shadows on themselves or onto other objects. However, you have complete control over this. LightWave lets you decide exactly which shadow options an object should employ. For example, you may wish for some objects not to cast 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.

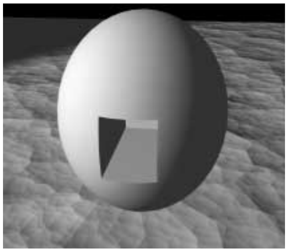
Deactivate **Self Shadow** if you do not want or need an object to cast shadows on itself. An egg is a good example of an object that cannot cast shadows on itself. A tree on the other hand is a prime candidate for **Self Shadow**.



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**) 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.

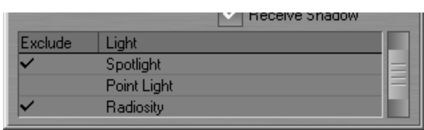
Shadow options apply whether LightWave is using ray-traced shadows or shadow maps. However, when you use shadow maps and you do not wish an object to receive a shadow, you need to deactivate both **Receive Shadow** and **Self Shadow**. Likewise, for an object that you do not wish to cast a shadow-mapped shadow, deactivate **Cast Shadow** and **Self Shadow**.

Receive Shadows and HyperVoxels

Currently, plug-ins 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 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.

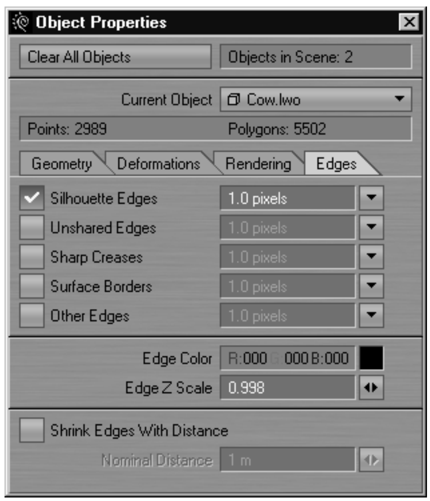


If for some reason you need the excluded lights to continue to cast shadows, deactivate the Shadow Exclusion option. When this *global* setting is not activate, all lights will cast shadows even if they are excluded. You will need to add the Shadow Exclusion command (Lights command group) to a keyboard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

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 Surfaces Editor's Advanced tab. LightWave also lets you add a **cel look** to your rendered images, like the look you see in cartoons.



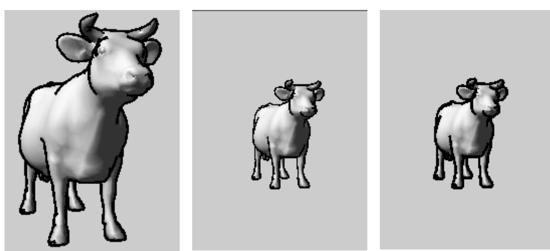
Polygon and surface edges are broken down into edge types that you can activate independently. You may also define a pixel width. You have great control over how cel-look objects are outlined.

The images from left to right are examples of different polygon edge effects:

- **Silhouette Edges** outlines the outer contours of the object.
- **Unshared Edges** are the places where two different surfaces meet, but do not share the edge of a polygon.
- **Sharp Creases** outlines places that have abrupt folds or wrinkles in the surface.
- **Surface Borders** is similar to **Unshared** Edges, but where the polygons share the edge.
- **Other Edges** are all polygon edges, except those where two surfaces meet.

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**.

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.



The far left image shows how the object looks with **Silhouette Edges**. The **Shrink Edges with Distance** option is active in the middle picture and the **Nominal Distance** is set to 3.6m, which coincidentally happens to be the same distance the Camera is away from the object. Moving the object away from the Camera, you can see the edges getting smaller. The picture on the right shows how the frame looks if the **Shrink Edges with Distance** option is inactive.

Edge Z Scale

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.

NOTE
It is unlikely **Edge Z Scale** values outside the range of .95 to 1.0 would ever be necessary.

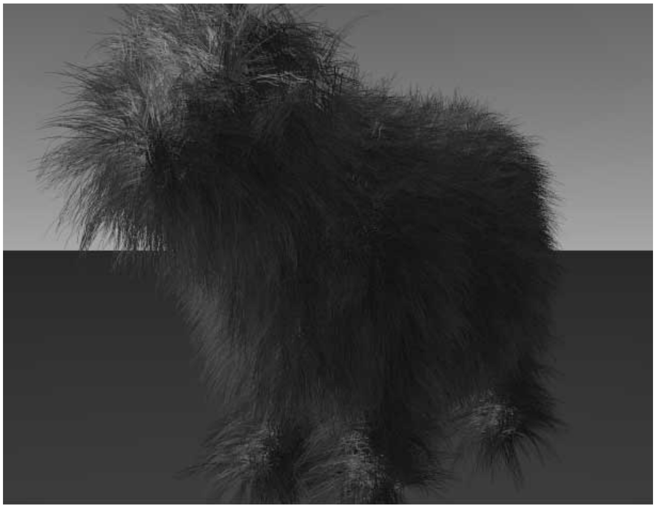
Sasquatch Lite

Sasquatch Lite allows you to quickly and easily create a variety of effects such as grass, fur, and hair on your objects. Sasquatch Lite uses its own rendering engine to render the hairs quickly and combine the results seamlessly with your LightWave objects. Sasquatch Lite is a very simplified version of the commercial plugin *Sasquatch* from Worley Laboratories.

To use Sasquatch Lite to create fur effects, you need to apply two plugins to your scene: the SasLite displacement plug-in (added on the Object Properties panel) and the SasLite pixel filter (added on the Processing tab of the Effects panel).

You “teach” Sasquatch Lite about what object(s) grow hair by adding the displacement plug-in on those objects. The hair attributes like color, length, and density are defined in the displacement plug-in's interface. Each object can have its own hair with its own style. You can even add multiple copies of the displacement plug-in to an object to add layers of hair.

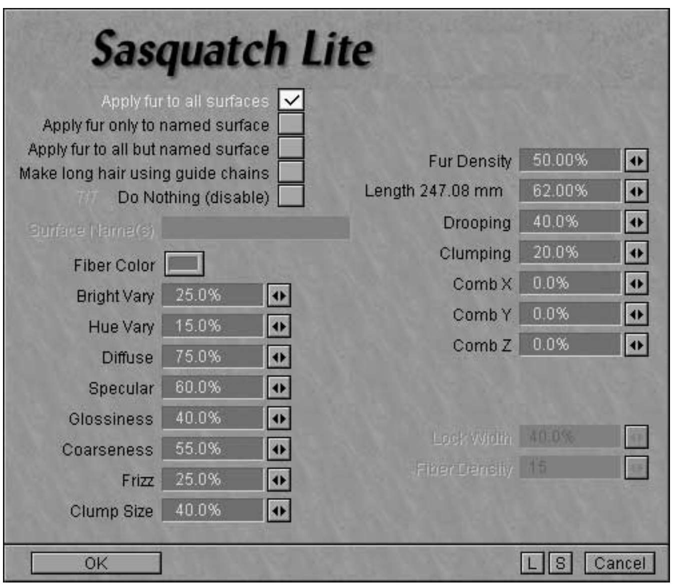
The rendering engine is controlled by the single pixel panel interface. This panel allows you to set global options such as antialiasing, and shadow control.



Default settings turn the cow into a sheepdog!

Designing Fur and Grass

The easiest effects to apply are fur and grass. The default settings of Sasquatch Lite will cause your objects to grow short brown fur all over the surface of your object. The fur itself is quite customizable, allowing you to make a huge variety of appearances.



The SasLite displacement plug-in panel

The large radio buttons in the upper left of the interface allow you to restrict the fur to appear only on certain surface(s) by entering the name in the **Surface Name(s)** field. You can use the wildcard character "*" to select multiple similar surfaces, so "Fur*" will select surface names like "Fur_back" and "Furry head". The entry is not case sensitive.

The basic color of the fur is selected with the **Fiber Color** setting. Real fur and hair have a range of hues, and you can approximate some of this variation with the **Hue Vary** and **Bright Vary** controls. These will vary the color shade and brightness of each hair randomly to give a more natural range.

The fur will react to the lights in your scene. Fur is usually much more sensitive to lighting direction than normal solid surfaces, so you often need to spend extra time to get the effects you're looking for. This sensitivity is inherent in real hair and fur too; we're used to looking at solid surfaces like wood and stone, but the thin fibers of hair react very differently. SasLite's shading captures these differences, but it does mean that you have to learn to understand the lighting more carefully than normal LightWave shading.

Like normal surfaces, SasLite fibers have **Diffuse**, **Specular**, and **Glossiness** attributes. These allow you to change the way the hairs react to light. **Specularity** is particularly important, but is also very sensitive.

The thickness of individual fibers is controlled by **Coarseness**. Human hair is very thin, but you can create much wider strands, even thick porcupine quills or grass stalks by using large enough values.

Fur and hair fibers are rarely completely straight. You can make the fibers “wiggle” more by using the **Frizz** control.

Fur Density determines how many fibers are added to your object surface. **Length** is the maximum length of each fiber. Both of these controls are relative to the basic size of your object. This makes setting values much easier, since you don't have to think about the size of your object as much. Both **Length** and **Density** are *non-linear*, so doubling the value will give you much more than double the amount of fur or length. This makes adjustments easier overall, since it eliminates extreme cases which might require bizarre settings like 100,000 percent!

The length of each fiber is partially randomized to give a more natural variation The length you specify in the interface is the maximum length.

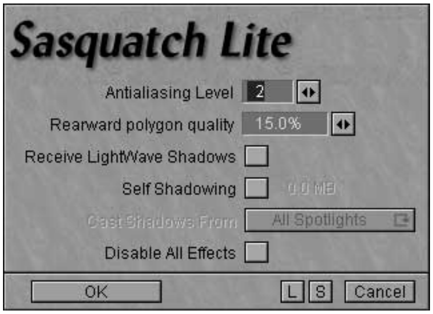
Drooping controls the way fibers bend downwards from gravity.

Clumping makes the fibers cluster together in groups. This control sets the number of fibers in each group, and **Clump Size** sets the width of each clump.

The direction of the fibers can be biased in a “combing” direction. This combing is in object coordinates, so it is safe to move and rotate your object in Layout without worrying about the fibers changing their orientation. The **Comb X**, **Y**, and **Z** controls let you specify this combing direction and strength.

Rendering

The SasLite pixel filter controls control the rendering engine used to generate the final fur effects. This engine is independent of LightWave's, so it has its own controls for antialiasing and shadowing.



The SasLite pixel filter

Higher **Antialiasing Level** values will make SasLite work harder to produce a smoother output image, though rendering may take longer. This setting is independent of LightWave's antialiasing controls.

SasLite speeds up most fur rendering by skipping the fibers on the backside of your object which are probably hidden by the object itself. This can increase rendering speed considerably, but if the fibers are very long, you may find that even the backside fibers should be visible. You can set the quality of the “backside” fibers with the **Reward Polygon Quality** setting. If your scene appears to be missing any fibers, you can increase this value to tell SasLite to work harder and be more conservative when rendering.

You can also speed up fiber rendering by telling SasLite to ignore the effect of LightWave geometry shadows. This means that objects won't cast shadows onto fibers, but your rendering speed will be considerably faster. The **Receive LightWave Shadows** control toggles this option. When active, the fibers will be affected by shadows.

Self-shadowing is the very important ability for fibers to cast a shadow onto themselves. It's important because the deep complexity of fur is very sensitive to self shadowing. Without it, fur and hair looks flat and doesn't “feel” natural. (Sparser effects like grass still look good though, since it's not as dense and there are fewer shadows).

SasLite uses shadow maps (similar to LightWave's) to allow the fibers to cast shadows onto themselves. This means that any shadow casting lights must be spotlights. You'll get the best quality if you adjust the cone angle of the spotlights to be as narrow and focused on your object as possible. You can choose to use all spotlights as shadow casters, or just the ones LightWave uses as shadows itself.

Long Hair

SasLite has a very simplified version of Sasquatch's system for rendering long hair. The shape and path of the hair is defined and animated by the use of “guide geometry,” which are chains of two-point polygons. When you render, SasLite adds bundles of hair which follow these guides. The advantage to this method is that you can create, style, and animate the hair using all of LightWave's tools.

You can create guides easily in Modeler. Just make some points. Then, select the first two points and hit the P key to make the polygon. Select the next set of two and hit P again. Do this for the length of your guide. This creates the proper kind of guide that SasLite needs. You can clone the chain, or load pre-made chains, to define the hair shape. (Use the Polygon Statistics panel to select and delete any non-two-point polygons.)

You can make the guide as simple or complex as you like (including making knots, bows, or even braids), but it has to be a chain of two-point polygons. The root end of the chain must have a different surface name than the chain itself. SasLite needs this marked end so that it knows which end of the chain to grow from.

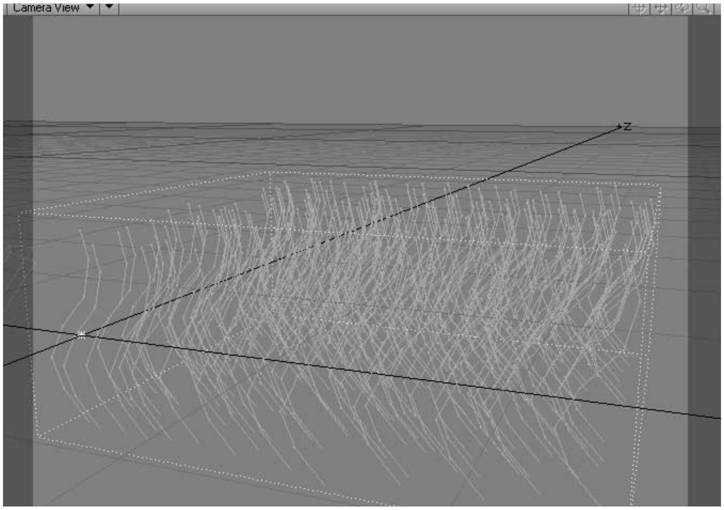
This end surface can be as simple as an extra one-point polygon placed at the root end of the chain. In practice, it's easiest to make one extra point at the end of the chain, then name the last two-point polygon something like “Root”. You can then move the first point on top of the second point to hide this extra link if you don't want it to show. You might name the surface of the rest of the chain “Hair”.

To use guides, enable the **Make long hair using guide chains** option on the displacement plug-in. Enter in the surface name of the “hair” guide in the **Surface Name(s)** field. Remember that the hair needs to be attached at the root to another surface (any surface) so SasLite knows which direction to grow from.

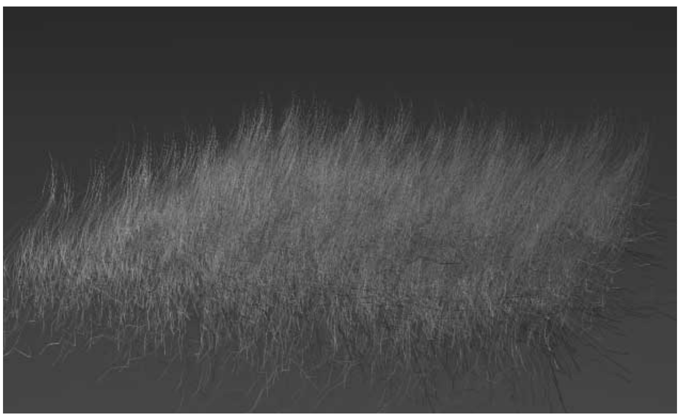
Lock Width specifies how wide the bundle of hair is relative to the length of the guide. **Fiber Density** controls how many fibers are created along the guide you create. This may be a low number if you have many guides, but you can make thick ropes by using a high enough setting. Clumping is still active with long hair, so adjusting **Clump Size** can change the way the fibers cluster together. Smaller values tend to look best.

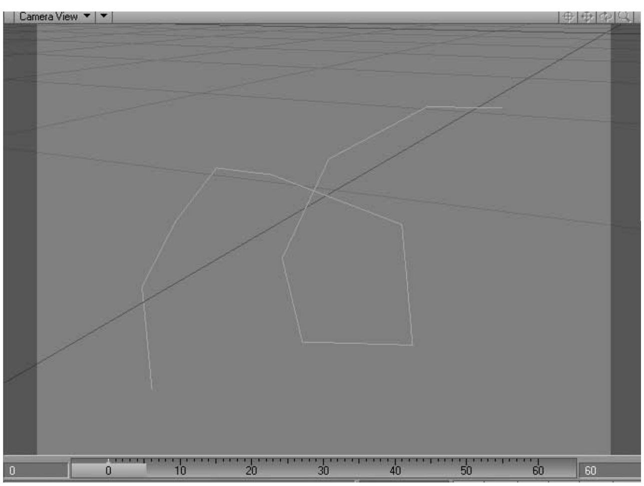
If your first trials with hair create “explosions” of hair, you should decrease the **Clump Size** and **Lock Width** of the hair until it's under control and following the guide more closely. You can then increase the values (slowly) until you get the volume and thickness you want. SasLite’s settings are also greatly influenced by the scale of your objects. You may find that you have more control using smaller objects.

Also, often your guides will be hidden in the hair. If they are visible, just dissolve them out as you would any object.

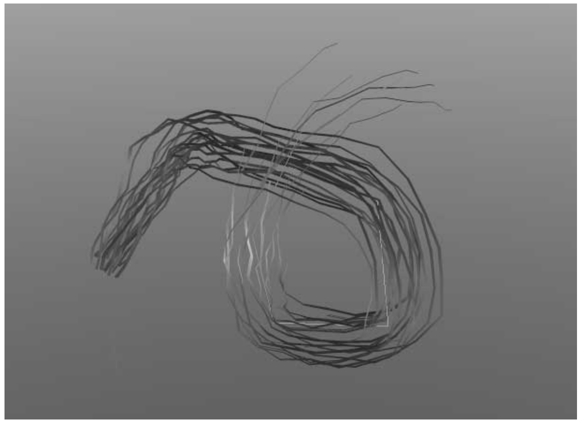


Field of hair guides


SasLite long hairs (object dissolved out)



A lone strand of hair

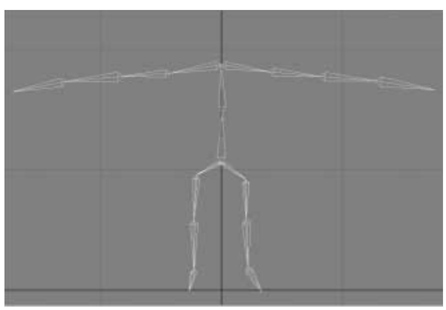
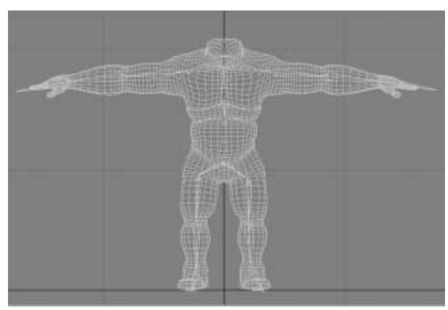


Gnarly hair from single guide

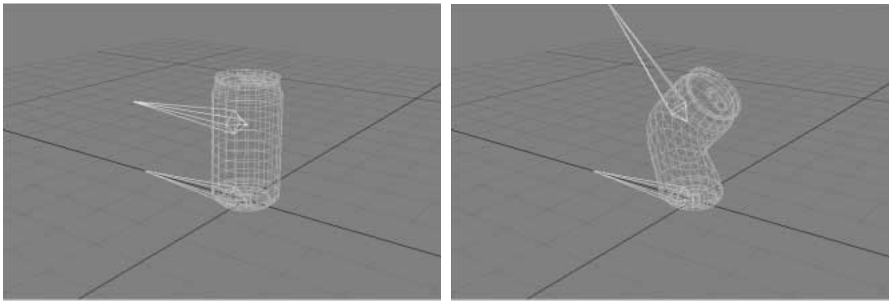
chapter 10
Bones and Skelegons

Chapter 10: Bones and Skelegons

Bones isn't a reference to a certain doctor on a certain sci-fi TV show. Adding bones to an object lets you animate the object by deforming its shape in a very controlled manner. Dancing gas pumps and credit cards, beating hearts, and undulating ghosts can all be animated with bones. In many respects, LightWave bones operate just like the bones (and muscles) in your own body, which when moved, change your own physical shape. In fact, bones are referred to as an object's skeleton. They are often set up like an actual skeleton, particularly when used inside an object that might have a bone skeleton in real life, like a person. Here all of the bones have parent/child relationships. This works well because when you, for example, rotate the bone in the upper arm, all of its children (farther down the chain) rotate with it.

However, a *skeletal* setup is not required. Here, we have just two bones in this soda can with no bone parenting. The lower bone acts as an anchor here, while moving and rotating the upper bone deforms the upper portion of the can.



Two bones deforming a soda can

Bones function by first holding a default *rest position* and *rest rotation*. After this, anytime a bone is moved, rotated, or sized away from its *rest* settings, the object will deform. How much the object deforms is based on the number of bones, their ranges of influence, and their relative strengths. It helps to think of bones as magnets that attract parts of an object (those closest to the bone) as the bone is moved, rotated, or sized away from its rest settings. Bones play against each other, so you usually need at least two. A single bone will affect every point within an object (unless an influence range is used).

NOTE
Because bones are generally set up in a hierarchy, they are often used with LightWave's Inverse Kinematics system.

NOTE
As with displacement and clip maps, bone information is saved as part of a scene file and not as part of an object. In order to load an object and its bones into a scene, use **Load From Scene**.

BONE-FRIENDLY OBJECTS

Since objects can be greatly deformed by bones, it is a good idea to get into the practice of using three-sided polygons in order to keep rendering errors from appearing. Tripled polygons, by their very nature, cannot be non-planar and therefore will render correctly.

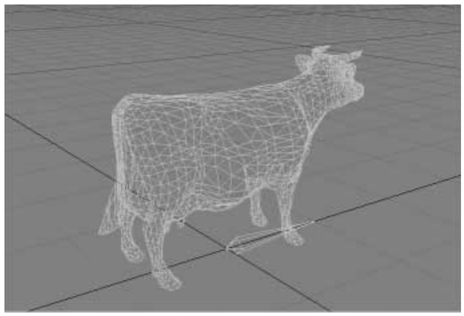
SELECTING A BONE

You can select a bone to edit as you would any item in layout. The Bones panel can be displayed by selecting a bone and then pressing the P key. From this panel you can switch between different object skeletons (i.e., bone hierarchies).

NOTE
It is very important to understand that each object in a scene can have its own independent bone hierarchy.

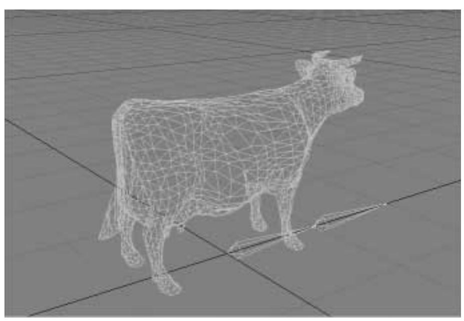
To add a default bone to your object:

Select the object and then choose **Items** > **Add** > **Bones** > **Add Bone**. The bone will be added with its base at the object's local Origin, along its positive Z axis.



To add a child bone to an existing bone:

Choose **Items** > **Add** > **Bones** > **Add Child Bone** and an identical bone will be added with its base at the tip of the selected bone. The selected bone is automatically the parent. As you create bones, the last created one is always selected. Thus, you can quickly add child after child. If there is no bone selected, this is the same as just adding a bone.



HINT
Since you are often adding multiple child bones, remember the shortcut (=) to immediately add a child bone.

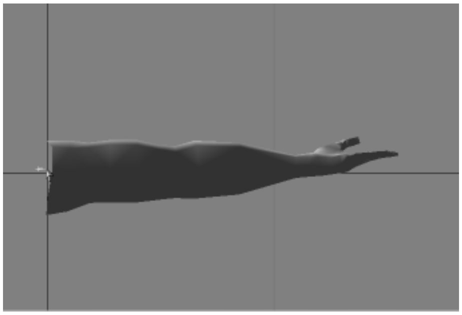
NOTE
You will be prompted for a name when adding bones. Click **OK** to accept the default.

To draw a bone in a viewport:

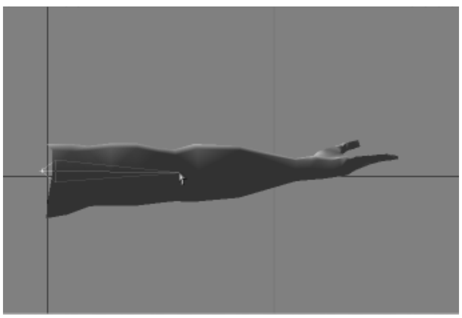
1 In any Layout orthogonal-view viewport, select the object, or existing bone if you are drawing a child bone.

2 Select **Items** > **Add** > **Bones** > **Draw Bones**. (Since this is the initial bone, you can alternatively choose **Draw Child Bones**.)

3 Place your mousepointer where you want to place the base of the bone.



4 Drag your mouse to create the bone.



To open the Bones panel:

Select a bone and press the P key or click **Item Properties**.

BONES FROM ANOTHER OBJECT

You can *borrow* the bones from another object by selecting the *lending* object in the **Use Bones From Object** pop-up menu.

ACTIVATING AND RECORDING REST POSITION

Once a bone is placed <u>and keyframed</u> in the desired rest position 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.

To change only the activation state of a bone (and not adjust the rest position), you can: use CTRL+R to toggles the state, use the **Bone Active** option on the Bones Properties panel, or use the activation checkbox on the Scene Editor.

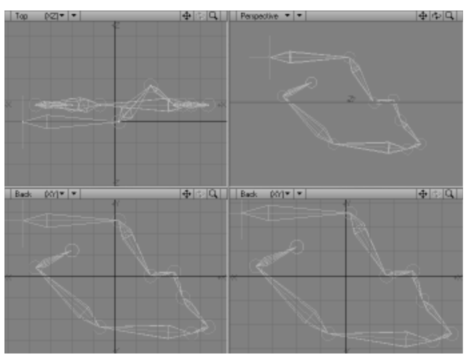
NOTE
When adjusting a chain of bones individually you'll need to keyframe each bone (or use **Auto Key Create** on the General Options tab of the Preferences panel) before moving on to the next bone.

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 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.

SKELEGONS

Another way to create a bone hierarchy is to use LightWave's *skelegon* feature in Modeler. Skelegons are a polygon type that can be added to your object in Modeler; they *look like* bones. You can create a complete skeletal structure out of polygons, hence the name skelegons. Skelegons set up only *placeholders* for the bones. When you load an object with skelegons into Layout, the skelegons must first be converted to bones before you can use them for skeletal deformation.



There are several advantages to this system. First, you can use existing points in a character to create the skelegons, if you wish, or you can draw your own skelegons manually. Having this function as a modeling tool makes the creation process much easier, as building a skeleton is inherently a modeling task. Since the skelegons are considered polygons, you can use any of the modeling tools to create and modify the skeleton.

Another benefit is that the skelegons are saved in the object file itself. This means that you no longer need to create a separate scene file to hold bone positions.

One of the most beneficial features of this system is that when you modify the shape of the model, the bone structure can be modified at the same time, so you do not need to make secondary modifications to that skeleton.

Imagine building a character and spending the time to create the bone structure. Now the client decides they would rather have the character with shorter legs and really long arms. Typically, the modeling changes would be simple but recreating the skeletal system would be an arduous task at best. With skelegons the process becomes one simple step of modifying the geometry.

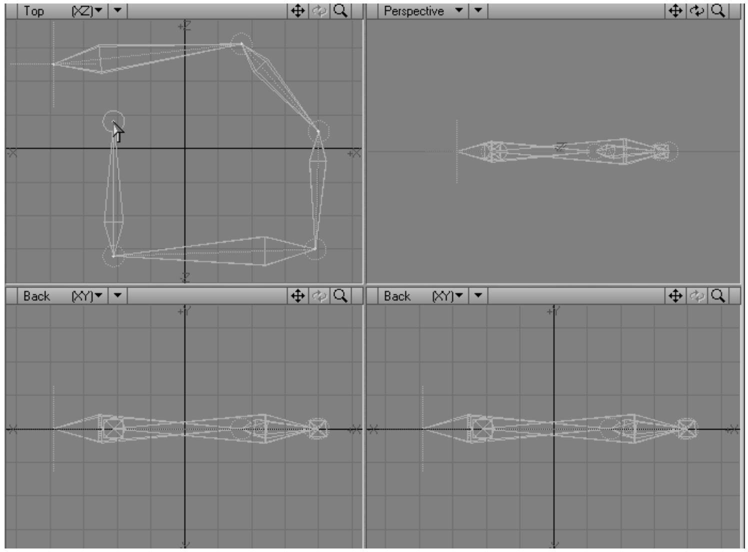
With skelegons it is also feasible to have a series of base skeletons that you store and use as templates. The rubber band editing style of the skelegons makes it very simple to modify an existing skeleton to fit all types of characters.

NOTE

Remember that skelegons are only placeholders for bones. Also, there is no way to convert bones back into skelegons.

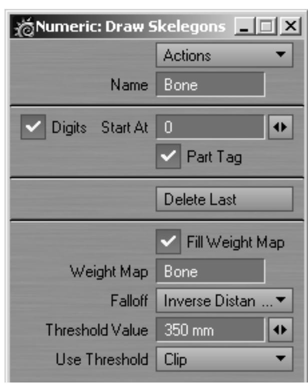
Drawing Skelegons

Like drawing bones in Layout, you can use the Draw Skelegons tool to draw skelegons directly into a Modeler layer. Just choose **Create** > Elements: **Skelegons** and drag your mouse. When you release the mouse button a skelegon is created. With the tool still active, you can drag your mouse again to create yet another skelegon attached to the first. Select the tool again to turn it off.



Drawing Skelegons

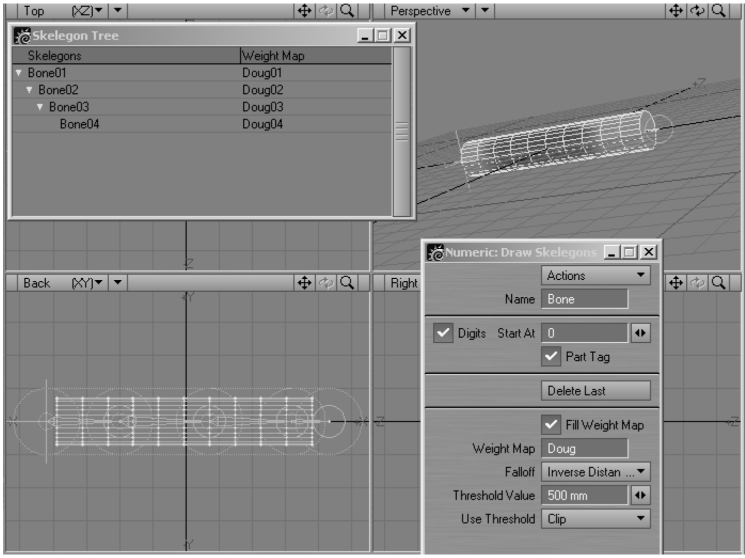
You can customize the **Name** used for each skelegon, as well as the **Weight Map** name, on the numeric panel. When **Digits** is active on the numeric panel, bone names are incremented, like Bone01, Bone02, and so on. You can specify the initial number in the **Start At** field. If the **Part Tag** option is active, the bone name will also be used as a Part name automatically. Click the **Delete Last** button to delete the last skelegon being created.



Draw Skelegon numeric panel

The lower portion of the dialog combines the functions of the **Bone Weights** plug-in (**Map** > Weight & Color: **Bone Weights**), discussed later. Activate **Fill Weight Map** to gain access to these functions. Remember that weight maps are part of geometry, not skelegons. Thus, you need some geometry in the foreground to use this feature. The weight maps will be automatically created for the geometry.

Note that you can also activate the **Fill Weight Map** option after you have finished drawing your Skelegons or even toggle it on and off. This, however, must be done prior to making the Skelegons, by dropping or switching the tool.



Creating weight maps with Draw Skelegons tool

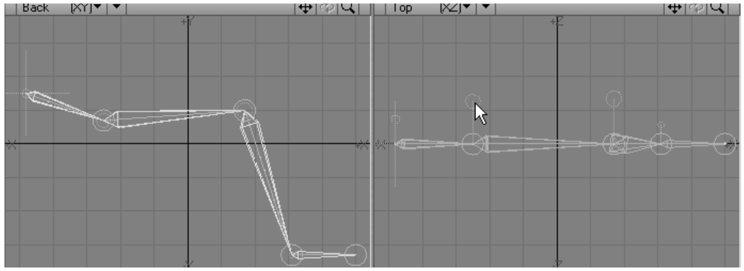
Define the base weight map name in the **Weight Map** field. The maps will be named sequentially (e.g., Bone01, Bone02, and so on). The **Falloff** setting corresponds to the falloff that you can use on bones normally when you are not using weight maps. (See “Falloff Type and Limited Range”, later in the chapter). The resulting weight maps will give you a nice starting point for modifying bone influences.

The **Threshold Value** is a distance from the bone that defines a capsule-shaped region.

The **Use Threshold** pop-up has several settings. This will take the weight value at the **Threshold Value** distance and either: ignore it (**Off**); set all weights outside the distance to 0 (**Clip**); subtract the threshold weight from all weights, so the weights go smoothly to 0 at the threshold, then start to become negative (**Subtract**); or subtract the threshold weight value as above, then clip the negative weights to 0 (**Blend**).

Bank Rotation Handle

When you edit skelegons, a bank rotation handle will appear. The default angle for the handle is perpendicular to the viewport the skelegon is first drawn in. Drag it to change the bank rotation to apply when it’s converted into a bone.



The mousepointer is over the bank rotation handle

NOTE
The resulting bones will have only a 0 degree bank if they are drawn in the Top view, assuming you do not adjust the skelegon rotation handle.

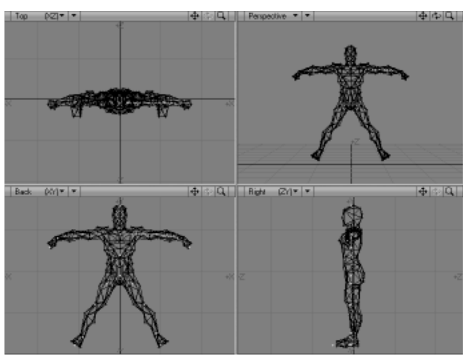
Make Skelegon

Another 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 **Make Skelegon**, you can even use points from the object to create the curves.

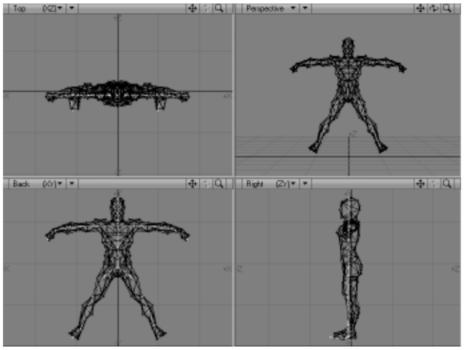
To use Make Skelegon:

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.)

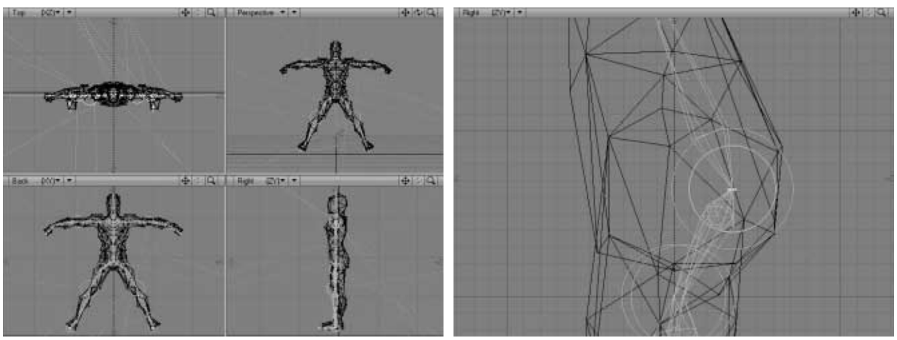
2 Select points, working from the (theoretical) base of the first child bone to the tip of the last bone in the chain. Then choose **Create** > Elements: **Make Curve** > **Make Open Curve** to create an open-ended curve with your points.



3 Repeat the previous step for all chains, if you have more than one. You can use the same point in multiple chains.



4 Choose **Construct** > Convert: **Make Skelegons** to create the skelegon from the curves. You can edit the base/tip of the bones using standard modeling tools. The skelegon bones will stay attached to each other.



If you desire, you can select and delete the curve afterward.

5 If you want, you can optionally cut and paste the skelegons into the layer with the normal object geometry.

Editing Skelegons

You can edit existing skelegon (chains) by simply selecting the lowest skelegon in the chain you wish to edit. Then, choose **Detail** > Other: **Edit Skelegons**. You can also edit skelegons along with an object's normal polygons by selecting them at the same time.

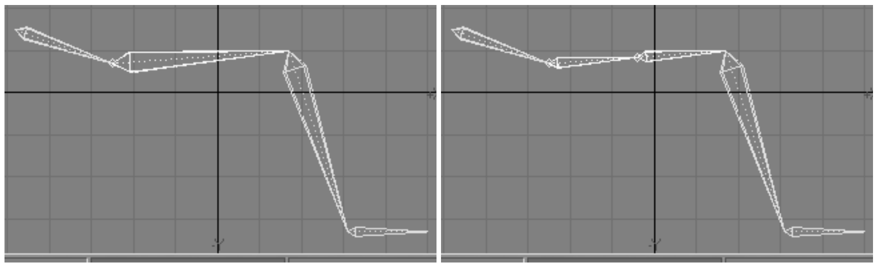
If you want the skelegons to use the same points as your object, copy the points you want to use as beginning and end points for the skelegons to an empty layer. Use the **Make Skelegons** command. Then cut and paste the skelegons into the normal object layer. Finally, merge points.

Changing Skelegon Direction

The **Flip** command (**Detail** > Polygons: **Flip**) will flip the direction of selected skelegons. (This command also flips the direction of a polygon's normal.)

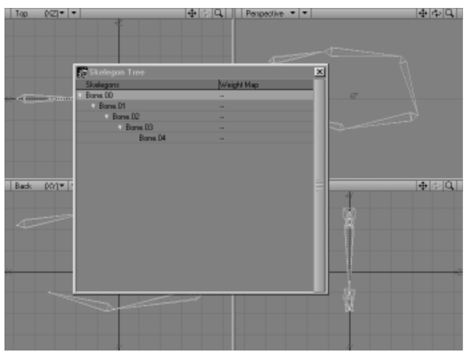
Splitting Skelegons

You can split selected skelegons in two using the Split Skelegon command (**Construct** > Subdivide: **Split Skelegon**).

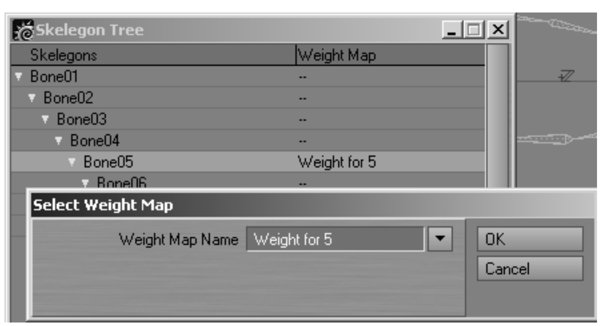


The Skelegon Tree Panel

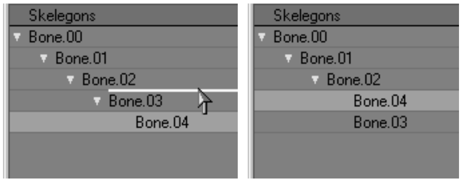
If you open the Skelegon Tree panel (**Detail** > Other: **Skelegon Tree**), you can view the hierarchy of skelegons. Skelegons will be automatically parented if Draw Skelegons was used, for example.



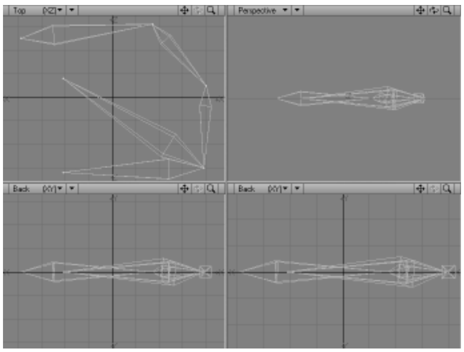
The Weight Map column lists associated weight map names. Double-click in the Weight Map column to specify a weight map.



You can also change the hierarchy by dragging and dropping individual skelegons in the list.

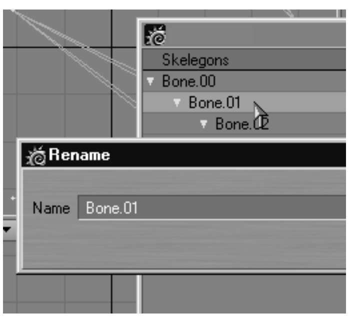


The same points will be used, but the connections will change to reflect the revised hierarchy.



You can rename a skelegon by double-clicking its name in the list and entering the new name in the dialog that appears. The names you assign become bone names when the skelegons are converted to bones in Layout.

You can also rename a skelegon by selecting it in the viewport and then executing the **SkelegonRename** command.



Converting Skelegons to Bones

Skelegons cannot be used directly in Layout. They must be converted into bones first.

To convert skelegons into bones:

1 In Layout, select the skelegon object.

2 Choose **Items** > **Add** > **Bones** > **Convert Skelegons into Bones**. This converts the skelegons, which are only placeholders, into actual bones. Note that the bones are immediately ready for action. Rest positions are set automatically and bones are activated.

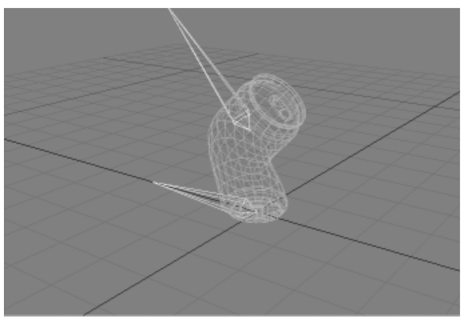
3 If your skelegons were in a Modeler layer by themselves, you need to point the regular object to the newly created bones. Select the regular object and then open its Bone panel. Select the skelegon object in the **Use Bones from Object** pop-up menu.

NOTE
If you want, you can simplify the process by cutting and pasting the skelegons into the same layer as the object before saving. Then you can avoid the **Use Bones from Object** feature.

BONE STRENGTH

Bones work against each other. Thus having only one bone with unlimited range in an object is the same as having no bones—the entire object is affected by the changes to the bone. The real power of bones appears when you have more than one bone in an object.

When you have more than one bone present, each bone will affect the points closest to it more strongly (again, assuming unlimited range for each bone). For this reason, it is common to place strategic *anchor bones* that do not move, to keep selected parts of objects from being deformed by other bones in the object.



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.

A bone with a **Strength** value double that of another has twice as much influence over points than the *weaker* bone. Pulling these two bones equally in opposite directions would cause the object to move toward the bone with greater **Strength**. The object, however, won't move completely with the stronger bone because the weaker bone still exerts some influence. If the bones had equal **Strength**, the object would not move at all, as each exerts the same amount of **Strength** on the object.

L-R: All bones equal; Middle bone stronger; and Middle bone weaker

Scaling the Strength

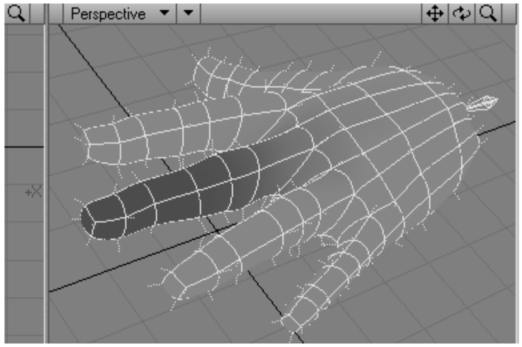
The **Multiply Strength by Rest Length** option (Bones panel) causes a bone to multiply its **Strength** by its **Rest Length** to determine the influence of the bone. Bones with larger rest lengths will exert greater influence over other bones with equal **Strength** values, but smaller rest lengths.

Using Weight Maps

Weight maps scale falloff-based bone effects rather than completely replace them. *Boned* characters can then be easily *cleaned up* with weight maps to eliminate unwanted influences. (See Chapter 28 for more information on weight maps.)

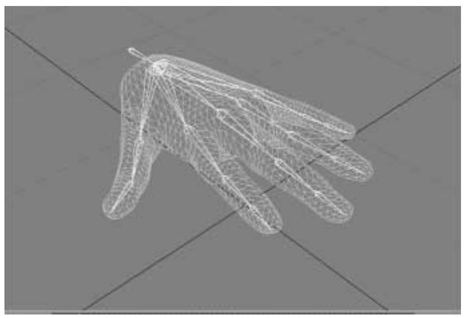
NOTE
Remember that weight maps are created in Modeler. The maps are assigned to bones in Layout.

Each bone affects each point according to a *weight*, which is either a function of distance or a weight map value. Here is a nice hand object. The middle finger weight map is shown—actually each finger has its own weight map.



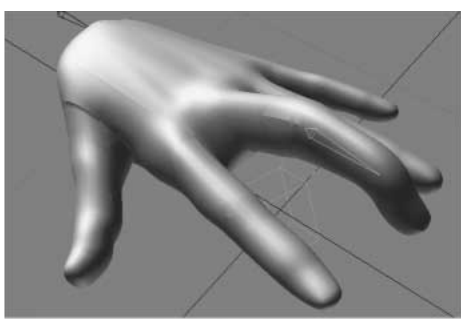
Weight map on finger

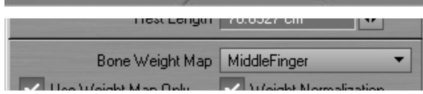
A simple bones hierarchy was added to this object in Layout. Normally, if we tried to bend the bones in one finger, we would still get some movement in other fingers.



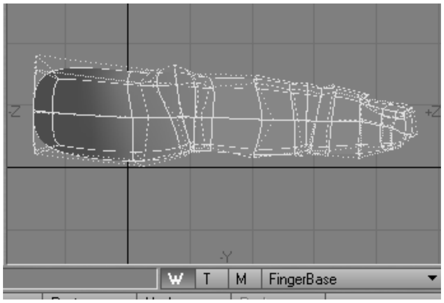
Bones added to hand

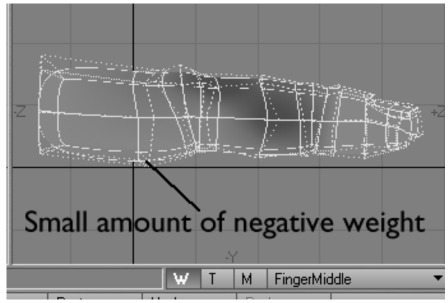
Below, the appropriate weight maps have been assigned to each bone using the **Bone Weight Map** pop-up menu on the Bones panel. So now if you bend the bones in the middle finger, they affect only points with non-zero weights in the weight map for the middle finger.

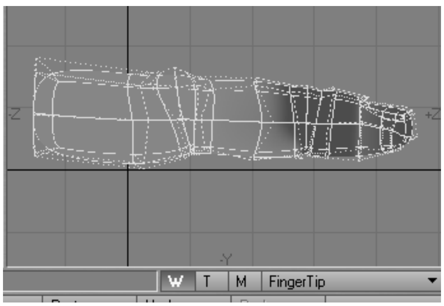




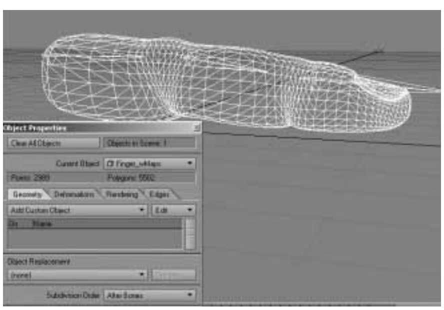
Below, three different weight maps have been applied to a finger; one for each bone. Since the screenshots are not in color, it is difficult to see the true falloff of the weight map; however, you can see that the different weight maps will allow deformation more in some areas and less in others.

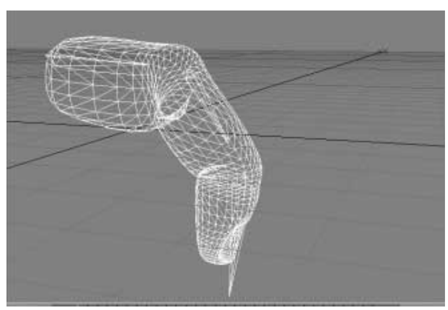






With SubPatch objects, you should set the **Subdivision Order** to **After Bones** on the Object Properties panel to avoid weird polygons after the bend.

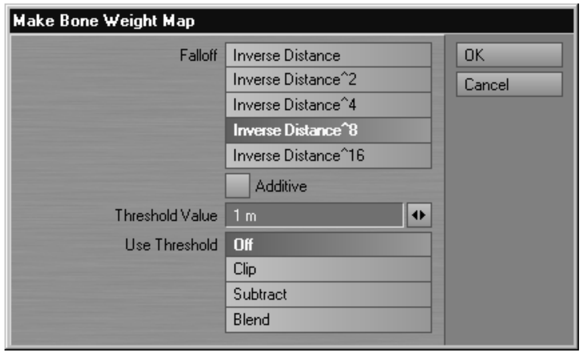




Bone Weight Help

Modeler's **BoneWeights** plug-in (**Map** > Weight & Color: **Bone Weights**) uses a **Falloff** algorithm to fill a weight map when you use the Skelegons in the foreground. The BoneWeights correspond to the falloff that you can use on bones when you are not using weight maps. (See "Falloff Type and Limited Range", later in the chapter). This plug-in gives a handy starting point for modifying bone influences. The weight maps match the bone names.

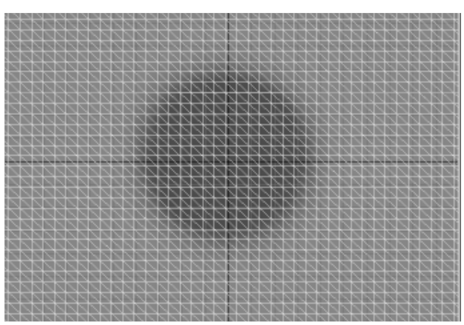
Remember that weight maps are part of geometry, not skelegons. Thus, you need some geometry in the foreground to use this feature. The weight maps will be automatically created for the geometry.



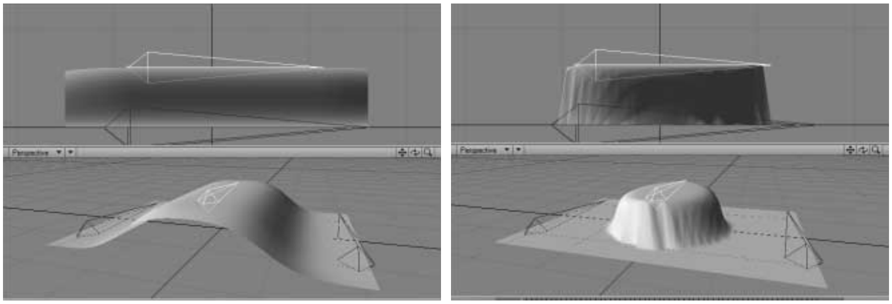
The **Additive** option adds the weight to any existing value. So if both left and right arm bones were named armbone.01, the weight map would show the effects from both arms. The **Threshold Value** is a distance from the bone that defines a capsule-shaped region. Using **Threshold Value** will take the weight value at this threshold distance and either: ignore it (**Off**); set all weights outside the distance to 0 (**Clip**); subtract the threshold weight from all weights, so the weights go smoothly to 0 at the threshold, then start to become negative (**Subtract**); or subtract the threshold weight value as above, then clip the negative weights to 0 (**Blend**).

Weight Normalization

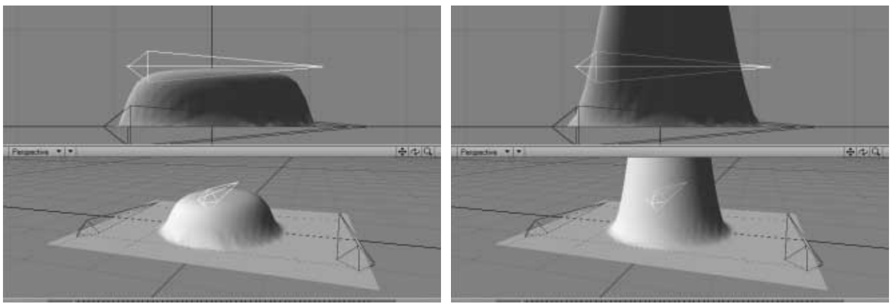
If for some reason you want weight maps to be the sole control for influence, activate the **Use Weight Map Only** option. You should also activate the **Weight Normalization** option, and your bone influences will normally be scaled based on distances and weight map. If you deactivate **Weight Normalization**, only the bone's weight map is used, which can cause wild influences unless the map is made of precise values.



Weight map on flat subdivided plane



Left: Normal effect lifting a center bone. *Right*: Using the weight map

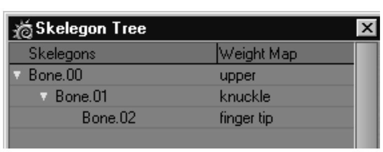


Left: **Use Weight Map Only** active. *Right*: **Weight Normalization** inactive

Auto Weight Maps with Skelegon

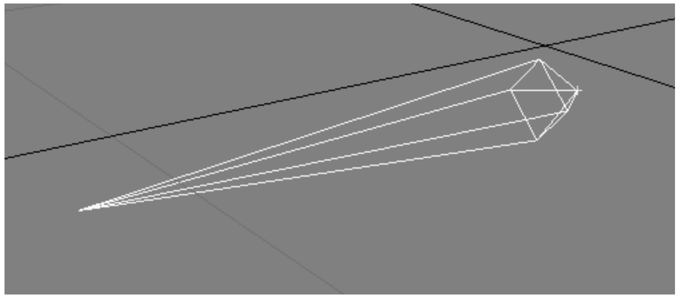
If you have created a skelegon hierarchy in Modeler, you can automatically create the weight map names, one for each skelegon, by running the **Set Skelegon Weight** command (**Detail** > Other: **Set Skelegon Weight**). Note that this merely sets up the association to map names. You still need to create and edit the weighting values as you would normally. The associated map names are automatically assigned to the bones and will appear as the **Bone Weight Map** setting on the Bones panel.

The previously described SkelegonTree panel will list map assignments.



AREA OF INFLUENCE

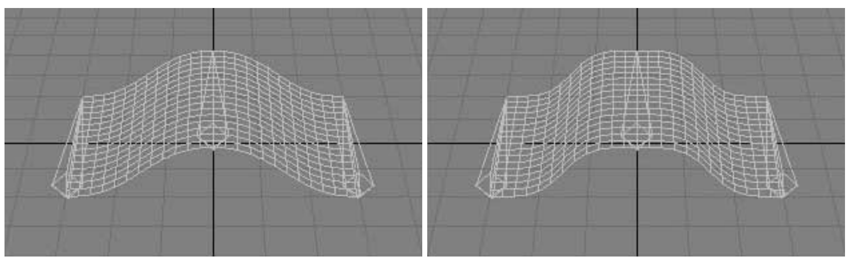
Bones are represented in Layout as pyramids that connect at the base. Although it appears as if a bone has a specific base and tip (and for rotational purposes, it does), the area of influence of a bone is equal around it. The length of a bone is adjusted by changing its Rest Length. You can use either the **Bone Rest Length Tool** (**Objects** > Bones: **Rest Length**) or adjust it numerically on the Bones panel.



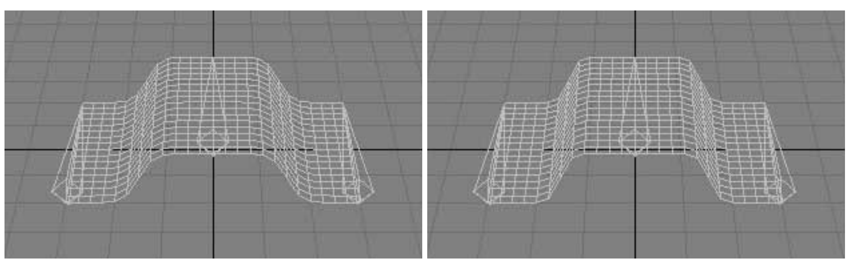
A bone

FALLOFF TYPE AND LIMITED RANGE

The **Falloff Type** (Bones panel) setting determines how the influence of bones weakens over distance. The settings indicate the mathematical formula used to determine the falloff of the influence. **Inverse Distance ^ 4** is the default setting and should work in most cases. The higher the exponential number (i.e., ^2, ^4, ^8, ^16), the faster the falloff of the bone's influence. Because the falloff is exponential, as opposed to linear, control over nearby points can be maintained with much less influence over distant points.



Left: Inverse Distance^2. *Right*: Inverse Distance^4

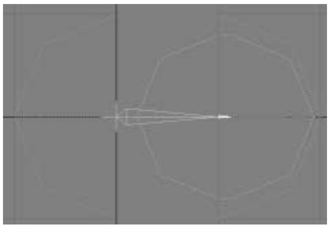


Left: Inverse Distance^16. *Right*: Inverse Distance^64

NOTE

Generally, proper use of **Falloff Type** will make using **Limited Range**, discussed subsequently, unnecessary in most cases. The **Falloff Type** affects all bones in a particular object skeleton.

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 less 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.



Min is very powerful and points within this range are greatly influenced by the bone, following the bone movements exactly (unless an extremely small bone **Strength** is applied to the bone). **Max** determines the gradual *falling off* point of a bone's influence. The strength of the influence falls off from the edge of the **Min** down to 0 at the **Max**. Any points falling outside of this range are not affected at all by the bone.

Generally, to get the best effects from using bones, you should either use bones that all have unlimited range, or that all have limited ranges. Character animation is generally easier using limited ranges on all bones.

Often you will want your **Min** and **Max** to be equal. Typing in a **Min** that is bigger than the **Max** will increase the **Max** to the same size. Likewise, typing in a **Max** smaller than the **Min** will force the **Min** down.

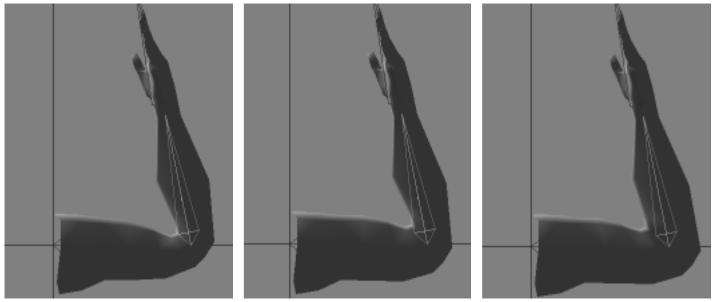
JOINTS AND MUSCLES

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 Compensation for Parent (Bones panel) 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.

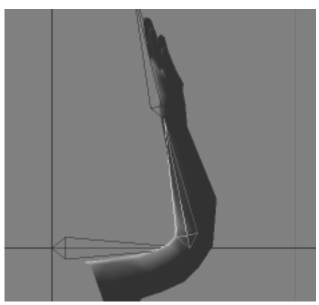
The default value of 100% was chosen as a value that is applicable for most instances. Negative values can also be used, which will increase the pinching effect.

NOTE
The joint is always located at the base of the current bone. Adjusting **Joint Compensation for Parent** is not the same as adjusting the parent bone's **Joint Compensation**.


L-R: Off; Joint Compression 100%; Joint Comp for Parent 100% added

LightWave also makes it easy to create realistic flexing muscles when you use bone structures in the object. When you activate **Muscle Flexing** (Bones panel), the points affected by the bone will flare out as the bone is rotated. The effect simulates muscles bulging under the skin.

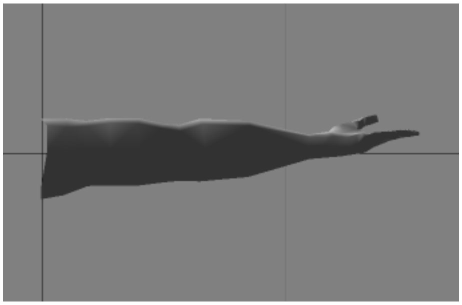
Muscle Flexing for Parent (Bones panel) is generally more common than **Muscle Flexing**. For example, when your forearm rotates your biceps flex. The forearm is the child and the bicep is the parent. It would not look natural for the forearm to bulge as it rotated, but it looks perfectly natural when the biceps flex.



NOTE
Joint Compensation and Muscle Flexing react only to pitch rotations. To react to other rotations (or just for more precise control), you could model a *muscle* Endomorph object and use ChannelFollower (or an expression) on the morph target's MorphMixer channel (on the Graph Editor) to follow the bone's rotation. You'll need to set ChannelFollower's **Scale** down to something like one percent or less.

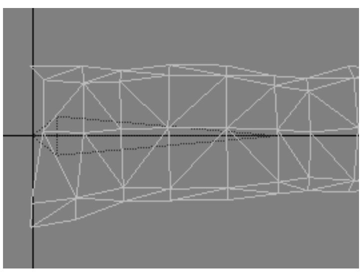
Exercise: using bones

1 Load the ARM_LOWPOLY.LWO object (TUTORIALS folder) into a clear scene in Layout. Use the Side view. Note that it just happens to be modeled along the positive Z axis. Just what we need. What a coincidence!



2 Select the arm object and choose **Items** > **Add** > **Bones** > **Add Bone**.

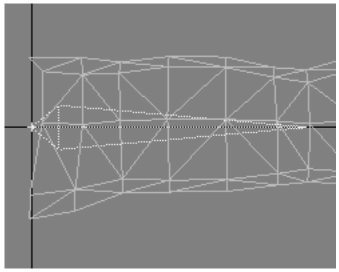
3 If you are using a shaded display, you probably won't see the added bone. You can make it visible by making it the current item, but since we plan to add several bones, using a wireframe display works best. So, change the viewport's **Maximum Render Level** to **Front Face Wireframe**. You will see the bone inside the arm—try looking at the object using different views to see that the bone is indeed inside the arm.



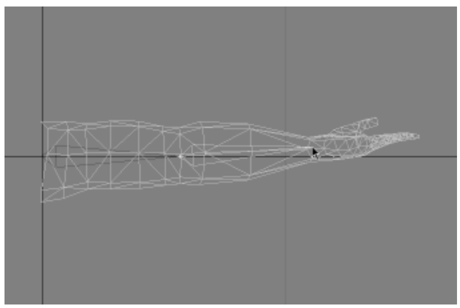
4 Notice that the tip of the bone is not quite at the elbow. Since we need it to bend there, we must increase the bone's Rest Length. Activate the **Bone** edit mode button at the bottom of the interface and then click **Objects** > Bones: **Rest Length**.

The arm was the current item, so activating the **Bone** edit mode lets us edit its bones. If other objects were in the scene, we would need to select the bone directly in the viewport or select the appropriate object first.

5 Drag your mouse to the right until the **Rest Length** is about 1.13 (the indicator is in the lower-left corner). The bone's tip should be just about at the elbow. Note that we don't need to keyframe this change.

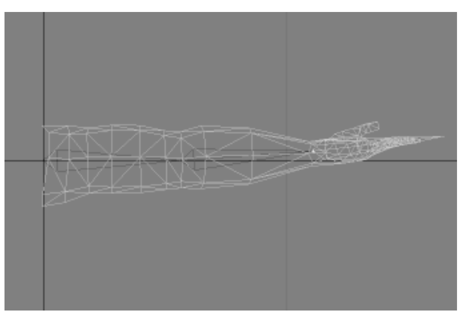


6 Next, let's add a child bone and for fun, we'll draw it. Choose **Items** > **Add** > **Bones** > **Draw Child Bone** and drag your mousepointer to the the wrist.



7 Now add another child bone, but don't draw it. Choose **Items** > **Add** > **Bones** > **Add Child Bone**.

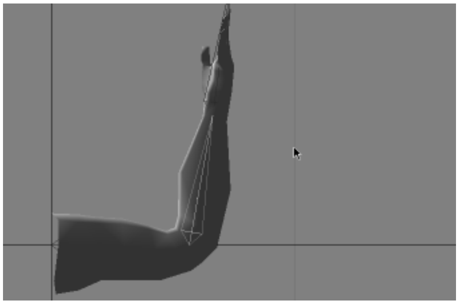
8 The last bone will extend past the tip of the fingers but that's okay. However, select **Items** > Tools: **Rotate** and change its **Pitch** a little (-2.4 degrees) to align the bone with the hand a little better. You do need to re-keyframe changes to position and rotation, so re-create a keyframe for this bone after the rotation change.



As you can see, you position bones the same way you position objects.

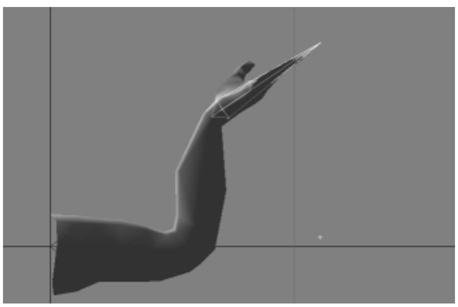
9 Press the R key to set the rest position and activate the current bone. Notice that the bone's outline becomes solid and bright. Press the UP ARROW key to select the middle bone and activate it. Do the same for the first bone.

10 Now for the fun part! Select the middle bone and rotate its Pitch. (Switch the viewport back to **Textured Shaded Solid**.)



Cool!

Select the last bone and rotate its Pitch.



OK. Now walk like an Egyptian... (stop whistling!)

11 Select the middle bone and play with the **Joint Compensation** and **Muscle Flexing** options (Bones panel) to see their effects.

12 Try varying the bone **Strength** for different bones and observe the effects.

chapter 11
Item Motion Options

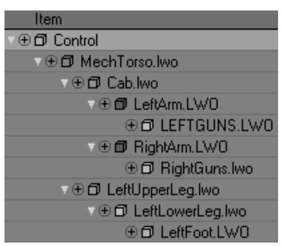
Chapter 11: Item Motion Options

Generally, the Motions Options panel controls the overall motion of an item in your scene, as opposed to individual channel control, which is accomplished using the Graph Editor.

PARENTING

Parenting refers to LightWave's ability to create hierarchical relationships between items, much like files use directories. These relationships can grow into a complex hierarchy with multiple items parented to an item and the *child* items being *parents* themselves of other items.

You can parent just about anything to anything. That is, lights can be parented to objects, objects can be parented to a camera, and so on. You can also parent items to bones; however, bones must remain under the object they relate to. Generally, you'll restrict parenting to the same type of item (e.g., objects to objects); however, there are often times when, say, parenting a camera or light to an object can be very handy.



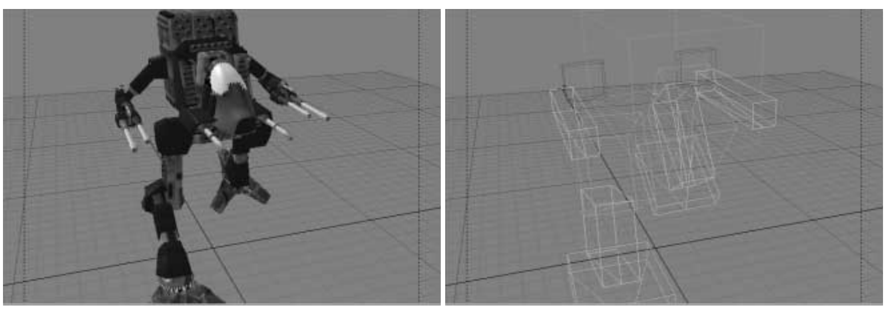
Scene Editor List showing indented hierarchical relationships

When you initially parent an object to another, you align its local Origin with the local Origin of the parent object. The child inherits the position, rotation, and scale of its parent. You can also animate the child independently, but its movement will be *added on to* the parent's.

A simple example involves a camera parented to a car object. As the car moves throughout the animation, the camera follows its movements, thus providing the look of a camera mounted to the car. Additionally, the camera may have its own motion, so it could go up and over the car while still moving along with it.

Camera parented to car

A more complex example involves a robot walking. The hands of the robot are parented to their respective forearms, which in turn are parented to their respective upper arms and so on. As the main parent is moved (in this case, perhaps the torso of the robot), all the child objects follow. Of course, all objects can have their own motions, which lets you flail the robot's arms about as it runs.



Left: Solid textured display. *Right*: Bounding boxes show individual objects

HINT

If possible, try to set up your hierarchical relationships before you animate the individual items. Then work from the *top down* (i.e., parent to child).

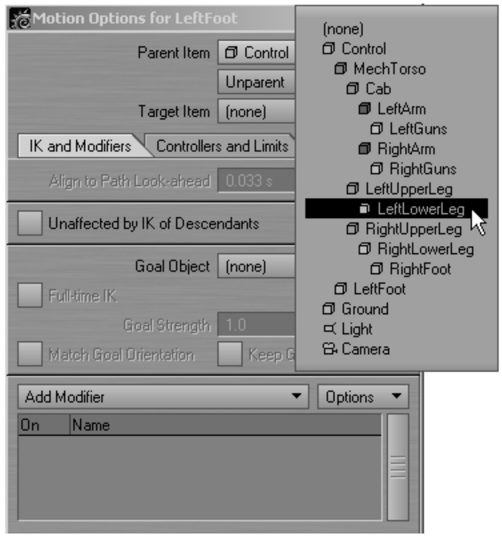
To change hierarchical relationships:

Use the Scene Editor to quickly change hierarchical relationships by dragging items. (See Chapter 12 for more information.)

To use the Motion Options panel to change an item's parent:

1. Select the item to be parented (i.e., future child).
2. Choose **Items** > Motions: **Motions Options** (or M key).

3 Select the parent item from the **Parent Item** pop-up menu. (Yes, in the LightWave world, children can pick their parents. And what a wonderful world it is too...)



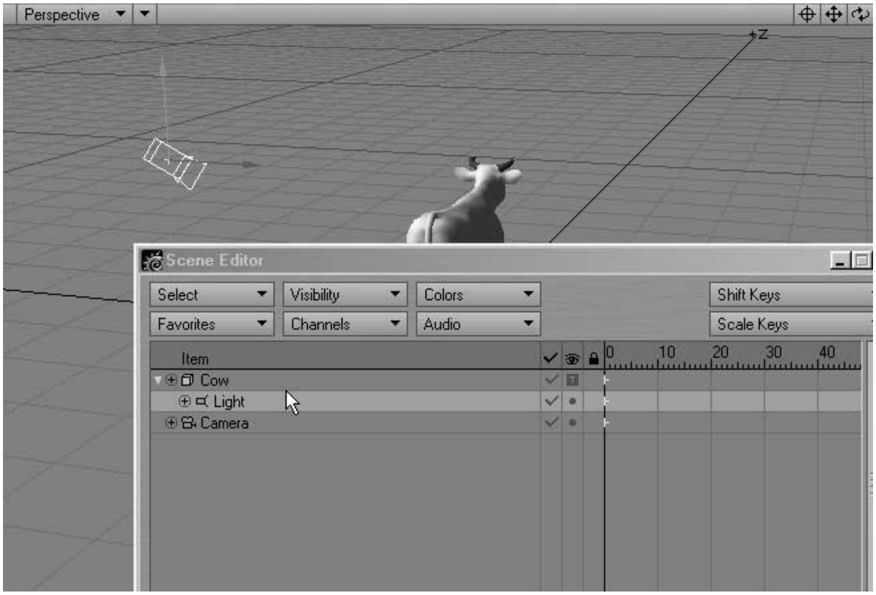
HINT

Use **File** > **Load Items From Scene** to load in an entire chain of parented objects. This way, you can have default scenes set up with all of the objects parented and in place. You can even have motions set up so you could load in a walking woman or a police car with revolving lights atop. If you do load a scene that has motion for certain objects, it may be a good idea to have the items in the loaded scene parented to a null object that you can easily reposition once it's loaded in the new scene.

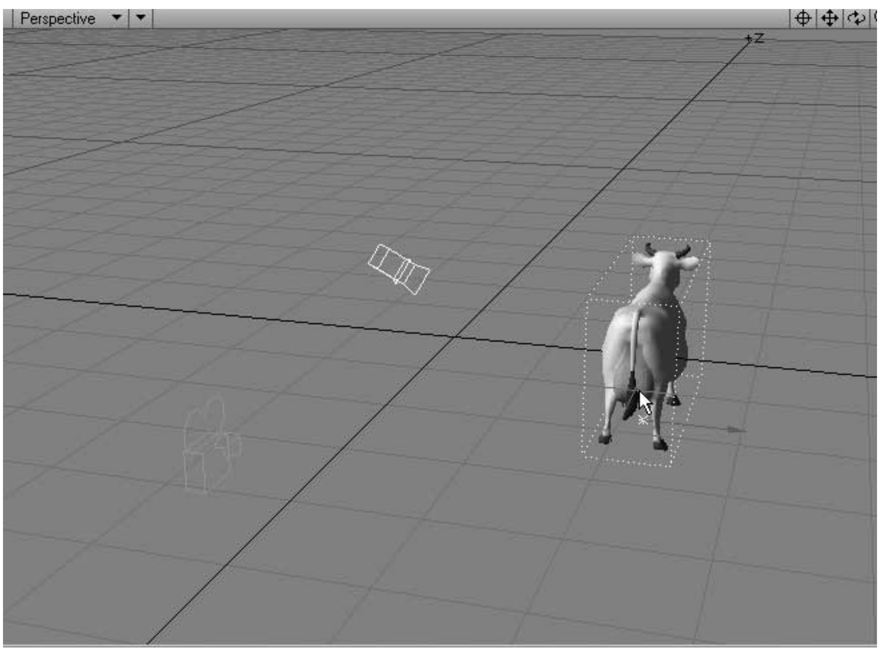
Exercise: parenting:

1 Load the COW.LWO object (ANIMALS subdirectory).

2 Parent the light to cow. (You can just drag it under the cow.)



3 Drag the cow around and notice how the light moves right along with it.

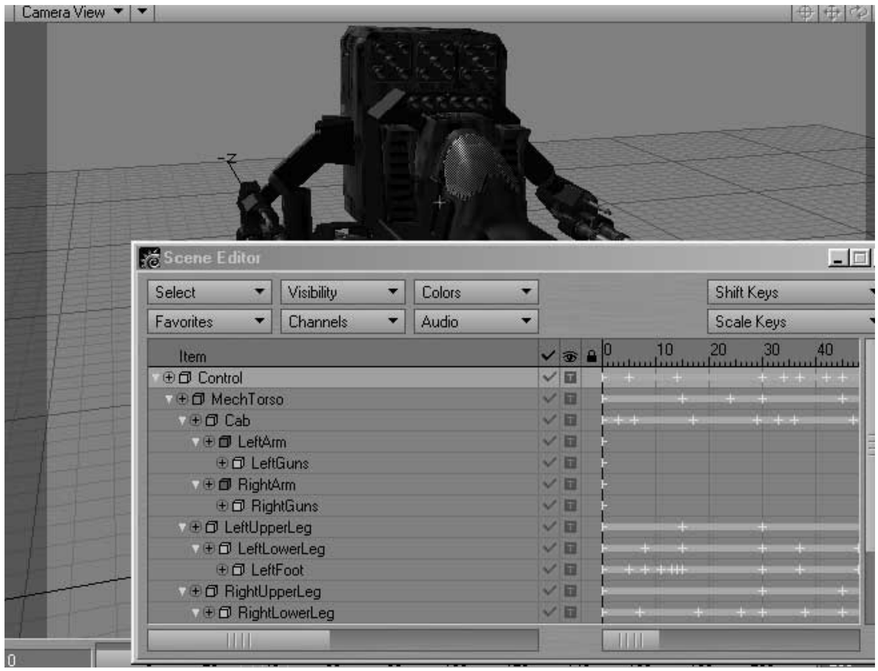


Example: complex parenting:

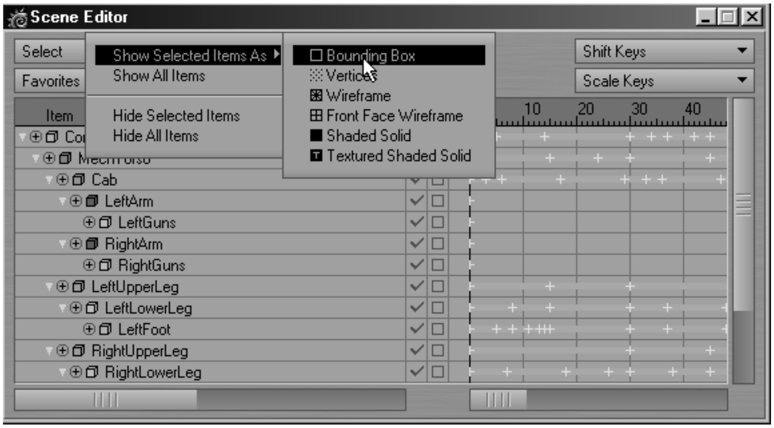
For a good example of parenting, let's look at the Mechwalk scene that comes with LightWave.

1 Load the MECHWALK.LWS scene (ROBOTS subdirectory). This machine is composed of many objects linked together in a hierarchy.

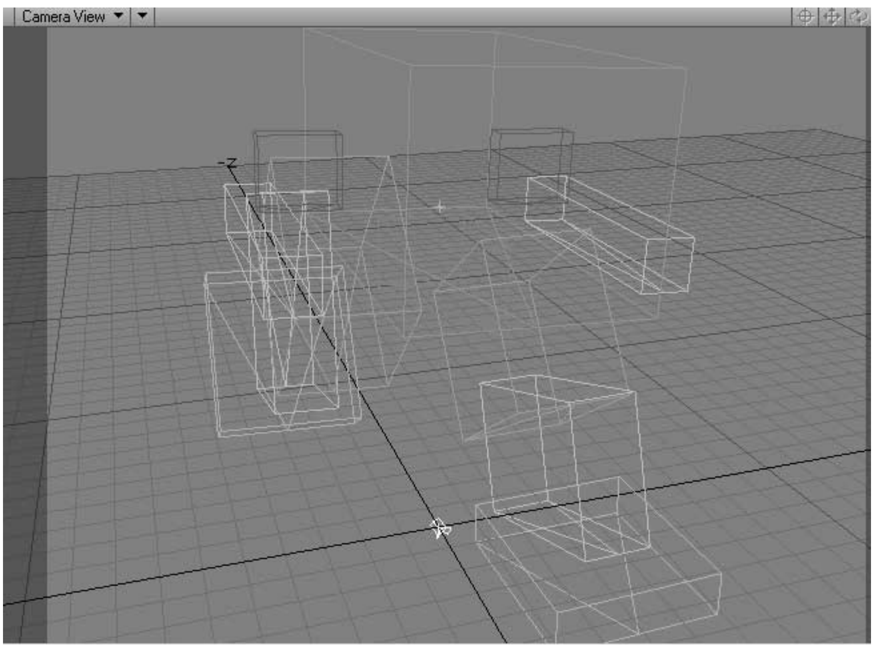
2 Open up the Scene Editor panel (**Scene Editor**) and look at the object hierarchy. The *Control* object—really just a null object—is the master parent. Repositioning it would move all of the children objects.



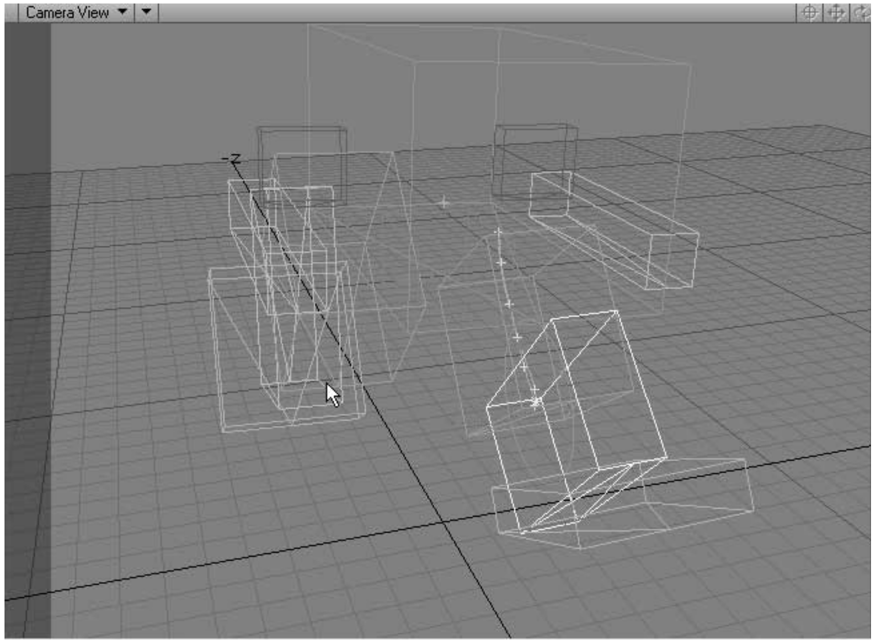
3 Select all objects (**Select** pop-up) and set their visibility to bounding box (**Visibility** pop-up). This lets you see the individual objects more easily in the viewport.



4 Select the **Light** edit mode to avoid displaying all objects as selected. All of the different objects in the viewport should be different colors.



5 Try moving and rotating various objects and see their effect on their children.



NOTE
Generally, the way a scene like this is created is that the individual objects are created in a standing position in Modeler. After all objects are loaded into Layout, the pivot points are then moved to the proper positions. Remember, pivot points should always be set prior to any rotating or moving.

Unparenting

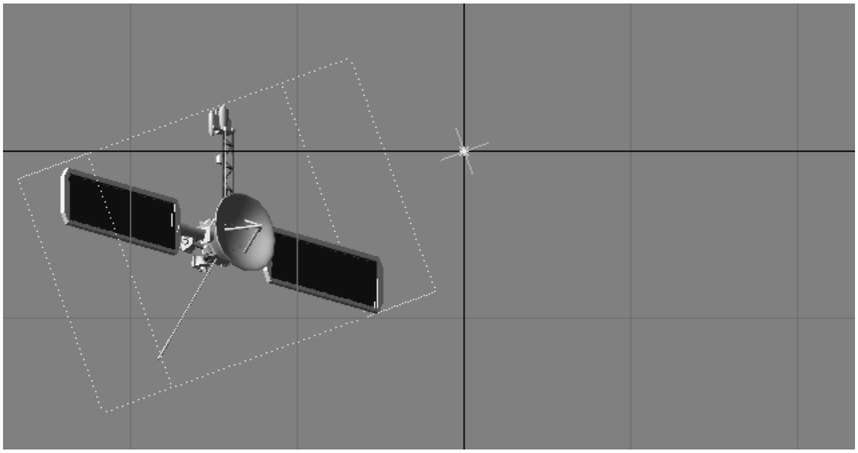
You can unparent an item by:

- clicking the **Unparent** button,
- selecting **(none)** from the **Parent Item** pop-up menu, or
- dragging in the Scene Editor.

Local Axis Rotation

We stated earlier that rotation generally occurs relative to the global axes. However, remember that child items inherit their parent's rotation. This, in effect, lets you use parenting to rotate an item relative to something other than the global axes. To do this, add a null and parent it to the object. Rotate the null to the initial rotation you want the object to have. Now, you can rotate the object itself and it will all happen relative to the null's rotational values.

For example, let's say you wanted a satellite to orbit a planet at a 20-degree angle. Simply add a null and rotate its bank to 20 degrees. Load the satellite and move its pivot point away from the object, a distance equal to the radius of the orbit. Set the satellite's position back to 0, 0, 0 (remember it gets moved because of the pivot point move). Parent the satellite to the null and rotate the satellite's heading.



Orbiting satellite

Parent in Place

If the **Parent in Place** option is active on the General Options tab of the Preferences panel (**Display** > **General Options**), an item will maintain a constant position, size, and orientation when it is parented or unparented. Adjusted position keyframes for the child will be created at the current time, but its position at other times may still change. It is best to be at frame 0 and to perform parenting before animating when this option is active.

TARGETING

Sometimes you will want an item to continually *point to* or *follow* something. A common example is when a stationary camera tracks a car speeding down a street. Obviously, you could keyframe the necessary camera rotations, but wouldn't it be easier to just say, "camera, follow that car." Well, in LightWave, this is called *targeting*. Any scene item can be targeted by another item.

Car targeted by light and camera

Targeting automatically controls an item's heading and pitch rotational values based on the position of a specified object. Banking is unrelated to the targeted object, so you can still control that channel.

A dashed line will appear connecting any item with its target. There is a display option to turn this off or on (**Display** > **Display Options**).

NOTE
Targeting overrides any actual heading and/or pitch channel values.

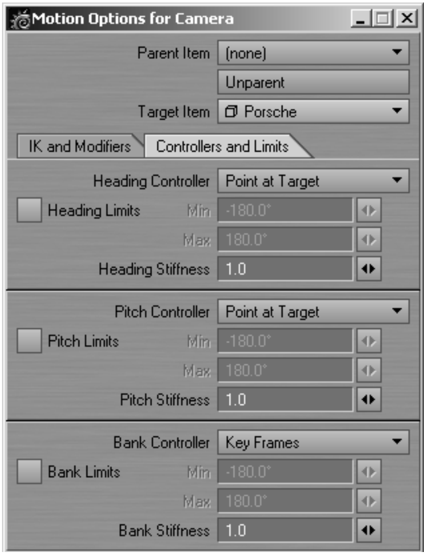
To target an item:

1 Select the item to be targeted toward something.

2 Open the Motion Options panel (**Items** > Motions: **Motion Options**).

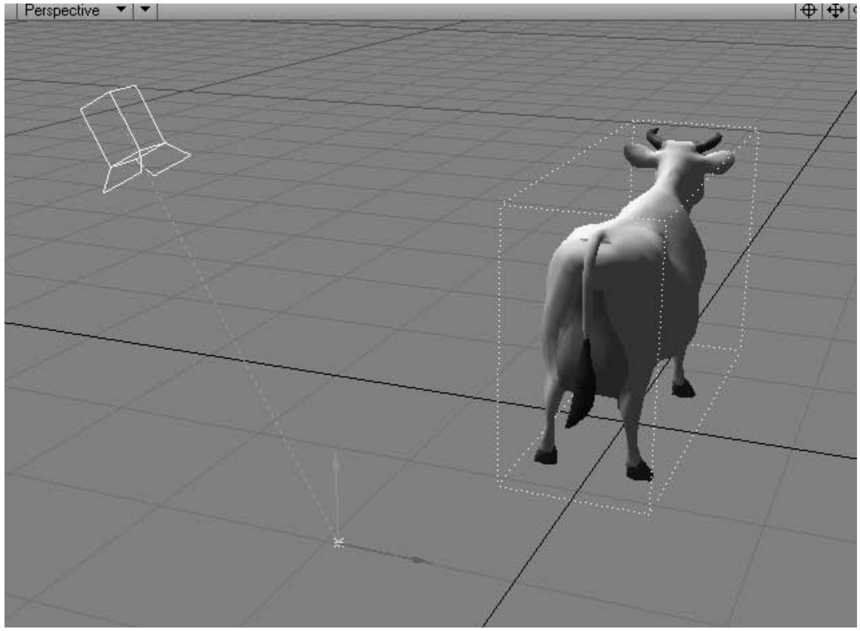
3 Set the **Target Item** pop-up menu to the desired object you want to point to.

4 Both the **Heading Controller** and **Pitch Controller** (Controller and Limits tab) will automatically set themselves to **Point At Target**. If you desire, you can set either back to **Key Frames** for manual control.



Where Does It Point?

A targeted item always points towards an object's pivot point. This may not always be the exact location you want to point to. There are two solutions. You could move the item's pivot point, but this might cause other undesirable ramifications. Another method is to parent a null object to the target and move it to the desired target location. Then, select the null as the target.

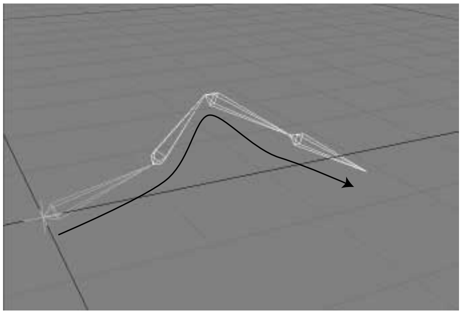


Light is targeted at cow, but cow's pivot has been moved off the object

INVERSE KINEMATICS

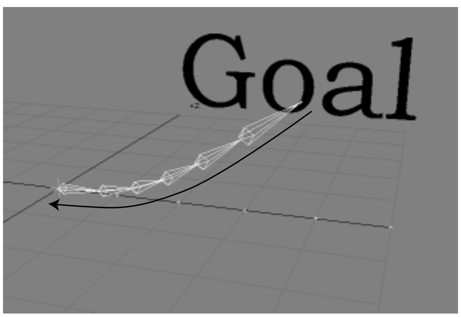
Kinematics describes motion, or the study of motion. It is motion in the most abstract sense, without regard to force, mass, velocity, acceleration, or the positional values that produced the motion. In computer animation, you employ kinematics when you animate links of parented objects such as those used to make up a robot's arm or leg, or when you work with a chain of parented bones inside a one-piece limb. With these types of joints between objects/bones, we generally deal with rotational values for the individual objects/bones in the chain.

Forward kinematics is the standard type of motion generated in LightWave when you rotate/move any parents in a chain. The animator decides what will be the motion of all the joints. The motion of the last item in the chain is determined indirectly by the accumulation of all motion of any parent items up (i.e., *forward*) through the chain. (Like if you move your upper arm, your forearm and hand also move with it.)



Motion accumulates forward through the chain of bones

LightWave also features *inverse kinematics* (IK), sometimes called *goal-directed motion*. IK occurs when you have a hierarchy of two or more objects/bones and a *goal object* selected for an object or bone in the chain. LightWave can calculate (*solve* for) the motion of the chain based on the goal's position. Usually, the last item in the chain (i.e., the last child) is the one that receives the goal, but this is not necessarily so, as you can have multiple goals for a chain. (Like if Billy P's mom grabs his ear, his head, neck, and body must usually follow.)



Motion is computed from goal back up through the chain of bones

You can even use forward kinematics (normal keyframing) and IK simultaneously for an incredible hybrid motion system. You can set kinematics independently for each item and rotational setting!

Full or Part-time?

By default, LightWave does not compute IK continuously. Thus, animating the goal object will not do anything. You position the goal to pose your chain and then keyframe all of the items in the chain. This method uses LightWave's IK as an animation aid. In effect, you are using IK to pose your chain, but forward kinematics when you render your animation.

Alternatively, you can use **Full-time IK** for any item in the chain with a goal. When active, LightWave continuously computes the chain's motion based on the goal. Thus, you can animate the goal object, and the chain will react accordingly. **Full-time IK** also works well with multiple active goals. This means you can have an active knee goal (middle of the chain) as well as an active foot goal (end of the chain). When you move one goal, LightWave takes into account the position of the other goal.

IK and Modifiers tab

Creating key frames for all items in the chain can become tedious if you make a lot of changes and must keep deleting and re-creating key frames for all items in the chain. Thus, you may want to use LightWave's **Full-time IK** option.

Because you can have multiple goals, you can activate **Full-time IK** for some goals and not for others. A full-time IK item further <u>down</u> the chain will always override a non-full-time IK item further <u>up</u> the chain. In practice, however, you'll generally have **Full-time IK** on or off for all items in the chain.

NOTE

If you are not using **Full-time IK**, you do not need to create key frames for the goal object, but it can be useful to help *remember your place*.

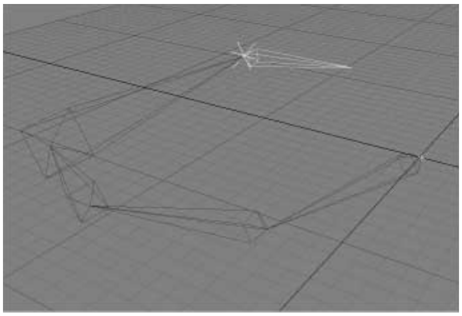
HINT

An easy way to create keys for all items in a chain is to select the parent of the chain and create the key frame using the **Current Item and Descendants** option (Create Motion Key requester).

LightWave also provides a global IK on/off switch: **Items** > IK: **Enable IK**. Use this to toggle IK solving off or on for all IK chains in the scene.

Pointing the End of a Chain

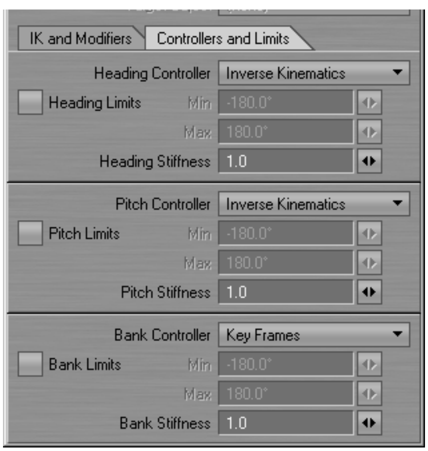
A bone rotates at its base. So, to get the tip of a bone to point towards something, you need to add an extra child bone whose base acts as a pointer (or you could parent a null). For aesthetic reasons, you'll probably want to reduce this pointing bone's **Rest Length** to something really small. If it is a chain of objects, parent a null object to the end of the chain.



Selected bone is "pointing" at goal (null object) using its base

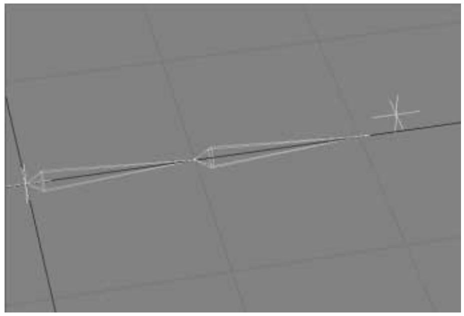
Mix and Match

To have an entire chain solve for IK goals, you should set each item's **Heading Controller** and **Pitch Controller** to **Inverse Kinematics** (Motion Options panel). As such, you can also have any of the controllers (or the item) not use IK by setting the controller to something other than **Inverse Kinematics**. This is how you can mix IK with forward kinematics!



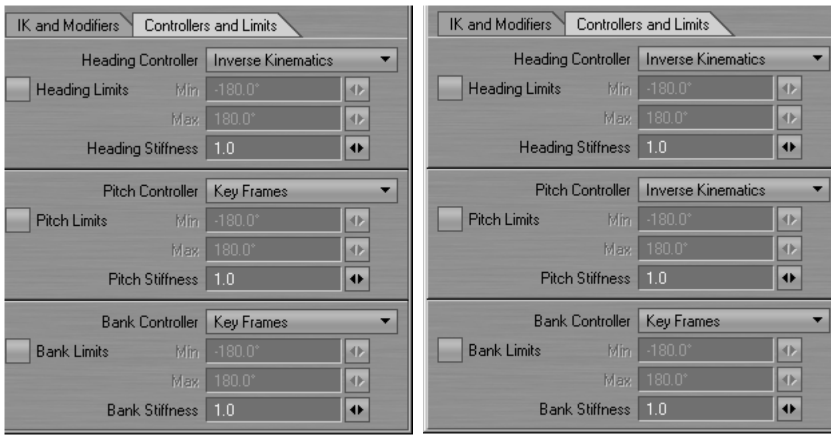
Mixing IK and Key Frames

Here is a basic scene with three bones attached to a null object and a second null object that will be the goal object for our IK chain. (Note: the third bone, which is small and nearly invisible, is at the tip on the right side.)

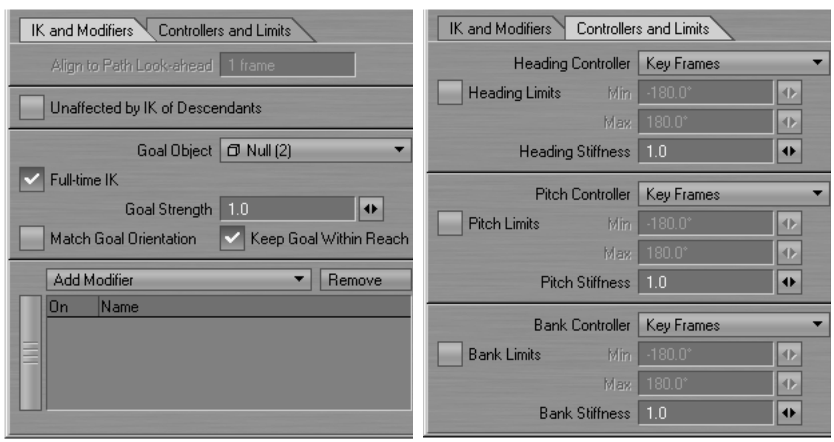


Three-bone chain. Tip bone is very small

Below are the Motion Options for the bones.



Left: Bone (1). *Right*: Bone (2)

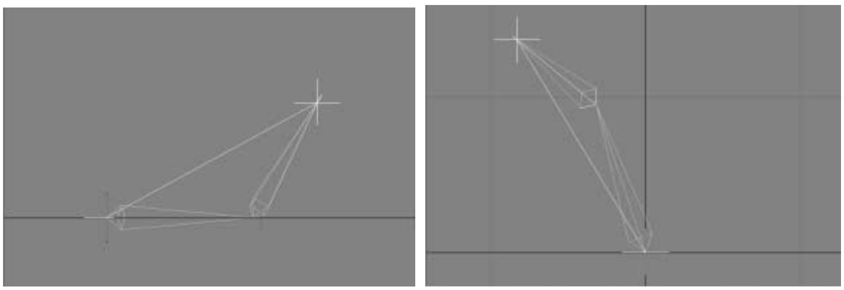


Bone (3)

Bone (1)'s **Heading Controller** is set to **Inverse Kinematics**, but its **Pitch Controller** is set to **KeyFrame**. This means that only its heading will be affected by IK influences. Bone (2)'s **Heading Controller** and **Pitch Controller** are both set to **Inverse Kinematics**. This means that both its heading and pitch will be affected by IK influences.

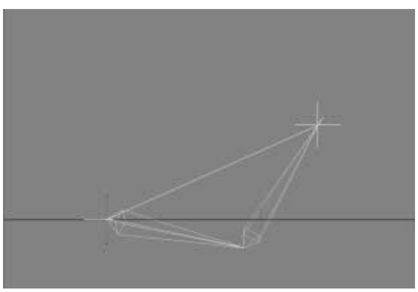
Bone (3) doesn't use any IK controllers because it merely acts as a pointer to the goal object. Note that it does have Null (2) selected as the **Goal Object** and **Full-time IK** is also activated.

Now, as the Goal object is moved around, Bone (1) moves only on its heading, while Bone (2) moves along both heading and pitch. (Note that the **Show IK Chain** option on the Display Options tab of the Preferences panel (**Display** > Options: **Display Options**) and **Keep Goal Within Reach** option on the Motion Options panel's IK and Modifiers tab, are also active. **Enable IK** is also on, but you knew that, right?)



Left: Side view. *Right*: Top view.

Finally, if you rotate Bone (1)'s pitch (as you would normally), you can see it change. This shows how you can now use a hybrid of forward and inverse kinematics.



Forward and inverse kinematics in one chain

Setting Your Goals

Setting a goal object is similar to targeting a camera to an object: the item in the chain with the goal will try to *point* at the goal object. This is called *reaching* for the goal. Items up the chain will move and rotate to keep the chain connected.

If physically possible, the goal object will be *stuck* to the goaled item. Moving it beyond its *reach* will detach the goal object. You can use the **Keep Goal Within Reach** option to affix the goal object to the item, if desired. This may make monitoring multiple goals easier, but it is up to you.

Keep Goal Within Reach is active

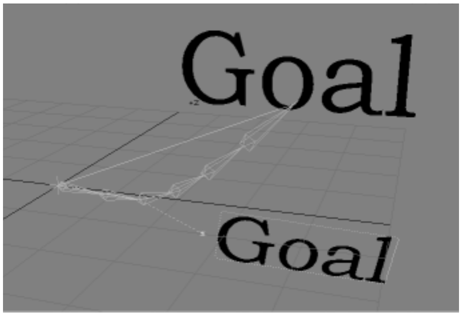
Moreover, let's say you build an arm and instead of parenting the hand to the forearm, you just make the hand a goal for the arm. With **Keep Goal Within Reach**, you can insure that the hand stays locked to the forearm. (According to Paul, doing this manually would be a pain and you'd end up adding a lot of extra keyframes.)

NOTE
If you move the IK object with goals that have **Keep Goal Within Reach** active and you are using **Full-time IK**, those goals will be dragged along for the ride. Wheeeeee!

Although any object can be a goal, null objects are used most often because they do not render. However, you can also use a visible object effectively as a goal. Let's say that you have a searchlight object that you always want to point at a blimp flying overhead. By modeling the searchlight with its origin at one end (or simply moving the pivot point to the end) and parenting a null object at the other end, you can select the blimp as the null object's goal. The searchlight will now follow the movement of the blimp.

Dueling Goals

When you create an IK chain with multiple goals that may compete, you must set strengths for these goals. The strength will determine the relative *magnetism* that goal has for the shared objects in the chain. To increase the strength of a particular goal, raise the **Goal Strength** value.

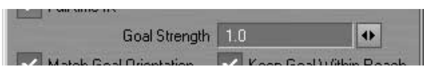


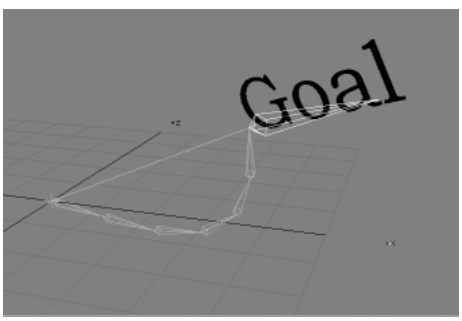
Multiple goals on one chain

Goal Strength comes into play when the distance between the competing goals becomes too great to solve the IK calculation, or when the distance exceeds the length of the object chain. As the goals move past this distance, the objects will move away from their original positions. The distance they move from both goals is equal as long as the strength values are equal. If the strengths are not equal, the difference in distance is equal to the ratio of the two strength values. If one goal's strength is set to 1 and the other is 10, the object's distance from the weaker goal is ten times that of the stronger goal.

Goal Orientation

Items in an IK chain that have a goal assigned can match the orientation of the goal. You do this by activating the **Match Goal Orientation** option.





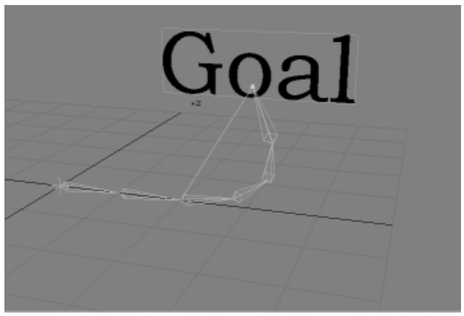
Note: The rest length for the bone with the goal has been increased for illustration

You should note that this option also matches the goal object's *scaling*, overriding the (goaled) item's scale setting. If this adversely affects your animation, one solution is to copy the (goaled) item's scale channels to the goal object. Another solution may be to add a null object to the IK chain and use that as the goaled item instead.

HINT
Use Match Goal Orientation to prevent a character's feet from going through the ground plane.

Breaking the Chain

When you activate the **Unaffected by IK of Descendants** option for an item in an IK chain, the selected item is *locked* from the effects of an IK goal. This, in effect, makes that item the new *base* of the IK chain, and only items farther down the chain are affected by any IK operations.

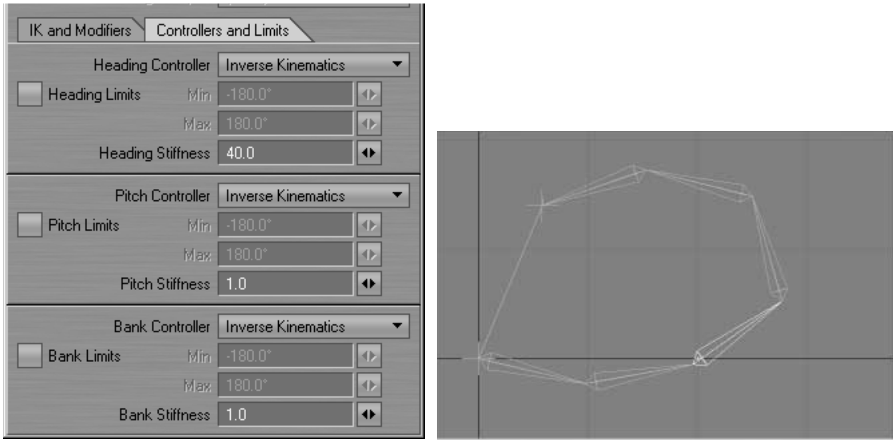


Chain broken with Unaffected by IK of Descendants

A good example of using **Unaffected by IK of Descendants** 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 tail bones are affected by a goal object (i.e., the "tail wagging the dog"). You activate **Unaffected by IK of Descendants** for the bone at the base of the tail to accomplish this.

Stiff Stuff

You may also add a **Stiffness** value to items in the IK chain. Items with a higher stiffness value will tend to be more resistant to bending caused by IK.



The third bone has a high stiffness setting (for heading in this case). Note how it barely flexes.

HINT
When you use IK, you will often want to set rotation limits, as discussed later in the chapter.

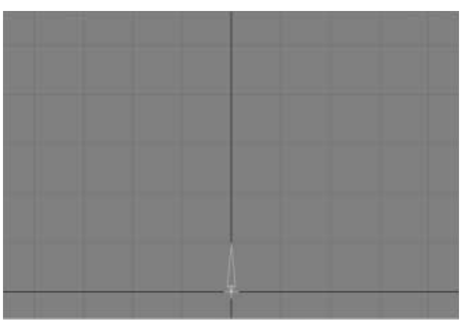
Exercise: setting up a bone IK hierarchy

1 Bones must be added to an object, so add a null object to a clear scene and call it BASENULL. Add another null object and call it GOALNULL.

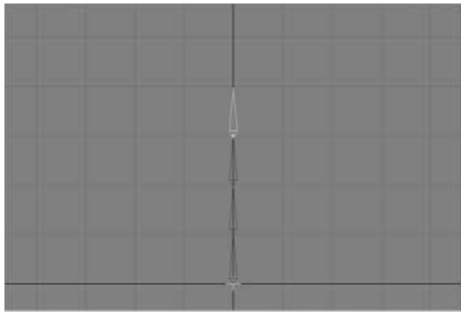
HINT
Naming nulls is a good habit to get into. Although not necessary, it can avoid confusion when you use them later.

2 Change to the Top view and move it so that the nulls are at the bottom of the Layout window.

3 Choose **Items** > **Add** > **Bones** > **Add Bone** to add a bone to the null object.

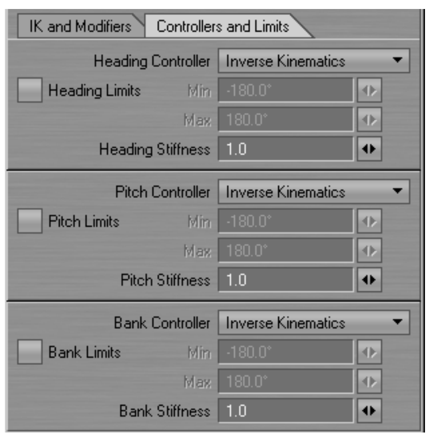


4 Add three child bones using the shortcut (=).

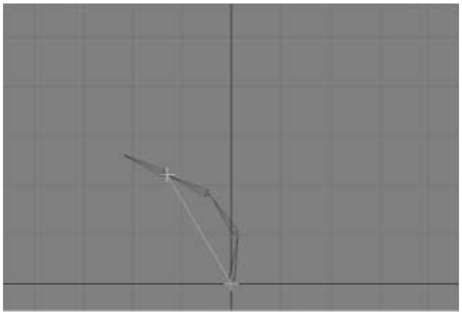


5 With the last bone (4) still selected, open the Motion Options panel (M). Select the GOALNULL as the **Goal Object** and set all of the Controllers to **Inverse Kinematics**.

6 Using the **Current Item** in the main interface, select each of the other bones and set all of their Controllers to **Inverse Kinematics**, too.

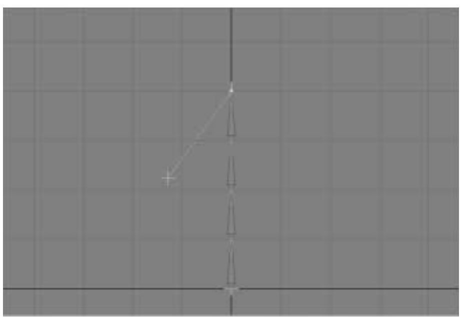


7 Now, select the GOALNULL and drag it around the viewport. You should notice that the chain follows the object; however, the last bone does not seem to point to it and just hangs there.

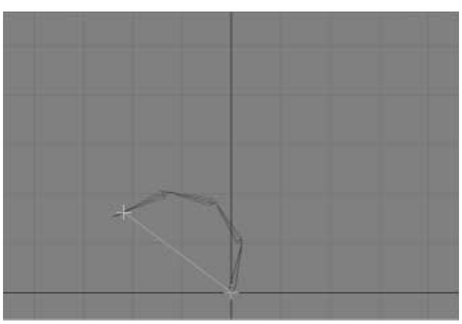


8 To create the illusion that the tip of the bone (4) is pointing to the goal, you need to add another child bone, whose <u>base</u> will act as a pointer. Select the last bone and add a child (=). Size the new bone's **Rest Length** down so that it is more or less a point.

9 Now, set Bone (4)'s **Goal Object** to **(none)** and set the new bone's (5) **Goal Object** to GOALNULL.



10 Now move GOALNULL in the viewport. That's more like it. You should now see the bones bend around like a snake with the end pointing to the goal. Notice how the GoalNull tends to *stick* to the base of Bone (5).

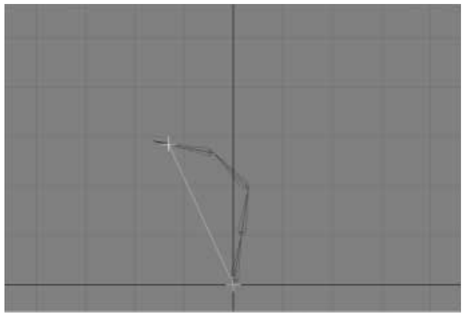


Exercise: bone stiffness

1 Continue on from the previous exercise.

2 Select Bone (2) and set the **Heading Stiffness**, **Pitch Stiffness**, and **Bank Stiffness** (**Motion Options** panel) all to 100. This makes the bone very inflexible and is probably a much higher value than you'd use normally.

3 Move the GOALNULL around. Notice that the stiffened bone's joint (base) is much more resistant to bending compared to the others.



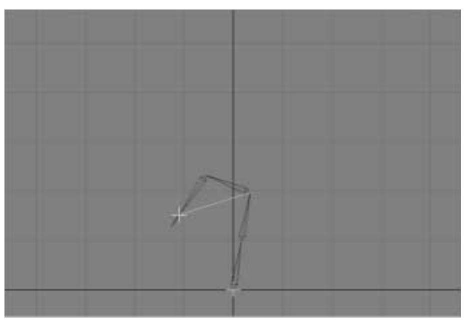
Exercise: limiting chain movement

1 Continue from the prior exercise.

2 Reset Bone (2)'s **Stiffness** values to 1 (Motion Options panel).

3 Activate **Unaffected by IK of Descendants** for Bone (2) (Motion Options panel).

4 Now move the GOALNULL around. Bone (1) through Bone (2) should be stationary and the snake should bend beginning with Bone (3).

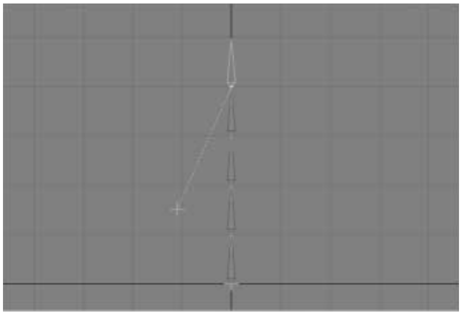


Exercise: To match goal object orientation:

1 Continue on from the previous exercise.

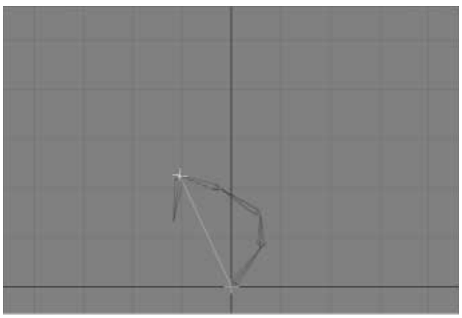
2 Deactivate **Unaffected by IK of Descendants** for Bone (2) (Motion Options panel).

3 Select Bone (5), click **Objects** > Bones: **Rest Length** and then click **Items** > Tools: **Reset** to restore its size.

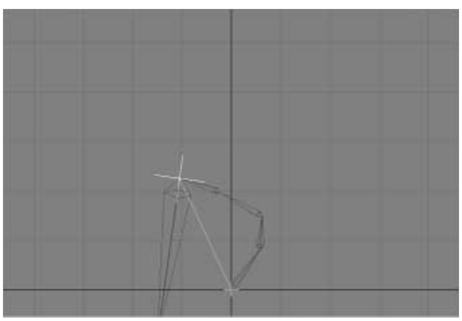


4 Activate Bone (5)'s **Match Goal Orientation** (Motion Options panel) option.

5 Move and rotate the GOALNULL around. Notice how the orientation of the *goaled* bone matches the orientation of the goal object.



6 Size the GOALNULL object up and see how the bone sizes up too.



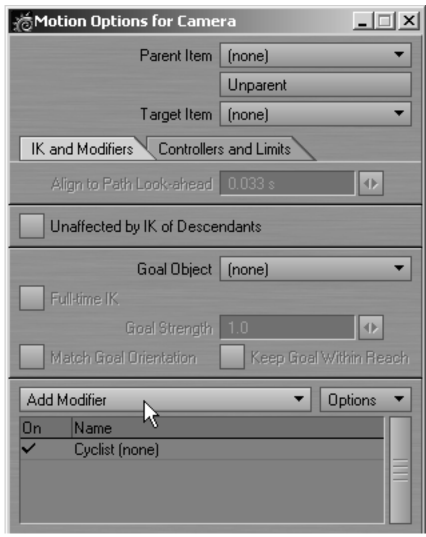
7 Set the **Keep Goal Within Reach** option for Bone (5).

8 Move GOALNULL again. Notice how it is now permanently attached to the goaled object and can no longer be moved off it.

ITEM MOTION MODIFIERS

Item motion modifiers are added on the IK and Modifiers tab of the Motion Options panel. These generally control the motion at the item level, as opposed to strictly the channel level, which are handled by Graph Editor modifiers (see Chapter 8). However, they often have individual motion channel control. Item motion modifiers can also get motion data after IK is applied, which is not possible with channel motion modifiers.

To display this panel for the current scene item, choose **Items** > Motions: **Motions Options** or just press M.



Motions Options panel

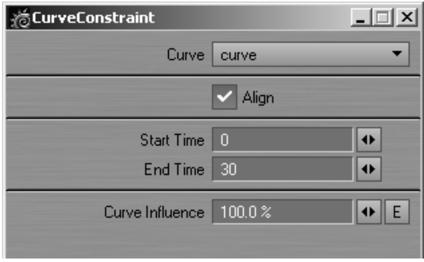
To use an item motion modifier, select the modifier from the **Add Modifier** pop-up menu on IK and Modifier tab. Once added, click on its name in the list to access its settings, if any.

NOTE
Modifiers that do not have an explicit additive option are generally additive in nature.

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.)

Enabling **Align** will rotate the object so it faces along the curve in the traditional “Z axis” manner. The **Start Time** and **End Time** values determine how long the item takes to traverse the curve. The **Curve Influence** percentage blends the curve motion in or out.



CurveConstraint options

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, and eliminate 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), like with a roller coaster.

Cyclist

Cyclist will associate the action of one item with a repeatable action (an animation cycle) of another item.

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.

The **Cycle Frames** define the action that you want repeated (i.e., the animation cycle).

The **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.

Essentially, **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-type 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.

Similar to motion graphs, **End Behavior** determines what happens after the first cycle is completed.

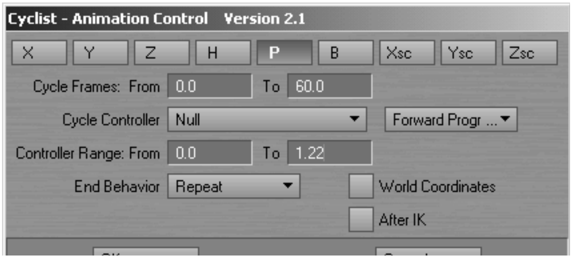
Activate the **World Coordinates** button to use the **Cycle Controller**'s actual (world) coordinates, rather than its relative coordinates. These could be different if, for example, the object is parented to another object.

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.

Exercise: Cyclist

1. Add a Null object.
2. Load the WHEEL.LWO from the TUTORIAL folder into Layout.
3. Parent the Wheel to the Null. Create a keyframe for the Wheel at frame 0.
4. Go to frame 60 and rotate the wheel's pitch 360 degrees and create a keyframe.
5. Open the Graph Editor and set the **Pre Behavior** to **Linear** for the wheel’s pitch channel. This sets up the Wheel's *cycle* (frames 1 to 60).
6. Set the Last Frame to 180 and go to frame 0.
7. Move the Null to 0m, 19.5cm, 1.4325m, and create a keyframe.
8. Go to frame 103, move the Null to 0m, 19.5cm, -21.1209cm, and create a keyframe.
9. Go to frame 126, move the Null to 0m, 19.5cm, 35.175cm, and create a keyframe.
10. Copy this keyframe to 140. To do this, while still at frame 126, press the ENTER key to bring up the Create Motion Key dialog. Enter 140 in the **Create Key At** input field and click OK. This copies the keyframe info from 126 to 140.
11. Go to frame 180, move the Null to 0m, 19.5cm, -1.44m, and create a keyframe. Adjust your Perspective view so you can see the null's entire motion path.

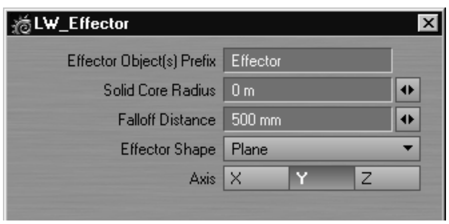
12 Select the Wheel and open its Motion Options panel. Add the Cyclist modifier and set the settings as shown. (Note: Since the wheel is about .194m in diameter, the circumference is about 1.22m (2 * pi * .194).)



13 Close the Cyclist panel and drag the time slider. You will see the wheel turn backward and forward as it moves.

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.



The **Effector Object(s) Prefix**, with the default of Effector, is a name prefix, as such, 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. 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.

You also have a choice for the **Effector Shape**. It can be a **Point** or **Plane**. If it is a **Plane**, you need to specify the **Axis**.

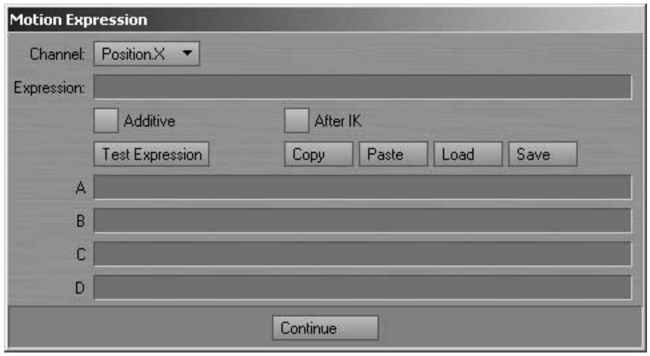
The impact of the effector object is set and animated by keyframing its XYZ **Size** channels. Positive values repel and negative values attract.

NOTE

See also the Effector custom object plug-in discussed in Chapter 9.

Expression

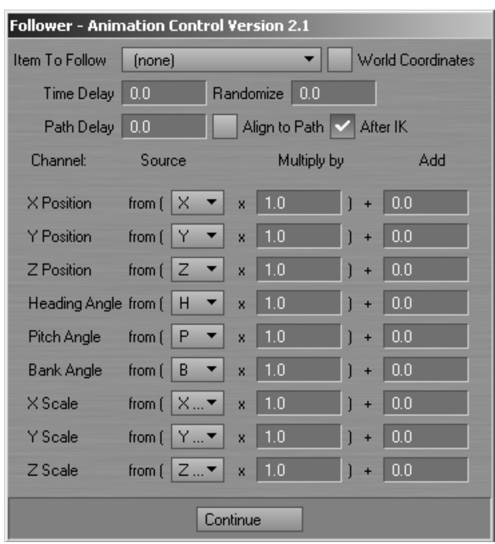
This is a motion-modifier version of expressions, which can get the results of other motion modifiers, as well as IK. This is not possible with a channel modifier. (See Chapter 8 for more information.)



You can select multiple channels on the **Channel** 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 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.



Use the **Item To Follow** pop-up menu to define the item whose motion you want to use. This is the *leader*.

Activate the **World Coordinates** 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.

The amount of seconds entered into the **Time Delay** field is added to the current time. This number may be negative.

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.

The **Path Delay** 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.

The **Align to Path** option will align the follower object to its new path.

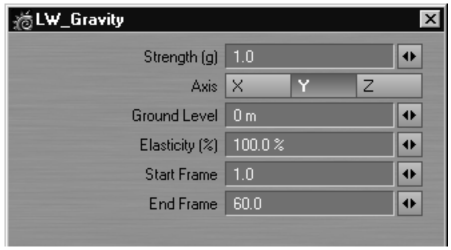
The **After IK** option allows you to apply this modifier taking (or not taking) into account changes from IK.

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.

The value can be scaled by inputting a factor other than 1 in the corresponding **Multiply By** field. The 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 is the acceleration due to gravity in units of 'g', the Earth's standard gravity. The object's position will be altered along the specified **Axis**. The **Ground Level** value determines at what height the object bounces. 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.

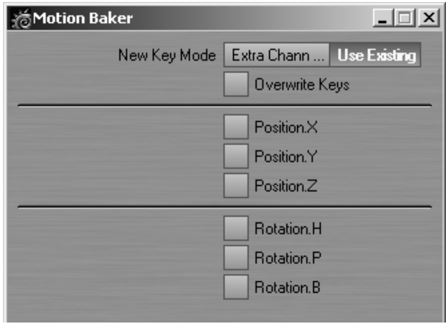
The **Start Frame** and **End Frame** parameters specify when the effect is applied.

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.

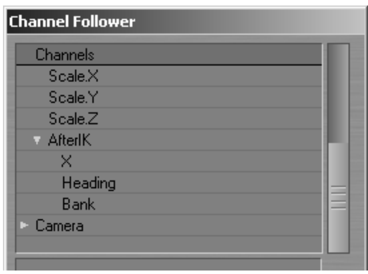
Motion Baker

Motion Baker (formerly IKCapture) 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. The computation is executed by dragging your frame slider or playing the scene.



Motion Baker panel

Selecting the **Use Existing** mode will create normal keys for the selected position or rotation channels. The **Extra Channels** mode places the keys in a special *AfterIK* channel group—you might use these with expressions, for example.



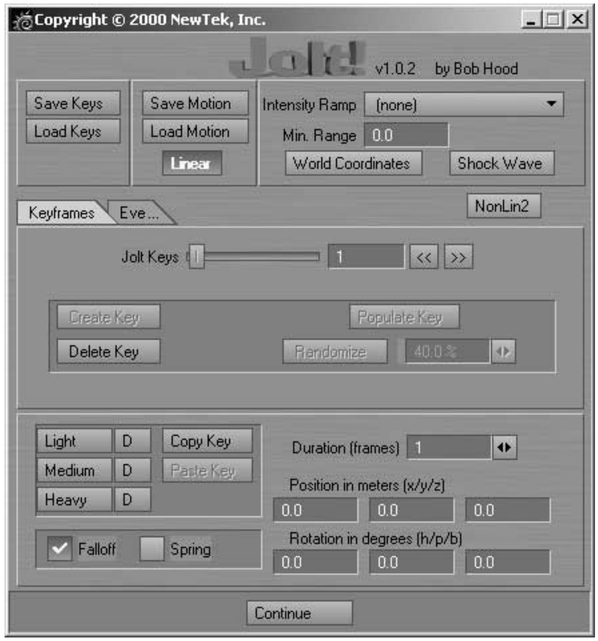
AfterIK group shown in ChannelFollower channels window

If **Overwrite Keys** is active, 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.

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.

Jolt!

Jolt! is an item motion modifier that simulates the chaotic and uncontrolled movements that occur when physical impacts have enough weight and velocity to cause jarring vibrations. For example, the collision of a celestial body striking another, the impact of a robot's foot with the earth, the collision of a laser beam with a passing spaceship, and so on. All of these events are good candidates for Jolt!



Jolt! panel

Jolt! also lets you specify vibration events using keyframes that indicate when a vibration will begin—and be at its most intense. You can also specify the duration of the vibration and its initial intensity. Intensities can be set based on the object's position (X, Y, and Z axis) and its rotational values (heading, pitch, and bank).

NOTE
Jolt! does not provide the basic motion of an item, but rather modifies an existing motion. Jolt! does not modify existing motion except to radically deviate from it temporarily at specified intervals.

Global Options

The **Save Keys** and **Load Keys** buttons let you save and restore Jolt! settings. The **Save Motion** and **Load Motion** buttons save and load, respectively, standard LightWave motion files based on the keyframe data. The **Save Motion** button is ghosted if no keyframes are set, and both buttons are ghosted when the Events tab is selected. Motion files saved with the **Linear** button active have their keyframes set to the (spline control) linear mode.

Activating **Intensity Ramp** tells Jolt! to track the movement of a LightWave item (i.e., Camera, light, object, or bone) and based upon its distance from the Jolt! object, increase or decrease the effect of the vibration proportionally. **Minimum Range** is the radius in meters of the effect range. LightWave items outside of this range will not create a visible effect. If you forget to set this value when you leave the Jolt! interface panel, you will be reminded and Jolt! will disable **Intensity Ramp** (with a minimum range of 0.0, it is ineffective anyway).

If you activate **Shock Wave**, vibration keyframes are adjusted internally to offset for the distance of the ramp object. In effect, the further away the ramp item is, the longer it will take for the vibration event to actually trigger, and the later it will trigger from its indicated start time. This option is useful if you use Jolt! on items in your scene other than the Camera. Items using Jolt! that are closer to the ramp item begin their vibration events sooner than items that are further away. With items positioned correctly, this option can produce a visible shock wave effect from the ramp item's location.

If you select an item that is part of a parental hierarchy, Jolt! alerts you and—if it isn't selected already—suggests that you turn on the **World Coordinates** option. This option ensures that you get proper movement information from the child item.

The **NonLin2** button activates an alternate calculation for Jolt!'s motion calculations. It creates motions that are less harsh.

Keyframes Tab

The Keyframes section houses the Jolt! key controls. When you want the jolting to occur on specific keyframes, you will utilize this tab. (If you want the effect to occur based on the position, rotation, and/or scale of an item, this is set on the Events tab. You can set both.) The **Jolt Keys** slider selects the current frame. The range of the slider will exactly coincide with the number of render frames that have been specified on LightWave's Render Options panel—not necessarily the same value that is used in the Preview settings. The **<<** and **>>** buttons will jump to the previous or next keyframe, if any exist.

Clicking the **Create Key** button makes a keyframe at the current frame. Use **Delete Key** to remove an existing keyframe. Note that the current frame must be a keyframe to do this.

Clicking the **Populate Key** will take the settings for the current keyframe and copy them to every existing keyframe. In other words, it *populates* all keyframes with the current settings and saves you from the tedious task of copying and pasting settings, frame by frame.

Randomizing Keys

Randomizing (or *jittering*, if you prefer) provides a means of *breaking up* potentially monotonous key settings. Although Jolt! will internally randomize settings to some degree as it applies them to the item, unless you are using intensity ramping, the actual key values themselves will not be altered. By using the **Randomize** button, you can generate variances such that the motion of subsequent keys does not look so similar.

The randomizing process requires two or more keys in order to function. The settings of the first key are never altered (nor is there any reason to alter them because no other key will look exactly like them after the effect is applied). The degree of variance can be altered by using the Threshold control input field to the immediate right of the **Randomize** button. This control lets you specify, as a percentage, the maximum amount that each altered key will deviate from its current value.

Jolting Effect

The bottom half of this tab sets the actual jolting effect. Clicking **Light**, **Medium**, or **Heavy** will update a keyframe's **Position** and **Rotation** values to reflect preset values for a light, medium, or heavy vibration. This feature can be used for starting points or final settings.

WARNING
Please be aware that when you press any of these buttons, any existing keyframe settings will be destroyed.

Clicking the **Copy Key** button will copy all of the Jolting effect settings to an internal memory buffer. Clicking the **Paste Key** button will paste the settings into the fields currently visible. Note that this can affect either the Keyframe or Events tab interchangeably.

Using Preset Values

You can alter preset values. If you press the **D** button to the immediate right of any of the presets, then you store the current settings for the key as the default values for that preset. These new default values will persist between sessions with LightWave 3D. (Jolt! stores its preset defaults in a file called JOLT.PRE. You can restore the built-in Jolt! preset values at anytime by deleting this file; however, it can be stored in various places depending on the system configuration. You will need to search your hard drive to find it.)

Applying Turbulence

Jolt! can apply turbulence to your item's motion path in several ways. When you select **Falloff**, the turbulence applied will gradually decrease throughout the duration of the event. In other words, at the first frame of the event, the position and rotation values you have entered for the event will be at their strongest, while at the last frame of the event they will be at their weakest. If you do not select **Falloff**, then Jolt! applies the event values at their full strength at each frame throughout the event duration.

If you select **Spring**, Jolt! applies turbulence uniformly, and makes it appear as though your item is supported and buffered by springs. Without **Spring**, Jolt! applies turbulence more chaotically or randomly, which produces more of a true vibration effect.

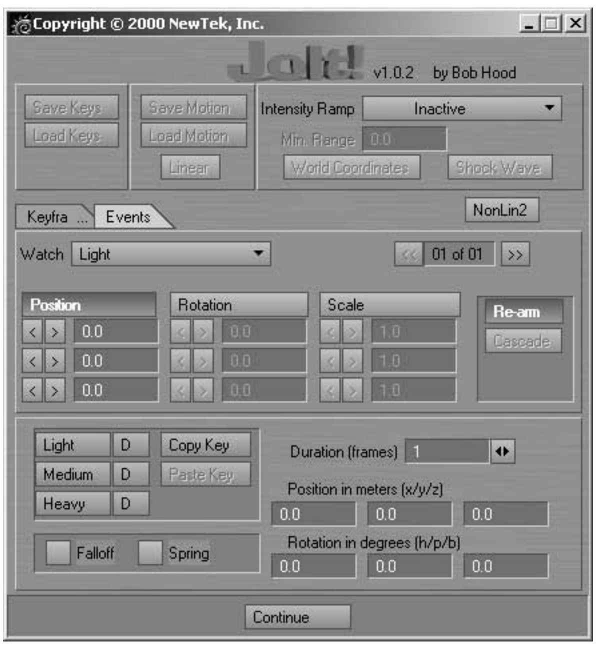
Key Settings

A **Duration** value must be specified, in terms of the number of frames, within which Jolt! must complete the effect. Because the you can specify duration and location of vibration events, keyframe data may overlap. Jolt! handles this situation by warning you about the overlap condition. The overlapping key will start before the preceding key ends. If a keyframe is deleted, Jolt! will recalculate all keys to ensure that any overlapped keys are corrected.

The **Position** controls let you to define the maximum deviation on each of the three axes for the item at the current frame. These values are all specified in meters. The higher the number, the more dramatic the initial movement in that direction. A value of zero (0.0) in any position will prevent the item from deviating in that direction. The **Rotation** controls work similarly, but set deviation values for heading, pitch, and bank (in degrees).

Events Tab

On the Events tab, you can cause jolting based on the position, rotation, or scale of items. (If you want the effect to occur on specific keyframes, this is set on the Keyframes tab. You can set both.) The **Watch** pop-up menu gives you a list of all the items currently in your LightWave scene. You may select any available item from this list as the *watched* item, that is, the item that triggers the event.



With the watched item selected, click on the **Position**, **Rotation**, and/or **Scale** button to activate the watched attributes. Click either the

< (less than) or > (greater than) button next to the input field you wish to set. The **Position** and **Scale** fields correspond to X, Y, and Z, from top to bottom and the **Rotation** fields correspond to H, P, and B.

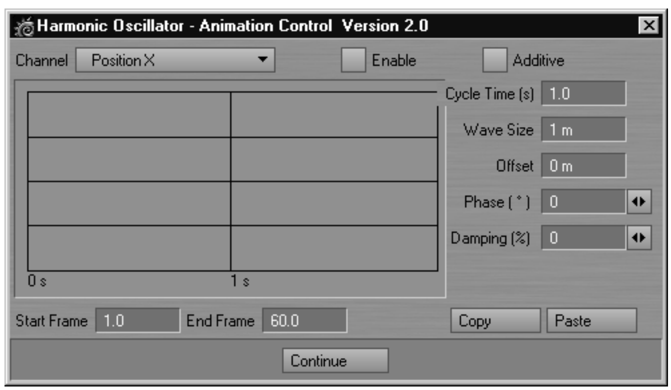
The **<<** and **>>** buttons let you watch more than one item. To add another watched item, simply click the >> button. The informational display to the left will tell you what item is currently selected and how many there are in total (e.g., 01 of 03). The **<<** and **>>** buttons are also used to navigate through existing watched items. If you add a watched item by mistake, set **Watch** to **(none)**. You also cannot add another watched item if the last existing item is set to **(none)**.

Activate the **Re-arm** button if the item repeats its motion and you want it to trigger the event again. If **Cascade** is active, Jolt! will ignore the event (i.e., not evaluate it) until the event immediately preceding it has occurred at least once.

The settings on the lower half of the tab work as described for the Keyframes tab (see *Jolting Effect*, above).

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.



Use the **Channel** pop-up menu to select an animation channel to be affected. 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 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 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.

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 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.

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.

The **Start Frame** and **End Frame** parameters specify when the oscillator is applied.

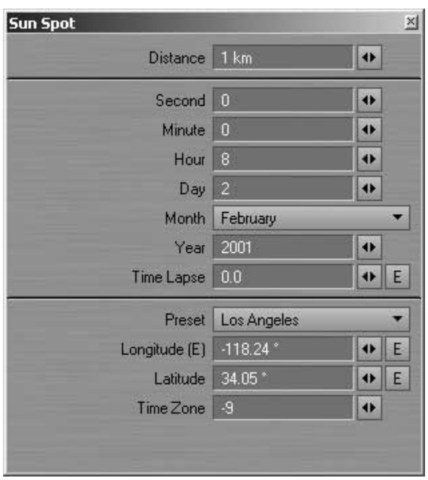
Copy and **Paste** use an internal clipboard to move oscillator settings between channels or Layout items.

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 is 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.

The *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**.

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 minimize any impact on rendering time from moving lights.

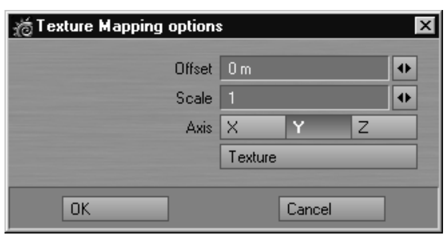
You can accelerate the movement by increasing **Time Lapse**. For example, 86400 (60 seconds * 60 minutes * 24 hours) will cause one day's rotation to occur in one second.

Set the **Longitude** and **Latitude** for the part of the world your sun is (theoretically) shining on. **Time Zone** is +/- Greenwich Mean Time (GMT). There are presets provided.

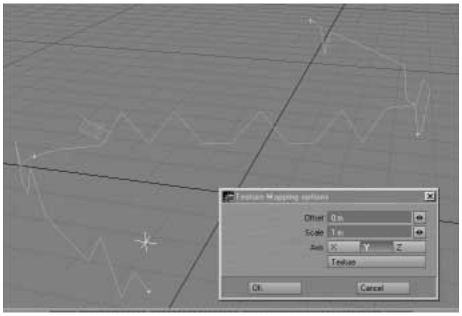
During *night* hours, the sun will stay at its lowest point at the end of its cycle and then *pop* to the starting point at the beginning of the next cycle. In other words, it will not revolve in a 360-degree circle.

TextureMotion

Texture Motion 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!



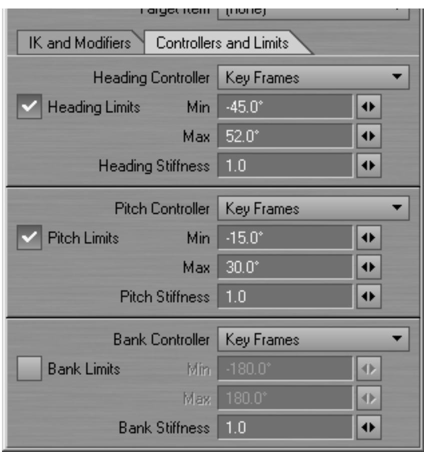
The **Axis** determines the perpendicular mapping axis, just like with an image map. The **Offset** lets you move the motion in the positive or negative direction along the selected **Axis**. **Scale** acts as a multiplier for the motion. A value of 1 has no effect.



Here a simple Checkerboard texture has been applied

ROTATIONAL LIMITS

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 football injury). To use the limiting feature, activate the desired **Limit** option(s) and set the related **Min** and **Max** values on the Motion Options panel (M) for the particular item. Setting rotation limits can be particularly important when using inverse kinematics. (Note that the maximum value must be greater than the minimum.)



You can interactively set the **Min** and **Max** values by rotating the item to the desired minimum direction and choosing **Items** > IK: **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 **Items** > IK: **Limits** > **Record**

Maximum Joint Angles will automatically input the values into the maximum limits for the active rotational parameters. You can deactivate **H**eading, **P**itch, and/or **B**ank to avoid setting limits for the deactivated setting.

If any of the rotational limits are active on the Motion Options panel, you can access angle limit dialogs from the **Items** > IK: **Limits** menu.

Heading Angle Limit dialog

ALIGN TO PATH

Another way to automatically control the rotation of an item is to align itself to the motion path that it is traveling. This feature aligns the local Z axis of the item (through its pivot point). A classic example of this is when you have a car speeding down a curvy road. Sure, you could keyframe the rotational changes, but its much easier to have LightWave align the car to its path.

NOTE
This is one reason most vehicles (and other objects) are modeled along the Z axis, facing the positive side.

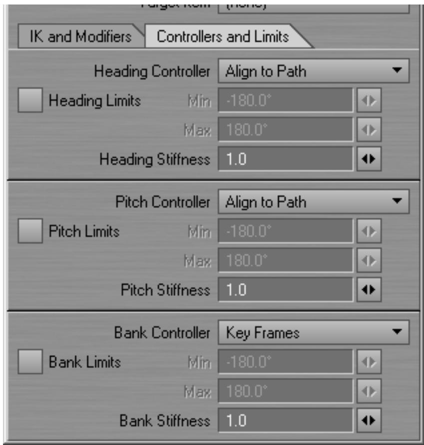
This feature automatically controls an item's heading and pitch rotational values based on its motion path. Banking is still independent, so you can still control that channel manually.

NOTE
Align-to-path overrides any actual heading and/or pitch channel values.

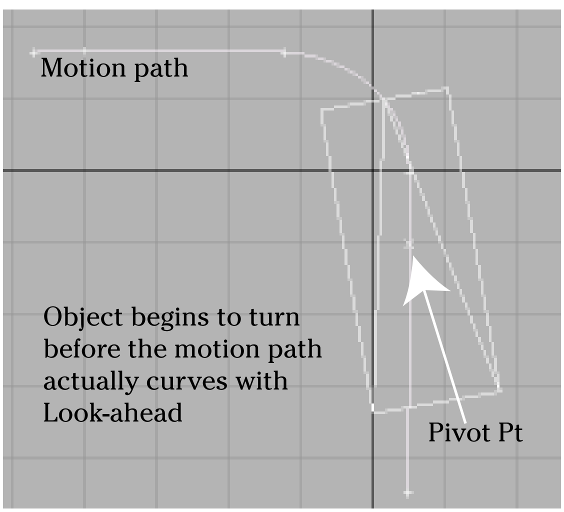
To apply align-to-path to an item:

1 Select the item to be aligned.
2 Open the Motion Options panel (**Items** > Motions: **Motion Options**).

3 Generally, you'll want to set both the **Heading Controller** and **Pitch Controller** (Controller and Limits tab) to **Align to Path**. If you desire, you can set either back to **Key Frames** for manual control.



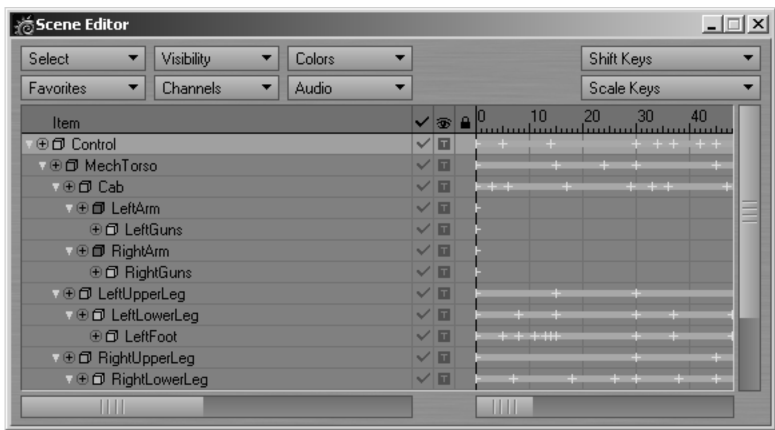
The **Align to Path Look-ahead** value on the **IK and Modifiers** tab determines how far ahead of itself an item *looks* as it travels a path, when **Align to Path** is activated. Like a racecar driver, the item can anticipate the turn and begin turning before the motion path actually curves. An object looking three frames ahead will smoothly round a corner as it travels through the turn, whereas as an object looking fifteen frames ahead may appear to *turn out* of the corner before it has actually traveled through the turn itself.



chapter 12
The Scene Editor

Chapter 12: The Scene Editor

The Scene Editor gives you the *big picture*. Here you can see a list of all of the items in your scene arranged hierarchically. You can see all of the individual enveloped channels, perform global edits on keyframes, and change hierarchy. Here, you also set the wireframe colors, change visibility modes, and even load a reference sound file.

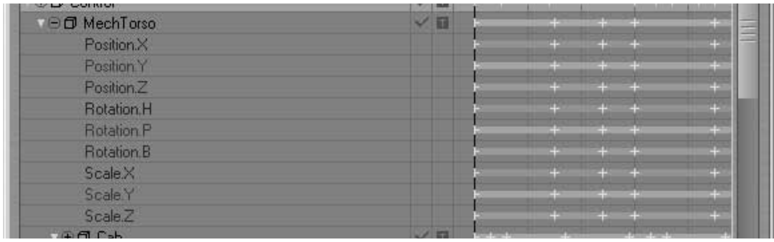


Scene Editor

BASIC FUNCTIONS

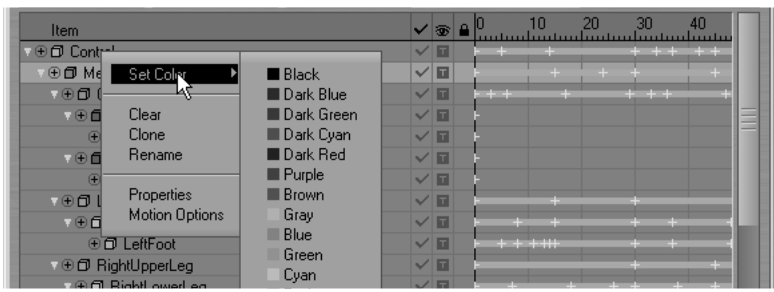
The scene list is a standard LightWave list window. You can expand and collapse groups as well as subordinate items (e.g., child objects) by clicking the arrow icon that appears to the left of the item name.

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.



Expanded channels

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.



Changing item color

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!

Shrunken Scene Editor become 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 somewhat 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.

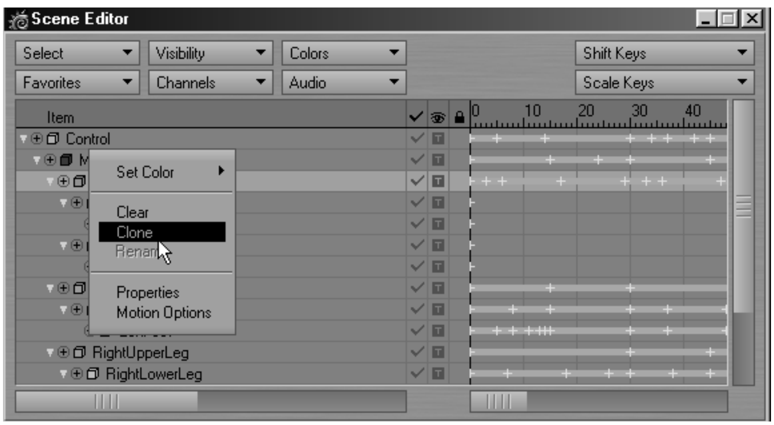
Light is hidden and camera is visible

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.

The lock icon indicates a locked item

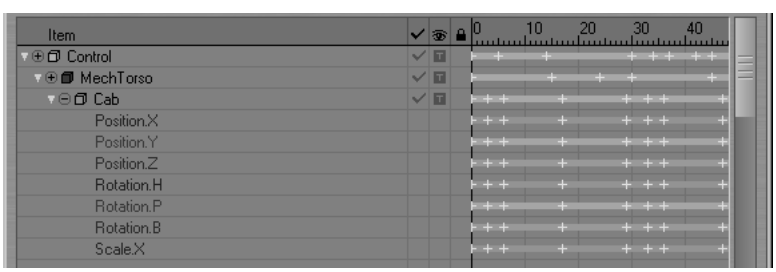
POP-UP MENU

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.

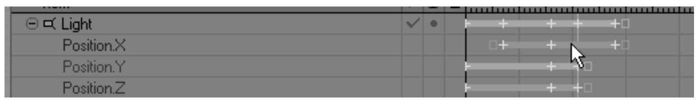


The line with the item's name is the master channel and will show a composite of all of the keyframes in any of the underlying channels.

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.

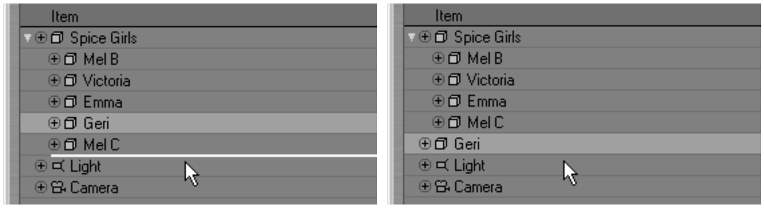
There are square handles at the beginning and end of every bar. You can drag these to scale all of the keys in the bar.

Drag handle at end to scale

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.



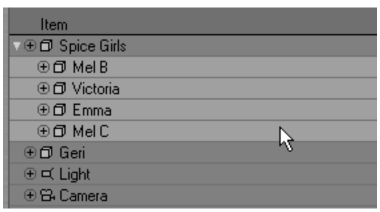
Unparenting object by dragging

For more information on parenting, see Chapter 11.

NOTE
An item inserted between a parent and child will always become another child.

EFFECTS OF SELECTION

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.

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.

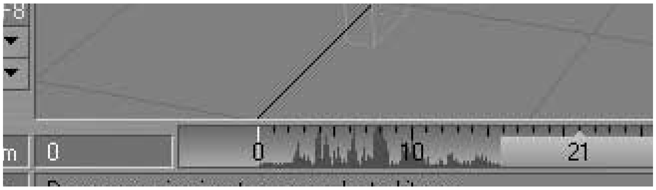
Some operations can affect frames outside the specified range. For example, shifting a range of frames in the middle of a motion path will cause keyframes after the range to shift so they are not overlapped by the newly shifted keyframes.

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 <u>other than motions and envelopes</u> 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.

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 though 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.



Audio graph visible behind frame slider

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).

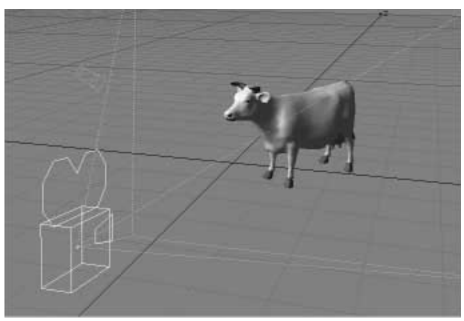
chapter 13

Camera Basics

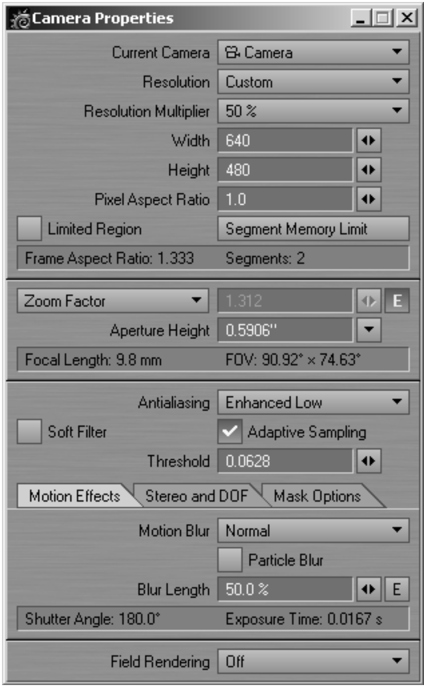
Chapter 13: Camera Basics

When you create a LightWave animation or still image, you will have a display device in mind. In the case of video, it's probably a TV set or monitor. You might also be creating something that will appear only on a computer screen. Other possibilities include film or print media.

Whatever the ultimate destination of your images, two things will become very important before you can render your images: pixel aspect ratio and resolution. Essentially, these are the shape of the pixel and number of pixels, respectively.



Whenever you render a frame in LightWave, you render it from the camera's point of view. LightWave's Camera Properties panel lets you change any and all settings of the camera. To open it, just select a camera and press the P key. Among other things, you can control the resolution of the rendered output, the antialiasing amount, motion blur, and the depth of field settings.



Camera Properties panel

MULTIPLE CAMERAS

You can add additional cameras by choosing **Items** > **Add** > **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.

RESOLUTION

The **Resolution** pop-up menu contains several standard resolution presets. Selecting one will automatically set the **Width** and **Height** fields, and the **Pixel Aspect Ratio** field (more on this parameter later).

Width and **Height** fields determine the precise number of horizontal and vertical pixels, respectively. Any values between 16 x 16 and 16,000 x 16,000 are acceptable.

The **Resolution Multiplier** scales the underlying selected **Resolution** up or down, with **100%** as the default full resolution. This feature lets you quickly reduce or enlarge the resolution of your rendered images. You will often want to render test animations at a smaller resolution for test purposes. Particle, line, and edge thickness, as well as the glow radius, are scaled by the **Resolution Multiplier** during rendering.

Manually entering values directly into the **Width** or **Height** fields will override any selected **Resolution** (which will then display the word **Custom**). The **Resolution Multiplier** will continue to operate, automatically scaling these values if you alter its setting.

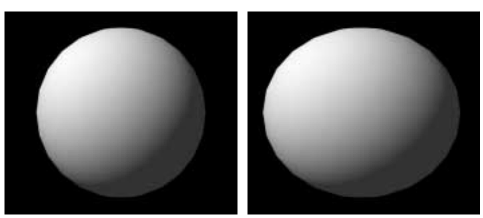
Deciding on a resolution is generally a matter of output quality and intended use. If you are rendering for video, there is no reason to use anything other than the appropriate NTSC or PAL settings. If you are rendering for film or print, you'll want to use a higher resolution. For computer animations, lower resolutions may be adequate.

PIXEL ASPECT RATIO

A pixel is a pixel, right? Wrong. Although pixels are generally square on your computer monitor, some display devices, like television displays, use rectangular pixels.

The **Pixel Aspect Ratio** refers to the shape of the pixels on the target display device. It is calculated by dividing the width of the pixel by its height. For instance, a square pixel will give you a **Pixel Aspect Ratio** of 1.0, and this is nearly always used for images that are viewed exclusively on computer monitors. For NTSC video, **Pixel Aspect Ratio** is about .86 to .9.

So why do you have to worry about this? Well, remember that a pixel is just a dot with some color information. So let's say you had a ball that was exactly 50 pixels high and 50 pixels wide. Generally, the ball would be perfectly round if you displayed it on a computer monitor. However, if you showed this same image on an NTSC video display (e.g., a TV set), the ball would appear vertically elongated. That's because NTSC video's **Pixel Aspect Ratio** is about .9. Conversely, if the ball appeared round on an NTSC monitor, it would appear squashed on a computer monitor.



Left: .9 Pixel Aspect Ratio on video monitor. *Right*: Same image on computer monitor.

Fortunately, selecting one of the **Resolution** settings will also set the default **Pixel Aspect Ratio** to the proper value. If you are working on an animation that will be shown on video, you'll have to get used to the stretched/squashed imagery when viewing it on your computer display.

FRAME ASPECT RATIO

Do not confuse **Pixel Aspect Ratio** with the more common term *frame aspect ratio* (sometimes just referred to as *aspect ratio*). Frame aspect ratio is the pixel width divided by the pixel height times the **Pixel Aspect**

Ratio, usually rounded to whole numbers. For example: a standard computer resolution is 640 x 480, which would have a frame aspect ratio of 4:3 or 1.333 in decimal form. (640/480 reduces down to 4/3, which equals 1.333) The frame aspect ratio is displayed for informational purposes on the Camera Properties panel.

LENS SETTINGS

The zoom factor pop-up menu lets you set the equivalent real world camera lens using **Zoom Factor**, **Lens Focal Length**, or **Horizontal FOV** (Field of View), and **Vertical FOV** values. The default, **Zoom Factor** of 3.2, is equal to a 24 mm camera lens, a mildly wide angle lens. Users familiar with camera lenses may find the **Lens Focal Length** the easiest to use. Long-time LightWave users will be more comfortable with **Zoom Factor**. The FOV settings set this parameter using the degrees of view, which take into account the **Width** and **Height** settings.

Smaller **Zoom Factor** and **Lens Focal Length** values produce a wide angle lens effect, while larger settings produce a narrow (telephoto) lens effect.

A close-up using an 8mm lens

Create an envelope for special effects such as quick zooms in and out, or zooming in on a subject while pulling the camera back.

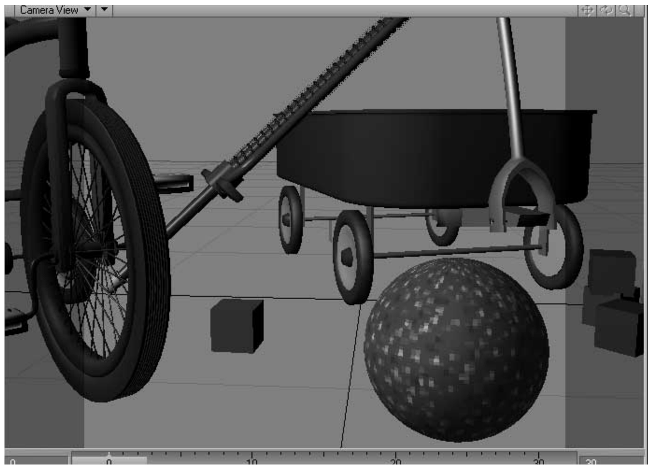
Aperture Height

In LightWave, you can change the size of the virtual film that the camera uses by selecting the **Aperture Height** button. Changing this setting affects only the strength of the **Depth of Field** effect (see below) and the **Lens Focal Length**. When you generate LightWave imagery for film photography, use the same film size used for the project's real world camera, so you match the optical characteristics of that camera.

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

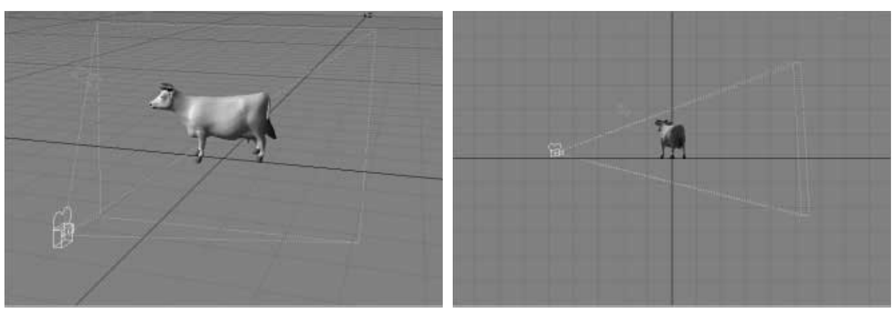
CAMERA SETTINGS IN A VIEWPORT

When you use the Camera view, the areas of the viewport outside of the frame are shaded with the overlay color, marking the edges of the camera frame. Items (or portions of items) in the shaded areas will not be seen when you render the frame.



These areas can be either horizontally or vertically oriented depending upon the resolution and pixel aspect you choose (and the shape of your interface screen).

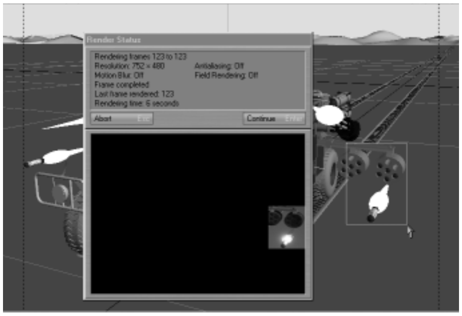
If you aren't using the Camera view, the camera representation includes a *camera pyramid*, with its tip lying on its side. The camera pyramid shows the frame aspect of the image that will be generated.



Camera pyramid

RENDERING A LIMITED REGION

You can use a *limited region to* mark off a specific rectangular area to be rendered. This is a great feature when you wish to test render an area of the frame without rendering the entire image.



Two controls are available for using limited region. If you activate the **Adjust Limited Region Tool** in the Camera view, a yellow dotted-line bounding box will appear. This is the limited region rendering area. To change the area, drag either the corners or the sides of the rectangle. Place the cursor inside the rectangle to move the entire rectangle around. Choose a new tool, like Move, to stop editing the limited region.

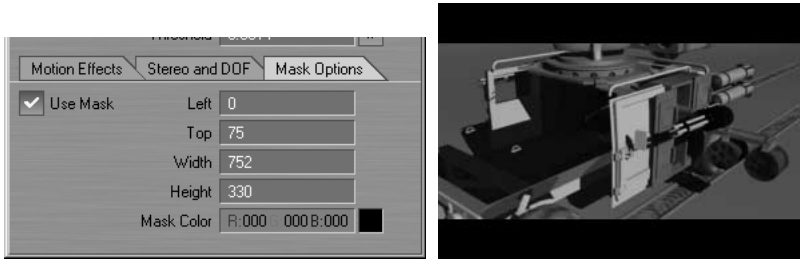
The second control is the **Limited Region** option on the Camera Properties panel and if you activate it, only the limited region will render. Pressing CTRL+L will toggle the state of this setting.

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.

MASKING OUT A REGION

The **Use Mask** is similar to **Limited Region**, but masks the outer areas with a color instead of cropping them out. The mask size, defined by the **Left**, **Top**, **Width**, and **Height** settings, are pixel accurate and considered part of the camera setup (along with the resolution, pixel aspect ratio, etc.). The outside area can be set to any color using the **Mask Color** option. Use this feature to get a *letterbox* effect.



Camera mask and resulting rendered image

SEGMENT MEMORY LIMIT

When LightWave renders an image, it can render the image in segments. Segmental rendering saves memory but usually takes longer. Select **Segment Memory Limit** to input a value (in megabytes) to determine the maximum amount of memory LightWave can devote to segments. If the value is lower than that required to render an entire frame, based on camera settings, images will be rendered in multiple segments. The default is 8MB and the minimum is 1MB.

NOTE
The segment memory value should be considered an upper bound (only the exact amount of memory needed will actually be used).

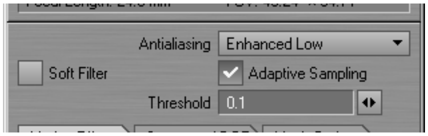
Ideally, you would want to use a large enough value so the number of segments is reduced to one. This is not always possible (especially when using lots of image maps), but can speed up rendering a great deal when using features such as Motion Blur.

NOTE
On platforms that support virtual memory, you may get better results using smaller segments that fit within available RAM. (Using one segment that may not fit entirely in RAM forces you to page to the hard disk and slow down rendering). You may need to experiment with segment values to find a useful setting.

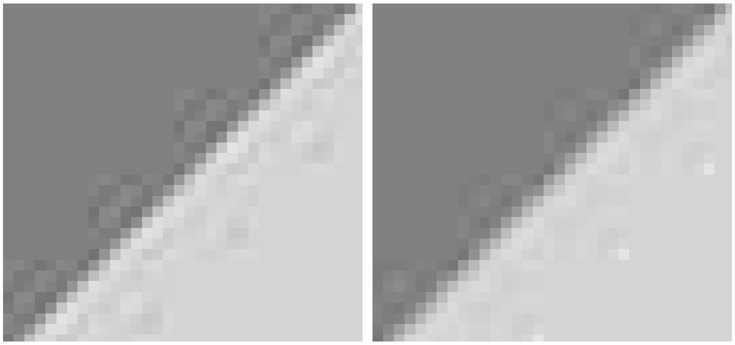
After changing the **Segment Memory Limit**, a requester will appear that lets you save the new value as the default (which is stored in the config file).

ANTIALIASING

Aliasing is a common computer artifact and occurs when a line is seen diagonally. Because a computer's pixels are rectangular in shape, a line seen at a diagonal angle will appear to *stair step* or *alias*. LightWave can *antialias* an image by comparing the RGB values of two neighboring edge pixels and then adding an RGB value between the two. This fools the eye into seeing a smooth edge.



Antialiasing set to Enhanced Low


Without (*left*) and with (*right*) antialiasing

The level of antialiasing performed by LightWave is based on the number of antialiasing passes you tell LightWave to make. Each higher setting performs more passes.

For most video work, low or medium antialiasing is acceptable. For print work, sometimes rendering a higher resolution with no antialiasing is faster and looks as good as or better than a lower resolution with low antialiasing.

All antialiasing levels have an enhanced option. This greatly improves antialiasing quality by using almost twice as many samples for each pixel and filtering them more intelligently, yielding a better rendered image. Selecting an enhanced antialiasing level will add to render time, although with **Motion Blur** or **Depth of Field** activated, this increase is small.

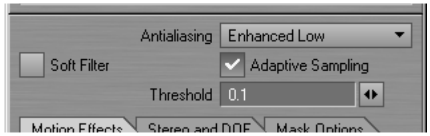
NOTE
The Extreme antialiasing settings are beneficial only if Motion Blur or Depth of Field is being used.

HINT
Try rendering in lower resolution with a high antialiasing level and then scaling the image up in an image processing program to obtain a very soft look (works well for underwater scenes).

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.



Adaptive Sampling active

The adaptive sampling **Threshold** functions by comparing the brightness of two neighboring 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 less 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 Options panel, **Rendering** > **Render Options**) 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.

HINT

When you turn off **Adaptive Sampling**, images take longer to render, but you will achieve better antialiasing results. If using **Adaptive Sampling** does not give you the results you wish or fine lines are being missed (even at a low or zero sampling **Threshold** level), disable **Adaptive Sampling**.

APPLYING A SOFT FILTER EFFECT

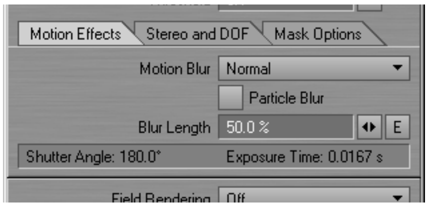
Selecting **Soft Filter** instructs LightWave to render objects with a slightly softened look. This is handy to recreate the *soft* look of film.

NOTE

Soft Filter affects only objects. Any background images will not be affected by the softening effect.

MOTION BLUR EFFECTS

Your eyes naturally blur fast-moving objects. A film camera recording objects that move faster than the camera's shutter will also produce motion blur. In order to simulate this effect in LightWave, use some of the options on the Motion Effects tab on the Camera Properties panel. Using **Motion Blur** can help overcome the unnaturally sharp *computer feel* inherent in many computer generated animations.



Motion Blur set to Normal

Motion blurred moving car

LightWave's motion blur system handles everything that changes over time, like surface changes, shadows, light intensities, etc. It even accounts for curved motion and does not just perform the blur in a linear fashion.

HINT
Try to render with one segment (see **Segment Memory**) to greatly speed up rendering with **Motion Blur**.

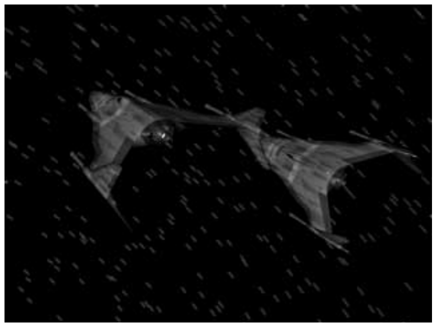
LightWave simulates **Motion Blur** by producing a number of semi-dissolved images of a moving object. In the **Normal** mode, the number of images is determined by the number of **Antialiasing** passes. In fact, you must have some level of **Antialiasing** in order to use **Motion Blur**. The **Dithered** mode greatly enhances the quality of the effect by doubling the number of images.

HINT
Dithered Motion Blur will give you better results than using the next higher **Antialiasing** setting, but will take only about one-third of the amount of extra time the higher **Antialiasing** would require.

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

Because **Dithered Motion Blur** uses a process related to field rendering to compute its effect, you may not use it and **Field Rendering** simultaneously.

If **Particle Blur** is active, LightWave will blur the movement of particles (single point polygons) during the course of an animation (whether they are moving or the camera is moving). This helps simulate the effect of moving quickly through star fields or simulating underwater particles, rain, and fireworks. The blur of a particle will appear as a streak emanating from the particle itself.



Notice the blurring of the star particles

HINT

Sometimes when you use particles in an animation, you may need to move them on or off screen or move them some distance instantaneously (i.e. over one frame). If you don't want the blurring effect to happen during this move, use **Linear** for the **Incoming Curve** (Graph Editor) on the particles over that one frame.

Blur Length and Real Cameras

The shutter in a real camera usually uses a rotating mechanism, making one complete revolution each frame. Thus, the amount of time the film is exposed is actually expressed as an angle (in degrees). (A shutter angle of 360 degrees would be one full rotation.) The exposure time for a single frame can, therefore, be calculated as follows:

Exposure time = 1 / (360 degrees / *Shutter angle* x *Frame rate*)

Now, even though a (feature) film camera takes 24 pictures every second, the film is exposed for a much shorter time than 1/24 second to allow for the camera mechanism to advance the film. In fact, usually it is about one-half (i.e., 50%) that amount: in this case, 1/48 second or a 180-degree rotation (i.e., shutter angle). This is why the **Blur Length** value defaults to 50%. Perceptible movement during this exposure time is what causes motion blur in the real world.

NOTE
In LightWave you can go beyond what is possible in the real world and set the **Blur Length** beyond 100 percent.

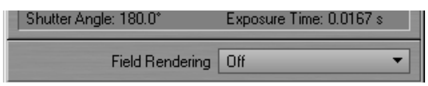
A **Blur Length** of 100 percent will streak the distance of its current location back to the previous frame. Likewise, 50 percent will streak it back half the distance between the two frames and 200 percent will spread it back the distance of the last two frames. A negative value will streak a particle forward.

On the **Motion Effects** tab, there is an information display for **Shutter Angle** and **Exposure Time**. The **Shutter Angle** is the **Blur Length** multiplied by 360 degrees. The **Exposure Time** is computed using the formula mentioned previously.

NOTE
Also see the VectorBlur image filter in Chapter 14.

Rendering Video Fields

LightWave normally renders in frames. However, a frame of video is actually composed of two interlaced fields—one consisting of the odd-numbered scan lines and one consisting of the even-numbered scan lines. If you ever freeze a frame of fast-moving video on a VCR, you may notice a flicker in the image. This is because you are seeing the two individual fields. In LightWave, with normal rendering, each field of video is rendered identically and you will see no flickering.



Field Rendering option

NOTE
Field Rendering is designed for animations going to video tape. It does not work well with images going to film or other non-video playback.

When selecting **Field Rendering**, you instruct LightWave to render each frame as two distinct fields. The motion of an object is divided in half for each frame of its movement. The first half is rendered to one field and the second half is rendered to the other field. LightWave then interlaces the two fields and displays it as a single frame. The **Field Rendering** pop-up menu lets you set whether the even or odd field is created first.

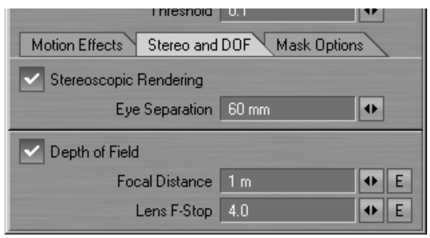
The result is that instead of 30 distinct images per second (normal video rate), you actually see 60 distinct images per second. This makes for much smoother motion and prevents strobe-like effects that the human eye can pick up. (Note that the two fields are still combined into a single file.)

Using **Field Rendering** does *not* greatly increase rendering times.

NOTE
Some video display devices/recorders require even lines to be recorded first while others require odd lines first. If you notice incorrect strobing in your recorded video output, simply select the other **Field Rendering** menu option. NewTek's VideoToaster requires even fields first for 480-line images and odd fields first for 486-line images.

STEREOSCOPIC RENDERING

LightWave offers a **Stereoscopic Rendering** option on the **Stereo and DOF** tab. Rendering with this option active will result in two images for each frame, which depict the scene as viewed from positions to the left and right of the single camera that would normally be used. The frame numbers in the filenames of the saved images will be followed by either "L" or "R" to distinguish these views. The **Eye Separation** input field defaults to 60mm, which is about the average distance between a human's eyes. We suggest that you use a null object as a Camera target and place it at the point where you want eyes to converge.



NOTE
Also see the **Anaglyph Stereo: Compose** filter, discussed in Chapter 14, which will create *red-blue glasses* style images.

NOTE
For an in-depth look at stereography, get out your red-blue glasses and head over to HTTP://WWW.AIFX.COM/3D_LW_A.HTML. Tony Alderson provides some insightful information.

DEPTH OF FIELD

Normally, everything in a LightWave scene is completely in focus, no matter what the distance is to the camera. When rendering with **Antialiasing** greater than **Low**, you have the added ability to use the **Depth of Field** option, which will render images with some objects out of focus.

Depth of Field in action!

NOTE
See also the Digital Confusion image filter in Chapter 14.

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 1/4 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.

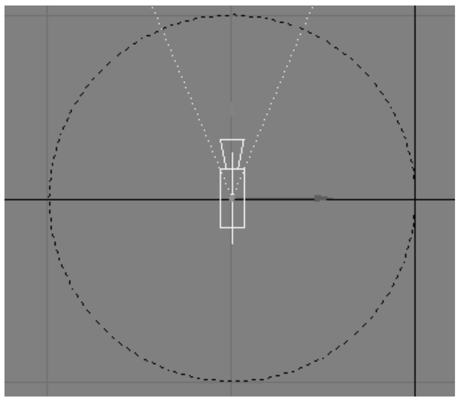
The larger the **Lens F-Stop** value, the larger the depth of field, that is, the greater the distance between the near and far distances where objects appear in focus. Conversely, the smaller the **Lens F-Stop**, the smaller the range of focused area.

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 the depth of field is more pronounced.

HINT
Use the **Quickshade Rendering Mode** (Render Options panel) to check out depth of field renderings quickly.

If **Depth of Field** is active, the focal distance is shown in orthogonal viewports as a dashed circle around the camera.

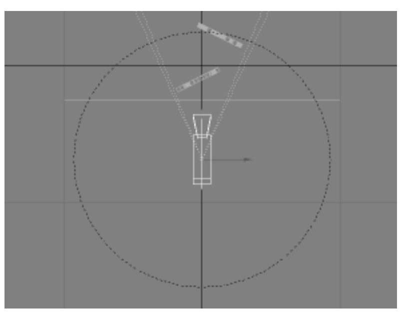


Depth of Field circle around camera

Exercise: depth of field

1 Load the DOF.LWS scene from the BENCHMARK subdirectory.

2 Switch to the Top view and adjust your view. Use the **Front Face Wireframe** for the viewport's **Maximum Render Level**.



If you use the Grid's size of 10 as a gauge, the camera appears to be a little less than 10 meters from the object farthest away along the Z axis. (Remember the camera's position point is actually near its center.) Note the position of the DOF ring.

3 Switch to the Camera view. You should see two objects, aptly named Fuzzy and Sharp.

4 Select the camera and open the Camera Properties panel. **Antialiasing** should be set to **High**. **Depth of Field** should also be active with a **Focal Distance** of 9.2632. This should make the Sharp object in focus, since it is about that distance away from the camera, and the Fuzzy object out of focus, since it is much closer to the camera. The small **Lens F-Stop** setting constricts the *in focus* range to a very small area, effectively limiting it to just the Sharp object.

5 Do a quick render (F9).

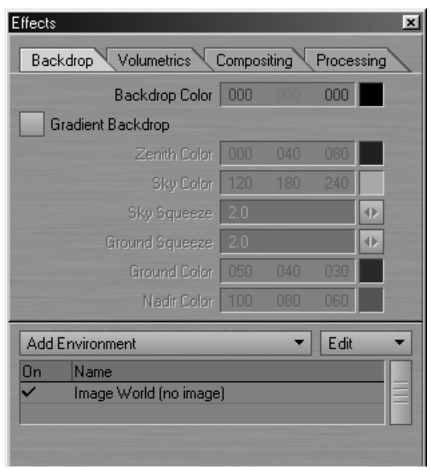
HINT
You can get an accurate measure of the distance between the camera and your selected object using the Ruler or Range Finder custom objects discussed in Chapter 9.

chapter 14

Backdrops and Image Processing

Chapter 14: Backdrops and Image Processing

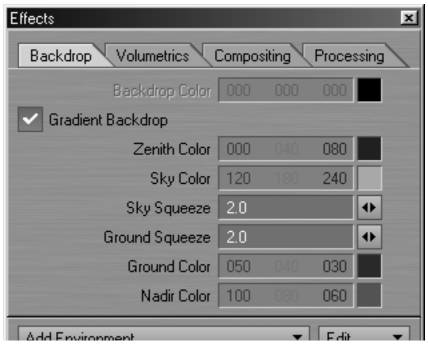
LightWave lets you change the colors of the backdrop behind all of the objects in a scene. The settings are on the Backdrop tab of the Effects panel (**Scene** > Effects: **Backdrop**). You can elect to have a solid color using the **Backdrop Color** setting or a gradient color backdrop. By default, 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'll likely not 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.



Gradient Backdrop active

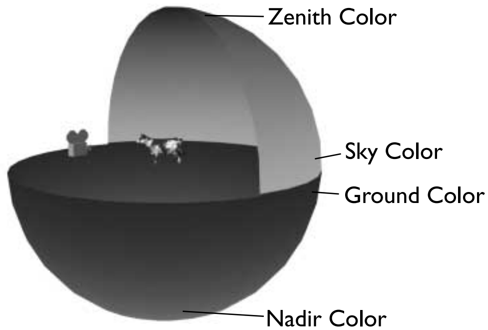
There are actually two gradient areas. Think of it as a huge sphere cut into two halves. One half rests on top of the Y axis plane, and the other half sits directly beneath it.

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

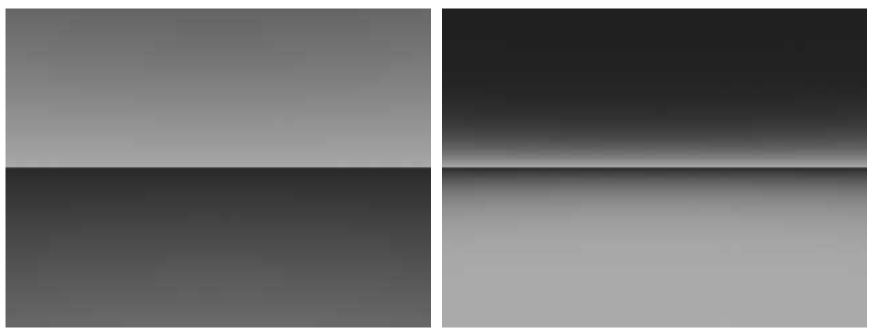
If you want the **Gradient Backdrop** to blend everywhere with no sharp color change, make the **Sky Color** and **Ground Color** the same.

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.



Gradient areas

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.

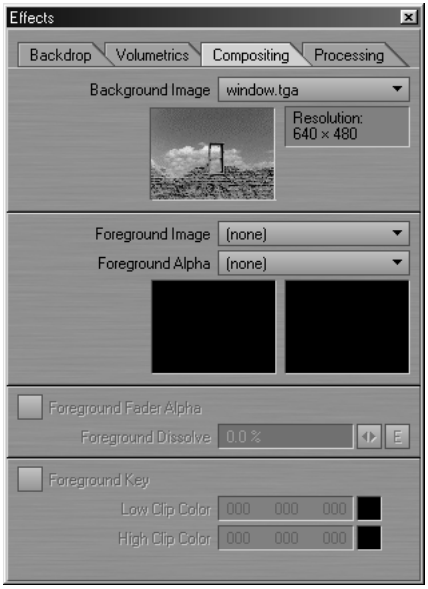


Left: Normal squeeze. *Right:* **Sky Squeeze** and **Ground Squeeze** both at 20.0

HINT
The default gradient colors are useful when simulating a reflective chrome surface.

BACKGROUND IMAGE

The **Background Image** is similar to the gradient backdrop; however, it is always *registered* to the camera. That is, it will always appear in exactly the same position/location no matter which way you tilt or move the camera. You set this option on the Compositing tab of the Effects panel. Background images are considered infinitely distant from the camera. You can never have an object behind a background image, nor can you light a background image or cast shadows upon it.



Background Image loaded

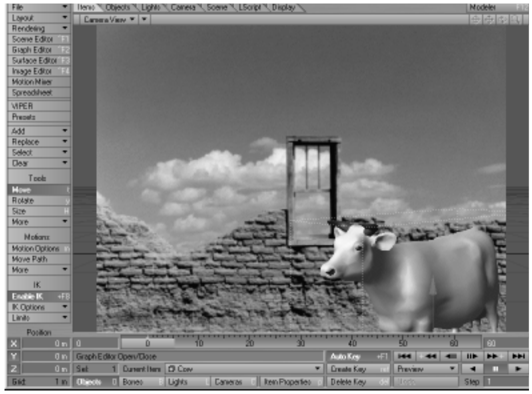
Background images stretch to fit the Camera resolution and frame aspect that you are using. Make sure to use similar-sized background images if you wish them to match.

You often use background images to merge live action and 3D elements, like creating the illusion that a UFO crashed into the Empire State building. An image of the New York skyline would be the background image and the UFO would be a LightWave object.

NOTE
Background images are visible only through the Camera View.

HINT
If you actually need your background image to interact with objects, use the image as a Surface Color Texture mapped on a flat plane object and place the plane at the back of your scene.

If you set **Camera View Background** on the Display Options tab of the Preferences panel (**Display** > **Display Options**) to **Background Image**, you will see the set Background Image in Layout's Camera View. Of course, actually seeing the background in the Layout window is optional. The background image will always appear in the rendered frame.



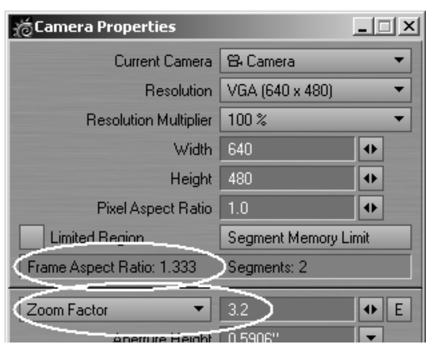
Cow object stands in front of Background Image

NOTE
Displaying background in the Layout window, particularly a color one, is processor intensive, so use this feature sparingly.

BACKGROUND PLATE

Sometimes you will want a flat plane to act as your background *instead* of using the **Background Image** option, described above. This will allow you to do things like cast shadows on the background or fly things behind it. To create a flat box object that exactly fills your screen, use the following steps:

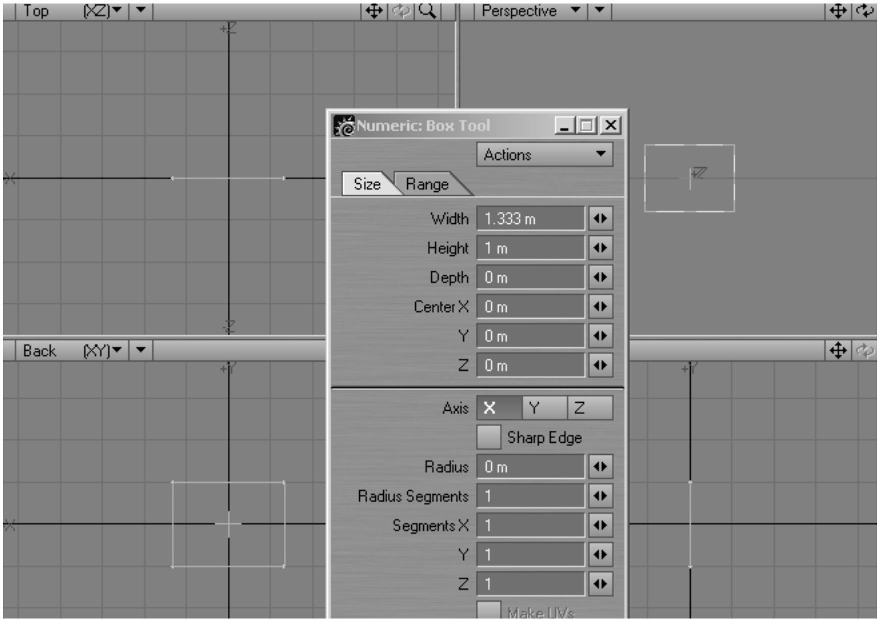
1 In Layout, open the Camera Properties panel. Make sure all settings are set as desired. Note the **Frame Aspect Ratio** and **Zoom Factor** values.



Frame Aspect Ratio and Zoom Factor on Camera Properties panel

2 In Modeler, select the Box tool and open the numeric panel. To make sure you are starting from the default settings, select **Reset** from the **Actions** pop-up menu. Then select **Activate** to gain access to the input fields.

Set **Width** to the **Frame Aspect Ratio** value you noted in the previous step and **Depth** to 0 m. Leave all other settings at their default values.

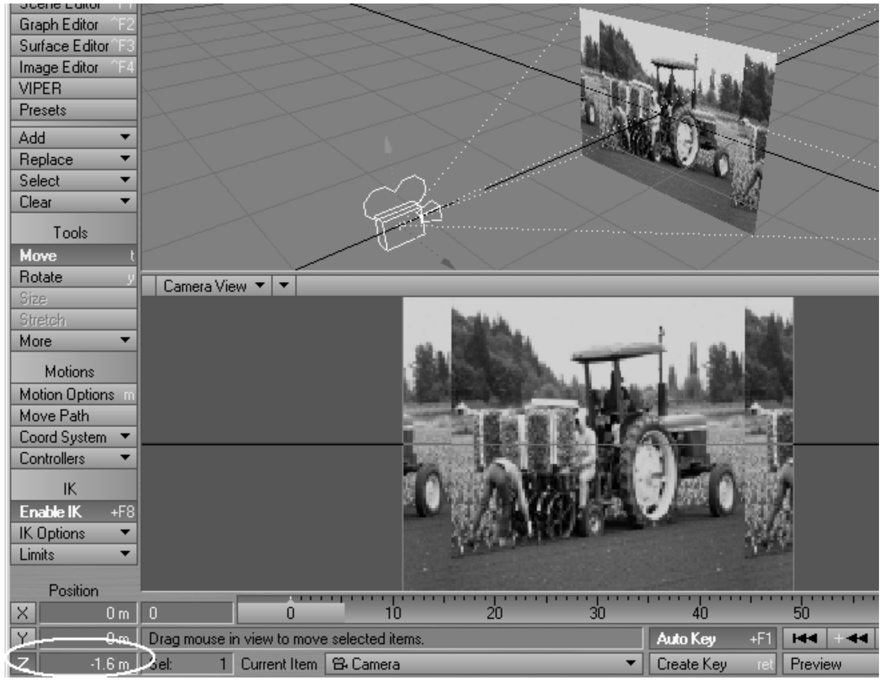


Make a box in Modeler

3 Deselect the Box tool to make the box. The polygon may be facing the wrong direction, so use the Flip command (**Detail** > Polygons: **Flip**) to face it in the negative Z direction. Save it and load into Layout.

4 Finally, the distance between the camera and background plate object must be one-half the Zoom Factor value, noted in Step 1. You can either move the camera or object. After that, the plate will now precisely fill the camera view.

NOTE
See also the Camera Mask custom object in Chapter 9.



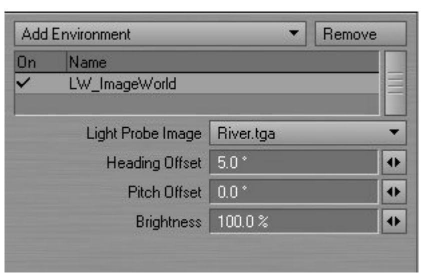
The Distance between the camera and object is one-half the Zoom Factor

ENVIRONMENTS

Several plug-ins may be used to create exciting backgrounds for your animations. You access these plug-ins 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.



ImageWorld

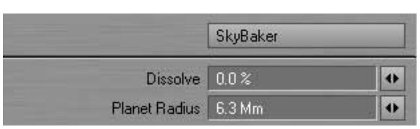
SkyTracer2

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 2-D effect) within an existing scene, or saved as image files to be composited or used as texture maps.

Sample skies created by Gregory Duquesne.

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.

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.



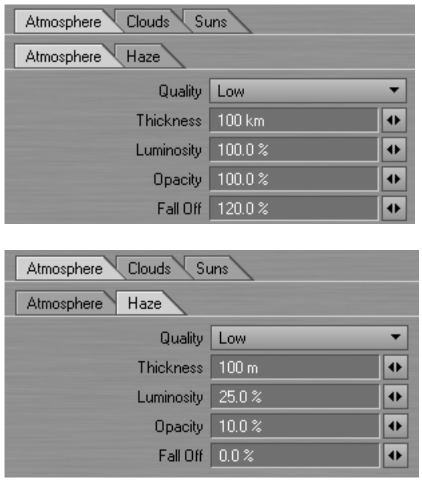
The top portion of the panel contains the settings for SkyTracer2's global parameters.

The **SkyBaker** button brings up the SkyBaker panel (see below).

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.

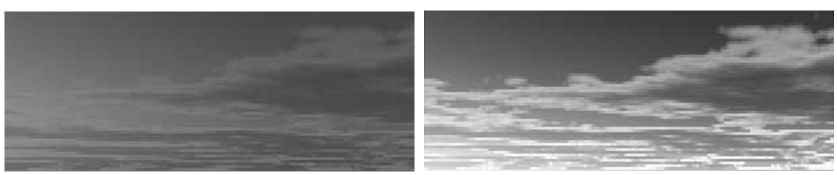
The Atmosphere Panel

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.

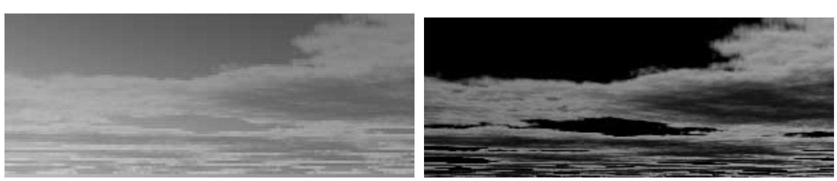


Atmosphere and Haze settings are adjusted in the Atmosphere panel.

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.


An atmospheric Thickness value of 100% and 50%.

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.


An atmospheric Luminosity value of 100% and 0%.

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.

An atmospheric Opacity value of 50% and 200%.

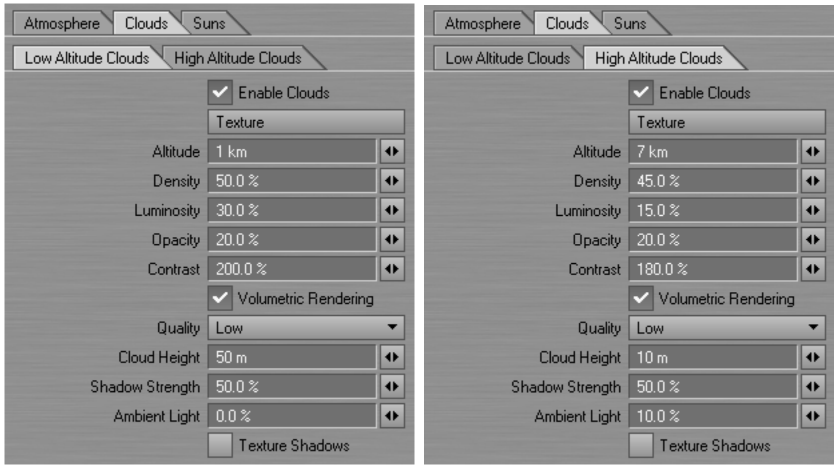
Opacity measures the percentage of light absorbed within the layer. High values attenuate light, and objects disappear at the horizon. When you use opacity with haze, you can create pollution-like effects. However when you adjust the opacity within the atmosphere panel, you can create strange, out-of-this world skies.

An atmospheric Fall Off of 75% and 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.



The Low and High Altitude Cloud panels.

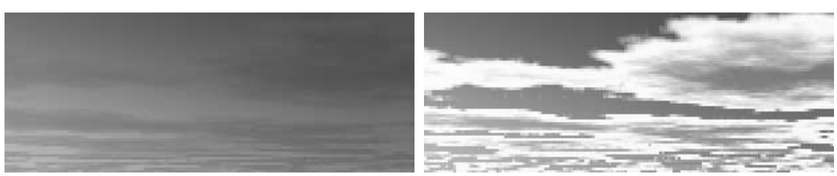
Use the **Enable Clouds** switch to turn the two cloud layers on or off.

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.

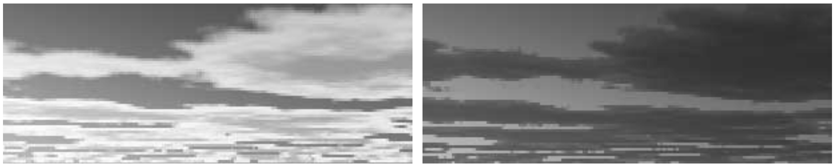
A low-altitude cloud setting Density of 20% and 200%.

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.



A low-altitude cloud Luminosity setting of 10% and 100%.

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.



A low-altitude cloud Opacity of 10% and 50%.

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.

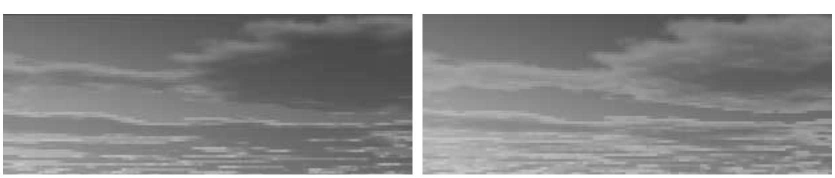
A low-altitude cloud Contrast setting of 100% and 200%.

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.

Volume Rendering activated and deactivated.

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.



The Textured Shadows option activated and deactivated.

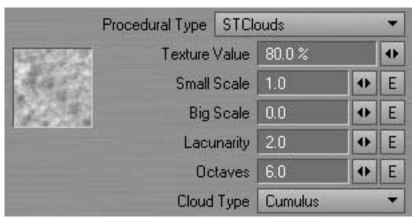
Textured Shadows activates the cloud layer's self-shadowing option, which generates more accurate shadows within the layer, but increases rendering times. **Textured Shadows** is similar to HyperVoxel's self-shadowing feature.

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 procedural texture STClouds' portion of the Texture Editor panel.

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.

For example, try the image DARKFRACTAL.TGA in the IMAGES\TUTORIAL FOLDER.

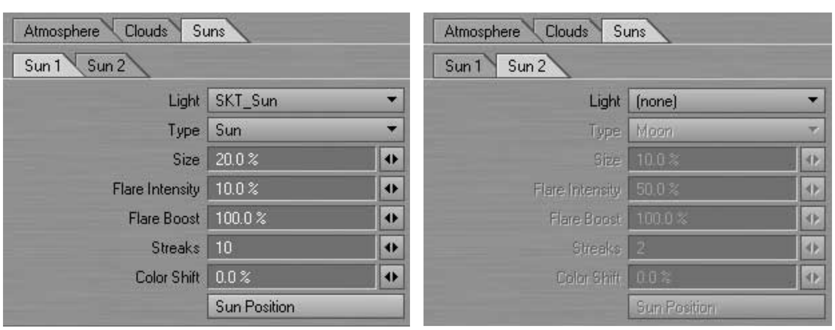
Use a gradient in any layer to modify previous layer's effect. The gradient **Input Parameters** are **Heading** (ray heading angle), **Pitch** (ray pitch angle), and **Height**. The height is the height inside the cloud layer.

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.



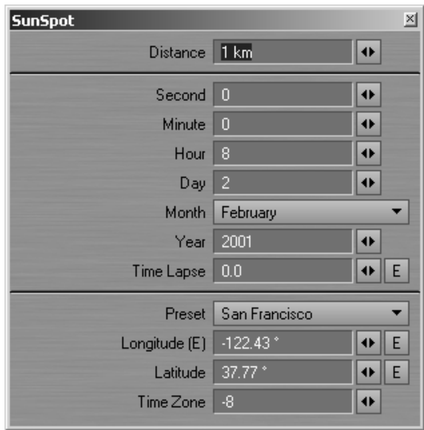
The Suns Panel adjusts the flare parameters used as the sun.

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. The Sun Spot motion modifier is detailed in chapter 11.



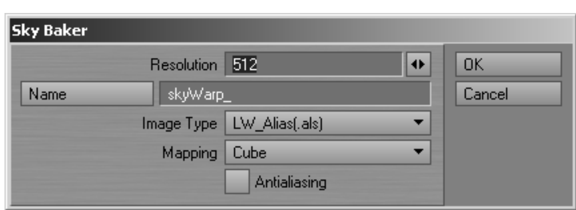
The Sun Spot motion modifier

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.

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.

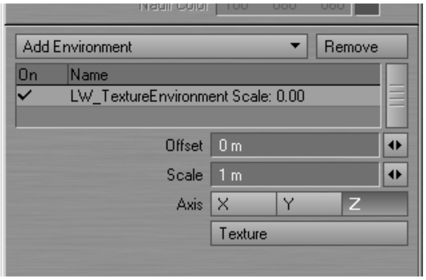
NOTE
The spherical projection has the advantage of being more compact and not distorted when compared to the cubic projection.

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.

Texture Environment

The Texture Environment Environment lets you use a LightWave texture as a background.

You can access the standard Texture editor, letting you mix images, procedurals, and gradients. You can create turbulent multi-colored skies with procedural textures, or manipulate images for incredible effects.



The texture is not *stuck* to the camera background, like a normal background image, so 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 cool nebula behind your starfield using a simple Turbulence or Fractal Noise procedural texture.

NOTE
For information on the Volumetrics tab, see Chapter 15.

COMPOSITING

The whole idea behind traditional image compositing is quite simple: take two or more images, and merge them into a new *composite* image. LightWave lets you do this, but also takes it one step further by letting you throw objects into the mix.

The images can also be a sequence of pictures, so you may use captured video footage as a background for your objects. A simple example would be a modeled UFO moving against a real sky and trees. Images can appear behind all objects, in front of all objects, or a combination of the two with objects showing in between.

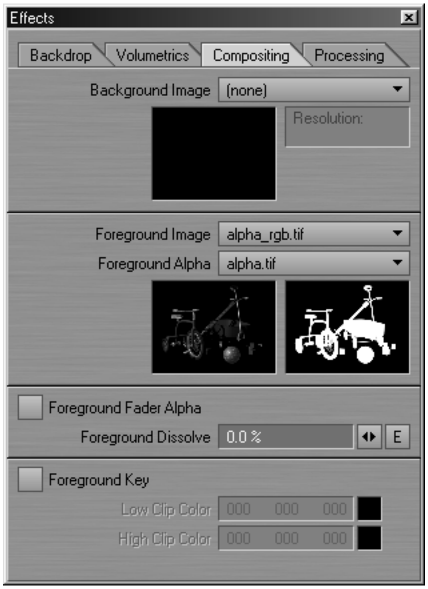
Compositing can be a render time-saver. If you set up a scene with many objects, but only a few are moving, you could render one frame with only the inanimate objects, and then remove all of those objects and render the animated objects against the single frame saved earlier. This is especially useful when the still objects are complicated or involve ray-traced shadows, refraction, and/or reflection.

Foreground Images

Placing an image in front of everything may seem like a silly thing to do. However, some options let you cut parts of the image away so you can see through it. You can also make the image partially dissolved or even envelope the dissolve. Dissolving in a black image in the foreground will produce the common *fade-to-black* effect, or reverse the dissolve to fade in from black.

You can make holes in the foreground image based on a defined color range in the image. However, the biggest drawback to simply *clipping* out portions of the foreground image is that you will have hard edges. A color is either clipped or it isn’t, so you see either the foreground image or the background image.

Use the **Foreground Image** pop-up to set the foreground image.



Foreground and alpha images loaded

Alpha Images

You may also use a special *alpha image* to merge the background and foreground images. This type of alpha image is very different from the type you use to set up a transparency surface texture. LightWave composites the foreground image over any objects or background by *adding* their colors to the foreground image. How much background is added is determined by the alpha image. The darker the area of an alpha image, the more the background is added to the foreground. The pseudo mathematical equation might look like:

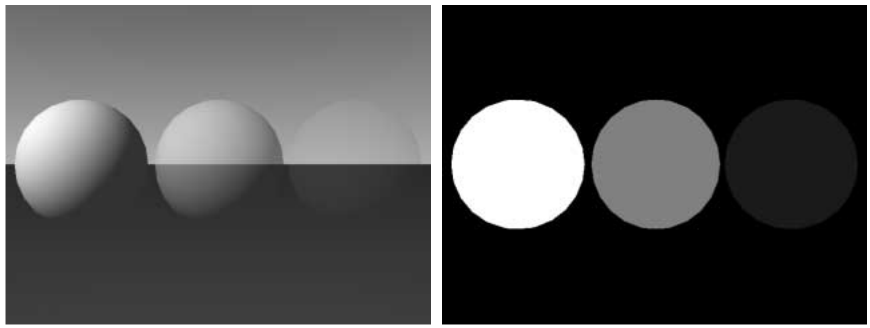
Foreground + (1 - Alpha) * Background

If you used the same exact image for both the background and foreground images, plus a solid black image as the alpha image, you will receive a final rendered image where every pixel is twice the color value it was. This results from the background image being completely added to the foreground image.

Creating Alpha Images

Generally, alpha images will be generated when you render a scene to create the foreground images. When you select **Save Alpha** on the Render Options panel's Output Files tab (**Rendering** > **Render Options**), LightWave will generate and save an alpha image in addition to the normal RGB output. The alpha image will be composed of grayscale values representing the opacity of any objects that were rendered in the scene.

An object that contains no transparent surfaces will be rendered as solid white. Transparent surfaces will render in some shade of gray depending on how transparent they are. One hundred percent transparent surfaces render as black. A 50-percent transparent surface will render as 50-percent gray. Using object dissolve, antialiasing, motion blur, and so on. will also give you values other than white in an alpha image. Any background (image or colors) will be black in the alpha image, as will any additive effects such as glow or lens flare.



Left: RGB image of three balls will differing transparency. *Right*: Associated alpha image

Since glows and lens flares are additive effects and are assigned a value of black in an alpha image, glows and lens flares in the actual foreground image will simply have the background values added, so they will appear brighter where the background is a value other than black.

HINT

Generally, due to LightWave's additive compositing method, foreground images are created using a solid black backdrop. This allows the composited background to show through unaltered when it is mixed 100 percent with the foreground.

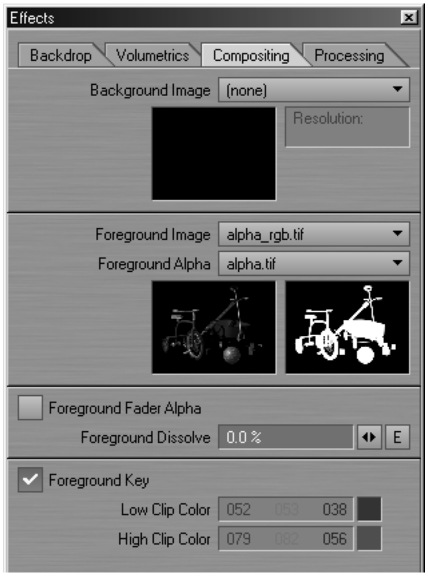
Foreground Fader Alpha

What happens if you want to composite a foreground element on top of objects or a background, but the foreground image was not rendered over black? In this case, LightWave provides a **Foreground Fader Alpha** button that can be selected. In this mode, LightWave will ignore any areas of the foreground image corresponding to black areas in the alpha Image. These areas of the foreground image will be faded *awa*y to nothing so you see 100 percent of the background instead.

When using **Foreground Fader Alpha**, glows and lens flares will not be added to the final rendered image (their corresponding alpha is black) unless you are using a different alpha image containing non-black areas in those locations. Additionally, antialiased object edges most likely will stand out because they contain bits of color from the original non-black background.

Foreground Key

Activate **Foreground Key** when you want to *key out* (i.e., not render or see) a color (or range of colors) from the foreground image. Use the same color value for both **Low Clip Color** and **High Clip Color** if you wish to key out one color only. Use different values to key out the **Low Clip Color** and **High Clip Color** values, and all those in between.



Foreground Key option active

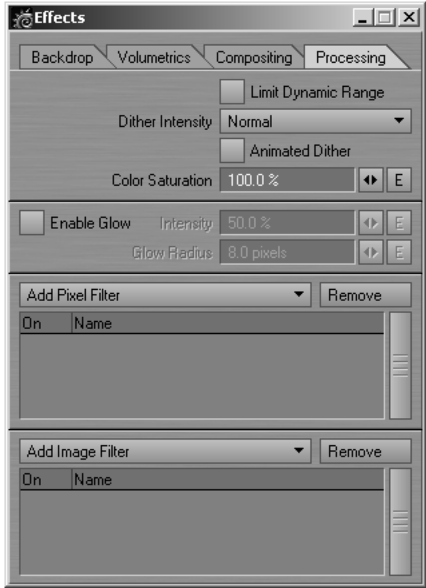
The **Low Clip Color** is the color value for the darkest color value that will be keyed out. **High Clip Color** is the brightest color value that will be keyed out.

HINT

If you want to create a giant space battle and lack the RAM to hold all of the objects and image files, you could composite layers of ships and achieve the same results. This is, in fact, how some broadcast TV shots were done using machines with only 32MB of RAM in the early days of LightWave.

PROCESSING EFFECTS

The Processing tab on the Effects panel contains functions that apply effects to the rendered image. Choose **Scene** > Effects: **Image Process** to directly bring up the Processing tab of the Effects panel.



Processing tab of the Effects panel

Limit Dynamic Range

Limit Dynamic Range clips the pixel color components of each rendering pass at 1.0, improving the antialiasing of extremely bright areas. This option should not be used with filters or image savers that expect high dynamic range data.

Dither Intensity

Dithering blends two colors to simulate a third color between them, forming a more realistic blend. **Dither Intensity** lets you set the amount of color blending used by LightWave when rendering an image. Even with 24-bits of color data, it is possible to perceive color banding where distinct changes in color or brightness occur within otherwise smoothly ramped colors. **Off** removes all dithering, and you will probably experience some color banding. **Normal**, the default setting, reduces banding to the point where it nearly disappears. **2x Normal** increases the dithering even further, which may be useful for high-end systems that still retain some appearance of banding in the final image. **4x Normal** boosts dithering so that the resulting image looks more grainy, like film, which may be a desirable effect (especially when used with **Animated Dither**, below).

Animated Dither

Select **Animated Dither** to change the dithering pattern used from one frame to the next. This ensures that dithering is randomly placed, so there is no apparent pattern to the dither blend. With a **2x Normal** or **4x Normal Dither Intensity**, this can be used to approximate the randomness of film grain moving through an image.

Color Saturation

Color Saturation lets you control the amount of color in a scene (or in an animation, if using an envelope). **Saturation** at 100% is a normal, full-color setting, while saturation at 0% is a black and white setting.

Glow Settings

When **Enable Glow** is turned active, LightWave can add a glow around surfaces with a (surface) **Glow Intensity** above 0%. Use the controls below to set up the amount of glow you wish to add to all such surfaces.

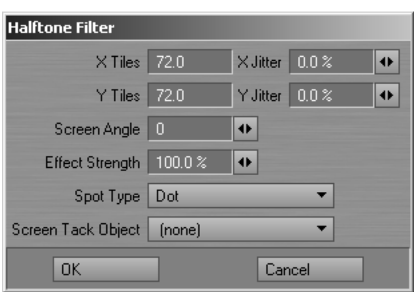
Glow Intensity sets the brightness of the glow, starting from the edge of the surface itself and fading away from there. **Glow Radius** sets the distance (in pixels) that the glow extends from the edge of a glowing surface. Note that different resolution settings will cause dramatically different results.

PIXEL FILTERS

Pixel filters let external applications affect LightWave's rendering engine. Filters in this class can be affected by motion blur and other sub-frame operations during the render rather than as a post process, as image filters are.

Halftone

In print, halftone screens are made up of dots that control how much ink is deposited at a specific location. Varying the resulting dots' size and proximities creates the illusion of variations of grey or continuous color.



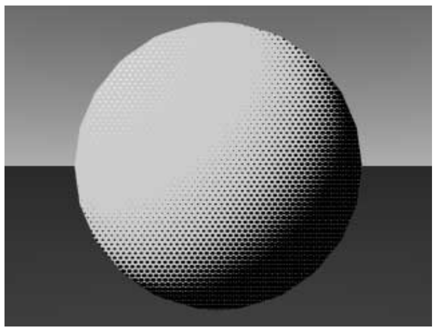
Halftone panel

The **X Tiles** and **Y Tiles** values determine the number of possible horizontal and vertical dots. If you would like to randomize either of these settings (throughout your animation), use the corresponding **Jitter** fields.

In the photography world, this type of effect is achieved with a (physical) screen that breaks up the image into dots. Think of the **Screen Angle** setting as the rotation of that screen. It controls the angle of the dots.

You can control the overall amount of effect by adjusting the **Effect Strength** setting. Settings below and above the default of 100% are allowed.

You can change the pattern by changing the **Spot Type**. Specify and animate a **Screen Tack Object** to animate the pattern position. The **OK** button stands for Ocular Kinesthetics. It lets the filter change pixels that will then be perceived by the ocular nerves.



Halftone effect

SasLite

See the discussion on the “Sasquatch Lite” displacement plug-in at the end of Chapter 9.

IMAGE FILTERS

Image filters post-process rendered images. All pre-processor (image filters added on the Image Editor) and post-processors (image filters added on the Effects panel) are the same plug-in class. However, some image filters require post-process data and cannot be used on the Image Editor.

Anaglyph Stereo: Compose

When you use this filter in conjunction with the **Stereoscopic Rendering** option (Camera panel, Stereo and DOF tab), it saves out the left and right channels *stitched* together as the right channel (*red-blue glasses* style), and the normal RGB image as the left channel.

This can be used with the **QuickTime Stereo** animation **Type** (Render Options panel) (Mac and Intel only) to save out a stereo QuickTime movie.

Anaglyph Stereo: Simulate

This filter *fakes* having the **Stereoscopic Rendering** option turned on in the Camera panel by making the image a stereo image. The options default to 1.1 (meters) in **the Eye Separation** field. In many cases, you may want to increase this value, to exaggerate the effect.

NOTE
You must render in one segment to use this filter, so you may have to adjust your **Segment Memory Limit** (Camera panel) accordingly.

Bloom

Sometimes light reflections in the real world are so bright that too much light will enter a camera lens, over-saturate areas, and create a glare or a glowing halo. Shiny metallic objects like cars and water often exhibit this phenomenon. Bloom will mimic this effect.

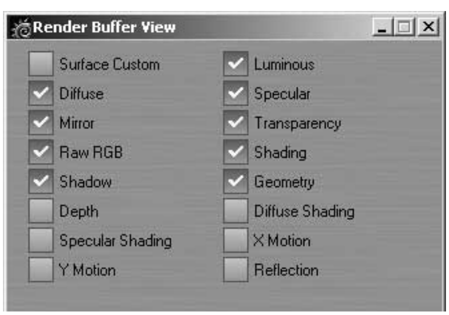
Bloom

Threshold determines how bright a pixel must be before it *blooms*. **Strength** is the strength of the *bloom brush* compared to the pixel it blooms. **Size** is the radius in pixels of the brush at a 640 by 480 resolution. If the resolution is different, the brush will be adjusted so that the effect always looks the same at different resolutions.

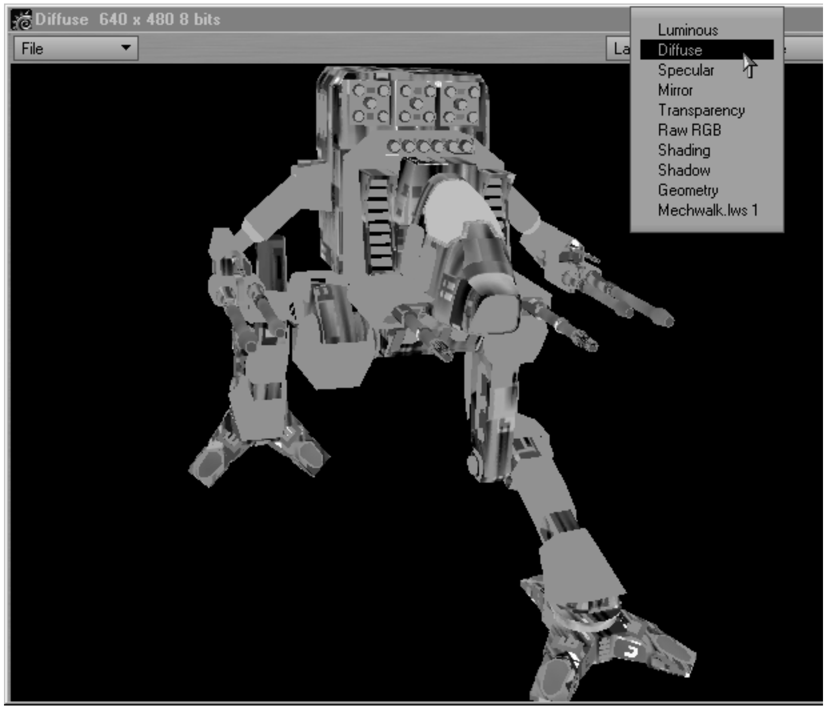
NOTE
Use Bloom for a simple over-exposure bleed effect. For a more feature-rich version, use Corona.

Render Buffer View (Post-processing only)

This filter makes the selected internal buffers visible as separate images when you use the Image Viewer.



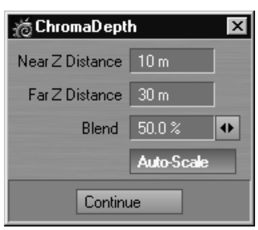
Render Buffer View dialog



Viewing the buffer images with Image Viewer Layer pop-up menu

Chroma Depth (Post-processing only)

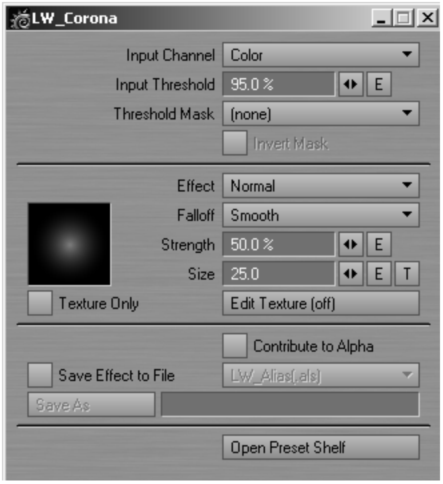
The ChromaDepth filter makes a stereo image for use with ChromaDepth™ glasses (see www.chromatek.com). Basically, color determines the apparent depth in the image. This filter re-colors the scene based on the Z-depth; it spreads the spectrum of colors from the **Near Z Distance** to the **Far Z Distance**. **Blend** dissolves the ChromaDepth image in with the regular colors to make the depth effect more subtle. **Auto-Scale** finds the actual *near* and *far* for you when it renders—you can render once with this option on and the calculated values will appear in the fields.



Chroma Depth settings

Corona

Corona is an extension of Bloom. Essentially, it scans the **Input Channel** and whenever the input value exceeds the **Input Threshold**, Corona applies some effect.



Input Settings

Input Basis lets you choose which internal buffer to read for the bloom effect, so you can bloom on specularity, bloom on diffuse, and so on. The **Input Basis** acts like a trigger: when its value exceeds the **Input Threshold**, the effect is applied to that pixel.

Color uses raw pixel values—essentially any pixel on screen that is bright enough gets bloomed. **Alpha** uses alpha's pixel values—0 to 100% for the image. **Specular Shading** uses 0 to 100% of surface specularity as *shaded* during the rendering. This varies over a given surface and is different from the Specular surface channel, which is uniform over a surface. **Diffuse Shading** is similar, but uses the diffuse surface property.

Geometry uses the normal of object surfaces, where 100% indicates that the normal points at the camera. **Inverse Geometry** is similar, but 100% indicates that the normal points perpendicular to the camera. These are easily demonstrated using a sphere. For **Geometry**, the center of the ball would trigger Corona, while **Inverse Geometry** would result in the effect along the edges.

Special uses the surface Special Buffer feature on the Surface Editor. The value of the Special Buffer is compared against the threshold and when it exceeds that value, the Corona filter is applied.

The input can also be *masked* to skip areas of the input altogether. **Threshold Mask** is basically an alpha channel. Brighter areas will be susceptible to the mask, while darker areas will not.

Effect Settings

The **Effect** pop-up menu selects your blending mode. **Additive** yields very hot (white) results where triggering pixels are closely grouped. This is useful for, say, obtaining the look of metal being heated. The center of a block of metal will become super hot while the edges do not. **Normal** is similar to **Additive**, except in the case of heated metal, the effect at the center and the edges tends to grow more evenly. **Maximum** takes the maximum of contributive pixels. This yields an effect like applying balls of cotton to brightly colored particles, whose effects start to merge as the particles become closer to each other.

The **Falloff** pop-up menu lets you select how the bloom brush falls off. The preview window will give you an idea of what the falloff will look like.

Strength is the strength of the *brush* compared to the source pixel. **Size** is the radius in pixels of the brush at a 640 by 480 resolution. If the resolution is different, the brush will be adjusted so that the effect always looks the same at different resolutions.

The **Edit Texture** button can modulate the color of the bloom brush with a texture.

When **Texture Only** is inactive and **Edit Texture** is off, the effect uses the color of the original image. If texture color is available, the effect uses the color of the original image plus the contribution of the texture. When **Texture Only** is active and there is also a texture color, the effect uses the value of the texture only.

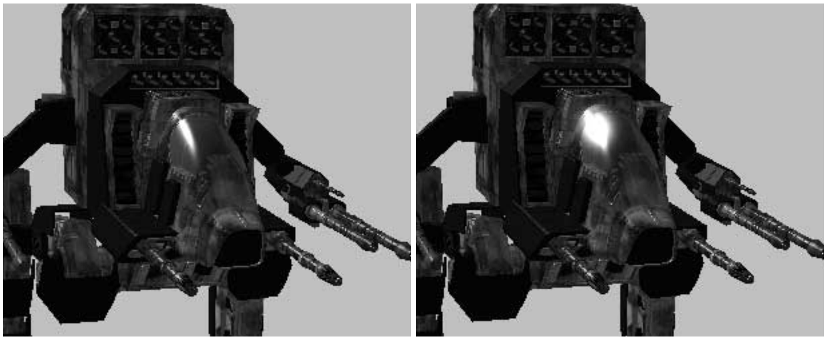
Other Settings

The Corona filter will affect your alpha channel if you activate the **Contribute to Alpha** option.

Use the **Save Effect to File** option to save just the corona effect to an image file when rendering. Note you must also choose a file format and define a filespec.

When you use gradients with Corona, you will have additional options for the **Input Parameter**. These options let you customize how the corona effect is applied. For example, the size or intensity of the effect can grow or diminish based on an object’s proximity to another object, center of the image, and so on.

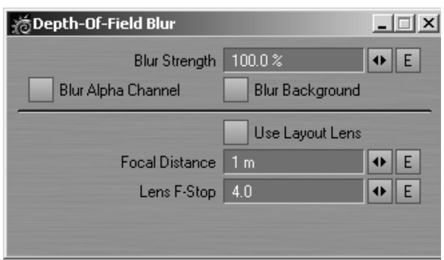
You can use the standard Preset Shelf if you want to save and recall settings.



Left: Without Corona. Right: With Corona (Note: Applied as a post-processing filter)

Depth-Of-Field Blur (Post-processing only)

This filter lets you add a depth of field effect that is based on a fast image filter, without requiring multi-pass antialiasing like normal depth of field does (Camera panel). You can adjust the overall strength of the blur, as well as independently choose whether to blur the alpha channel and background.

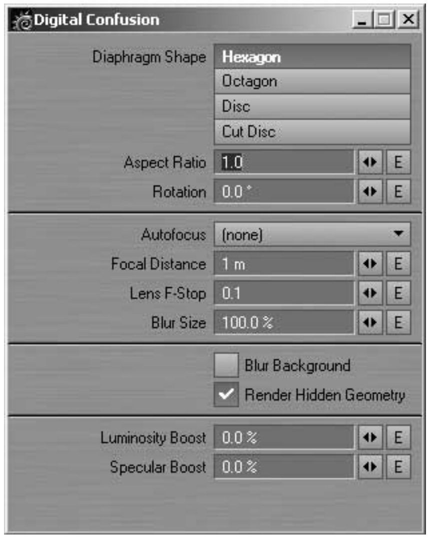


See the discussion on normal Depth of field (DOF) in Chapter 13 for information on the **Focal Distance** and **Lens F-Stop** settings. You can also activate the **Use Layout Lens** option to use the Camera Properties DOF settings.

Digital Confusion (Post-processing only)

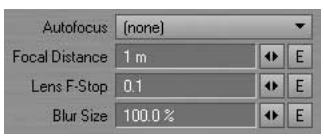
Lightwave's built-in Depth of Field effect (Camera Properties panel) adjusts which pixels the rendering camera considers in and out of focus. The Digital Confusion image filter creates the same effect using the similar controls, but offers several extra features. Adjusting the **Focal Distance** and **Lens F-Stop** still easily controls the range of focus, but added options include camera lens parameters, auto-focusing, and ways to fine-tune surface properties.

Since Digital Confusion is added during the anti-aliasing rendering pass, it will respect and can actually improve oversampling methods like motion blur and Depth of Field effects. However, Adaptive Sampling (Camera Properties panel) may not function correctly with this filter. In this case, you should use an Enhanced Antialiasing setting (Camera Properties panel).



The Digital Confusion interface.

The four **Diaphragm Shape** settings, **Hexagon**, **Octagon**, **Disc**, and **Cut Disc**, determine which pattern Digital Confusion will use when “defocusing” the rendered image. These settings correspond to the actual shape of the camera lens used during this effect. The **Aspect Ratio** and **Rotation** angle of the camera lens can also be adjusted to create even more specialized effects.



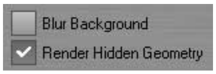
Controlling the Blur Effect

When using Lightwave's built-in Depth of Field controls, it is sometimes difficult to keep a moving object in focus. Now, instead of using envelopes to animate the focal distance, with Digital Confusion you can simply select a reference object from the **Autofocus** pop-up menu and the proper focal distance will be computed automatically. This reference object can be either the (target) geometry in the scene, or a null object used to dynamically adjust the focus. When an object is selected, the **Focal Distance** field becomes disabled.

The **Focal Distance** setting represents the distance from the camera to the point in space that is in focus. Objects that fall either in front of or behind this area will be rendered out of focus. Just how far out of focus is determined by adjusting Digital Confusion's **Lens F-Stop** setting. By changing this value, you are adjusting the radius of the area that is in focus. For this value, the smaller the **Lens F-Stop** size, the smaller the in-focus area will be.

NOTE
For more information on using the Depth of Field controls, refer to Chapter 13, "Depth of Field."

The **Blur Size** setting acts as a multiplier to Digital Confusion's "defocusing" effect. Adjusting this control is similar to adjusting the **Lens F-Stop** setting, but instead of changing the size of the in-focus area, it determines the amount of blur these pixels will receive. By entering a **Blur Size** value of 50%, the area out of focus, defined by the **Focal Distance** and **Lens F-Stop** setting, will receive only half the computed blur.

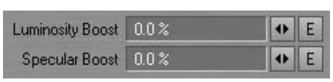


Digital Confusion's miscellaneous features

If a background image is used in your scene, you may want to activate **Blur Background** to blur it. However, because the background image is at an infinite distance from the camera, it will always receive the maximum blur amount and will result in much longer rendering times. A more efficient solution is to simply blur the background image in a paint program.

The **Render Hidden Geometry** feature forces LightWave to ray trace any geometry behind objects in case they go transparent when being blurred. This is a much more accurate representation of the depth of field effect, but can increase rendering times. In multi-pass rendering and compositing, it may be acceptable to not activate this feature, but normally it should be activated.

NOTE
Double-Sided geometry will not work correctly with the **Render Hidden Geometry** activated.



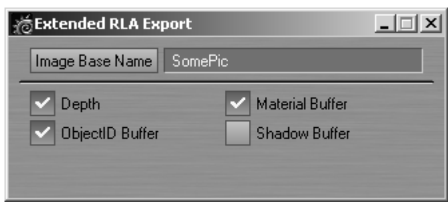
Fine-tuning surface properties

Sometimes defocusing the rendered image can cause the effect of surface specularity or luminosity to diminish unacceptably. To offset this effect you can adjust the **Luminosity Boost** or **Specular Boost** multipliers located at the bottom of the panel. Any pixels rendered with these surface properties will have their intensity adjusted accordingly.

Digitally confused cow

Extended RLA Export (Post-processing only)

This filter saves images in the *Extended* RLA format, popular for 2D/3D compositing work. The image includes optional depth buffers, as well as masks for which object or surface (material) a pixel came from. Enter the **Image Base Name** in the field or use the file requester button.



Full Precision Blur

This filter will soften an image by blurring it. Change the **Size** values to increase the amount of blur horizontally or vertically. The **Strength** settings determine the amount of the effect. You can also choose whether to affect the RGB (color) and/or alpha data.

You can use the **Rendered Alpha** (channel) or a **Special Buffer** as a mask using the **Use Mask** pop-up menu. If you want the mask computed prior to the blurring, check the **Pre-Blur Mask** option, otherwise the mask accounts for any blurring. To reverse the mask, check **Invert Mask**.

The **Special Buffer** setting on the Advanced tab of the Surface Editor can have a value from 0 to 1, with 0 meaning no blur and 1 meaning full blur.

Full Precision Gamma

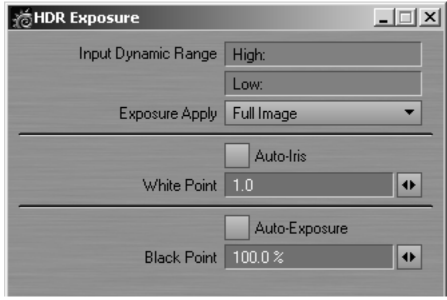
Display devices have a non-linear relationship between pixel values and physical light intensity—they do not excite the display phosphors linearly. This non-linearity must be compensated for to correctly reproduce intensity. The **Gamma Correction** value is the exponent value in the correction formula and determines how to convert pixel values to light intensity. The default, 2.2, is a common value used on images bound for video, but is not necessarily the optimum value.

HDR Exposure

This filter normalizes high dynamic range (HDR) images for using as image maps and HDR output for displaying in applications that are not HDR-savvy. The intensity mapping is non-linear, similar to the light response process in the human eye.

This filter processes the HDR output created by radiosity renders into better-looking, brighter pictures. It does this without impacting the accuracy of the lighting simulation, which can happen if you add ambient light or *crank up* lights unrealistically. It is really an essential part of the camera simulation, for a perfect digital camera. (The Virtual Darkroom filter is similar, but more complex. It simulates the two-stage process of film response to light, and print emulsion response to projection through the film negative.)

Although you can add this filter on the Image Editor, it is of limited use there and more useful as an Image filter on the Processing tab of the Effects panel. This is mainly because most images you load are not HDR images, so pre-processing is not necessary and normal gamma should probably be used, if necessary. Moreover, if you do load an HDR image, it's probably because you want the extra data. (Using the HDR Exposure filter will eliminate some, if not all, of the extra data.)



HDR Exposure

The **Input Dynamic Range** is an informational display showing the High and Low pixel-intensity values encountered in the last processed image. Note that when the panel first appears, this information is not yet known.

If you do not want the filter applied to the **Full Image**, set the **Exposure Apply** pop-up menu to **Foreground** to apply it only to scene items or **Background** to affect only the background (i.e., where your alpha would be black).

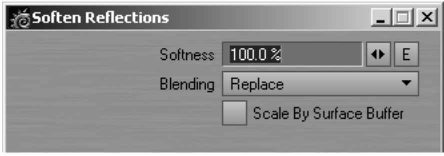
The **White Point** is the input intensity considered to be the hottest white. Anything above this will be the same white in the output. This control is overridden by the **Auto-Iris** option, which sets the white point based on the actual input image data. Adjusting the white point is similar to cranking down an iris on a film camera to limit how bright parts *blow out* in a photograph.

The **Black Point**, expressed as a percentage of the nominal black point (1/255), is the darkest non-black pixel level in the input image that will be preserved. Anything darker will come out black. The **Auto-Exposure** option overrides Black Point by using the actual image data to determine a black point in the incoming data. Lowering the black point is similar to increasing the exposure time for a photograph.

Once these values are set, the filter translates the incoming image intensity—in a process very similar to gamma correction—so that the darker colors get more of the output range than the brighter colors. In other words, the filter changes the spacing of intensity levels so more levels are devoted to low intensity, darker details.

Soften Reflections (Post-processing only)

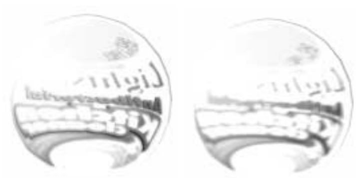
This filter will blur reflections. The **Blend** control will **Replace** the reflection with a blurred version, **Average** them together, which is more subtle, or use the **Maximum** of the replace result and the original, which avoids a dark halo, at the cost of a lighter image. You can also blend based on the **Alpha** channel or the intensity of the reflection (**LumaBlend**).



Soften Reflections

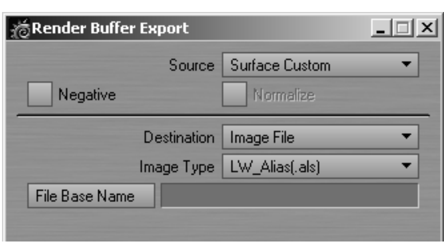
Soften Reflections can scale the blur based on the surface's value in Special Buffer 1 (Advanced tab of the Surface Editor), if you check the **Scale By Surface Buffer** option. (A value of 1 means 100 percent.)


Left to right: SoftReflections off, Replace, Average and Maximum


Left: Alpha. Right: LumaBlend

Render Buffer Export (Post-processing only)

This filter lets you save images from one of LightWave's internal buffers (**Source**). The **Surface Custom** option on the **Source** pop-up menu will create a grayscale image, where each object surface can have a unique grayscale value. This was designed to allow post-processing effects to be applied on a surface-by-surface basis. A surface's grayscale value (0-255) is assigned using the **Special Buffer** option on the Advanced tab of the Surface Editor.

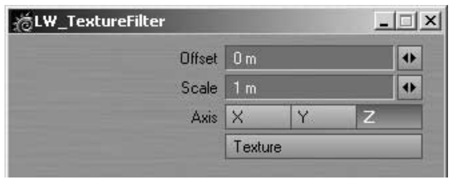


To invert the image data, check the **Negative** option. The **Normalize** option is available only for certain **Source** selections that don't normally provide image data, like **X Motion**. **Normalize** scales the values to be 0 to 1.

With the **Destination** pop-up menu, you can save the selected buffer image as a separate **Image File**, or replace the **Rendered RGB** or **Rendered Alpha** image data. (If you choose **Image File**, set the **Image Type** and enter the full path and filename in the **File Name** input field.)

TextureFilter

Use TextureFilter to add the selected texture to the image before use. You could use this filter to add, say, an animated Fractal Noise pattern to a simple black image. 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.)



To see procedurals in viewports:

You can use TextureFilter to see procedural textures in your Layout viewport! Basically, you apply the procedural texture(s) to the image using TextureFilter and then map the image to the surface. Here’s how you do it:

1 First, you need to load an image into the Image Editor. It really doesn’t matter what image you use since it will be obscured by the textures.

2 On the Image Editor’s Processing tab, add TextureFilter. Double-click it in the list window to access its options. Click the **Texture** button to access the Texture Editor.

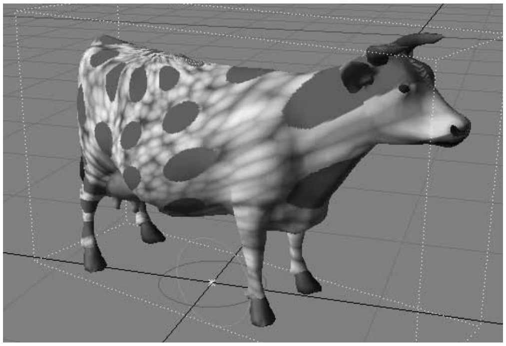
3 Change the default initial **Layer Type** to **Gradient**. This provides the underlying color. You can leave it white or change it.

4 Add one or more procedural texture layers and set as you would normally.

5 Load your object and open the Surface Editor.

6 Click the **Color** attribute’s **T**exture button. Leave **Layer Type** set to **Image Map** on the Texture Editor that appears.

7 Set **Projection** as you would normally and select your image from the **Image** pop-up menu. The procedural will now appear in your viewport.



Dots and Crumple textured on cow's hide

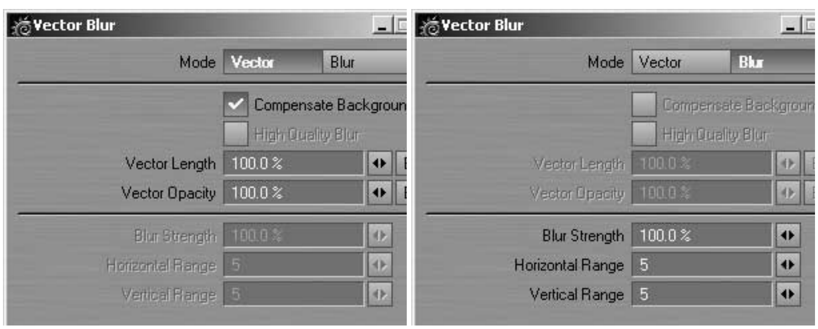
This operation requires a lot of computations and Layout may seem sluggish as it computes the texture.

NOTE
The image preview window on the Image Editor will show the texture as well. Thus, you can double-click the window to bring up the Image Viewer, from which you can save this compiled image. This can then be image mapped back onto the surface without all of the calculations required with TextureFilter.

If you use an image sequence instead of a still image, you can even see an animated texture! Note that if an animated texture is applied to a still image, it will not appear animated.

Vector Blur (Post-processing only)

LightWave's normal motion blur (Camera properties) may need long render times because of multi-pass antialiasing. Vector Blur, on the other hand, can achieve great-looking motion blur in a fraction of the render time.



The two modes, **Vector** and **Blur**, use motion data that identifies each pixel's immediate motion in the horizontal and vertical directions. The **Vector** mode smears the pixel out based on the motion information, while the **Blur** mode uses the horizontal and vertical data to *smush* together the surrounding pixels with that pixel. **Blur** affects the pixels around it including backgrounds, while **Vector** alters only the pixels of the moving object.

The **Blur** mode should be used in conjunction with the normal motion blur; however, the **Vector** mode can be used by itself—you don't even need to use antialiasing! The result can be drastically reduced rendering time.

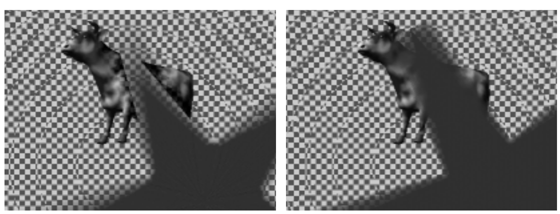
Below is a comparison between regular motion blur and the two modes of Vector Blur.

Left: Regular. Middle: Blur mode. Right:Vector mode

Overlapping Objects

Since objects become transparent with motion blur, the filter needs something in the background of the blur. When **Compensate Background** is active, the filter does <u>not</u> use the backdrop and attempts to compensate for this absence. This works in most cases, but may not give a very realistic motion blur.

When **Compensate Background** is not active, the filter uses the backdrop. However, it will not show an object behind another object.



Left: Compensate Background is off. Right: Compensate Background is on

If you have overlapping objects, you may want to do some test renders to see if Vector Blur will provide acceptable results. If not, use LightWave's normal motion blur.

High Quality Blur

If you uncheck **Compensate Background**, you can activate the **High Quality Blur** setting. This provides better quality, but takes longer to render. In this mode, you will only be able to set the **Vector Length** setting.

Limits

The important thing to understand about using Vector Blur is that it is a *post* process. As such, hidden geometry can't be blurred and you may see problems with motion blur on shadows and moving textures. However, it can be a great help when used in conjunction with normal motion blur, by giving you better quality with lower antialiasing settings (Camera panel).

Video Legalize

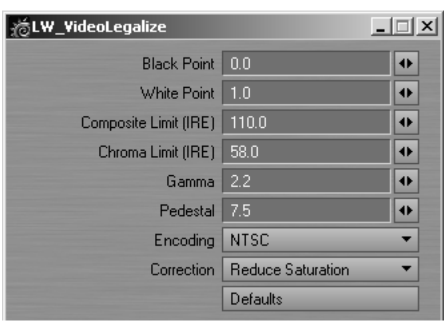
The Video Legalize filter might be more appropriately named *Hot Video*, since it assumes that pure black, RGB 0, 0, 0, is mapped to the proper *pedestal* by the encoding device (e.g., 7.5 IRE for NTSC). The encoding device may not, however, correct hot pixels—that is, colors that *exceed* certain video specifications. This is where VideoLegalize steps in.

Pixel values are *generally* scaled into IRE units using the **Black Point**, **White Point** and **Pedestal** settings as follows: Level = Pedestal + (100 - Pedestal) * (pixel - Black) / (White - Black). White is always 100 IRE and NTSC black is 7.5 IRE. Thus, for NTSC, the level would be computed: Level = 7.5 + 92.5 * (pixel - Black) / (White - Black).

Normally, an RGB level of 0.0 is black and 1.0 is white. If those are used for the **Black Point** and **White Point** settings, the level will be 7.5 + 92.5 * (pixel - 0.0) / (1.0 - 0.0) or just 7.5 + 92.5 * pixel. When the pixel is 1.0, the level is 100 IRE, and when the pixel is 0.0, the level is 7.5 IRE.

NOTE
Note that the actual computation is more complex than the above since other operations, like gamma correction, can happen.

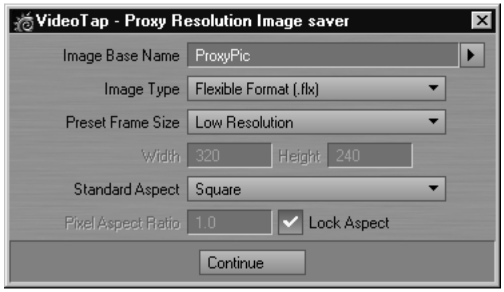
The settings default to **NTSC** encoding, but you may also select **PAL** from the **Encoding** pop-up menu. (Note that you also need to click the **Default** button after selecting a new **Encoding** item.) You may change the individual settings from their defaults, if desired.



The **Correct** pop-up menu determines how the image is corrected to fall within the specified limits.

NOTE
It is highly recommended that you use VideoLegalize (as a post-process image filter) if you plan to use your images for video.

Video Tap (Post-processing only)

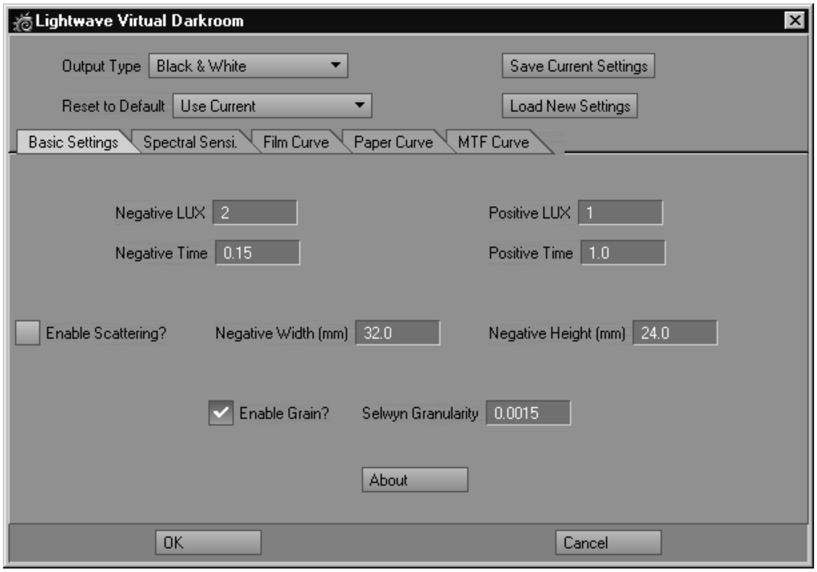


The Video Tap filter will save a second image using different camera and render option settings. This is perfect for times when you render a film resolution scene, but you want to preview a video resolution version on your Video Toaster or other digital disk recorder.

FUN FACTS
Movie-makers often attach a video camera to a film camera, so they can watch dailies without developing the film—saving time and money. This filter's namesake is the *video tap* on the film camera. (Clever idea by Brad. More clever name by Arnie.)

Virtual Darkroom

The Virtual Darkroom filter simulates the photographic capture of images. It is based on *A Model for Simulating the Photographic Development Process on Digital Images*, by Joe Geigel and F. Kenton Musgrave in the SIGGRAPH '97 conference proceedings.



Virtual Darkroom

Global settings are the four controls located at the top. **Output Type** specifies whether the output image is **Black and White** (single-plane grayscale) or **Color** (three-plane RGB). Use **Reset to Default** to reset all controls to default values. You can save and load settings using the **Save Current Settings** and **Load New Settings** buttons.

Basic Settings Tab

Basic Settings control everything except for those settings on other tabs. **Negative LUX** is the illumination value for the negative pass—analogous to scene capture with a camera—which will affect overall brightness. **Negative Time** is the exposure time for negative pass, essentially the exposure setting on the virtual camera. **Positive LUX** is the illumination value for the positive (printing) pass. Think of this as the brightness of the bulb in the enlarger or printing mechanism. **Positive Time** is the exposure time during the printing pass of virtual enlarger.

Enable Scattering will activate the internal scattering effect. **Negative Width** and **Negative Height** are the width and height, respectively, in millimeters, of the virtual negative. These values are used in scattering and grain calculations. **Enable Grain** will activate the grain effect. **Selwyn Granularity** controls the intensity of the grain. Increasing this value increases grain, decreasing it will decrease grain.

Spectral Sensitivity Tab

If **Output Type** is set to **Color**, there will be six sets of RGB percentage controls. Each RGB trio specifies the percentage of the black and white output plane that is contributed by the input image plane named in the control. For example, the RGB trio in the upper-middle defines how the output from the negative passes on the spectral sensitivity module creates the green plane.

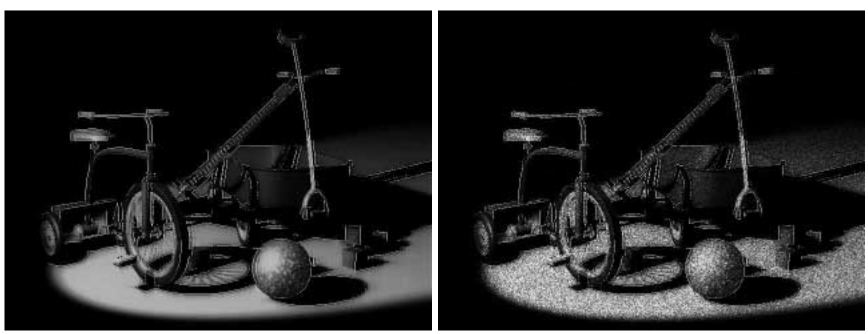
If **Output Type** is set to **Black & White**, you specify what percentage of the red, green, and blue planes of the input image are used when they are combined into a single black and white image. This transition takes place in the negative (first) pass. During the printing (second) pass, there is only a single input plane, so spectral sensitivity is not used.

Film, Paper, and MTF Curve Tabs

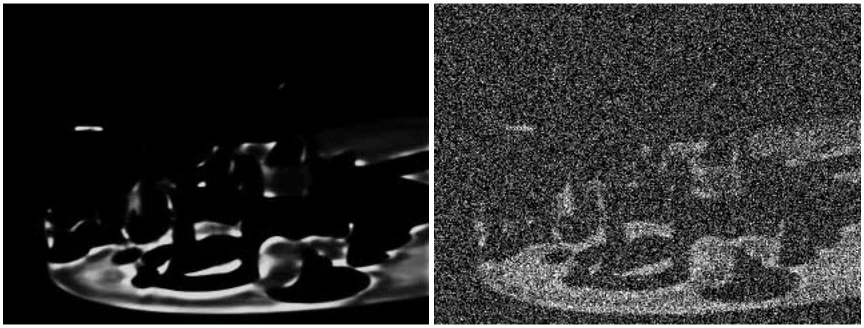
You can enter up to 21 pairs that define the characteristic curve of the film and paper, and the function for modulation transfer used for scattering calculations. For each pair, the first value represents log (exposure) and the second value represents the density.

For all Curve tab entries, points should be entered in order starting with Pt. 01. If **Output Type** is set to **Color**, curves must be set for each output plane by selecting the appropriate Red Plane, Green Plane, or Blue Plane sub-tab.

Original image.



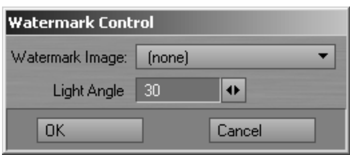
Left: Processing with **TMax 100/PolyMax** with scattering of a 3.2 mm x 2.4 mm. *Right*: Same with Grain enabled



Left: . *Right*: Processing with **TMax 100/PolyMax** with scattering of a .32 mm x .24 mm. *Right*: Same as above with Grain enabled

WaterMark

The WaterMark filter embosses an image in the lower-right quarter of the image. You can select any loaded image; however, grayscale images using a black background work best. The **Light Angle** field determines the direction of the implied light source.



NOTE
Other filters, like Pixel filters, may appear on the Image Editor in the list of image filters.

WAVEFILTERIMAGE

The WaveFilter Image image filter allows you to apply and adjust image filters, color filters, grain and matte filters to the entire image, the background, objects only, shadows only, a special buffer, or a user-definable color range. A powerful interface with color preview provides the control you need to tweak your image to perfection.

Image filters include blur, sharpen, edge blend, saturation, negative, high/low limit, palette, film grain, and flip frame. The edge blend filter provides additional and faster anti-aliasing control, allowing a possible 30% to 50% savings on rendering times.

Color filters controls for RGB values, as well as the contrast, midtone, gamma, luminosity and brightness of the image. The filters work on a percentage scale, but values above 100% can be entered to create all sorts of interesting effects.

Matte filters are used to create images for compositing. Portions of your images can be set to black or white for matte creation. If you do a lot of image compositing, you will find Matte filter to be an invaluable tool.

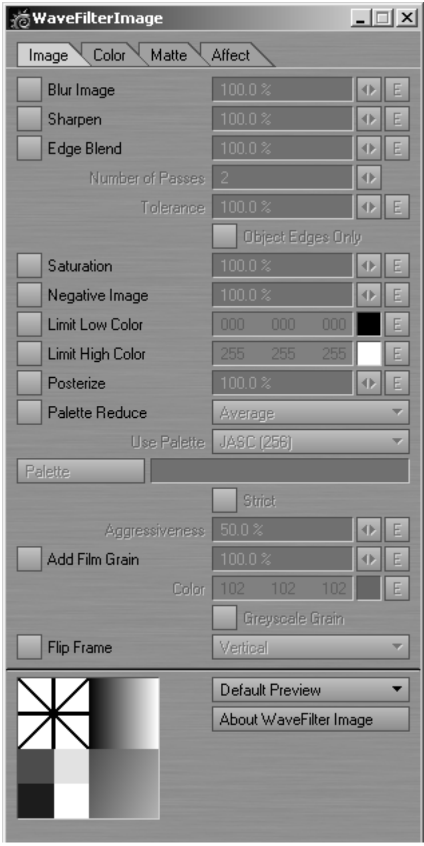
For even more control, WaveFilter Image can be added more than once to allow for multiple passes over an image, with each pass applying different settings to different parts of an image and creating different effects.

WaveFilter Image allows you to move many pre- and post-rendering operations directly into LightWave. It fully supports network rendering, animated settings and has batch processing capabilities.

NOTE
Special thanks to Mike Reed for his work on this plug-in.

Image Filters

Image Filters can be used individually or in combination with other filters to apply interesting effects. These filters use simple mathematical formulas for faster image rendering. Several adjustable presets are included for quick effects.



WaveFilter Image Image tab selected

Blur Image

Blur Image simulates out-of-focus effect and is applied equally across your image. This filter is commonly used on the background to help contrast your sharply focused objects in the foreground.

Sharpen

Sharpen is an edge-seeking (low-pass) detail-enhancing filter that will increase the sharpness between surface colors by exaggerating the difference in color between the two edges.

Edge Blend

Edge Blend is a softening tool that will smooth the edges between sharply contrasting surfaces. This helps to tone down the stair-step edges or jaggies found in computer graphics.

You may control the **Number of Passes** this filter makes. Higher values provide finer detailed blending of the edges. **Tolerance** allows you control the amount of edge affected. You may also limit the edge detection to **Object Edges Only**.

Multi-Pass Edge Blend

Some images with single-pixel details, like stars or highly detailed objects, lose desired details when the Edge Blend filter is set strong enough to remove unwanted jaggies. Increasing the **Number of Passes** can help to minimize this loss of detail. (Note that normally only a single pass is needed.)

When you select more than one pass, the filter searches for edges and applies the blend at a power equal to the percent you select divided by the **Number of Passes**. For example, if are using four passes at 100%, the filter will apply a blend of only 25% on each pass.

Once the first pass is completed, the edges are detected again, but this time fewer edges will be found—edges that needed only the light 25% blend will not be selected. This process is repeated for all passes. This results in the pixels that needed the least blending only getting a 25% blend, while those that needed a bit more get 50%, and so on. Only the worst jaggies get the full 100% blend.

Adding WaveFilter Image plug-in multiple times can also help in removing really bad jaggies. Just like using multiple **Number of Passes**, each instance of the filter will only select the jaggies that remain after processing in any previous instance. However, remember that the blend percentage will be applied fully for each instance, so you may want to keep the value as low as possible to avoid overly blurring the edges.

Using on Rendered images

If you render a large number of frames and find you should have used a higher antialiasing setting, you can simply apply the Edge Blend filter to them—WaveFilter Image does not require geometry to work. This could save you from re-rendering the original scene.

To do this, simply load the images as an image sequence and set the sequence as the background image. Then, set the camera to exactly the same resolution as used for the original images and turn off any normal antialiasing. Set the images to render to a new filename and you're ready to post process.

Saturation

Saturation allows you to control the saturation of the image's color.

Negative

Negative gives control of the amount of negative applied to an image. 50% negative is equivalent to RGB 128/128/128.

Limit High/Low Color

The **Limit High Color** and **Limit Low Color** settings limits the colors in your image to levels which are NTSC compliant. Only highlights are affected.

Posterize

The **Posterize** filter reduces the colors used causing a poster-like effect. Lower values allow more colors.

Palette Reduce

The **Palette Reduce** filter limits the colors in your image to smaller, evenly distributed, color palettes. Palette reduction method can be specified as a preference by choosing either **Average** (the default), **NTSC/PAL Luminance**, **HDTV Luminance**, **Nearest Red**, **Nearest Green**, or **Nearest Blue**.

Several palettes are available on the **Use Palette** pop-up menu. If **Custom** is selected, the **Palette** button and input field will be available. Here you can specify a Photoshop *.act file or an ASCII file that defines each color in the desired palette.

Film Grain

Check **Add Film Grain** to add a random noise effect. of selectable density to your image. The **Color** setting defines the grain color. Checking **Greyscale Grain** causes the grain to be various levels of gray.

Flip Frame

Flip Frame provides a quick method to vertically and/or horizontally flip the image.

Color Filters

Color filters allow global control of general color attributes.

WaveFilter Image Color tab selected

Contrast

The **Contrast** value sets the mid-tones of the selected part of the image either towards the highlight or shadow range. The **Center** setting controls the contrast tonal range.

MidPoint

The **MidPoint** value is similar to the **Brightness** setting, except it affects only the mid-tones within the selected part of the image. Use in conjunction with **Brightness** for a good overall lighting effect. The Center setting controls the **MidPoint** tonal range.

Gamma

Gamma allows control of the range between the darkest and lightest areas of your image.

Luminance

Luminance adds overall brightness or darkness to the frame. Shadows, mid-tones, and highlights will all be affected.

Brightness

Brightness multiplies the amount of light within the selected part of the image. Mid-tones and highlights are affected, while dark shadows are not affected. Use **Brightness** to adjust the lighting within your image without resetting the scene's lights or raising the ambient light level.

Adjust Color

The **Adjust Color** options provide different options for adjusting colors.

Mult: Pixel x Percentage looks at the color information in each RGB channel and multiplies the base color by the selected **Color**.

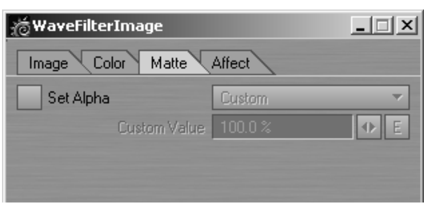
Add: Pixel + Color adjusts the color information by increasing the amount of R, G and B color in the image.

Sub: Pixel - Color adjusts the color information by decreasing the amount of R, G and B color in the image.

Replace: Pixel = Color replace will change the RGB color from the current base color to the new selected **Color**.

Matte Filters

The Matte tab contains options that affect the image's alpha channel. You can see the results in the Image Viewer when you render a frame (F9) by simply selecting **Alpha** from the pop-up menu in the upper-right corner.



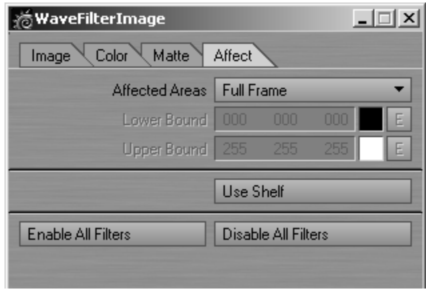
WaveFilter Image Matte tab selected

Choosing **Custom** will set the alpha channel for the affected area to a grayscale value defined in the **Custom Value** input field.

Choosing **All Black** sets the alpha channel for the affected area to black. Choosing **All White** sets the alpha channel for the affected area to white. Generally, with these, you will want **Affected Areas** (Affect tab) set to something other than **Full Frame**.

Invert reverses the alpha channel for the affected area.

The Affect Tab



WaveFilter Image Affect tab selected

Affected Areas

WaveFilter can apply most filters to a scene's images, surfaces, channels or objects. Often, when multiple WaveFilter passes are installed, you will select a different portion of the image to effect in each pass.

Objects applies the effect to all objects, including an image mapped polygon background.

Background applies the effect to the background only.

Shadows applies the effect to the shadow channel only.

Full Frame applies the effect to the entire frame.

High/Low Color Range applies the effect to a selected color range in all three color channels (RGB). The range is defined by the **Lower Bound** and **Upper Bound** colors.

Selected Surfaces applies the effect to *selected* surfaces. Surfaces are considered selected if they have a Special Buffer value greater than zero. To access, click the **Special Buffers** button on the Advanced tab of the Surface Editor.

Preset Shelf

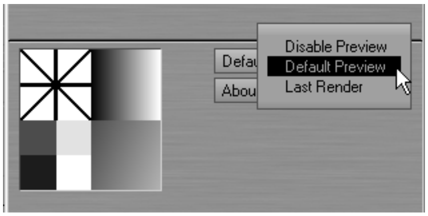
Click Use Shelf to access the standard Preset shelf to save and load presets. You can save WaveFilter settings to the shelf by double-clicking on the preview image.

Enable/Disable

You can quickly enable and disable all filters by clicking the **Enable All Filters** and **Disable All Filters** buttons.

Preview Window

The preview window will give you an idea of what the current settings will do. This can take a little time to compute so you can disable the preview. You may also use the last rendered image instead of the default image.



WaveFilter Image preview window and settings

chapter 15
Volumetrics

Chapter 15: Volumetrics

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.

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.

NOTE
Volumetric light effects, like a cone of light from a flashlight, are discussed in Chapter 32.

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 behavior 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 vaporized 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 behavior 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 gray areas and a thin white border. The dark gray 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

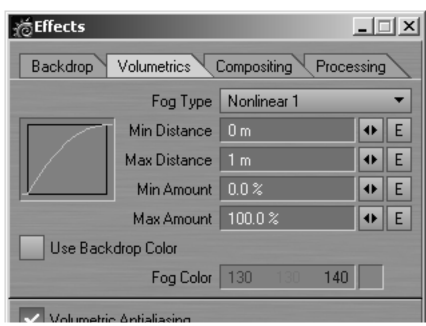
When you work with particles, a sphere of gas is associated with each particle. As a result, a particle cloud is really just a union of spheres. Computing the effect for every particle can be computationally intensive, particularly when their spheres overlap one another. The solution is the automatic particle sizing option, which evaluates a particle size so that each particle is close to another. The result is a dense cloud optimized 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 (**Scene** > Effects: **Volumetrics**). 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**.**



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 GroundFog 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, which 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 grayish white for a foggy day.

If you add the Texture Environment environment (**Scene** > Effects: **Backdrop**) 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.

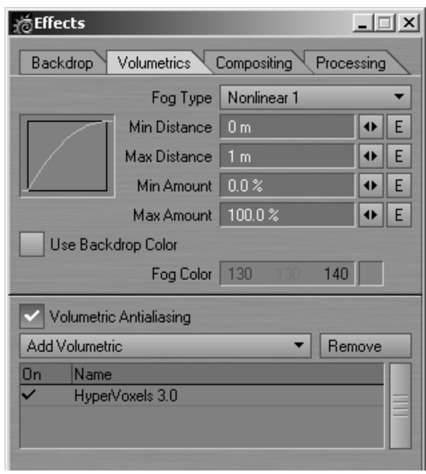
NOTE
Also see the discussion on the **Unaffected by Fog** option in Chapter 9 and the special display options for fog in Chapter 6.

VOLUMETRIC ANTIALIASING

The **Volumetric Antialiasing** option activates or deactivates the antialiasing of volumetric plug-ins *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.

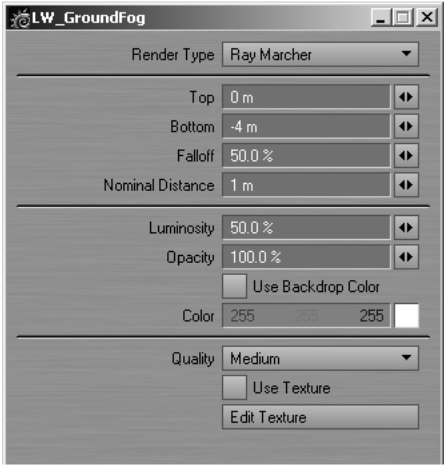
VOLUMETRIC PLUG-INS

Volumetric Plug-ins are added on the Volumetrics tab of the Effects panel.



GROUNDFOG

Use the GroundFog volumetric plug-in 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.



GroundFog panel

With the **Render Type** pop-up menu, you can choose between two different types of GroundFog. **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.

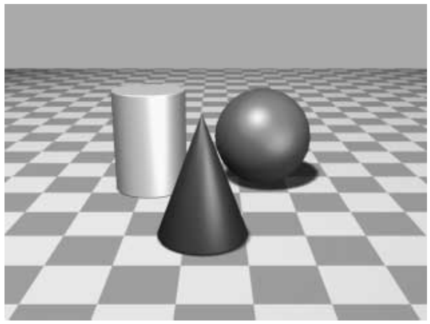
NOTE

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.

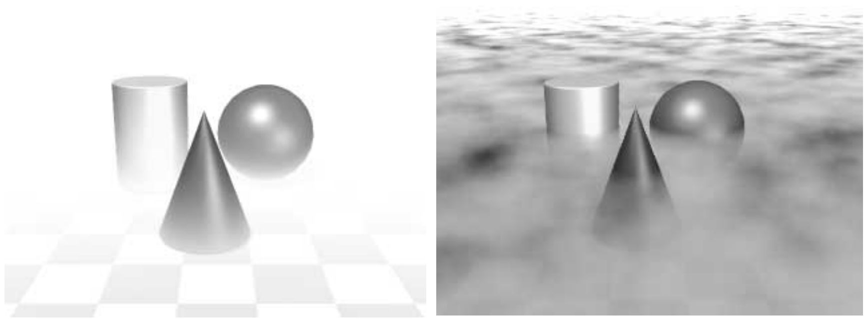
The **Luminosity** and **Opacity** values are the values where the fog is at its thickest.

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.



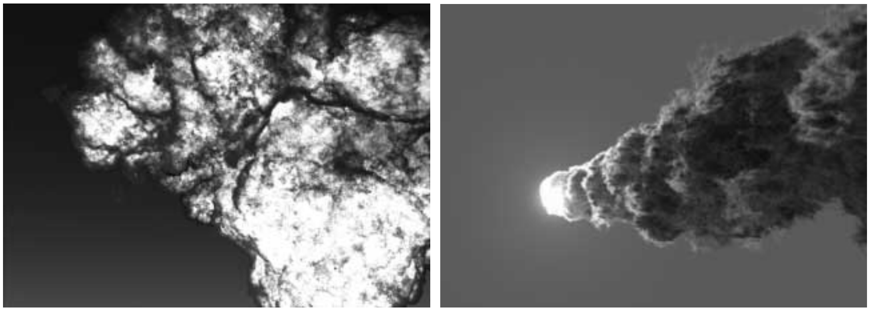
Without fog



Left: Fast Fog. *Right*: Ray Marcher fog using a texture

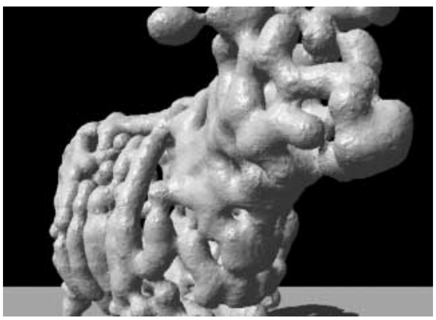
HYPERVOXELS

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, nebulas, contrails, fluids, smoke, ash, pyroclastics, gelatin, electromicroscopic images, rusted materials, detailed solid and rocky surfaces, and much, much more.



Example HyperVoxels

HyperVoxels have computed mass, as opposed to the modeled mass of normal LightWave objects. This opens the door to effects an order of magnitude greater in 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.



HyperVoxel cow

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 quicker, but lacks much of the volume mode’s 3-D quality.

HyperVoxel fog

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 3-D 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, plug-ins 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

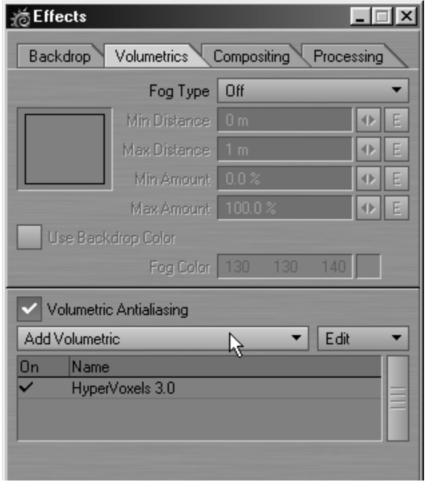
Here are some short exercises to jump-start you with HyperVoxels—more complex tutorials are included later.

Exercise: HyperVoxels basics

This exercise gives you a quick hands-on feel for how to use HyperVoxels.

1 Add a null object to an empty Scene in Layout.

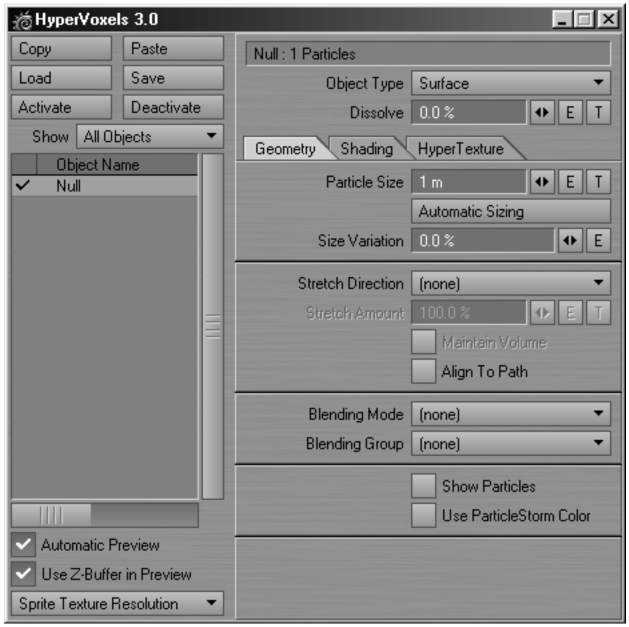
2 Add the HyperVoxelsFilter on the Volumetrics tab of the Effects panel (**Scene** > Effects: **Volumetrics**).



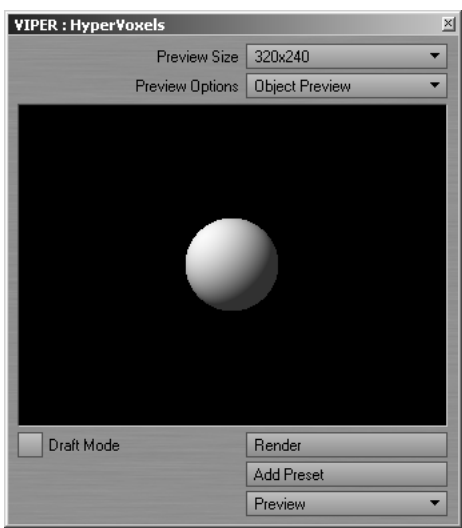
3 Double-click its name in the list to bring up its settings panel.

4 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.)

5 **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.

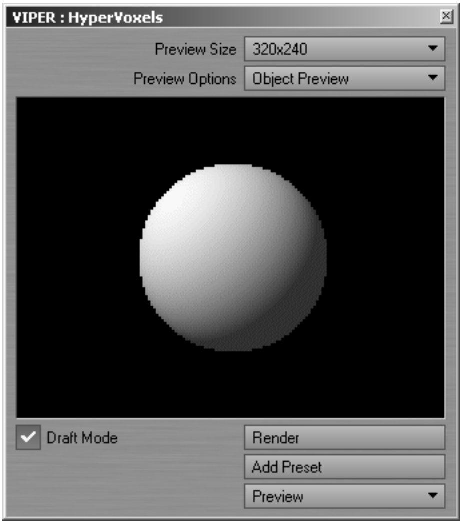


6 Make sure the **Automatic Preview** options are active so we can see the results of your efforts. **Automatic Preview** updates the preview as you make changes, so you don't have to keep clicking **Render**. Open VIPER (the button is on the main Layout toolbar). You should already see a rather dull ball in the VIPER window—of course it's a lot more interesting than a rendered null.

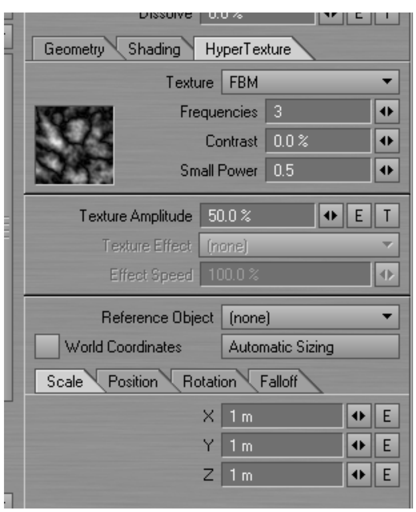


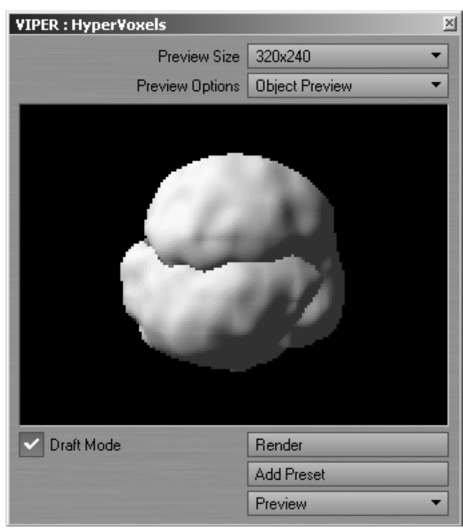
NOTE
You can preview animated textures with VIPER. See Chapter 3 for more information. Remember that VIPER uses the camera view, so make sure the camera can see your particles.

7 Change the **Particle size** to 2. The ball should be larger now. (Would you believe twice as big?) Activate the **Draft Mode** to speed VIPER up.



8 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.





9 Go ahead and play around with the various settings on this tab and watch the HyperVoxel change.

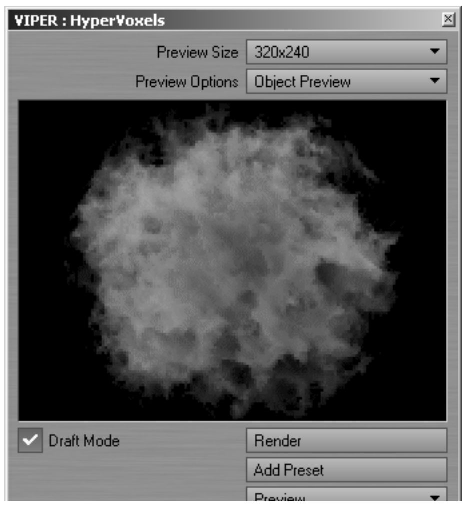
10 You can actually apply all of the standard surface attributes to HyperVoxels on the Shading tab. All of the settings should be familiar to you since they are identical in name and operation to the normal LightWave surface settings. Try changing the various values and see their effects on VIPER.

Exercise: HyperVoxel volumetrics

This exercise will give you a flavor for some of the volumetric features.

1 Using the HyperVoxel from the previous exercise, change the **Object Type** from **Surface** to **Volume**.

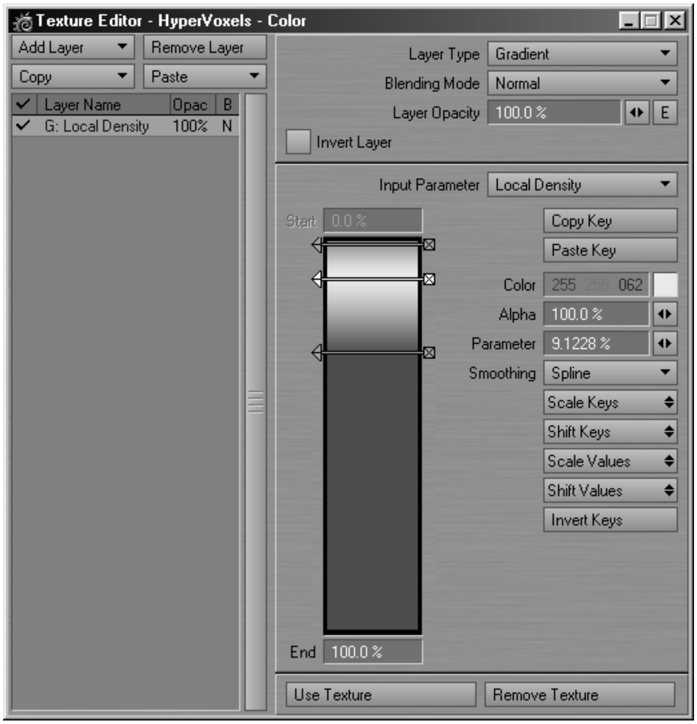
2 Go to the HyperTexture tab and select **Turbulence** as the **Texture**. You should now see a nice wispy volumetric cloud. This object is totally three dimensional. You could fly the camera into it and continue to see all of the details. How cool is that?!



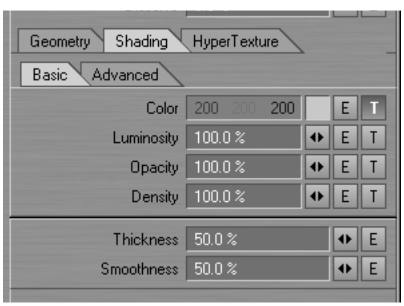
3 Let’s add some color. Go to Shading tab > Basic tab and click the **Color** Texture button.

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.

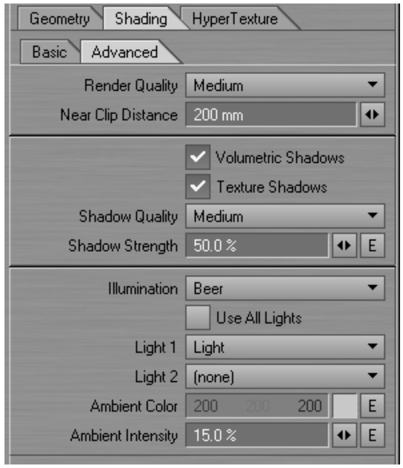
5 Create three keys: blue, yellow, and red, top to bottom. You will see VIPER updating the preview as you make changes. Click **Use Texture** to close the panel when you are done.



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.



7 Click on the Advanced subtab. 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.



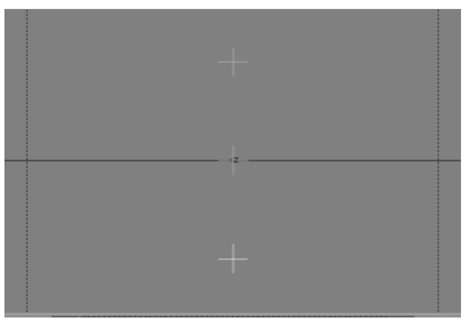
8 Try messin' with the various setting, including the Sprite mode.

Exercise: blending HyperVoxel objects

This exercise shows how different HyperVoxel objects can interact with one another.

1 Add three null objects to an empty Scene in Layout.

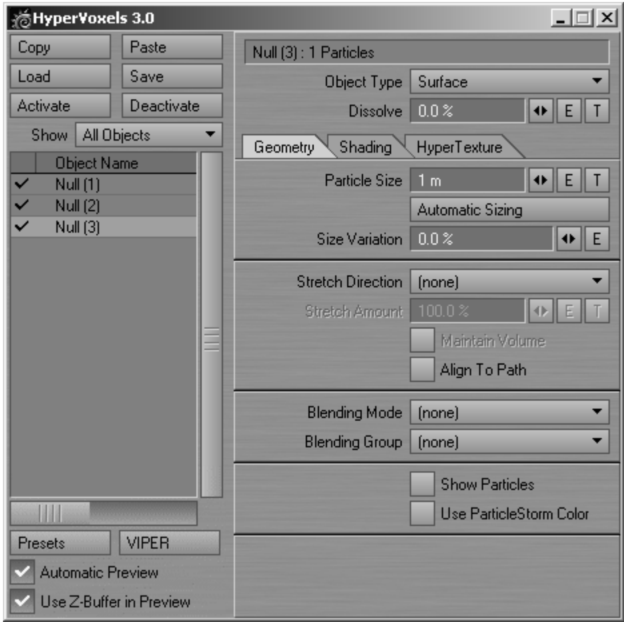
2 Move and keyframe Null (1) to Y = 1m and Null (3) to Y = -1m.



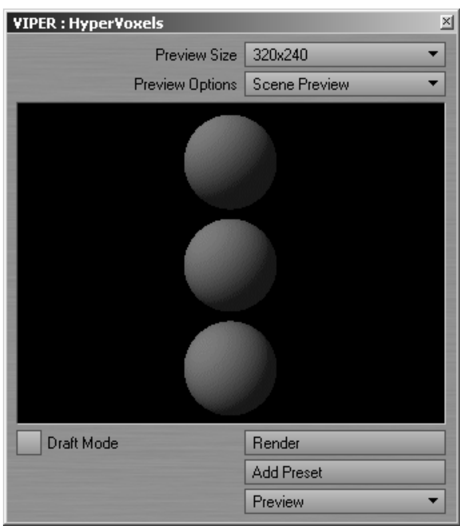
Camera view of three nulls

3 Choose **Scene** > Effects: **Volumetrics** and add the HyperVoxelsFilter and open its panel.

4 Activate HyperVoxels for all three objects.

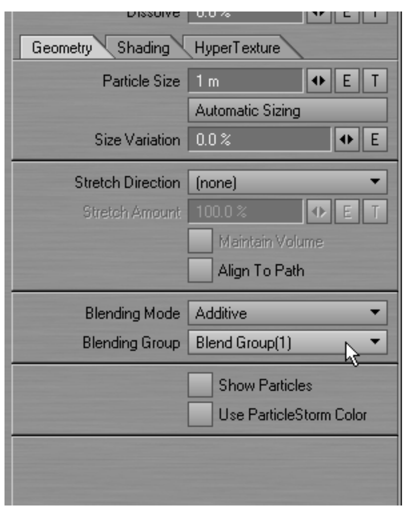


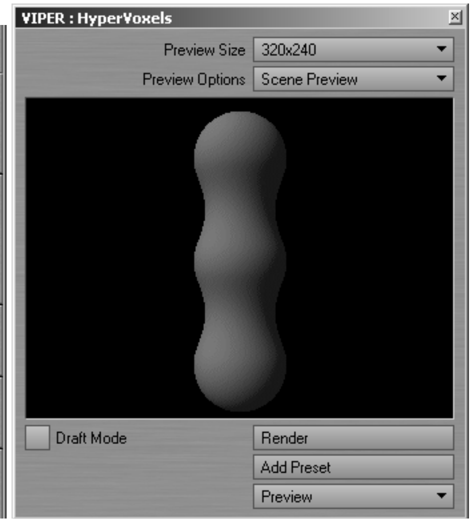
5 Open VIPER and set **Preview Options** to **Scene Preview**. This will give us a preview of all of the HyperVoxel objects in the scene.



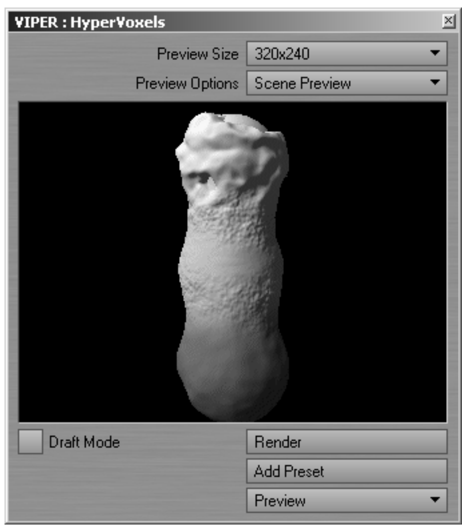
6 Select Null (1) and choose **New Group** from the **Blending Group** pop-up menu on Geometry tab. Accept the default name Blend Group (1) by clicking **OK**.

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.





8 Try changing the **Color** (Shading tab > Basic tab) of each null and giving them **Texture** (HyperTexture tab).



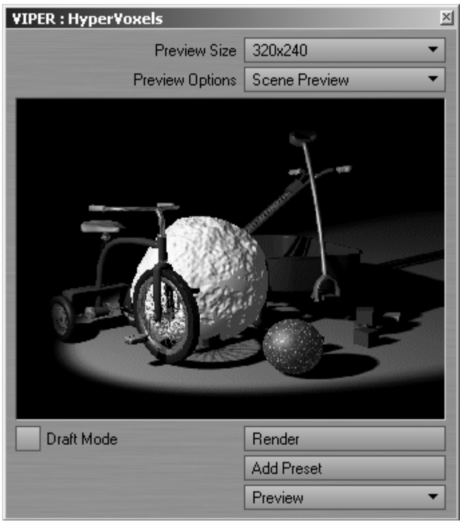
Notice how the texture surfaces blend together.

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 HyperVoxel objects 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.



The Use Z-Buffer in Preview option in action

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 allows you to retrieve and store HyperVoxels setting files. This can be used to create a library of attributes.

HyperVoxels Setting Management

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.

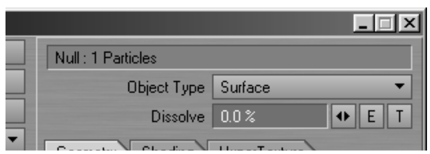
You can enable/disable HyperVoxels for an activated object by clicking in the checkmark column and you will not lose your settings. The **Activate** button will activate the selected object and set all settings to their defaults. The **Deactivate** button places the object in an inactive default state. 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.

Object Type

The **Object Type** pop-up menu lets you choose between the previously demonstrated **Surface**, **Volume**, and **Sprite** HyperVoxel modes. Use it for testing purposes when you are developing your HyperVoxel animation.

Dissolve lets you individually set an opacity value for each active HyperVoxel object: 0% is totally opaque and 100% is completely transparent.

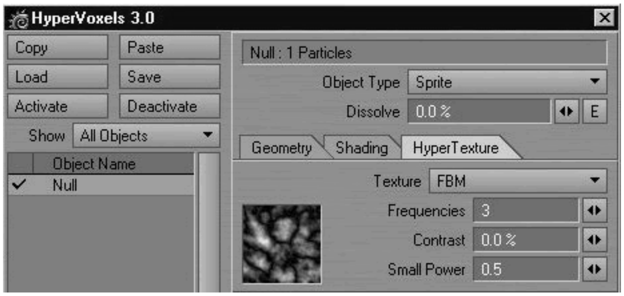


Object Type pop-up menu

Sprites

The **Sprite** mode is a simplified two-dimensional slice (or slices) of a HyperVoxel, so it can quickly produce results to give the user 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 3-D volumetric effect, like a smokestack in the distance.



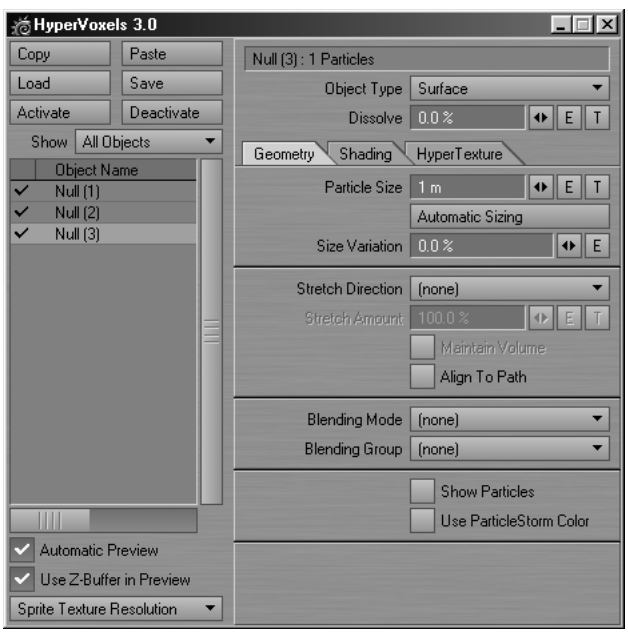
Sprite with hypertexture

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.



The Geometry tab

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 fewest 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.

Sprite stretched using Velocity

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

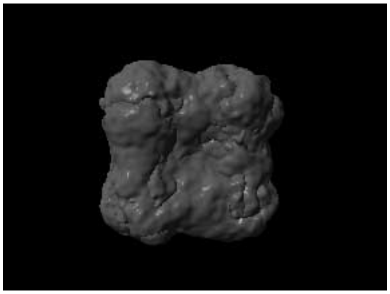
Align to Path causes the HyperVoxels to rotate as they follow a path. This can be particularly important when using particles.

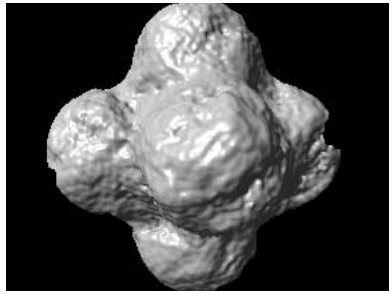
Blending HyperVoxels

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**

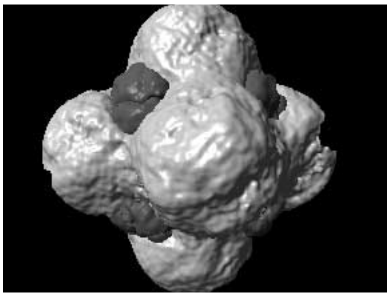
Here, the two objects are shown separately





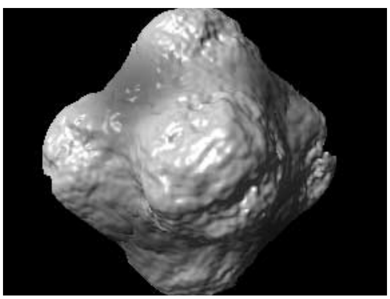
Left: Object 1. *Right*: Object 2

(none) means it will not interact with other HyperVoxel items.



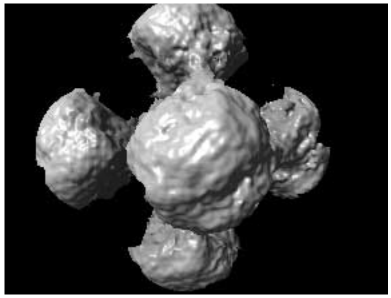
no blending

Additive will smoothly blend HyperVoxels.



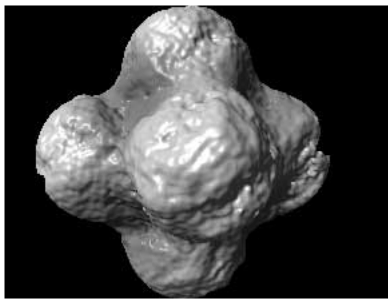
Additive on both objects

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.



Negative on object 1 and **Additive** on object 2

Effector is kind of a cross between **Additive** and **Negative**.

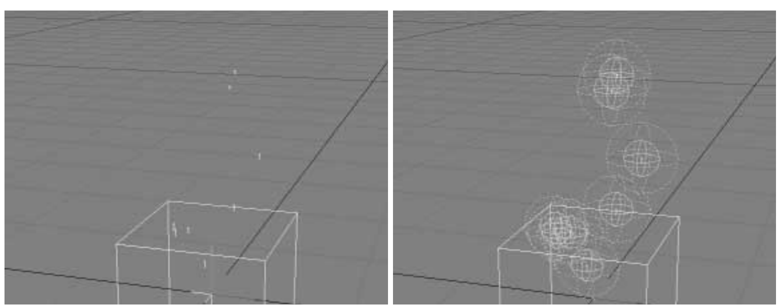


Effector on object 1 and **Additive** on object 2

A **Blending Group** is the group of HyperVoxel objects that 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.



Left: Show Particles off. *Right*: Show Particles on

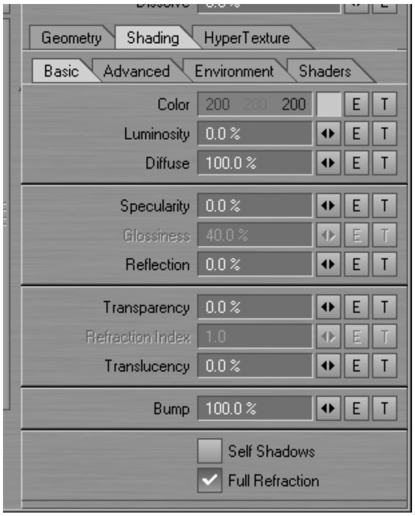
Use ParticleStorm Color

The HyperVoxels will use the color from ParticleStorm particles, if available.

NOTE
The HyperVoxelsParticles Displacement plug-in 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. (See Chapter 31 for more information.)



Shading Tab: Surface Mode

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 the **Color**, **Luminosity**, **Opacity**, and **Density** settings to change those parameters for the volumetric object.

Shading Tab:Volume Mode

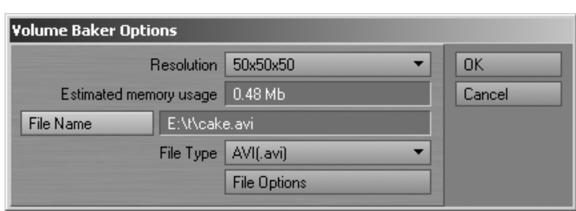
The **Thickness** setting, which defaults to 50%, lets you adjust the general *heaviness* of the volume. Higher **Smoothing** values yield less of a hard-edged look and volumes will appear more wispy.

Baking HyperVoxels

The baking feature subdivides HyperVoxels into a 3-D 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.)



Baking options dialog

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.

Volume Mode Advanced sub-tab

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.

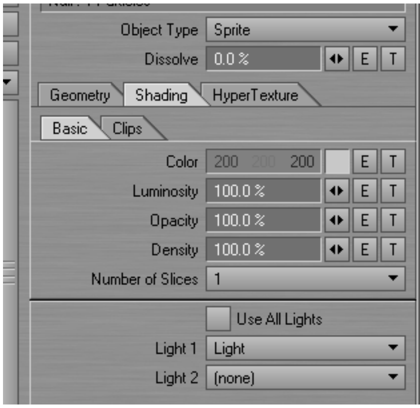
Ambient Color lets you specify the ambient light color of the volume. This setting works hand-in-hand with the **Color** setting on the Shader tab. **Ambient Color** is similar to ambient light in Layout, except it applies only to the selected HyperVoxel object. Basically, the shading process is very 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**.



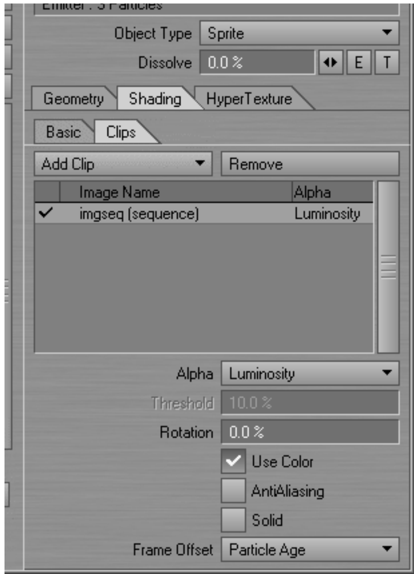
Shading Tab: Sprite Mode

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 to Right: 1-slice sprite, 5-slice sprite, and Volume mode

You may choose to light the Sprite using all lights in your scene or up to two specific lights.

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 subtab

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 grayscale 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.

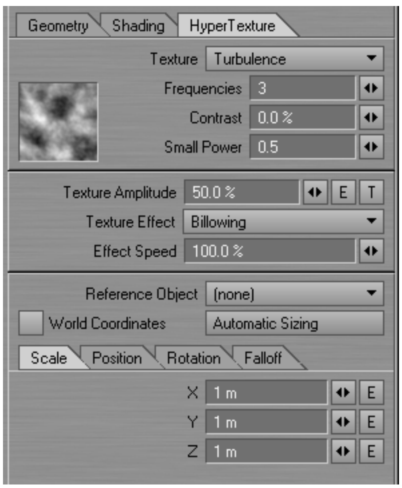
Sprite Clip Frame Offset

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.

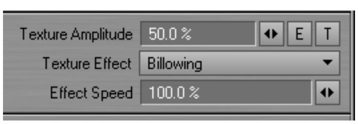
HyperTexture Tab

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. (See Chapter 31 for more information.)



HyperTexture Tab

The big difference is when you are using a **Volume** (**Object Type**) HyperVoxel. In this case, you can animate the texture effect.



Texture animation settings

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.

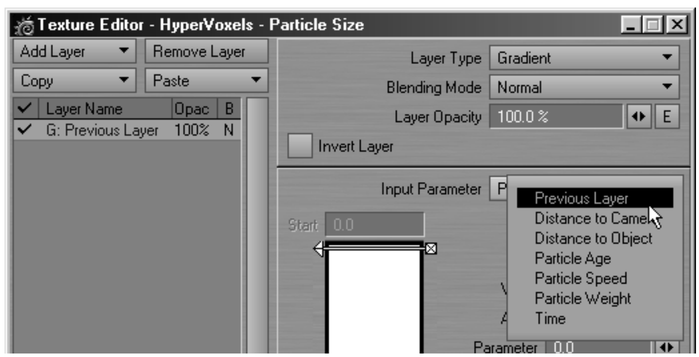
Velocity Translate moves the texture in the direction of the particle velocity. Since each particle has a different velocity, this can result in a cool and complex texture motion. **Dissolve & Expand** might be used on explosions. It dissolves the effect as it expands.Some trial and error may be necessary 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. See Chapter 3 for more information.

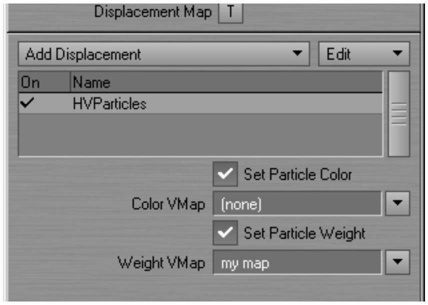
Gradient Input Parameters

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.



Gradient Input Parameters

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 HyperVoxelsParticles Displacement plug-in added. **Set Particle Weight** should be checked and the weight map selected.



Object Properties panel

To use the Particle Weight Input Parameter:

1 Create an object in Modeler.

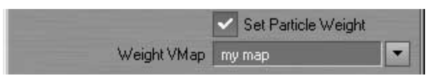
2 Create a weight map and add some random weighting to the points.

3 Load the object into Layout.

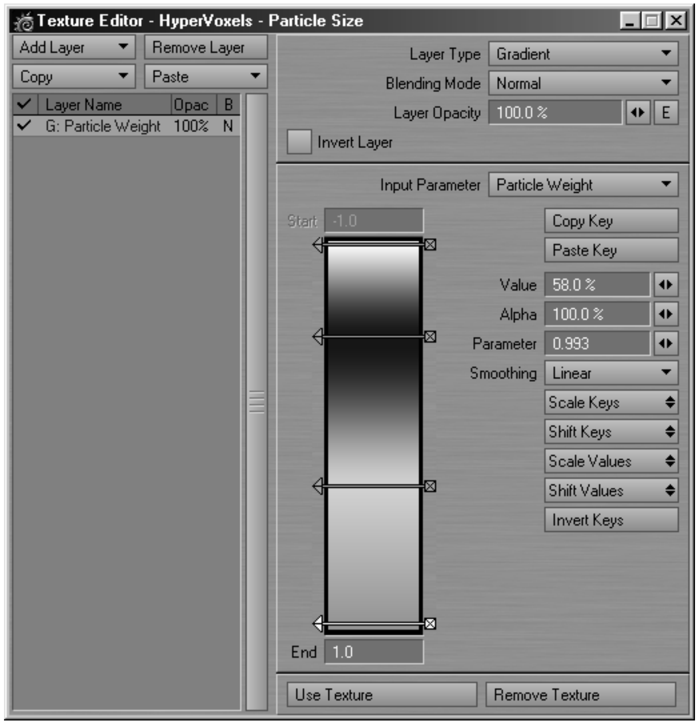
4 Choose **Scene** > Effects: **Volumetrics** and add the **HyperVoxelsFilter**.

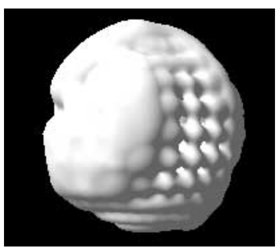
5 Open the HyperVoxels panel and activate the object.

6 Open the object's Properties panel, add the **HyperVoxelsParticles** Displacement plug-in. Open its options, activate **Set Particle Weight** and choose the weight map.



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.





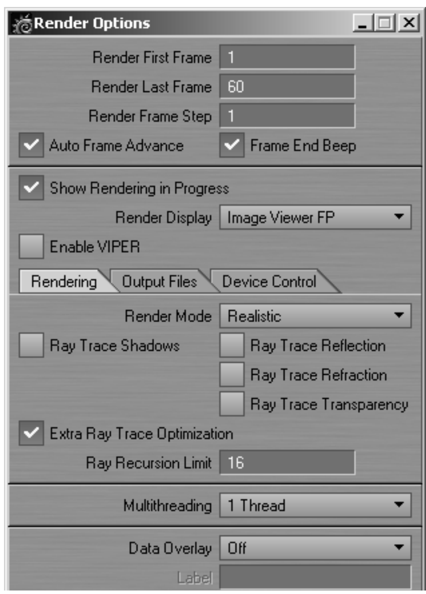
The ball.Your results will vary depending on your weight map

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.

chapter 16
Rendering Options

Chapter 16: Rendering Options

The Render Options panel (**Rendering** > **Render Options**) contains all of the settings that control your output. Generally speaking, it is here that you set how realistic you want your images to look. They can range from wireframe test versions to full ray-traced final images. You will also specify the file format for your output, ranging from single frames to an animation.



Render Options panel

RENDER FRAMES

The **Render First Frame**, **Render Last Frame**, **Render Frame Step** fields determine the range of frames you want to render as well as the frame increment. These are independent from the similar fields in the main interface, which control the playback of scene and Preview Animation creation.

AUTOMATIC ADVANCE

By default, LightWave will render only one frame at a time, even if you have set a range of frames to render. After each frame is rendered, you can click the **Continue** button on the Render Status panel (or press the ENTER key) to continue rendering the next frame in the range.

This would likely get tedious in a 900 frame animation, so selecting **Automatic Frame Advance** will continuously render all of the frames in the defined range without stopping after each frame.

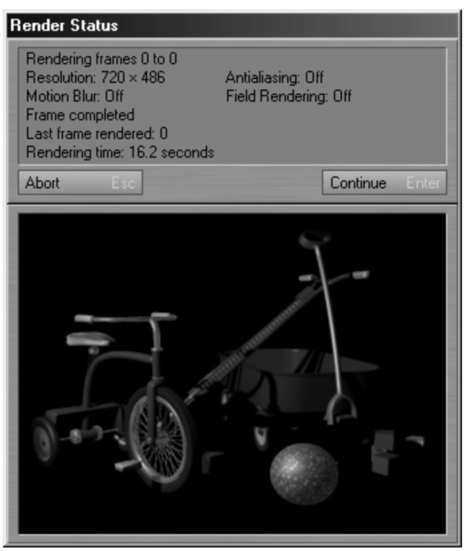
WARNING
When rendering, make sure you check the state of **Automatic Frame Advance** before you grab some shut-eye!

RENDER COMPLETE NOTIFICATION

By selecting **Frame End Beep**, you can instruct LightWave to have your computer beep whenever a frame has been rendered or a wireframe preview has been generated. This can be a handy feature to inform you that a process is finished. The status of **Frame End Beep** is saved as a default when you quit LightWave.

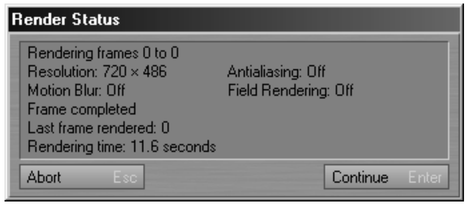
MONITORING PROGRESS

A watched pot does indeed eventually boil: when you activate the **Show Rendering in Progress** option, you add a preview window to the Render Status panel that displays the image as it is generated. The number of colors used in the window is determined by your computer's display setting. Of course, this is only a representation of the final image, which will be rendered in full color at the proper resolution.



NOTE
Using the **Show Rendering in Progress** option adds to rendering times. As such, this feature is best used for test rendering. You should deactivate it for the final rendering process.

If the **Show Rendering in Progress** option is not active, a similar dialog will appear, sans preview image.



You can close the Render Status window by clicking the **Abort** button or pressing the ESC key. You can also render the next frame (if you aren't using **Auto Frame Advance**) by clicking **Continue** or pressing ENTER.

VIEWING THE FINISHED IMAGE

Once LightWave finishes rendering a frame, the image is displayed using the selected **Render Display**. The **Image Viewer** option is described in Chapter 3.

NOTE
The FP (floating point) version of the Image viewer maintains the floating point image data. Use it if you want to save files from the viewer and need the FP data. Note that it uses more memory than the regular Image Viewer.

VIPER

VIPER (see Chapter 3) can use information from the internal render buffers to perform tasks like surface previews. When used in this manner, you must activate the **Enable VIPER** option. However, when you render you should disable this option, since it increases rendering time and memory consumption.

RENDERING

To render the current frame—determined by the position of the frame slider—choose **Rendering** > **Render Current Frame** or press F9. Select **Render Scene** to render the entire scene using the defined range of frames on the Render Options panel, or press F10. Make sure **Auto Frame Advance** is set as desired!

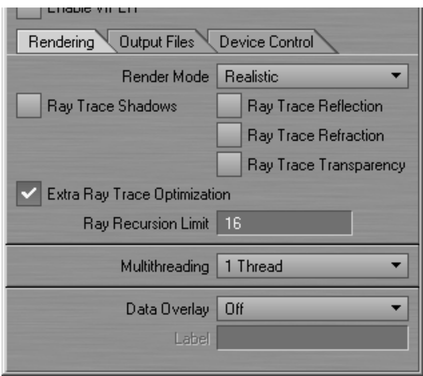
NOTE
Colors on your computer monitor will not match actual NTSC/PAL colors. If possible, always test frames from LightWave on a video monitor prior to final rendering.

You can also render only the selected object(s) at the current frame by choosing **Rendering** > **Render Selected Objects**. Unselected objects can still cast shadows or be seen in reflections on the rendered objects.

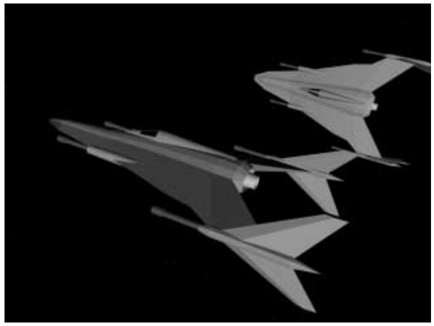
NOTE
The Render Status window is a modal window. Thus, you may be precluded from accessing options on other windows until it is closed.

RENDERING TAB

Depending upon your needs, LightWave can produce different types of output. Select a **Rendering Mode** to output images best suited for your project.

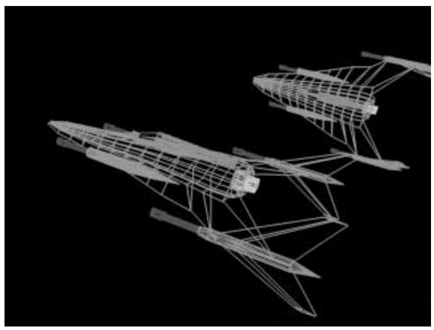


Selecting **Quickshade** will render the objects with no smoothing, textures, transparency settings, or object dissolve levels. It gives a better representation of the camera's view than **Wireframe** and can be handy for checking out rough lighting and object placement. Because of its speed, using **Quickshade** is a good idea for testing out CPU-intensive features such as Motion Blur and Depth of Field (Camera Properties panel).



Quickshade mode

Selecting **Wireframe** will produce wireframe representations of all of the objects in your scene. This will show all polygons of an object in the color chosen for **Surface Color** on the Surface Editor's Basic tab. The advantage of **Wireframe** is that it renders quickly and is ideal for showing the motions of objects in your scene without any of the detail.



Wireframe mode

HINT

To produce a solid wireframe animation, Use the **Unshared Edges**, **Surface Borders**, and **Other Edges** options on the Object Properties panel's Edges tab, and render in **Quickshade** mode.

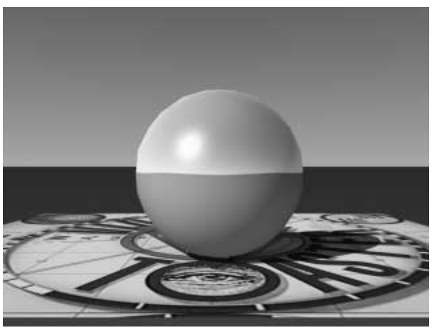
Realistic is the mode you will use most for final rendering. It renders images with full photo-realistic quality. Plus, you can use all ray tracing functions.

Realistic mode

Ray Tracing Options

When **Realistic** is selected as the **Render Mode**, you can also select any of LightWave's ray tracing functions. Ray tracing lets you render realistic shadows, reflections, and refractions.

Ray Trace Shadows causes illuminated objects to cast ray-traced shadows onto themselves or other objects.

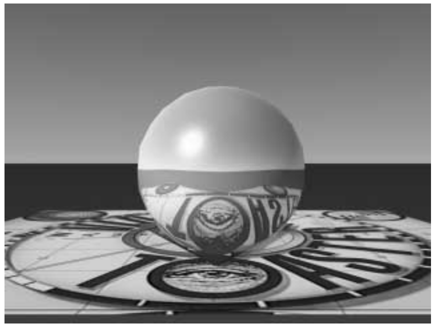


Ray Trace Shadows

NOTE
Lights must have **Ray Trace** selected as their **Shadow Type** (Light Properties panel).

HINT
Judicious use of the **Self Shadow**, **Cast Shadow**, and **Receive Shadow** options (Object Properties panel, Rendering tab) for individual objects can speed up rendering times dramatically when ray tracing shadows.

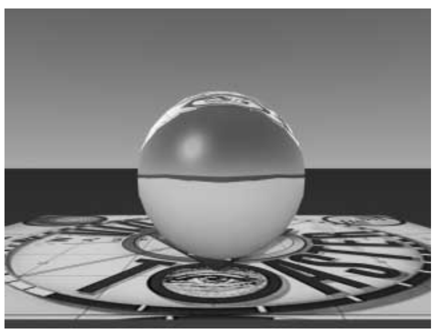
Choosing **Ray Trace Reflection** allows objects that contain surfaces with some degree of surface reflectivity to reflect any other objects around them.



Ray Trace Reflections

NOTE
The surfaces must have one of the Ray Tracing options selected as the **Reflection Options** (Surface Editor's Environment tab) in order to ray trace reflections.

Ray Trace Refraction instructs LightWave to calculate refractions for surfaces that have some degree of **Transparency**, and a **Refraction Index** (Surface Editor, Basic tab) higher than 1. Any objects seen through these surfaces will be distorted, such as a straw that appears bent in a glass that is half full of water.



Ray Trace Refraction

NOTE
Also see the discussion on the **Unseen by Rays** option in Chapter 9.

Ray Trace Transparency causes all transparent surfaces, not just refracting surfaces, to be ray traced. This allows volumetric effects (e.g., HyperVoxels) to appear behind such surfaces without forcing refraction ray tracing. Only use when necessary since this option will add to rendering time.

Ray Trace Optimization

Ray tracing in its simplest form must determine which polygon each ray hits. However, if all polygons were individually tested for every ray, ray tracing would be impractical for all but the simplest scene. Some sort of acceleration strategy is needed to make ray tracing practical.

LightWave's primary acceleration strategy is to compute a hierarchy of bounding volumes before doing any ray tracing. This involves some extra work initially, but it usually pays off when rays are actually traced. For example, if 1,024 polygons are contained within a box and a particular ray misses that box, there is no need to individually test whether the ray hits any of those 1,024 polygons, and the program can immediately try the next box. Moreover, if the ray does hit the box, it might still miss a smaller box within the first box, allowing it to skip the polygons in the smaller box, and so on. Obviously, such techniques can result in huge timesavings.

There are many ways to construct bounding volume hierarchies, some allowing much faster ray tracing than others. LightWave uses a simple and fast method by default. The **Extra Ray Trace Optimization** option activates a more sophisticated method, one that takes longer to compute but generates "tighter" volumes that can be ray traced more efficiently.

In scenes with heavy ray tracing, **Extra Ray Trace Optimization** can significantly reduce rendering time. However, in scenes with lots of geometry, but not much ray tracing, the extra preprocessing time might exceed any rendering timesavings. If there is no ray tracing at all in the scene, this option is ignored, even if active. However, it is not always obvious whether ray tracing is present. For example, surface shaders, volumetrics, pixel filters, and lens flares can sometimes trace rays even if all of the ray tracing options are inactive. In such cases, the preprocessing is still performed if **Extra Ray Trace Optimization** is active.

Here are some factors to consider in predicting whether **Extra Ray Trace Optimization** will help or hurt total rendering time:

- Bounding volumes are recomputed whenever time changes, so if motion blur is used, the preprocessing cost is multiplied by the number of passes and the number of segments, as opposed to being done only once per frame.
- The preprocessing cost depends roughly on the number of polygons in each object, but heavy geometry slows down ray tracing too, so that's not necessarily a reason to turn the option off.
- The benefits of using **Extra Ray Trace Optimization** depend on the amount of ray tracing needed, which is often proportional to the number of pixels that involve shadows, reflection, or refraction. As such, the benefit increases as the resolution goes up. Conversely, the

cost is still there even when rendering a 50% test image or a limited region, so any experiments to determine whether or not to use the option should be done at full image size.

Ray Recursion Limit

Ray Recursion is the maximum number of times you want light reflected off reflective surfaces. In the real world, things can be reflected an infinite number of times, like when you have two mirrors facing each other—you see an infinite number of reflections. In the world of 3D animation, you must set a limit for the number of reflections. The default value is 16, but you can set this field from 0 to 24. Lower numbers will yield faster rendering times.

HINT
If you do not have a lot of reflective surfaces in your scene, try using low values (even a value of 1).

Multiple CPU Systems

The **Multithreading** setting is used for systems with multiple CPUs. Normally, you should set this to the number of processors on your system; however, there are times when you will get faster rendering with this set at a greater number. You may want to test a few frames before beginning your final render.

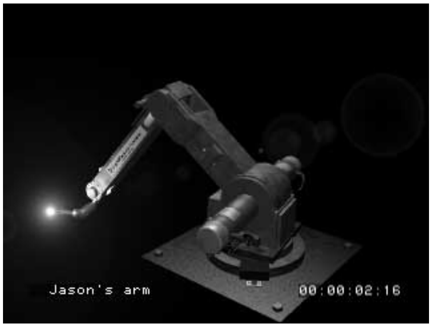
WARNING
A potential for problems exists with some plug-ins that were not designed with multithreading in mind. If problems occur, try using the 1 Thread setting).

When **Multithreading** is set to 1 and **Show Rendering in Progress** is enabled, you can watch the image being *built*. If a higher setting is used, the window will not update until the segment or frame is totally finished rendering.

Data Overlay

The **Data Overlay** option lets you place a descriptive title (of up to 20 characters) along with a frame reference in the lower portion of the rendered image. This is useful when you generate large numbers of animations for others to review and approve before final rendering. It can serve to identify the scene being rendered, and the specific frames you may wish to change. The **Data Overlay** pop-up menu setting determines the format for the frame reference. Enter the text in the **Label** field. If **Data Overlay** is not set to Off and the **Label** field is blank, the scene name will be used automatically.

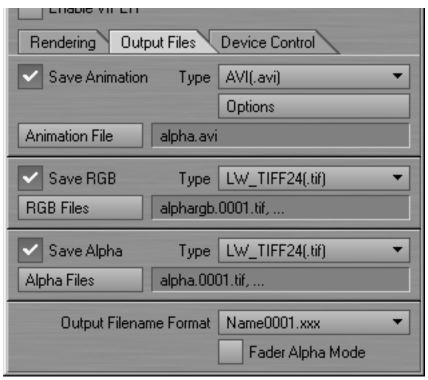
NOTE
If you have **Data Overlay** active, when you save your scene you can choose to have the scene name automatically set as the **Label**.



Data Overlay

OUTPUT FILES

Animations can be saved in two ways: as a single animation file (e.g., AVI) or as separate individual frame files. The two are not mutually exclusive.



Output Files tab

To save an animation file:

1 Click the **Save Animation** option to activate it.

2 If an acceptable animation file is <u>not</u> already defined (see the display to the right of the **Animation File** button), a file dialog will appear. Choose a location and a filename for the animation. Be sure to add the proper filename extension. (e.g., COOLANIM.AVI)

   If you wish to change the defined animation file, click the **Save Animation** button and define a new one.

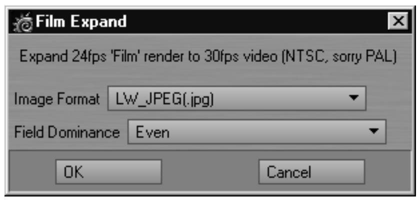
3 Choose the type of animation you would like to save with the related **Type** pop-up menu. The available options will vary depending on the platform you are using.

4 If the animation type has options, click the **Options** button to display a dialog with the available options.

You can activate and deactivate **Save Animation** without losing the defined animation file.

Special Animation Types

FilmExpand converts an animation from 24 fps (film speed) to 30 fps (NTSC video speed) by rendering 30 frames for every 24 of the original animation. The interpolation works best when you use **Field Rendering** (Camera Properties panel). When you select Field Rendering, an options panel appears where you can select the type of image to save and set the field dominance.



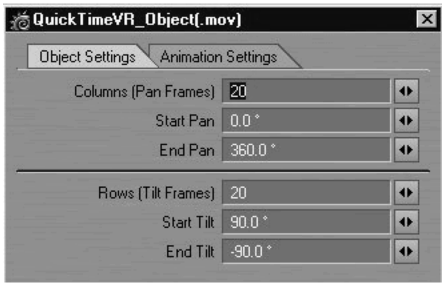
4XStoryboard creates JPEG images of the rendered animation frames at four times the camera resolution. You use this option when the **Resolution Multiplier** (Camera Properties) is set to 25% for test rendering of a scene.

Storyboard creates JPEG images that contain small versions of the rendered animation frames. The images are a five by six grid of thumbnails of the rendered frames. Every fifth frame is skipped.

QuickTime_Stereo is used with the **AnaglyphStereoCompose** filter, discussed in Chapter 14, for creating animations in the *red-blue glasses* style.

QuickTime Virtual Reality Object Saver

QuickTimeVR_Object is a *QuickTime Virtual Reality*™ object saver, which adds the ability to pan and tilt the camera inside the animation itself.



QuickTime Virtual Reality Object Saver

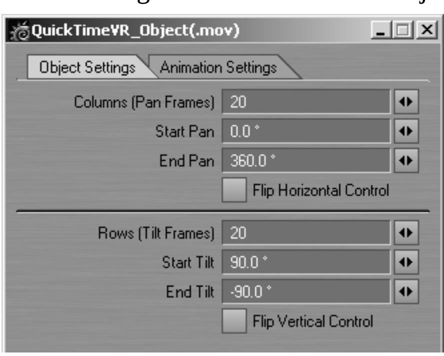
What is a QuickTime VR Object?

QuickTime VR Objects are QuickTime movies that display an object from multiple views. Special data embedded in an otherwise ordinary movie tells QuickTime how to play through the frames of a movie as a viewer moves the mouse across the image. When the viewer moves the mouse down, the player skips a large number of frames to find one that corresponds to a view of the same side of the object, from a lower position. This can simulate rotating the object in space, if the views are right.

Once a scene with the appropriate camera and/or object movement is created, open up the Render Options panel and select QuickTimeVR_Object(.mov) under the **Type** pop-up menu. Click the **Options** button to set the critical parameters for your animation.

Object Settings Tab

The image below shows the Object Settings tab.



Columns (Pan Frames) is the number of frames of horizontal views in the animation. If your view is from 0 degrees at frame 0 to a full 360 degrees at frame 20 (the same view as 0°), then you will have 20 columns (frames 000 to 019). In a typical movie, the row angle will step up after this many frames.

The **Start Pan** and **End Pan** values set the angle range of the horizontal views.

Flip Horizontal Control reverses the object's direction of rotation with regard to horizontal mouse movement. In other words, if you drag the object left and it turns to the right, you need to change this setting (or reverse your entire animation!).

Rows (Tilt Frames) is the number of frames of vertical views in the animation. In a QTVR Object Movie, the animation moves through all the pan frames in a row before moving to the next tilt frame, and does all the pan views again with that tilt.

If there are 20 pan frames (000 to 019, as above), the tilt angle will change every 20 frames, so sequential tilt values will be at frames 000, 020, 040, 060, etc. This means that your entire animation must be (tilt frames)*(pan frames) long.

HINT
The default Columns and Rows settings (20 x 20 = 400 frames) are more than enough for most uses and you may be able to use lower values to reduce the size of the file.

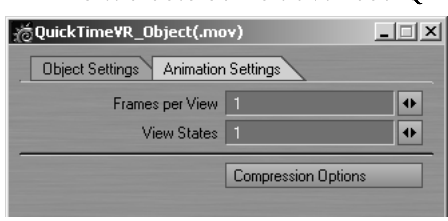
The **Start Tilt** and **End Tilt** values set the angle range of the vertical views.

NOTE
The *Start* and *End* settings may not seem to change simple viewing of the objects, but may be critical if the object is embedded in a larger QTVR world.

Flip Vertical Control reverses the direction of rotation of the object with regard to vertical mouse movement. If you drag the object up, and it rotates down, you need to change this setting.

Animation Settings Tab

This tab sets some advanced QTVR options.



Frames Per View sets the animation loop size that will run from each view. If this value is greater than one, then your animation should hold each pan view for this many frames, as the loop happens. This also means that your animation length must be multiplied by this number. You should not alter the pan or tilt frame counts; QTVR will handle that.

The QTVR object format features an *alternate state* that can be displayed based on some user input, for example, when the user clicks the mouse. The **View States** value supports this alternate state, but will also multiply your animation frame count. These extra frames take the form of an entire identical animation of the object in an alternate state, appended onto the first. Currently, the QuickTime player appears to support only two usable states, *not clicked* and *clicked*. In theory, higher values could be used.

The **Compression Options** button will invoke the standard QuickTime Compressor selection dialog. The choices are many and the differences subtle. The only relevant warning for QTVR is that you should set the compression to use keyframes at every frame, since these movies are not played in a linear fashion.

To set up a basic QTVR scene:

1 Add three null objects to an empty scene, named PAN, TILT, and TARGET.

2 Parent Tilt to Pan and parent the camera and light to Tilt.

3 Set the camera's position to 0m, 0m, -2m.

4 Unselect the X and Y motion channels, as well as all the rotation channels. Moving the camera on Z will determine its distance from the actual object—necessary for proper framing. All other camera motion is done with the Pan and Tilt nulls.

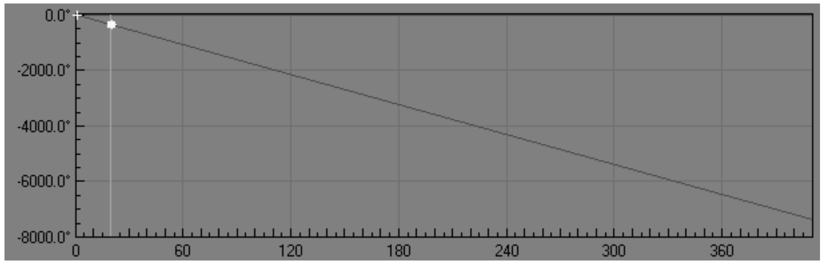
5 Parent **Pan** to **Target**, for future convenience.

At this point, you should decide how many rows, columns, view states, and animation loop frames you want. Here, we will use the defaults.

It is best to start on the innermost loop first. Since we're using defaults, with no extra animation loop frames (**Frames Per View** is 1), the inner loop is the pan motion.

6 Create a keyframe for the Pan object at frame 20, with the Heading at -360 degrees.

7 In the Graph Editor, select both keys and set their **Incoming Curve**, **Pre Behavior** and **Post Behavior** to **Linear**. This should give a constant heading rotation of one cycle every twenty frames.

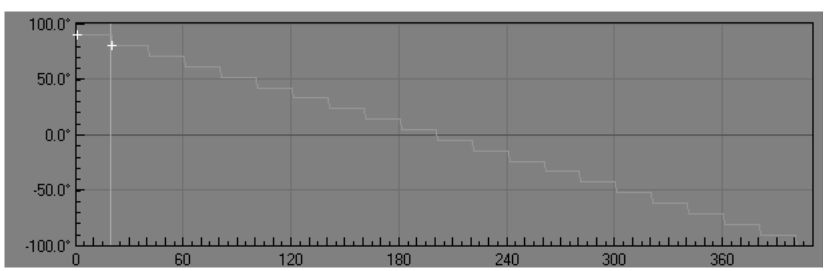


If we had chosen to use only one Tilt Frame, our work would be done. We chose, however, to create twenty Tilt Frames, ranging over 180 degrees. This means that each tilt frame should increase enough that the last row is at -90 degrees.

If we simply used, 180 degrees/20 or 9 degrees, the Tilt's pitch would reach -90 degrees at precisely 400, the first frame after our render! Using 180 degrees/19 = 9.47 degrees will get the tilt to -90 degrees at the end of the 19th row, so our animation will include a pan from directly below as well as directly above. Since there are 20 Pan frames, we must hold each tilt angle for that many frames.

In short, our motion curve should look more like stairsteps than the perfect line of the Pan Heading curve. To be more specific, Tilt's Pitch Angle curve should be stairs of a height of 9.47 degrees, and a width of 20 frames, going from 90 to -90 degrees. Fortunately for us, this is easily achieved.

8 In the Graph Editor, set the key at frame 0 in Tilt's Pitch channel to a value of 90 degrees and create a key at frame 20, with a value of 80.53 degrees (90-9.47). Set the **Incoming Curve** to **Stepped**, and the **Post Behavior** to **Offset Repeat**. The camera motion should now be complete.



9 You can save the scene as a basic QTVR template.

In order to create an interesting QTVR Object, you need something to render. If you load an object that is too large or too small you can move the camera along the Z axis (use the Local Coordinate System).

If the object is off-center, it should be moved. The object should fit in the frame of all the views, so it is important to check the ends and middle of the animation, but only set the Z keyframe position at frame 0.

Try loading the cow object and follow the preceding steps to create your own QTVR animation.

NOTE
There are many ways to set up a LightWave scene to render a QTVR Object Movie. The above procedure is only one method of many.

Saving Individual Frame Files

You can also save each frame as an individual file.

To save each frame as an individual file when rendering:

1. Activate the **Save RGB** or **Save Alpha** option.
2. Choose a location and a *base filename* (described below) in the file dialog that appears when you click the **RGB Files** or **Alpha Files** button. Using different names for RGB and alpha files is encouraged to avoid overwriting—no, let's make that a rule!
3. Choose the file format you would like to use with the related **Type** pop-up menu.

NOTE
See the *Appendix* for information on image types.

Using 32-bit RGB Formats

When a 32-bit RGB image format is selected, 8-bit alpha image data is automatically embedded in the same image file along with the 24-bit RGB data. Alpha channels are useful for creating masks to key an animation into another animation or to use with a video switcher to key the animation over live video.

Using 32-bit RGB image formats can cause unexpected results, like some or all of the image not appearing on surfaces when rendered, due to the alpha channel masking—the alpha channel will not be taken into account in OpenGL. (Of course, you can always disable the alpha channel in the Image Editor.)

Selecting a Filename Format

When you save an RGB or alpha image, LightWave will append a numbered extension to the "base name," based upon the frame rendered. The filenaming convention is determined by the **Output Filename Format** pop-up menu setting. "Name" will be replaced with the base name. The number one with the leading zeros (e.g., 0001) indicates the number of digits used for the numerical sequence. The standard PC file extension will replace “.xxx”.

For example, with **Name001**, frame 34 of an animation using the base name EXPLODE would save out as EXPLODE034. Using **Name001.xxx**, EXPLODE as the base name, and a Targa file format, the 56th frame would save an image named EXPLODE056.TGA.

NOTE
The .xxx formats are recommended for general use, since most applications require that you use the proper filename extension.

A filename example appears to the right of the save buttons. The last used **Output Filename Format** is saved as the default for the next time you start LightWave.

WARNING
If the images may be used on a platform that supports only “8+3” filenames, make sure to keep your base names to a maximum of four or five characters (depending on whether you select a Name001 or Name0001 **Output Filename Format**).

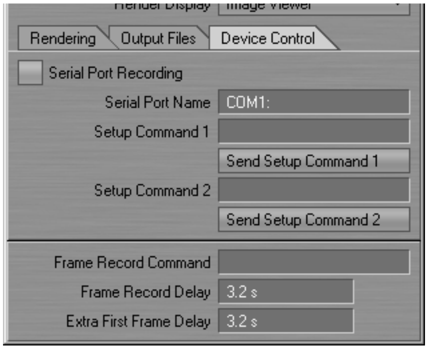
Fader Alpha

If you plan to use an alpha channel generated by LightWave along with an external video fader or linear keyer, or a different compositing program, you may need to activate **Fader Alpha Mode**. Certain switchers can use an alpha image as a fade control. When you select this mode, LightWave computes the saved RGB images and alpha images a bit differently. Transparent items in an RGB image render in a more intensified manner, with rough, aliased edges. Lens flares appear *overblown*. In alpha images, however, the transparency levels render properly so that the appropriate transparency levels are used when you combine the RGB images with the alpha channel.

WARNING
Do not use this mode if you plan to digitally composite images in LightWave or another digital compositing package that allows for additive compositing.

RECORDING IMAGES TO TAPE

With the **Serial Port Recording** option active, you can instruct LightWave to send record commands to a single-frame controller, enabling it to control a VTR deck to record frames. These record commands are sent after each frame is rendered.



The name of your serial port needs to be entered into the **Serial Port Name** field. On a PC, for example, you may use COM1: or COM2:. The name of your serial port is saved as a default when you quit LightWave.

Enter the appropriate commands into the **Setup Command 1**, **Setup Command 2**, and **Frame Record Command** fields. (These vary depending on the recording equipment's manufacturer: see its documentation for instructions and proper commands to use.) There are buttons to send the setup commands manually.

The **Frame Record Command** delays the playback of frames enough to give the VTR time to rewind and cue up for a single frame edit. **Frame Record Delay**, if needed, gives a deck enough time to pre-roll to the selected edit point before a frame is laid down. You can also add some additional time for the first frame.

Many single frame control devices need to know a starting position value (this is usually a time code number). To give a starting value, press the s key. The **Starting Position** requester will appear where you can input a value for the starting position. (Note: a **Starting Position** applies only when your **Frame Record Command** uses the # control code.)

To save wear and tear on your VTR, first render all of the frames to your hard drive. Then set up a new scene using the resulting image sequence as a **Background Image** (Effects panel, Compositing tab). Render the new scene and record the images.

chapter 17
Particle FX

Chapter 17: Particle FX

With LightWave's integrated particle system, Particle FX, 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.

Particle FX 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.

PARTIGONS

"Partigons" are a special object type for particles. Partigons provide a means for particle systems (like Particle FX) 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.

PARTICLE FX PANELS

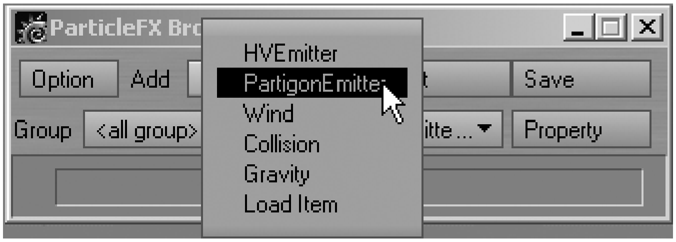
The Particle FX Browser panel is where you add the various controllers. From here, you can also open the property panels for the controllers that you add. To open this panel, choose **Scene > FX_Browser.**

ParticleFX Browser panel

HINT
Keep this non-modal panel out while you use particles; it will save you a lot of time adding controllers and changing properties.

To add a Particle FX controller:

From the **Add** pop-up menu, choose the controller you wish to add. Generally, the first controller you will add is an Emitter.



Adding a controller

NOTE
When you add a controller from the Particle FX Browser, what really happens is an object, usually a null object, is added to the scene. Then, a Particle FX custom object plug-in is added (e.g., FX_Emitter, FX_Wind, etc.). Just check out the Geometry tab of the Object Properties panel.

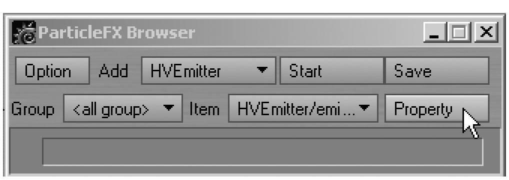
To delete a Particle FX controller:

Select and clear, just as you would any other Layout item (**Items** > **Clear** > **Clear Selected Item**).

To open a controller's property panel:

1 Choose the controller from the **Item** pop-up menu.

2 Click the **Property** button.



NOTE
You can also display the property panel for the currently selected controller by choosing **Scene** > **FX_Property**.

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, Particle 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 **Scene** > **FX_Start**.

NOTE
Many scenes don't need this step, but clicking **Start** won't do any harm if you are unsure.

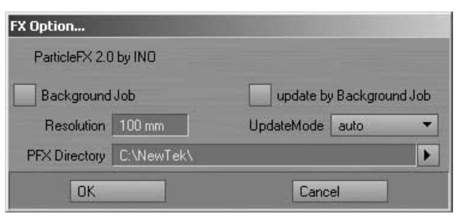
The Save Button

Clicking the **Save** button saves all emitter motions, using individual files, to your PFX directory (defined on the Option dialog). These can be loaded using the **Load Motion** option on the Emitter panel's File tab.

Options Dialog

The **Options** dialog has some options that affect the manner in which LightWave computes particles.

Options button



Options panel

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.

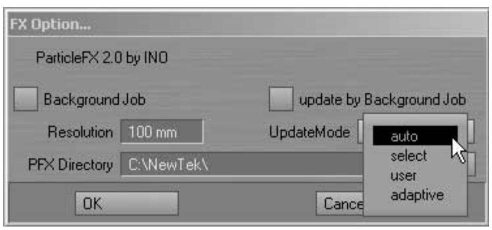
In this dialog you can also set the **PFX Directory**. This is the default directory used when saving controller settings.

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 emitters are updated using the **UpdateMode** pop-up menu.



UpdateMode pop-up menu on PFX Options panel

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.

LOADING/SAVING CONTROLLERS

All of the individual controller panels sport **save** buttons that let you save the controller's settings to a file. If you wish to add a new controller using the same settings, choose **Add** > **Load Item** on the Particle FX Browser panel. If you wish to use those settings on an existing controller, use the **load** button on that controller's panel.

Controllers also have **copy** and **paste** buttons, which let you copy settings between like-kind controllers.

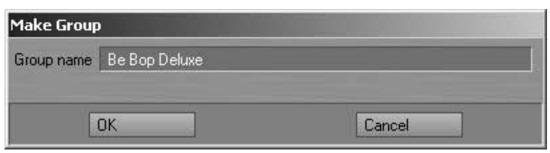
CONTROLLER GROUPS

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.

To create a group:

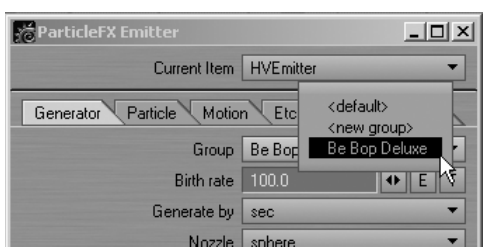
Choose <**new_group**> from the **Group** pop-up menu on the Browser panel or the individual controller panels. A dialog will appear that asks you to input a group name.



Make Group dialog

When you add a new controller, it uses the group selected on the Browser panel **Group** pop-up menu.

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.



Groups appear on Group button. Emitter properties panel shown.

NOTE
The Group feature works in conjunction with Motion Designer! Thus, if you add an emitter, a wind controller, and a soft body (Motion Designer) 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.

Emitter Controller

The Emitter controller is the main controller and the source for particles. The settings for this controller determine the *make up* of the particles, factors like how many particles are created, how they move initially, how long they last, and so on.

If you have multiple emitters, you can select a specific emitter from the **Current Item** pop-up menu at the top of the panel.

Emitter Types

On the **Add** menu for the Particle FX Browser panel, there are two emitter selections: **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. <u>To render particles from HVEmitters, you must add the HyperVoxels volumetric plug-in</u> 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.

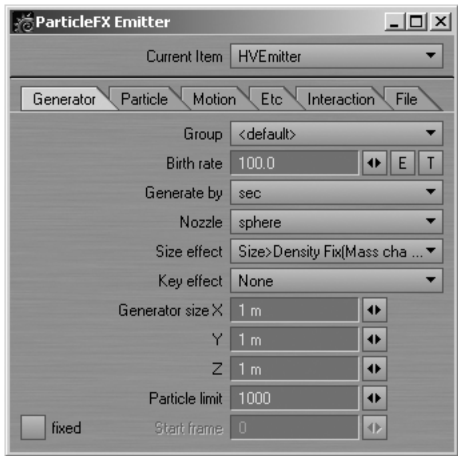
You can achieve many effects, like fireworks and sparks, by using PolygonEmitters alone (i.e., without HyperVoxels). You can save significant rendering time compared to HVEmitters, which require HyperVoxels for rendering.

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.

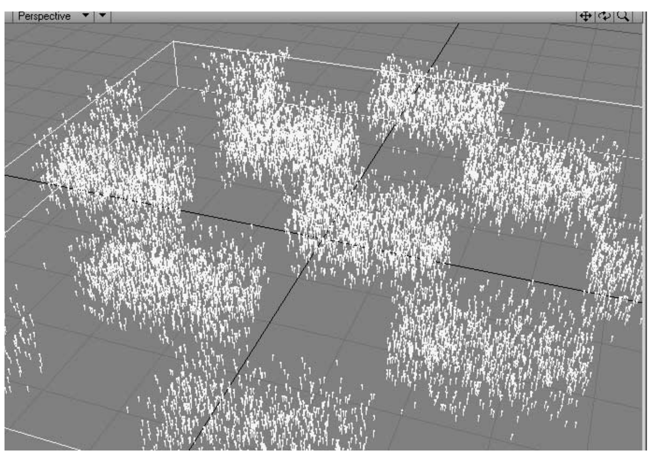
Generator Tab

On the Generator tab, you set parameters relating to the creation of the particles.



Generator tab of Emitter Properties

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.



Using Brick texture

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.

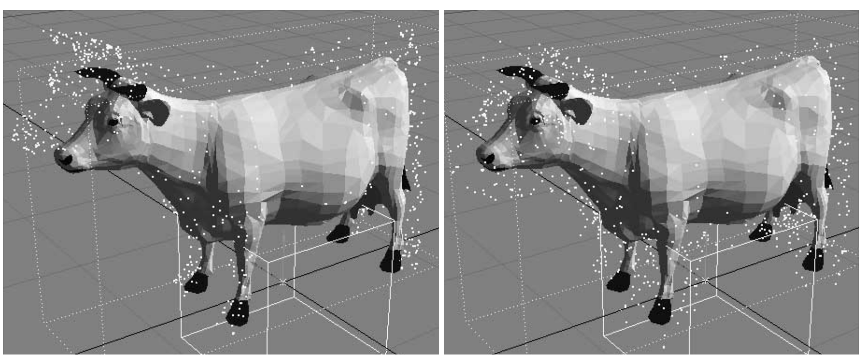
When **Generate by** is set to **speed**, it creates the particles based on the speed of the emitter, caused by moving it. You need to keyframe some movement for the emitter to use this setting. The faster the movement, the faster the particles are created. The effect is sort of like salt coming out of a salt shaker—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.

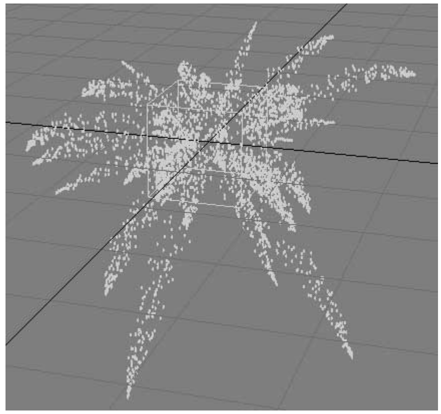
You can use geometry as an emitter, by adding the **FX_Emitter** custom object to any loaded object. This will allow you to use the **Object-vertices**, **Object-normal, Object-surface**, and **Object-line** nozzle types. When you use these nozzles, you can offset the center position by changing the **Center Position** values on the Misc tab.



Left: Object-vertices. *Right*: Object-normal

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.)



Firework-like effects can be created using the parent-emitter nozzle type.

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.

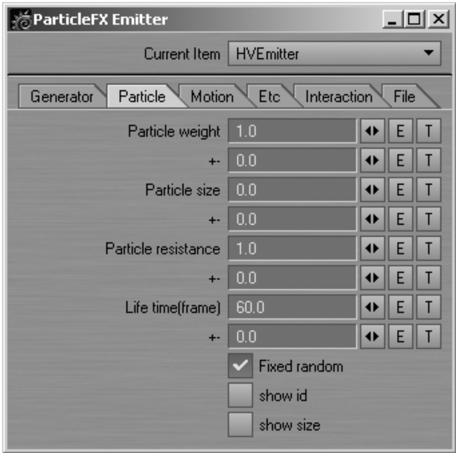
The XYZ **Generator size** fields set the default size of the emitter.

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.



Emitter Particle tab

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.

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.

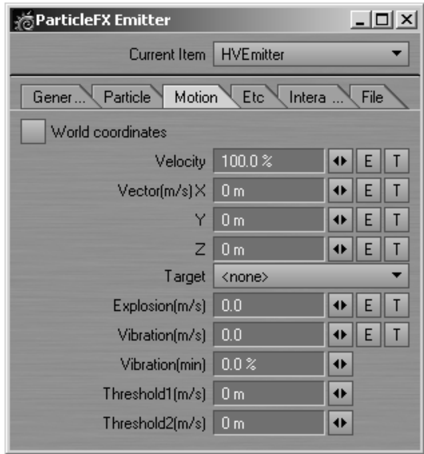
If you activate **Fixed Random**, random calculations are constant, so they yield more predictable results.

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.

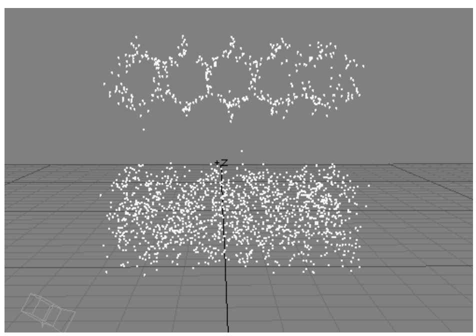
Motion Tab

On the Motion tab, you set how the particles are placed into motion.



Emitter Motion tab

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.



Honeycomb texture applied to velocity

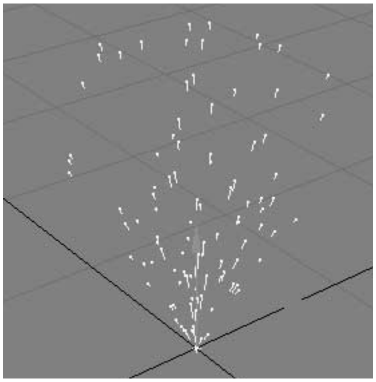
The XYZ **Vector** settings determine the initial direction and speed of the particle motion. Other factors, like gravity, wind, and so on, will impact the actual result, however. You may override the **Vector** settings and point the particles to an object in the scene by selecting it in the **Target** 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.



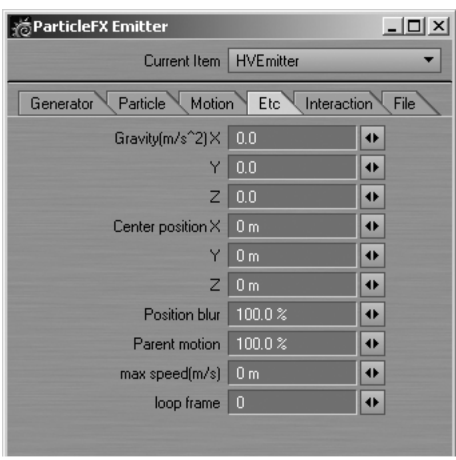
A fountain of particles

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.



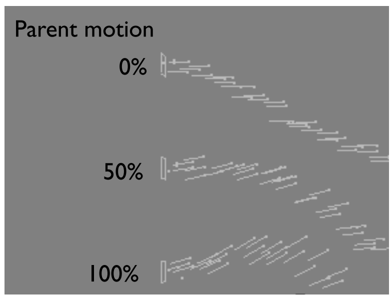
Emitter Etc tab

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.



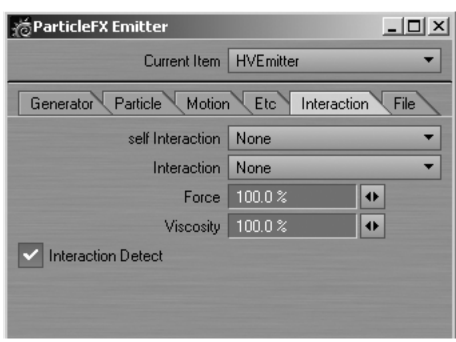
The three emitters are moving up in unison. Note how the 100% Parent motion particles also move up.

Setting the **Max Speed (m/s)** field regulates the top speed of the a 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 2 particles will be generated with the same birth location and motion.

Interaction Tab

The settings on this tab affect the way particles interact with other particles from both the same and other emitters of the group.



Emitter Interaction tab

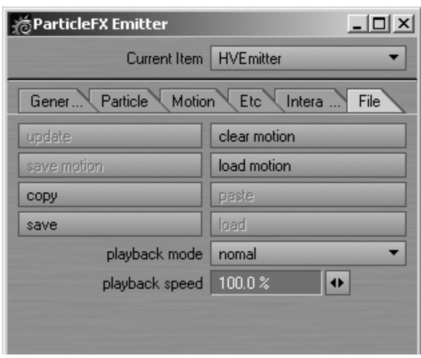
The **Self Interaction** drop-list determines how particles react to other particles of the *same* emitter. The **Interaction** drop-list determines how particles react to other particles from *different* emitters (within the same group). Options include having colliding particles **push** emitters, **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 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.



Emitter File tab

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, all will have no impact.

Save motion is a great time-saver for scenes that will be network-rendered 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 Particle FX 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.

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 Particle FX controllers to your scene.

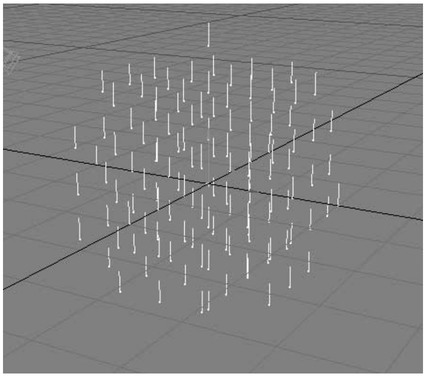
Using an Object as an Emitter

You can use a LightWave object as an emitter by adding FX_Emitter as a Custom Object plug-in on the Object Properties panel, Geometry tab. You must also use one of the *object* Nozzle settings discussed below under *Emitter Controller*.

NOTE
The custom object bounding box will continue to appear.

WIND CONTROLLER

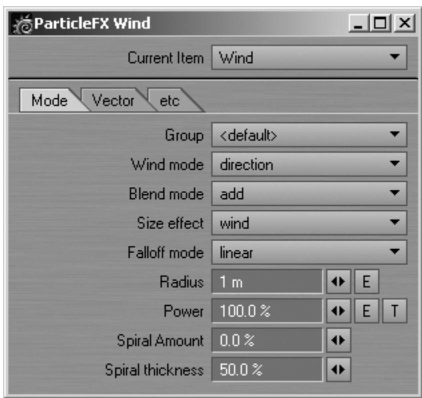
The Wind controller lets you add wind to blow your particles around. The Wind controller appears in a Layout viewport as a cloud of *wind indicators*—a bunch of short lines with pivot point dots at one end. The the length of the lines indicates the force of the wind: the longer the line, the more powerful the wind. The angle of the line indicates the direction of the wind at the line's location.



Wind object

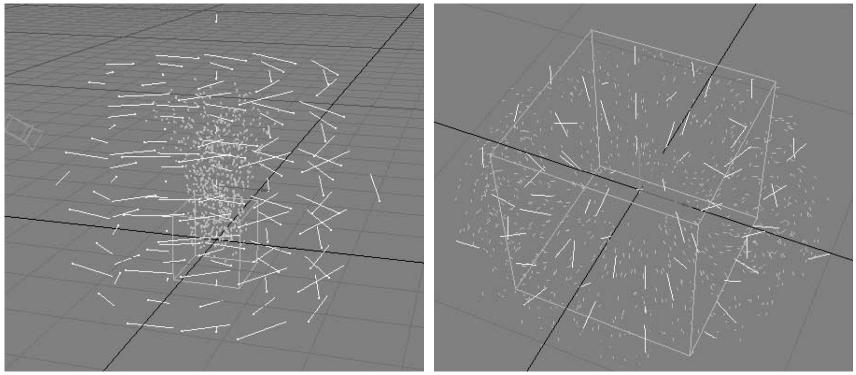
Mode Tab

On the Mode tab, you set the basic characteristics of the Wind controller.



Wind Mode tab

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.



Left: rotation(y). Right: doughnut

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.

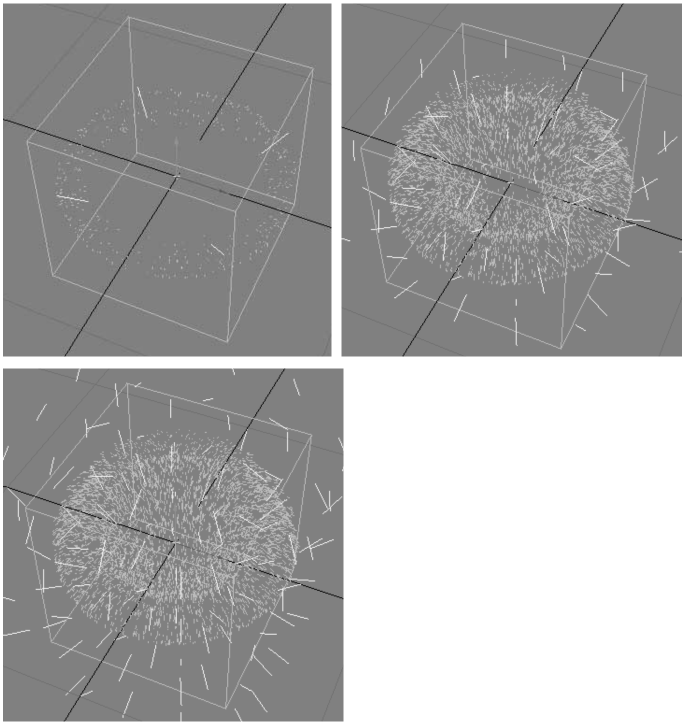
The **Falloff mode** setting determines how the wind's effect tapers off at its region borders, if at all. **Linear** means the wind is the same throughout its borders. The **Inverse Distance** setting tapers off the wind's power from its center. The **Distance** setting makes the wind falloff towards 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.

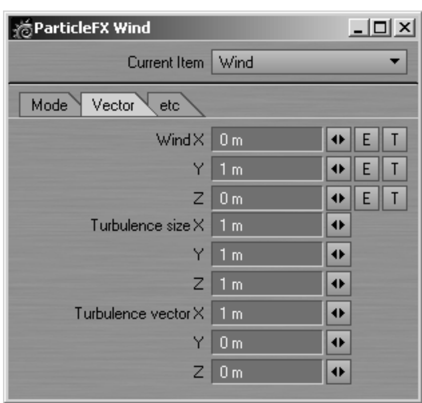
The **Spiral Thickness** setting controls the thickness of the doughnut wind.



Spiral Thickness. Left: 10%. Middle: 50%. Right: 100%

Vector Tab

On the Vector tab, you set the direction and force characteristics of the wind controller.



Wind Vector tab

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.

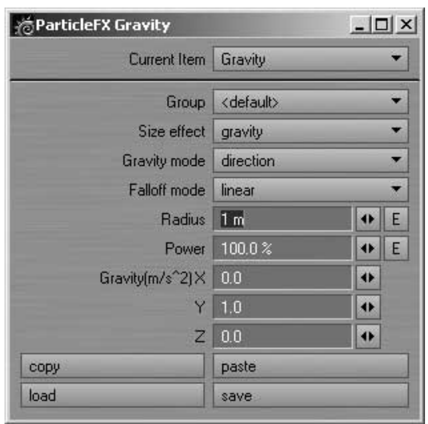
If you are using the **Direction** Wind mode, use the texture **T** button to add a 3D texture that affects wind direction and force. Note that the texture is added to the base **Wind** XYZ settings. To remove the texture, SHIFT + click the **T** button.

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.

GRAVITY CONTROLLER

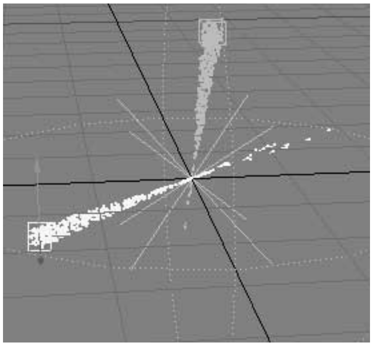
Add a Gravity controller to add gravity-like effects to your particle motions.



Gravity Controller

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 with **Power** = -100. Note how particles continue past center

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.

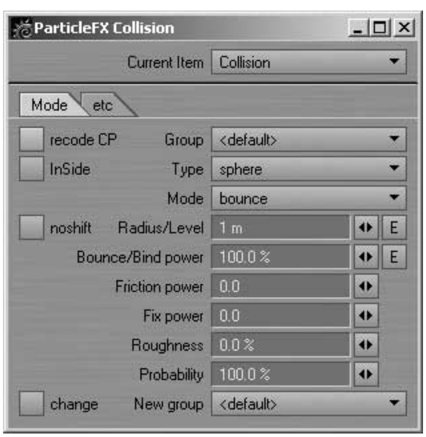
The **Radius** setting sets the radius of the controller region.

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.

COLLISION CONTROLLER

The Collision controller lets you add an element for the particles to bump into.

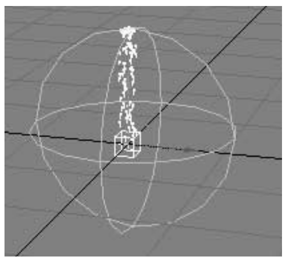


Collision Controller

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, which is discussed later. 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**, discussed earlier. 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.

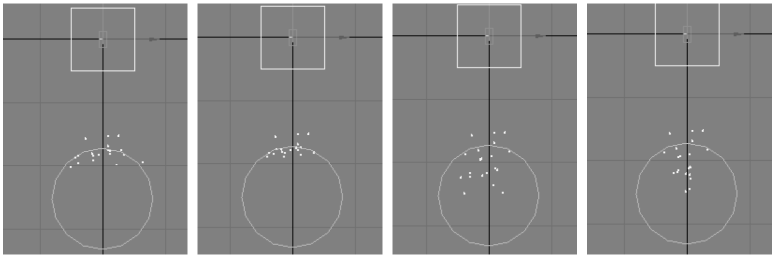


Using InSide

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.

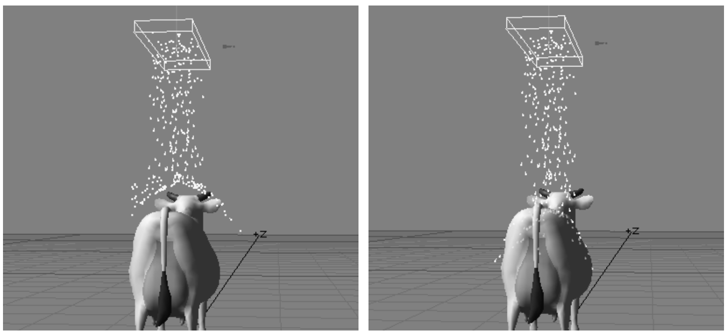
By activating the **Change** option, particles passing through the collision item can switch to a different group, to be affected by a different controller. The **New Group** defines which group the particles will switch to.

The **Mode** setting determines what happens when a collision occurs. The **Bound** 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.



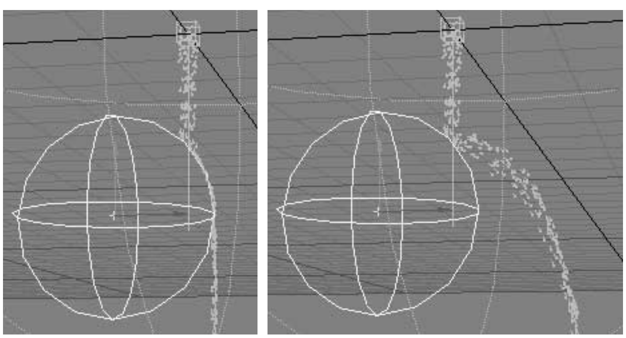
Left to right: Bound, Stick, Diffuse and Target modes

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.

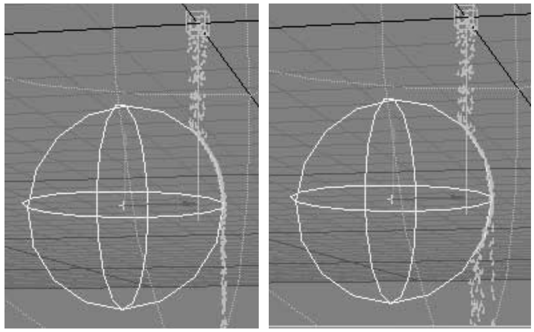


Left: Radius = .3. Right: Radius = 0

The **Bound/Bind power** setting controls how particles bound off of the surface, when **Mode** is set to **Bound**. The **Stick** mode controls the amount of *attachment* the particles have.

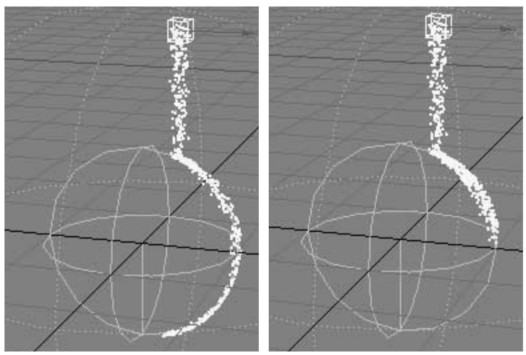


Bound/Bind power using Bound mode. *Left*: 0%. *Right*: 200%



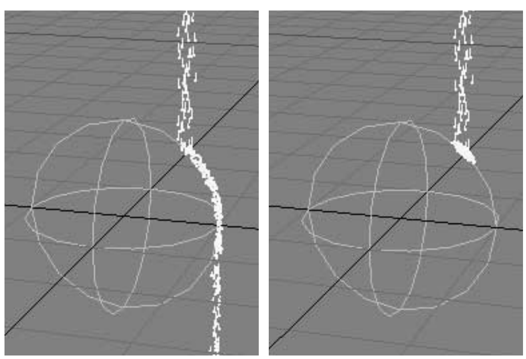
Bound/Bind power using Stick mode. *Left*: 75%. *Right*: 125%

Friction power adds friction to the collision surface when you use the **Stick** mode, which slows the momentum of particles sliding over the surface.



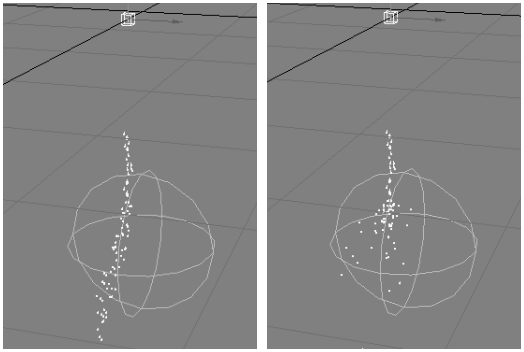
Friction power. *Left*: 0. *Right*: 2

Fix power causes the particles to stick to the surface and not slide around.



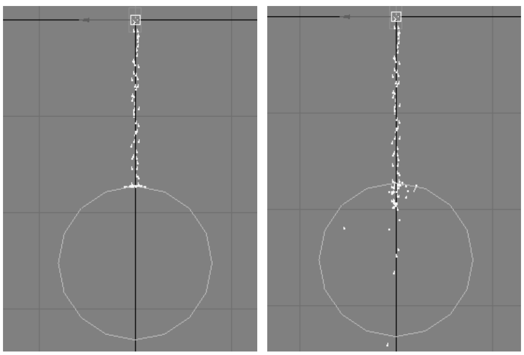
Left: **Fix power** = 0. *Right*: **Fix power** = .1

Increasing the **Roughness** setting *roughens* up the collision surface. Roughness changes particle motion after collisions, but it also depends on the Mode setting.



Using Bounding mode. Left: Roughness = 0%. Right: 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.



Left: Probability = 100%. Right: Probability = 10%

Using an Object for Collisions

You can use a LightWave object as a Collision object by adding FX_Collision as a Custom Object plug-in on the Object Properties panel, Geometry 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.

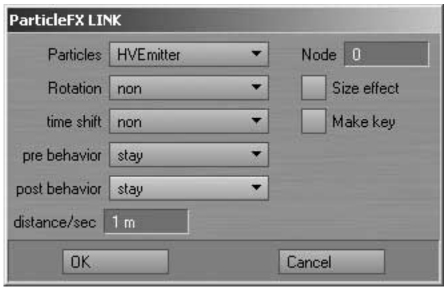
ITEM MOTION MODIFIERS

Particle FX features a few item motion modifiers that let Layout items move along with or instead of particles. You add these plug-ins to a Layout item by clicking the **Add Modifier** button on the item's Motion Options panel. After you add the modifiers, you can open an options panel by clicking the modifier's name in the list.

NOTE
Generally, these modifiers are applied to objects, but they will also work on cameras, lights, bones, and so on.

FX_Link

The FX_Link motion modifier links the motion of the particles with an object. To use FX_Link, just add the modifier on the item's Motion Options panel (IK and Modifiers tab)



FX_Link panel

Select the emitter on the Particles pop-up menu. The **Node** setting is the specific particle, and 0 is the first emitted. By activating the **Show Id** option in the emitter's particle tab, you can determine exactly which particle you want to use.

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.

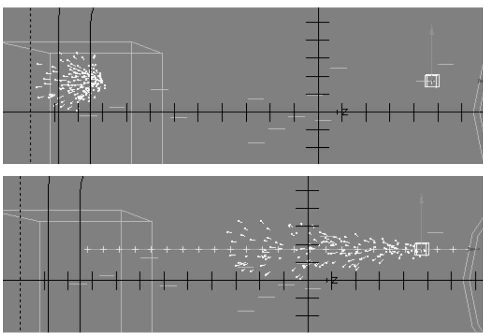
The **Stay** behavior will hold the first frame of the motion for the **Pre Behavior** and the last frame with the **Post Behavior**. The **Original** setting returns the object to its original state in either the beginning or ending of the animation. The **Size Dissolve** setting will dissolve the item 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.

NOTE
Make sure your particles have some Particle Size on the Particle tab of the Emitter property panel. By default, Particle Size is 0.

Make key will make a keyframe of the related particle motion on every frame—take care using this option because created keyframes don't go away when you uncheck this option. You sometimes need Make key because the movement of the linked object is only *apparent.* Internally, the object is still at its keyframed position.

In the top figure below, you can see that the linked object (an emitter) is on the right. However, its keyframed position is on the left as indicated by the emitted particles. In the bottom figure below, **Make key** is active. Thus, the keyframed position follows the motion of the linked object.

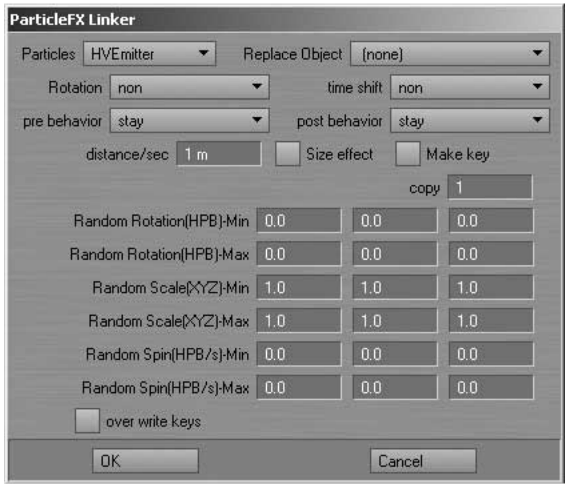


Top: Make key off. Bottom: Make key on

Make key is important for features like collision detection. The keyframed position will cause the collision event, not the linked object's apparent position.

FX_Linker

FX_Linker is a *Generic plug-in* that automates FX_Link setup on multiple particles. It basically clones the specified object and adds the FX_Link motion modifier using the settings you specify.



FX_Linker panel

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 **Scene** > **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.

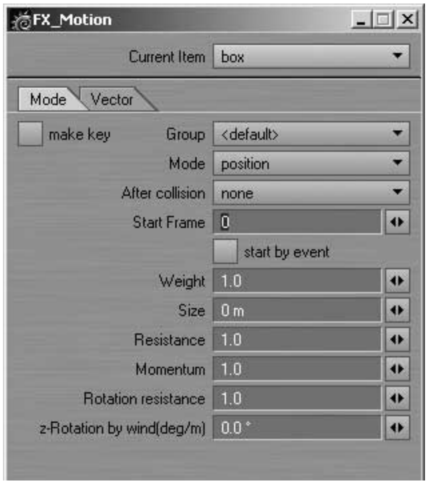
6 If you are using **Random** or **Size effect**, you may set maximum and minimum values in the provided fields.

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.

FX_Motion

When you add the FX_Motion motion modifier to an item, you can control it in a manner similar to particles.



FX_Motion

The Mode Tab

The **Mode** pop-up menu determines if you want the item's **position**, **rotation,** or both affected.

If you use rotation mode with a collision, you can set the **After collision** setting to determine what happens to rotation after a collision. The **None** settings will add no effect; **Reverse** reverses the incoming rotation; **Random** randomizes the rotation; and **Stop** stops rotation.

If you want the motion to begin at a certain frame, instead of at the time of collision, enter the frame number in the **Start Frame** field and make sure **start by event** is unchecked. If you want to trigger the motion upon collision, activate the **start by event** option.

Weight sets an arbitrary weighting value that will influence how factors like gravity affect the item.

Size affects the outer boundary used for collision detection. It can also be used by volumetric plug-ins like HyperVoxels.

Resistance adds an *air resistance* effect. Items will move slower as this value is increased.

You can set independent weighting and resistance settings for rotation using **Momentum** and **Resistance**.

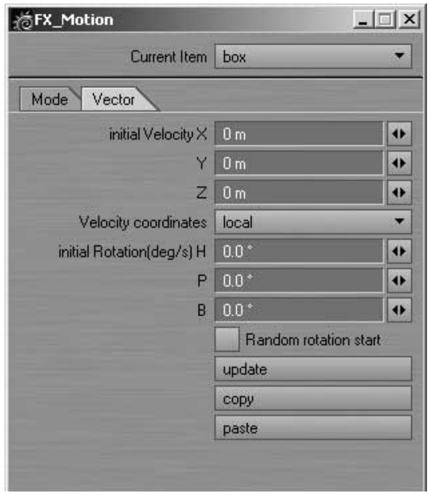
The **z-Rotation by wind** value makes the item rotate its bank as it moves or is affected by wind.

The **Make Key** option works same as in the FX_Link motion modifier (see above).

The Vector Tab

The **initial Velocity** XYZ settings set the initial direction and force. Use the **Velocity coordinates** pop-up to set whether these settings use the item's **local** axes or **world** coordinates.

The **initial Rotation(deg/s)** HPB settings set the initial rotation. If you want the rotation to begin at a random point, activate the **Random rotation start** option.

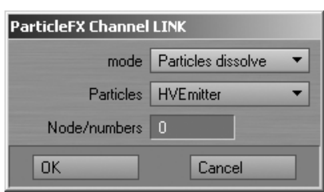


FX_Motion Vector tab

NOTE
The **Mode** setting on the Mode tab determines whether or not these initial settings have any effect. In other words, if you aren't using a rotation mode, the initial rotation settings don't matter.

FX_LINK CHANNEL MOTION MODIFIER

The FX_Link channel modifier has two functions. First, it can be used to dissolve out the object when the particles die.



FX_Link Channel motion modifier

To dissolve an object based on particle life:

1 Open the Object Properties panel for the object to be dissolved. (You'll also need an existing emitter controller.)

2 On the Rendering tab, click the **Object Dissolve E**nvelope button. This adds a Dissolve channel for the object and opens the Graph Editor.

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.

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.

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:

1 Open the Graph Editor and select the desired channel in the Curve bin.

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.

3 Set the **mode** to **Particles numbers** and select the emitter from the **Particles** pop-up menu.

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.

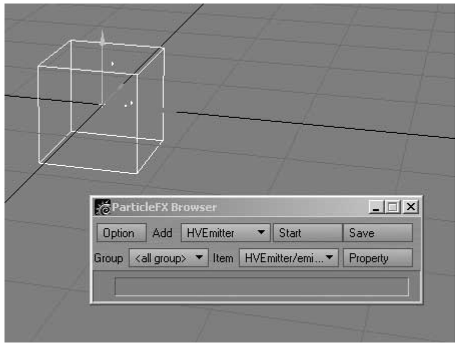
GETTING STARTED

Here are some very basic tutorials to get you started with particles.

Emitter and Wind

1 In Layout, choose **Scene** > **FX_Browser**. This brings up the Particle FX Browser panel.

2 Let's use the new partigon object type. From the **Add** pop-up menu, choose **PolygonEmitter**. This will add a PolygonEmitter controller to your scene. It will also list this emitter in the **Item** pop-up menu on the Particle FX Browser panel.

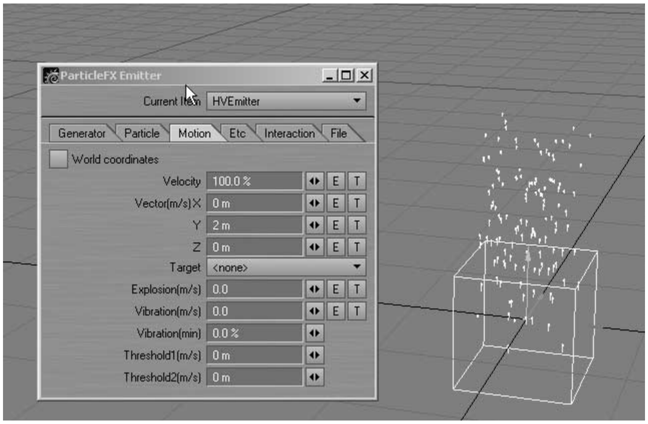


3 Click the **Property** button. This brings up the property panel for the controller shown on the **Item** pop-up menu, in this case, the emitter's.

4 Click the play button on the main Layout interface. This is the right-facing triangle in the lower-right corner—changes you make to controllers are updated in real-time. Right now, all you will see are the particles being created inside the emitter; there is no movement.

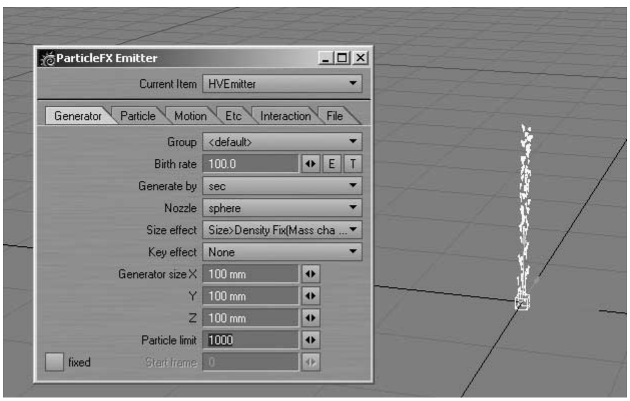
5 Open the Surface Editor. The PolygonEmitter object will have a default surface called *Partigons*. Just change the **Color** to a deep red. You should see the particles change color.

6 Click the Motion tab on the emitter's property panel. Set the **Vector Y** value to 2 to give the particles some vertical movement.

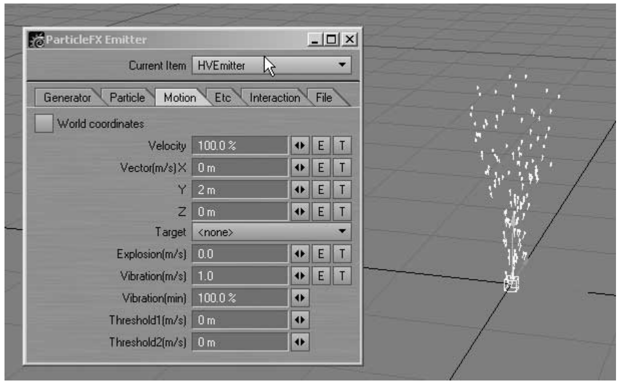


The particles at frame 41

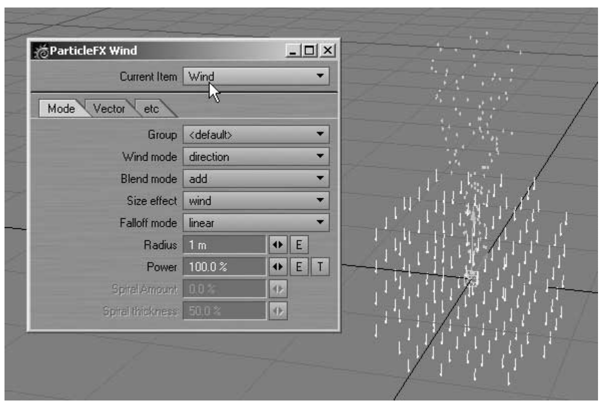
7 On the Generator tab, decrease the **Generator size** values to .1, .1 , .1. Notice the effect of changing the size of the emitter.



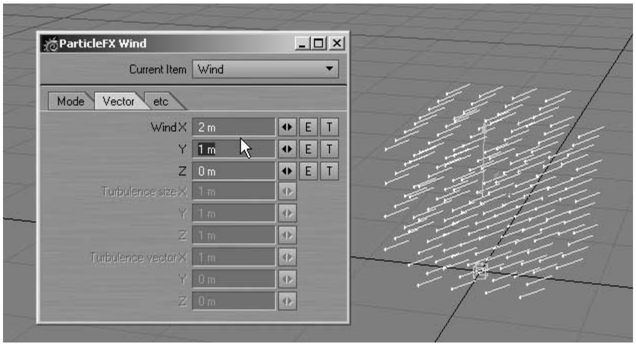
8 On the Motion tab, try different values for the **Explosion** and **Vibration** settings and note the results. When you are done fooling around, leave **Vibration(m/s)** at 1, **Vibration(min)** at 100%, and **Explosion** at 0.



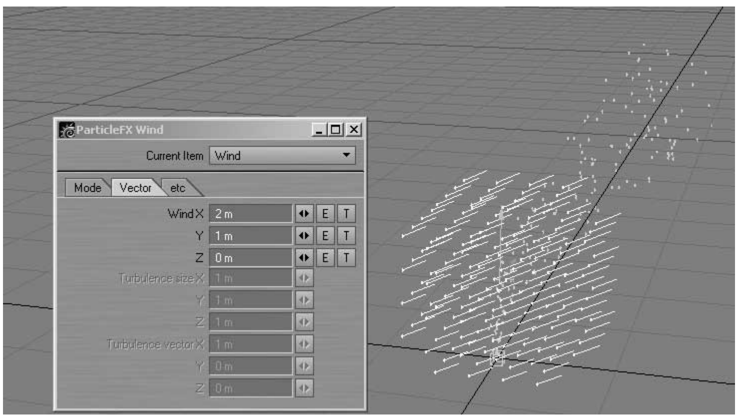
9 Go back to the FX Browser panel and from the **Add** pop-up menu, choose **Wind**. This adds a wind controller to your scene. The controller name, Wind, should appear on the **Item** pop-up menu, so click the **Property** button to bring up its property panel.



10 In Layout, stop playback and go to frame 0. Move the wind controller up a little (to about Y=700mm). Also, on the wind's property panel, change the **Wind X** setting on the Vector tab to 2m. Leave **Wind Y** at 1. You will see the little wind vectors change directions.



11 Play the scene again and see the particles change direction due to the wind.



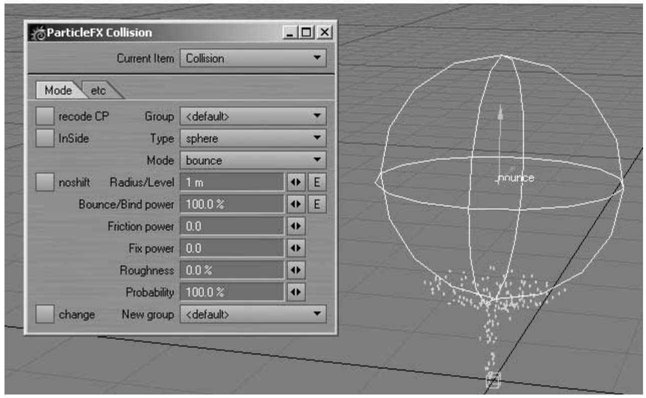
Go ahead and play with the wind and emitter settings. It's very easy. Try adding HyperVoxels to the emitter object. Remember, if you don't want to see the (partigon) particles when using HyperVoxels, use **Object Dissolve** on the emitter's Object Properties panel.

Collision Detection

1 Start with the emitter from the preceding tutorial, but clear the wind controller. (The wind controller can be cleared from the scene just like any object using **Items** > **Clear** > **Clear Selected Items**.)

2 On the FX Browser panel, choose **Collision** from the **Add** pop-up menu. This adds a collision controller to your scene. The controller name, Collision, should appear on the **Item** pop-up menu, so click the **Property** button to bring up its property panel.

3 At frame 0, move the Collision object up to about Y=1.725m. Play the scene and notice how the particles bounce off.



4 Play around with the Collision properties and note the effects. Also, adjust the emitter properties, like **Vibration(m/s)**.

More Tutorials

Obviously, these tutorials barely get your feet wet. However, because of the real-time feedback, you can mess around with the settings and get a good feel for how everything works. There are several example Particle FX scenes included on the CD. They will give you great insight as to how you can use ParticleFX. Also, please visit LightWave3D.com for additional tutorials.

chapter 18
Motion Designer

Chapter 18: Motion Designer

Motion Designer is a 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.

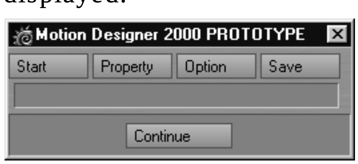
WARNING
Your Bounding Box Threshold (Display Options tab of the Preferences panel) must be set high enough to display your Motion Designer targets. If not, you may encounter a *can't scan objects...* message dialog.

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 **Scene** > Dynamics: **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.

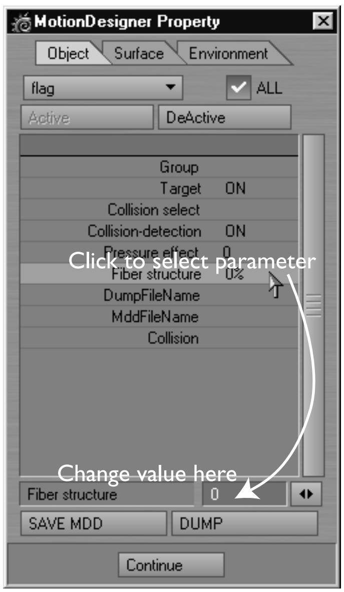
Click **Property** to bring up the MotionDesigner Property panel. Here you set the details of how MD affects the items in your scene.

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.



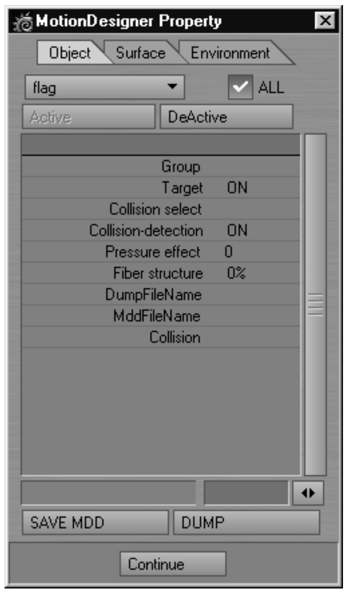
You can click some parameters in the list to toggle their state on or off. For filename parameters, you can double-click or press Enter to bring up a file dialog.

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

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.

Collision-Detection

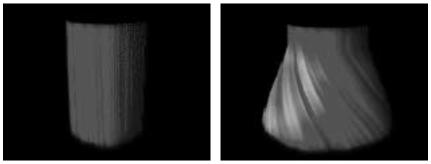
Use the **Collision-Detection** function to specify collision detection on an object-by-object basis instead of surface-by-surface. If this parameter is OFF, all of the object's surfaces are treated as if the (Surface tab) **Collision-detection** setting was OFF.

Pressure Effect

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.

Fiber Structure

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.



Left: Fiber structure = 0%. Right: Fiber structure = 100%

DumpFileName

DumpFileName is a special file that sets up an initial pose for the target object. This file is discussed below.

Collision

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:

1. Set your Last Frame in Layout to the pose you want to start with.
2. Start the calculation.
3. After the calculation, click **DUMP** and save the Dump file.
4. 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.

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

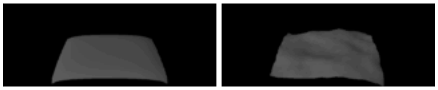
Weight

Weight defines the weight of the material. The flags below are the same except for the **Weight** parameter. The flag on the right hangs down more due to its weight and also will not flap as much.

Left: Weight = 1. Right: Weight = 2

Weight +-

Use the **Weight +-** parameter to add randomness to the **Weight** parameter.



Left: Weight+- = 0. Right: Weight+- = 1

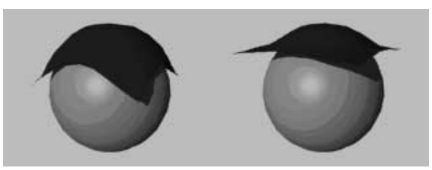
Spring

The **Spring** coefficient controls the *springiness* or stiffness of the material. The flags below are the same except for the **Spring** parameter. An extremely large **Spring** value on the right flag makes its cloth stiffer.

Left: Spring = 1000. Right: Spring = 10000

Viscosity

Viscosity controls the impact of a collision. 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.



Left: Viscosity = 1. Right: Viscosity = 10

If an object bounces, a higher **Viscosity** value will have less bounce motion because the bouncing force is *absorbed* by the **Viscosity**.

Resistance

Resistance controls the amount of air resistance. The flags below are the same except for the **Resistance** parameter. The flag on the right stretches more due to its greater air resistance.

Left: Resistance = 1. Right: Resistance = 2

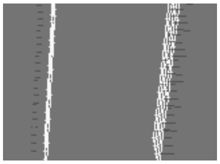
Parallel Resistance

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.

Two falling tiles. Left: Parallel-resistance = 100%. Right: Parallel-resistance = 0%

Back-resistance

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.

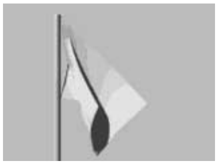


The back side for these objects is the side closest to the center of this image.

Structure Settings

Fixed

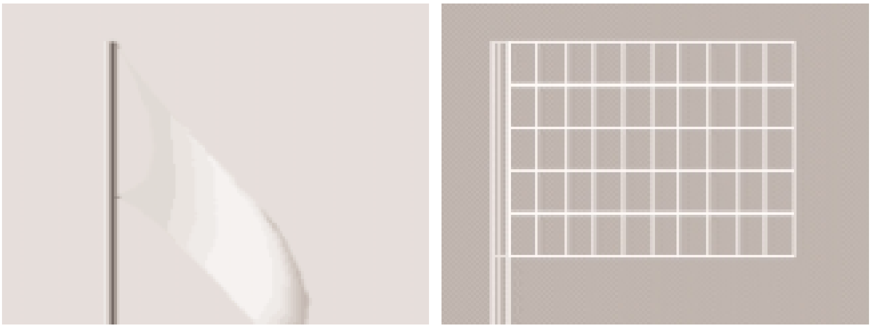
The shape of polygons with **Fixed** On remains intact. The flag pole below, for example, has **Fixed** On and the flag has it off.



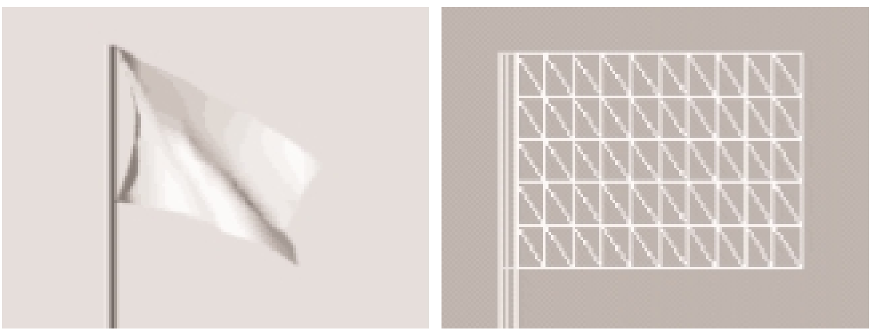
Sub-Structure

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.

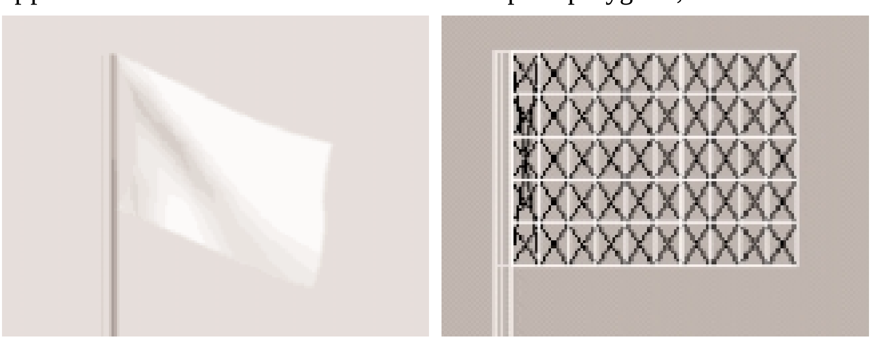
The flag below is made up of square polygons. Due to distorted polygons, the flag appears to be abnormally stretched as if it is made of a net.



The problem can be corrected (partially) by tripling the geometry in Modeler to eliminate the abnormal stretching. Unfortunately, triangles have a tendency to bend in one direction and cause wrinkles. They also resist bending in the other direction, which may result in an unnatural motion.



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. Notice below that the cloth appears smoother now than it did with tripled polygons, above.



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.

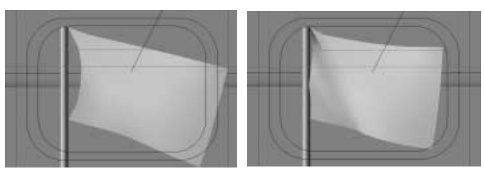
NOTE
The effect of Hold-structure is uniform throughout the surface, while with Sub-structure it is non-linear, which will often result in a more natural look. You can use a combination of both to achieve just the right result.

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.

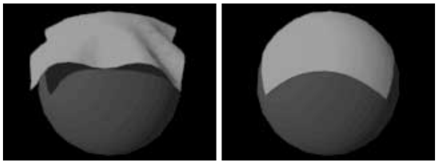


Left: Stretch-limit = 100%. Right: Stretch-limit = 20%

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.

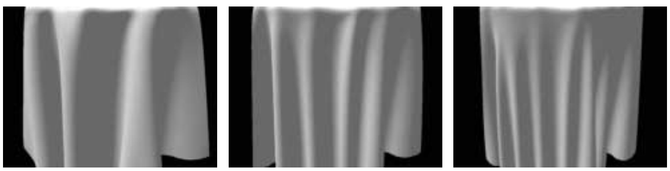
Compress Stress

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.



Left: Compress stress = 100%. Right: Compress stress = 0%

You can easily adjust the apparent thickness of a drape by adjusting **Compress Stress**, as shown below. (The drape object also uses these settings: **Weight** = 1, **Spring** = 200, **Resistance** = 2 and **Viscosity** = 1.)



Left: Compress Stress = 100%. Middle: Compress Stress = 10%. Right: Compress Stress = 1%.

Shrink

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.

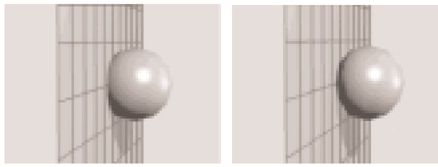
Self Collision

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.

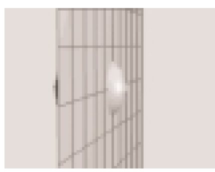
Collision Detection

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.



Left: No collision detection. Right: With collision detection

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 shown below.



As such, if you need collision detection inside a container, for example, you'll need to have interior facing surface normals.

Single-sided

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.

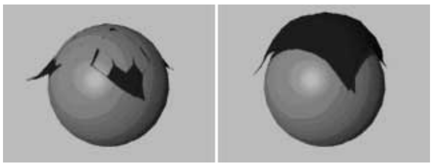
Left: Surface normals with the same direction. Middle: Single-sided = ON. Right: Single-sided = OFF

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.

NOTE
In the picture below, you can see the points on the cloth do not penetrate the ball, but the surface between the points does.

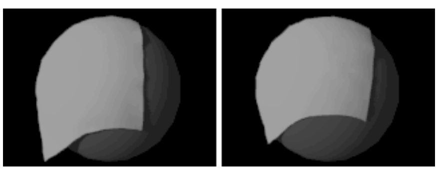
Use the **Skin Thickness** (Surface tab) setting to avoid undesired surface penetration like the example below. **Skin Thickness**, set in meters, creates a gap between the collision and target objects, within which a collision is deemed to have occurred. Make sure you specify **Skin Thickness** for the surface of the collision object, not the surface of the target object.



Left: Skin Thickness = 0. Right: Skin Thickness = 0.1(m)

Friction

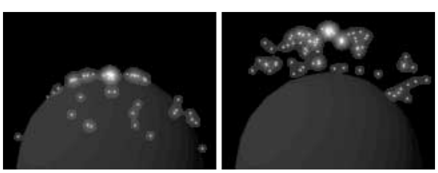
Like real world friction, the **Friction** parameter makes the surface less slippery. So if you want the target object to tend to slip off the collision object, set **Friction** to 0. If you want it to stick more, increase the value.



Left: Friction = 0. Right: Friction = 3

Bound Force

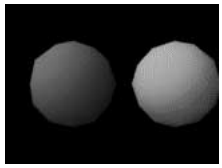
Bound force adds a *rebounding* speed change at collision, if set greater than 0. At 1, the speed is the same as the collision object. At 2, it rebounds at the collision speed.



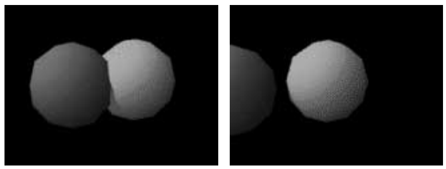
Particles dropped on a ball. Left: Bound force = 1. Right: Bound force = 2

Action Force

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.



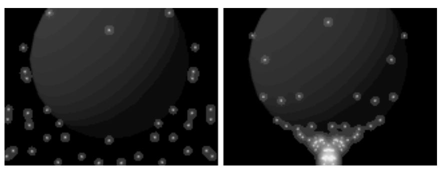
Object on left is stationary and object on right is moving left



After collision. Left: Action force = OFF. Right: Action force = ON

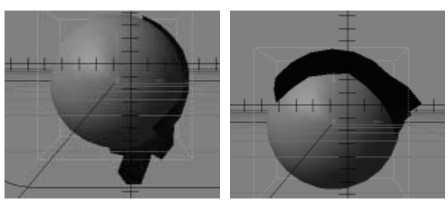
Bind Force

Bind force causes the target to adhere to the collision surface.



Particles dropped on a ball. Left: Bind force = 0. Right: Bind force = 1

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.



Left: Fix force = 0. Right: Fix force = 100

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.

COPY/PASTE

Use the **COPY** and **PASTE** buttons to copy and paste the current settings between surfaces.

SAVE/LOAD

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.

Gravity

Specify the direction and intensity of gravity along the world axes.

Wind1/Wind2

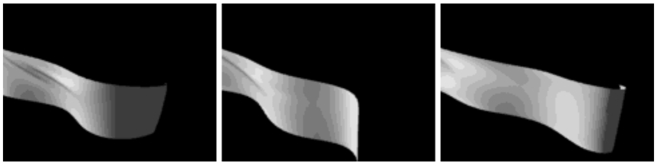
Specify the direction and intensity of two winds along the world axes.

Turbulence

Specify the direction and intensity of turbulence along the world axes.

Left: Turbulence = 0,0,1. Right: Turbulence = 0,1,0

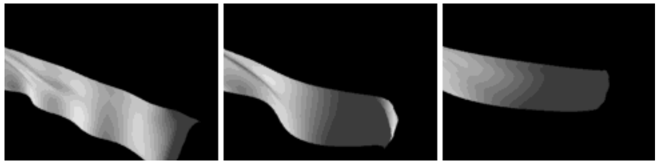
The strength of turbulence is affected by the wind, as shown in the examples below.



Left: Wind = 0,2.0,0. Middle: Wind = 0,5.0,0. Right: Wind = 0,8.0,0

Wavelength

Specify the wavelength of the turbulence.



Left: Wavelength = 0,0.25,0. Middle: Wavelength = 0,0.5,0. Right: Wavelength = 0,1.0,0

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.

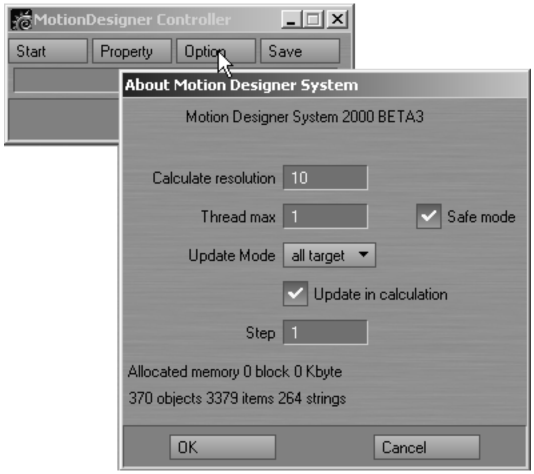
Since Motion Designer uses elastic body models as calculation models, based on mass, spring, and resistance, unexpected vibration may appear in the calculated motions. One of the following adjustments may help correct the problem:

- 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.

SUPPLEMENTAL MD DISPLACEMENT PLUG-INS

MD_Plug

MD_Plug is the displacement map plug-in that applies the motion data created by MD (MDD file) to the object in Layout.

NOTE
MD_Plug 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 plug-in manually.

Setting Options

Double-click MD_Plug after it has been added on the Deformations tab of the Object Properties panel to display its options panel.

The **MDD Filename** field specifies the Target's MDD file.

The **Action Start** field indicates the time (in seconds) at which the motions from the MDD file should start. This lets you delay the start of the effect.

The **End Behavior** setting determines what happens after the motion finishes. **Stop** holds the last state. **Repeat** repeats the motion from the beginning. Use **Composite** if you wish to *stack* multiple instances of MD_Plug and add the motions together.

If **Key-Move** is ON, keyframed motion is applied along with the MD displacement motion. Since the motion data created with Motion Designer (including MD_Scan) contains displacement of shapes using the keyframes, setting this ON can cause a doubling up of the keyframed motion. To prevent this problem, set **Key-Move** to OFF.

NOTE
When using MD with other displacement plug-ins (which should be loaded after MD) be sure Key-Move is ON. Motion Designer ignores any plug-ins loaded above MD_Plug.

MD_MetaPlug

MD_MetaPlug is a displacement map plug-in 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 plug-in 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 plug-in off without losing your settings.

MD_MetaPlug_Morph

MD_MetaPlug_Morph is a displacement plug-in 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 plug-in 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 plug-in.

NOTE
Using MD_Metaplug_Morph is CPU intensive, so be sure to select the appropriate Morph Mode.

MD_Scan

MD_Scan is a displacement map plug-in that incorporates motion data from Layout into Motion Designer. MD_Scan can handle motions, such as those of bones affected by displacement map plug-ins. The resulting data becomes an MDD file to be used with the main Motion Designer plug-in.

The **MDD Filename** field specifies the MDD filename.

Scan Points displays the number of points in the object to be saved.

Specify the frame where you want to begin recording in the **First Frame** field and the frame you want to end recording in the **Last Frame** field.

Specify the number of frames for which data should be saved in the MDD file in the **Frame Rate** field. Data for skipped frames is automatically interpolated. Increasing this number can reduce the data size when you save motions that are long but relatively slow.

Scanned Frame displays the fetched frame.

NOTE
When you use MD_Scan with other displacement map plug-ins, be sure to add MD_Scan last.

To use MD_Scan:

1. Add MD_Scan to the Deformations tab of the Object Properties panel for the object whose motion you wish to incorporate.
2. Double-click the added plug-in to open its Options panel.
3. Define the desired **MDD filename** and set the parameters. Click **OK**.
4. Click **OK** for the next dialog that appears.
5. Make a preview animation. (Yes, this is necessary!)
6. Open the MD_Scan Options panel again. This time a different version of the panel will appear. Click **OK** to save the MDD file data.

7 Remove MD_Scan or deactivate it.

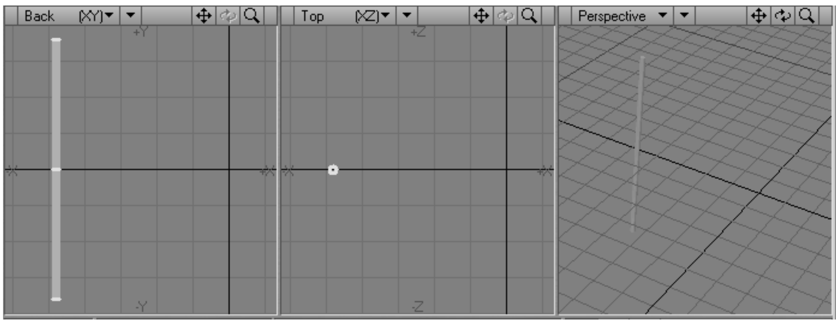
8 Load the file into MD_Plug.

TUTORIALS

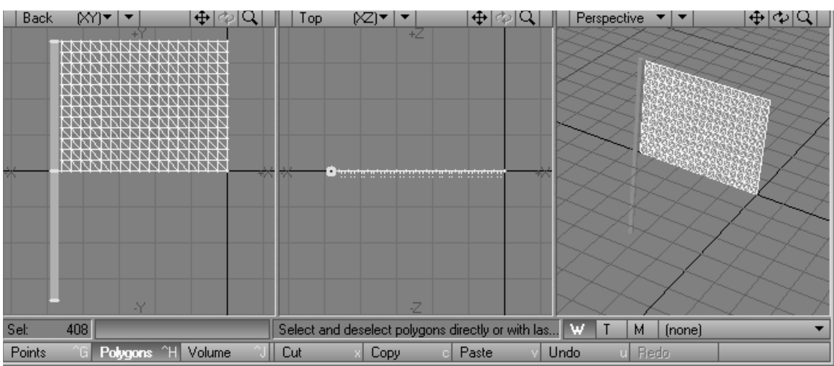
Here are a few basic tutorials to get you started with Motion Designer.

Waving a Flag

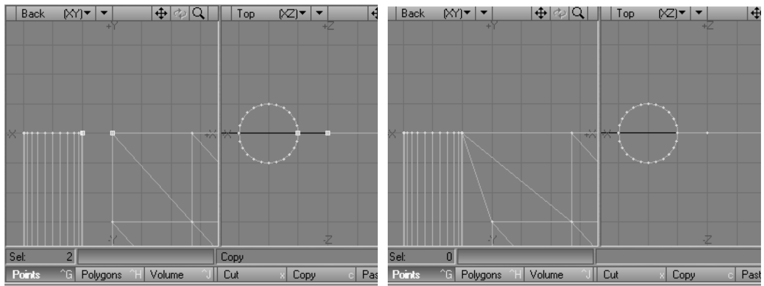
1 In Modeler make a flag pole using the Disc tool. Use two Segments. Name the surface *Pole*.



2 Using the Box tool, make the flag. It should use lots of X and Y Segments and line up with the top and middle of the pole. Triple the polygons (**Construct** > Subdivide: **Triple**). Select the flag polygons and name the surface *Flag*. Make sure the polygons face towards you. If they do not, flip them.



3 Zoom in and select the top-left point on the flag and then the nearest point on the pole. Make sure only two points are selected and choose **Detail** > Points: **Weld**.


Left: Before welding. Right: After welding

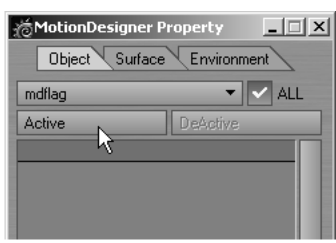
4 Weld the bottom-left corner of the flag to the pole in the same manner as above.

5 Save your object as MDFLAG.LWO and load it into Layout.

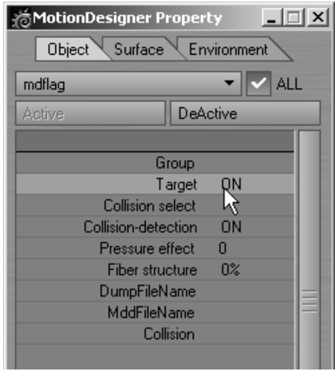
6 Choose **Scene** > Dynamics: **MD_Controller**. This brings up the MotionDesigner Controller panel.

HINT
For easy access, you may want to assign the MD_Controller command to a menu or keyboard shortcut.

7 Click the **Property** button on the panel to bring up the MotionDesigner Property panel. With only one object in the scene, our flag object will already be selected on the pop-up menu. Click the **Activate** button to use it with MotionDesigner.



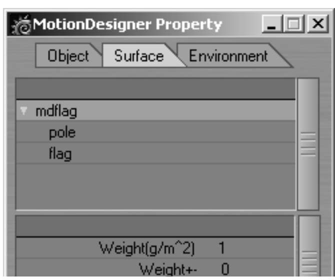
8 Next, double-click the **Target** line to turn this option on. Additional options will appear, but you can leave them at their default state.



NOTE

That is one click to select the option and another to toggle its setting. This is the way all of MD's On/Off settings work. If the item is already selected, you need to click only once.

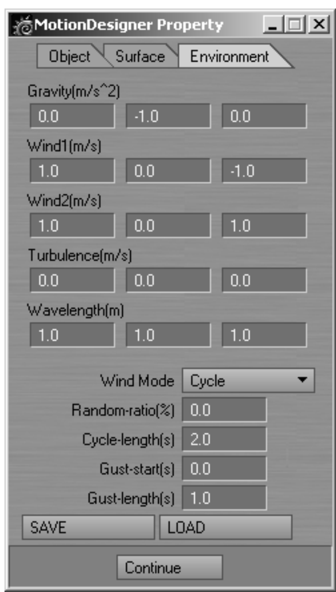
9 Click the Surface tab and then the MDFlag entry to display the two surfaces: Pole and Flag.



10 Choose the Pole surface and turn On the **Fixed** setting.

11 Choose the Flag surface and click the **Spring** setting. This setting requires a numerical input. Enter 1000 in the input field beneath the list of settings. A value of 1000 will make the surface resist stretching like a piece of cloth.

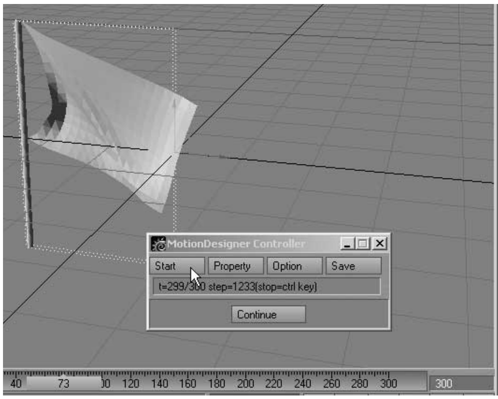
12 Click the Environment tab. Each setting has three fields that are for the X, Y, and Z axes, from left to right. Set **Gravity** to 0, -1, 0; **Wind1** to 1, 0, -1; and **Wind2** to 1, 0, 1. **Cycle-length** should default to two seconds; this value makes the environment settings repeat every two seconds.



13 Click the **Continue** button to close the MD Property panel.

14 In Layout, set your last frame to 300. Go to the Display Options tab of the Preferences panel (D) and make sure **Bounding Box Threshold** is high enough to display your MotionDesigner target.

15 Click the **Start** button on the MotionDesigner Controller panel and watch the magic!

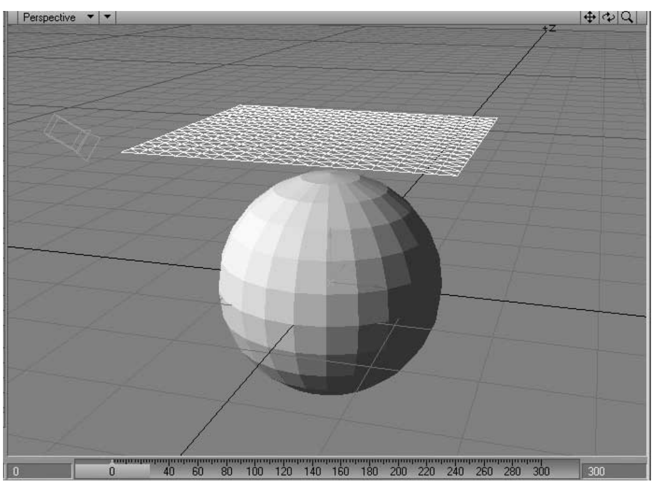


Go ahead and play with the surface settings to change the effect.

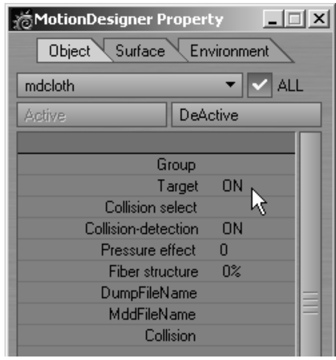
Collision Detection

In this tutorial, we will show you how to do basic collision detection.

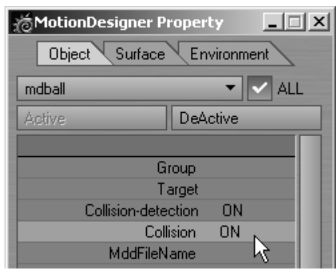
1 Here we have a simple flat segmented box with tripled polygons and a basic ball. The objects were modeled in these positions. The box object is called *mdcloth* and has a surface named *cloth*. The ball object is called *mdball* and has a surface named *ball*. (Clever naming, eh?)



2 Choose **Scene** > Dynamics: **MD_Controller**. From the MotionDesigner Controller panel, click the Property button. **Activate** the mdcloth object and turn the **Target** option on.

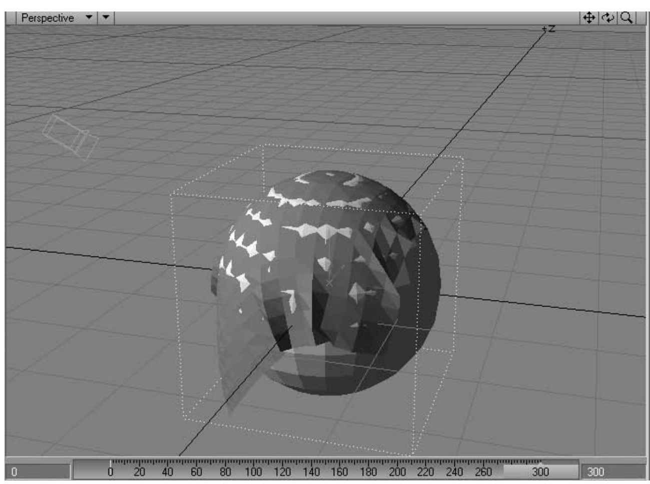


3 **Activate** the mdball object and turn the **Collision** option on.



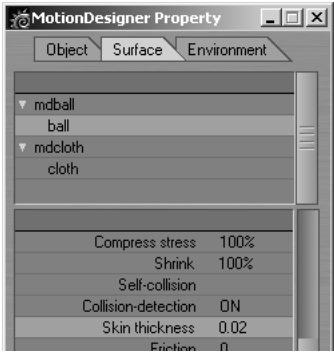
4 Go to the Environment tab and set **Gravity** to 0, -1, 0.

5 In Layout, set the last frame to 300 and click **Start** on the MotionDesigner Controller panel. You will see the cloth float down, due to the gravity, and deform around the ball.

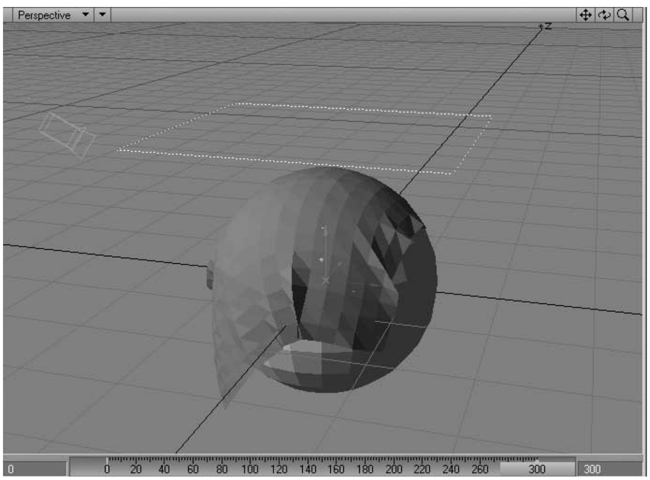


6 You should notice that some parts of the cloth surface go inside the ball. We can easily fix this by increasing the thickness of the ball surface for calculation purposes only.

Select the ball surface and increase the **Skin thickness** to about .02.



7 Click **Start** again and the cloth surface should no longer go inside the ball. If it still does, increase **Skin thickness** some more.



You should see your cloth slide off the ball towards the end of the animation. Play with the surface settings to change the effect.

More Tutorials

Please visit LightWave3D.com for additional tutorials.

chapter 19

Distributed Rendering

Chapter 19: Distributed Rendering

LightWave allows you to use the processing power of other computers on a network to render scenes. This is called *distributed rendering* or sometimes a *rendering farm*. There are a few basic approaches to distributed rendering with LightWave.

ScreamerNet uses Layout's Network Rendering panel (**Rendering** > **Network Rendering**) to control submitting scenes to networked computers running the ScreamerNet process. ScreamerNet can control up to 1,000 CPUs (Note: A single machine can have more than one CPU. Each CPU is counted as one). Each will render a frame from the animation until the scene is rendered.

NOTE
You can also use LightWave's distributed rendering feature to batch-render multiple scenes on a single machine.

Rendering Modes

You can choose from two distributed rendering modes: Original ScreamerNet and ScreamerNet II.

WARNING
Scenes, objects, and images should not have any spaces in their saved names; use an underscore '_' instead (e.g., MY_3DCAR.LWO instead of MY 3DCAR.LWO). This is vital for dependable ScreamerNet operation. If there are spaces in these names, ScreamerNet and LightWave may read them as corrupt files.

ScreamerNet Classic

Original ScreamerNet is designed to be used with networked CPUs that do not share directories. Original ScreamerNet has the disadvantage of requiring you to load the scene to be rendered on the control machine and sending the scene file, object files, and image files to each connected

CPU. When a frame is rendered by a remote CPU, it is sent back to the control machine whereby it is saved to a selected directory. This process, while it works, is often slow due to the constant file transfers.

NOTE
Original ScreamerNet is included primarily for backward compatibility with older rendering engines.

Since the Original ScreamerNet rendering method is somewhat outdated, those still using it will find little change in the controls for this method. Changes will be noted below. The majority of this chapter deals with the new ScreamerNet II rendering method.

SCREAMERNET II

ScreamerNet II does not require the scene be loaded on the control machine. As long as the scene, object, and image files, (as well as the save-to directories) are all accessible to each CPU on the network, each individual node machine can load the scene and save images directly to the shared drive(s). Additionally, ScreamerNet II can “batch render” up to 100 separate scene files.

The process of using ScreamerNet II is basically to run the rendering module (i.e., LWSN) on all of the computers that will be rendering. This includes the designated control computer. You then run LightWave on the control computer and identify the scene or scenes you want to render.

Since ScreamerNet communicates by writing files, NetBEUI and TCP/IP are not required. As long as each machine can *see* each other and write files across the network, ScreamerNet should function properly.

The LWSN program is supplied with LightWave. However, you will only receive the version appropriate for the platform for which you bought LightWave. For example, if you purchase the Intel version of LightWave, you will receive the Intel version of LWSN.

WARNING
All of the rendering nodes need not have the same architecture (i.e., Intel, Mac, etc.). However you must run the proper LWSN version on each type of machine. In addition, plug-ins used in the scene must also reside on each node. If not, some images would render using the plug-in and others wouldn’t because they couldn’t.

Using Screamernet II

The Host is the machine with Lightwave installed on it. Nodes are the other machines on the network used for rendering. Node machines do not need to have any NewTek software or hardware on them.

The key to understanding ScreamerNet is network drive mapping and sharing. There are a variety of ways to configure a network and thus a variety of ways to set up ScreamerNet.

ScreamerNet Rendering Requirements

To set up a ScreamerNet rendering farm you need to set up the following:

1. A common directory that the Host (controlling PC) and all rendering nodes can access. This could be physically located on the Host, but is not necessarily.
2. Each rendering node must be able to load a plug-in configuration file (e.g., LWEXT3.CFG) that has the plug-in files mapped with a path that the rendering node can access.

   Now, if all rendering nodes are on the same platform (e.g., Intel Windows), you could map the same drive letter to another common directory and place the plug-in, LWSN.EXE and configuration files there. The paths in the configuration file would obviously also use this drive letter.

   If you don’t use the common configuration file route, you’ll need to store a set of all of the files locally on each rendering node.
3. All of the rendering nodes need access to a Content Directory. Ideally this can be stored in one common directory since you want to make sure all of the rendering nodes are using the same objects and images.

NOTE
The Host machine can be a rendering node as well. It may be easiest to use the same mapping technique described above, although you could map everything locally if you want.

Getting the Nodes Going

The ScreamerNet utility is executed on each rendering node using a lengthy command. The following is an example. Note that the command has been broken up into separate lines for clarity. In reality, the entire command must be entered on a single line.

```
L:\ScreamerNet\LWSN.EXE -2
-cL:\ScreamerNet
-dL:\ScreamerNet\Content
L:\ScreamerNet\Common\Job1
L:\ScreamerNet\Common\Ack1
```

In the above example, we’ve assumed the following:

- All of the rendering nodes are Intel Windows machines.
- The drive letter L: has been mapped to the same network drive for all nodes.

- The LWSN.EXE and configuration files are all stored in L:\SCREAMERNET.
- The plug-in files are stored in L:\SCREAMERNET\PLUGINS and that is the path used in the LWEXT3.CFG file. (Note: that the plug-in files could still be in subdirectories (L:\SCREAMERNET\PLUGINS\ANIMATE).

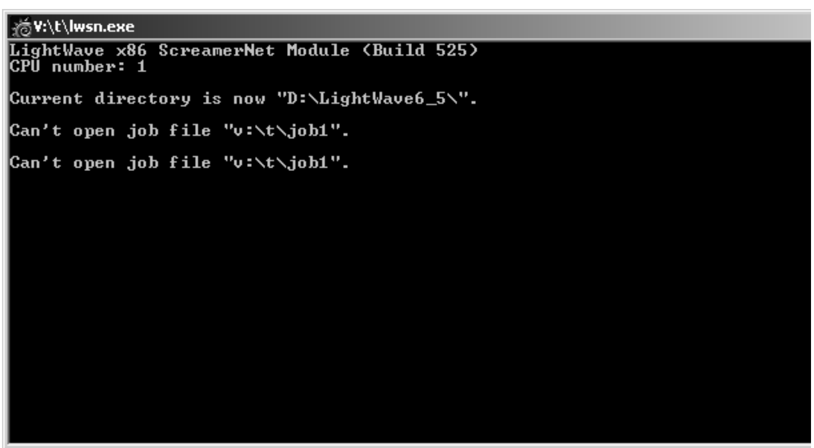
The -C parameter defines the directory where the configuration files are stored. The -D parameter defines the content directory to be used. (If this is not defined, the content directory defined in the scene file is used.)

The command for subsequent rendering nodes must increment the Job *and* Ack number. For example, node 2 would use JOB2 and ACK2, node 3 would use JOB3 and ACK3 and so on. The Job/Ack number equates to the CPU number used by ScreamerNet to identify the node and goes from 1 to the number set as your **Maximum CPU Number** on the Network Rendering panel, discussed next. (You must include the path for Job and Ack.)

To save time in the future, you will want to set these commands up as icon shortcuts.

NOTE
Mac users should use a command-line file, discussed in the Appendix, to set LWSN's parameters.

NOTE
After you execute LWSN.EXE you may see a message like *Can't open L:\ScreamerNet\Common\Job1*. This will be normal until Lightwave initializes ScreamerNet.



MS-DOS console window output

Scene Saving Info

1 When setting up your scene, you must use your content directory properly. Otherwise, some nodes may not be able to render the scene.

2 Set up all of your render options (e.g., file-saving) before saving the scene. You will probably want to save the rendered frames to a common directory, like L:\SCREAMERNET\FRAMES. Alternatively, they could be saved locally, but remember all rendering nodes will be getting the path to save the files from the same scene file.

NOTE
ScreamerNet does not render out animations. It renders only individual files.

Host Machine Setup

1 Start LightWave. Open the Network Rendering panel (**Rendering** > **Network Rendering**) and click the **Command Directory** button. Change this to Z:\LIGHTWAVE\PROGRAMS. You will get a requester that asks you to re-initialize; choose **NO**.

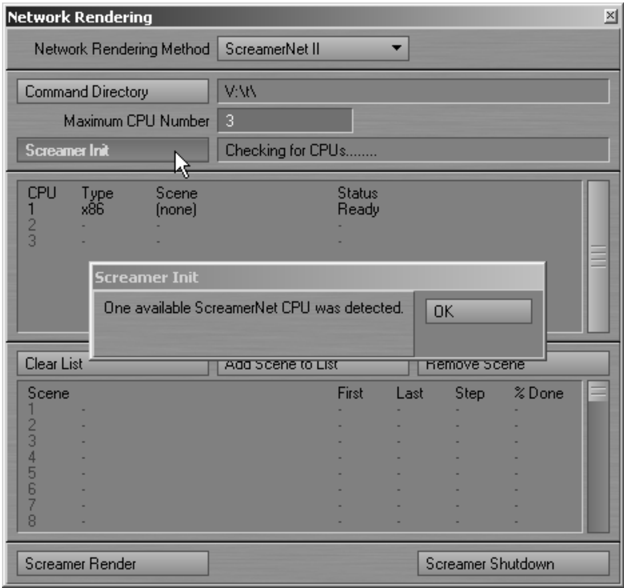
2 Set the **Maximum CPU Number** to your number of nodes/CPUs.

3 In the General Options tab of the Preferences panel, confirm that the **Content Directory** is set to the proper directory.

4 Quit LightWave and then start it up again to save changes to the configuration file.

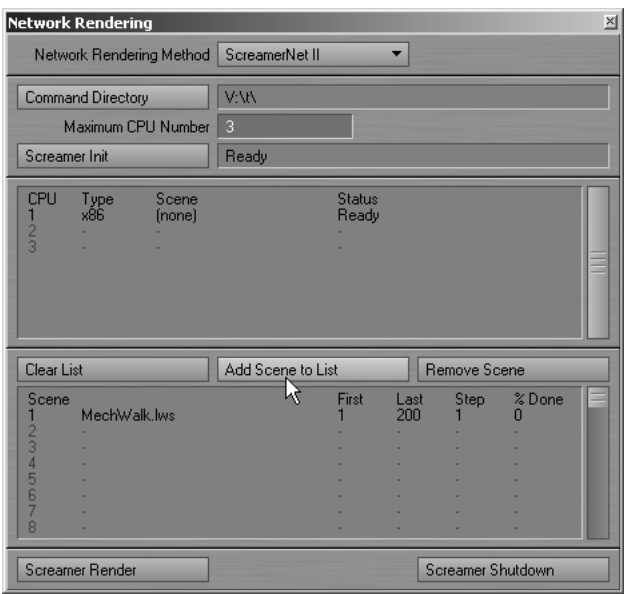
Controlling the Network Rendering

1 On the Host, open the Network Rendering panel (**Rendering** > **Network Rendering**) and click the **Screamer Init** button. At this point, all of your CPUs that are running ScreamerNet should appear in the window. (The job/ack number assigned to each node machine will correspond to the CPU number in the SN window.)



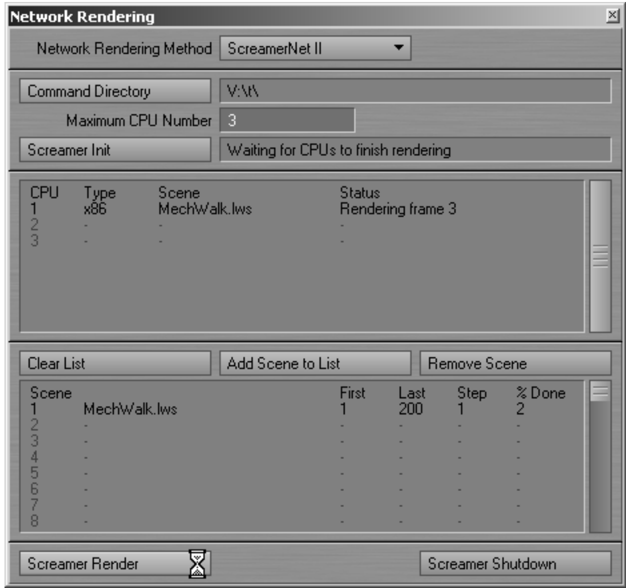
Network Rendering panel

2 Click **Add Scene To List** and add a scene(s).



Scene added for rendering

3 Click on **Screamer Render**. This will now render your scene(s).



ScreamerNet now rendering the scene

Shutting Down Nodes

When all CPUs are finished rendering, selecting the Screamer Shutdown button will close all windows on all ScreamerNet nodes and stop them. In order to start a new session, you must restart ScreamerNet on each CPU and re-initialize the CPUs from the control machine.

Aborting a Rendering Session

Press the ESC key to abort a ScreamerNet session. There may be a long pause before all of the ScreamerNet nodes are ready again, while each finishes up the current task. This is especially true when they are working on a complex scene. Note: ScreamerNet rendering is only abortable when the ScreamerNet panel reads *Waiting for CPUs to finish rendering*.

WARNING
If you close the DOS shell on any of the render nodes, you may have to restart all nodes and re-initialize them.

Changing the Number of Nodes

By changing the value in the Maximum CPU Number field from the default eight, you can tell the control machine how many nodes it should check for. The maximum number is 1000, but setting the number to the actual CPU's available nodes will save time. This number is written to the configuration file upon exiting LightWave.

ScreamerNet II Syntax

The command line parameters for LWSN.EXE -2 are:

LWSN -2 [-C<CONFIG FILE DIR>] [-D<CONTENT DIR>] [-Q] [-T<CHECK INTERVAL>] <JOB FILE> <REPLY FILE>

The check interval is defined in seconds. It specifies how often to check for commands during rendering. If the -T parameter is not used, rendering is uninterrupted and LWSN won't check the job file until its current render job is done. With -T, it will check every so often during rendering for only two commands: status and abort. (All other commands are ignored). (The actual check interval may be longer than specified, depending on what is currently being processed.)

BATCH RENDER ON ONE COMPUTER

You do not actually need a network in order to use the ScreamerNet. Just follow the network instructions as if the computer was the node and control machine. It's a little easier since you won't have to worry about shared directories and volume names. ScreamerNet is useful for rendering a series of scene files unattended, and it is most beneficial if you are using a dual-processor system—each processor could be treated as a separate CPU for rendering.

TROUBLESHOOTING

If clicking the **Screamer Init** button doesn't find the other CPUs, go back and start at the *Host Machine Setup* section.

If images seem to render unrealistically fast and no images are saved:

1. Check to make sure that you have full sharing access across the network. This is done through Windows NT Explorer. You can check this by copying a file at random back and forth across the network.
2. If the scene and/or objects were created without taking into consideration the new drive path names, rendering may occur on only the host machine.

The most common cause of ScreamerNet crashing is when too many computers try to write or read their information to/from the Host computers while the host renders.

- Do not use ScreamerNet to render on the host machine, but rather use it only as a server where the hard drives are found.
- Map a drive from a different computer as drive Y:, for example, and set your scene to save the animations to that drive. The computers don't know that drive Y: is not on the host, just that it's present.

Another problem occurs when the hard drive where you save the images or animation is full. This problem creates an error in LightWave and each ScreamerNet node.

Rendering Without LightWave

The LWSN program has a third option that lets you render a scene without running LightWave. There is no control machine and thus it is not a distributed rendering situation. You must tell the program specifically what to render. The method is run from a DOS prompt using the following syntax (one line):

LWSN -3 [-c<config file>] [-d<content dir>] <scene file> <first frame> <last frame> [<frame step>]

As you can see, you basically supply the program with the basic information needed to render a scene. An example would be:

LWSN -3 -cD:\Lightwave\Lw.cfg -d M:\Newtek Spicegirls.lws 1 900 1

In the example, the program would render frames 1 through 900 of the Spicegirls.lws scene using the Lw.cfg file stored in the D:\Lightwave and using M:\Newtek as the Content Directory.

The configuration file specification is optional, if the Lw.cfg file is in the current directory. Likewise, if the Content Directory is correctly specified in the configuration file, you do not need to give that parameter.

HINT

You can get the syntax for ScreamerNet by simply typing LWSN with no arguments.

chapter 20

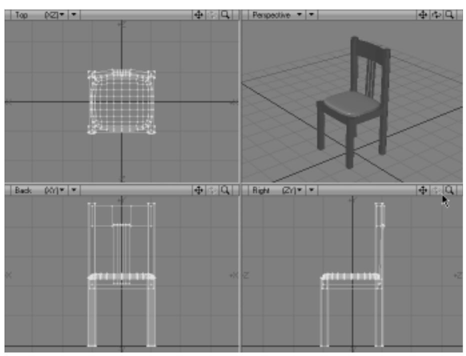
LightWave 3D Modeling

Chapter 20: LightWave 3D Modeling

LightWave Modeler lets you create objects from scratch or edit existing objects. Object modeling is the design and creation of wireframe objects from a simple shape, like an apple, to a complex shape, like a finely-detailed sports car. In Layout, you are the interior designer moving furniture and putting up pictures. In Modeler, you are the furniture maker and the builder of the house.

Components of a 3D Object

What makes up the 3D object? It has two traits: its physical shape and its visual appearance. The physical shape of an object comprises points joined by lines to form faces that we call polygons. The form that the polygons present to us makes up an object's shape.



The visual appearance of an object consists of color and texture qualities that connote realism to the eye, which are called surface attributes or material properties. When you enter a room and look at your chair you see a certain shape (that of a chair) and a certain surface (maybe wood or metal), so you think, "chair". In LightWave 3D you have the tools to create simple or complex objects and define their surface qualities with great precision and detail.

MODELING IN 3D

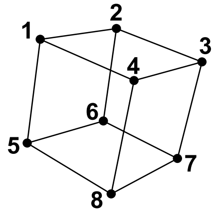
How would you make a mug? It depends; if you were a sculptor, you would mold a lump of clay. If you were a designer, you would draft it with paper and a mechanical pencil. If you were an artist, you would probably sketch it. But suppose you were using a computer. How would you make a mug then? What tools would you use?

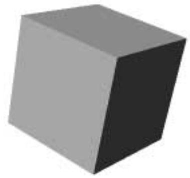
You want to portray the mug with realistic color, depth, and shading. You need drawing tools, yes, but also power tools! With the computer, you can create the mug in a variety of ways. LightWave recreates drafting and workshop tools on screen so that people who are neither sculptors nor traditional artists can use them just as easily. In fact, LightWave's tools do more than their real-life counterparts. There are several bonuses to computer design also. You can *undo* a mistake, something that is difficult to do in many of the arts. And there's no mess to clean up.

POINTS AND POLYGONS

Generally, 3D modeling is based on two simple elements: the *point* and the *polygon*. A point is a location in space, just as a point in a dot-to-dot coloring book represents a location on the page. Points are used as anchors to create polygons. Since points alone do not have height, width, or depth, they cannot be *seen* or rendered. However, you can load them into Layout and use points as invisible influences on other objects: as a center of rotation, or to play an invisible role of influence in an object hierarchy, for example.

The process of creating 3D objects is a lot like drawing in dot-to-dot coloring books—of course, with 3D modeling, the dots can also be placed in a third dimension (i.e., depth). Once placed, the dots can be connected with lines to form polygons. That collection of polygons forms a recognizable shape that the computer can draw and animate. Polygons are usually three-sided triangles or four-sided quadrangles (although they can consist of more sides).





You create or edit all objects using the same XYZ coordinate system used in Layout. The difference is that in Layout you are usually positioning an entire object, whereas in Modeler you are positioning the points that make up an object. Modeler also uses the XYZ coordinates 0, 0, 0 as the Origin.

NOTE
The Origin is the default rotation center (called the *pivot point*) for objects in Layout. Keeping this in mind as you model your objects will make the objects load into Layout properly without having to move the pivot point.

EDITING OBJECTS

Generally, to build a 3D object, you create, combine, and modify simple shapes into more complex shapes. This is known as editing. The basic building blocks (i.e., *primitives*) include cubes, spheres, cylinders, and other shapes. Using tools not unlike those found in a wood or metal shop, you form a realistic representation of the envisioned object.

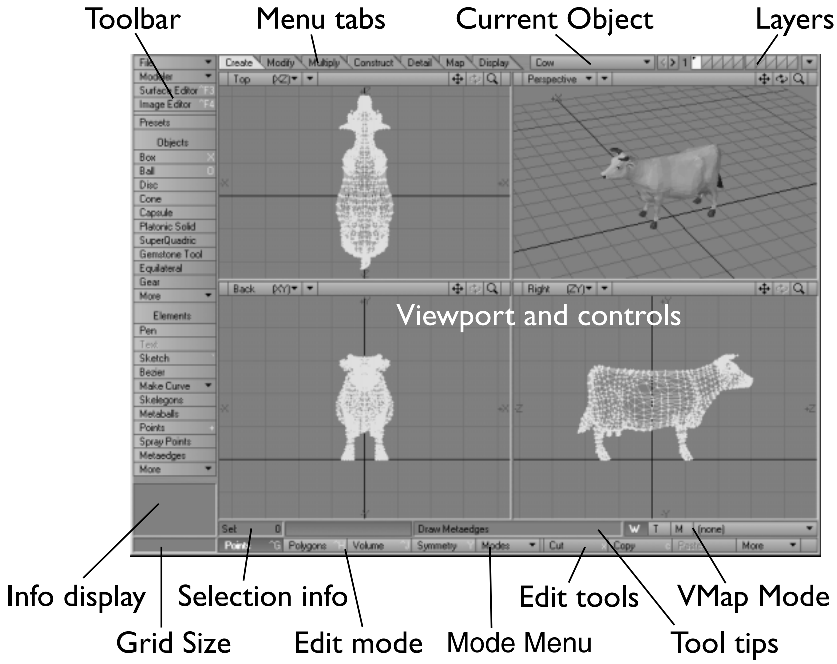
HAVE A PLAN

If you were building a desk, you would work from a sketch, a photograph, plans, or a small model of the desk you intended to build. This approach also works well for modeling 3D objects. Having the actual shape or a representation of the object in front of you while you design is invaluable, and it will aid you in determining size, shape, angle, color, and more. Whenever possible, work from pictures or small-scale models. Picture books, magazines, blueprints, and visual dictionaries are useful companions to have nearby.

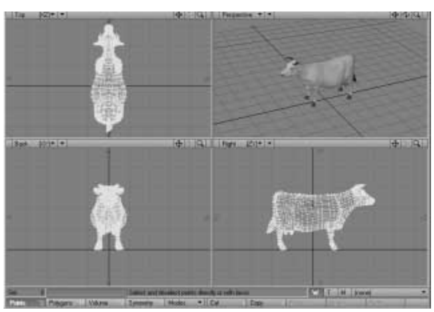
As in any project, you either begin from scratch or modify an existing object. If you're new to 3D, building from scratch will appear tedious at first. The Modeler exercises start off by using existing objects so you can get the hang of how modeling tools work.

THE MODELER INTERFACE

Modeler's screen is divided into several areas. By default, the workspace of the screen consists of four viewports. Modeler displays four simultaneous points of view (Top, Back, Right, and a forced-perspective view), each looking at the same portion of the workspace from a different angle. These are not unlike Layout's viewports. Changes made in one view are immediately updated in the others.



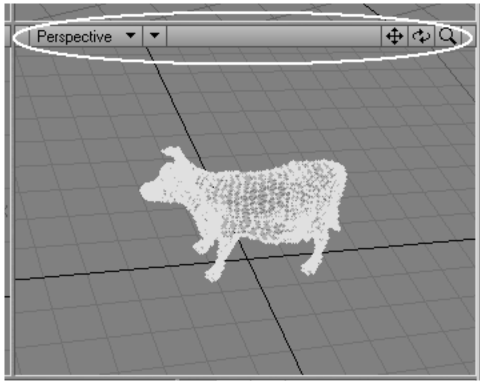
The toolbar sits at the side of the screen. The buttons presented will vary depending on which menu tab you select along the top. You can completely hide (or unhide) the toolbar by pressing ALT+F2 (or choosing **Modeler** > **Interface** > **Hide Toolbar On/Off**).



Without Toolbar

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. The buttons on the right side affect panning, rotating and zooming your view. See Chapter 27 for more information.



Viewport titlebar controls

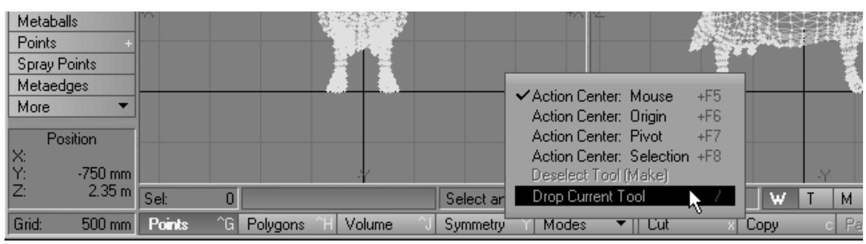
Modeler Menus

Most of the menus should be fairly obvious. The **Construct** menu contains commands that perform large-scale operations, grouped by action type. The **Detail** menu contains commands that change the properties of objects, grouped by object type.

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. So, to create your object you would start on the **Create** menu tab; to detail your object, you would move to the **Detail** menu tab, and so on.

The Utility group, at the bottom of the **Construct** menu, contains miscellaneous commands that do not fit elsewhere on the menus. The **Additional** pop-up menu resides here for unclassified commands.

Make particular note of the **Mode** menu along the bottom border. It's got some important stuff on it and is easily missed.



Other Interface Areas

Above the workspace on the right are the current object pop-up menu and the layer buttons. (More on this later.) In the bottom-left corner of the screen you can see the grid size information display. As in Layout, this tells you how big each square is in the Modeler workspace. Just above the grid display is an information display. Most of the time this gives you feedback on the position of your mouse. However, depending on what you are doing, it can also display a variety of other information.

To the right of the grid display are the three mode buttons. These are mutually exclusive buttons, so only one can be active at any one time. These determine when you are editing points, polygons, or a user-defined area. Above them is the selection info display, which normally tells you how may points/polygons you have selected.

Along the bottom-right are the cut-and-paste buttons, as well as the undo/redo buttons. In the row of controls above this is the tool tips display and the Vertex map mode (**W**, **T**, and **M)** controls.

RESETTING TOOLS

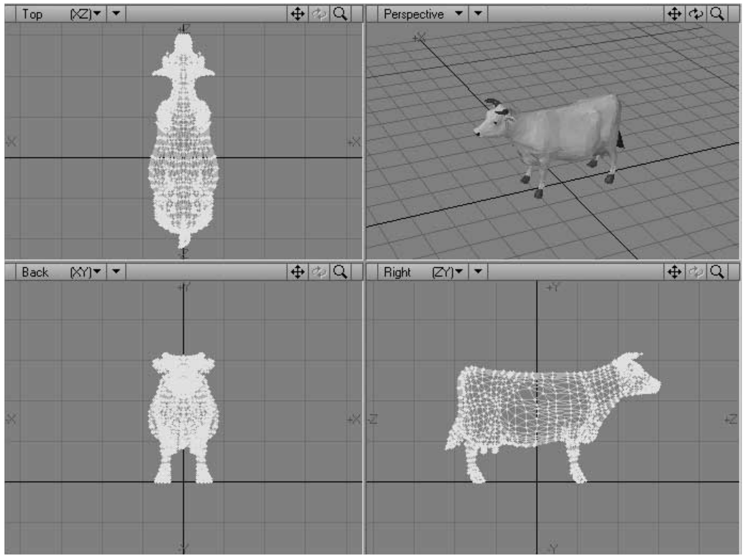
Non-interactive areas (e.g., not a button) on the toolbar and bottom border (including the left-corner information display and grid size) act as a reset button, much like you might use the ESC key on other applications. What happens when you click in a reset area depends on the current state of tools. For example, if you have points/polygons selected and you are in the Point/Polygon Selection mode, they become unselected.

All built-in tools (i.e., primitive and modifying) maintain their state, if they are used, unselected, and then used again. The default state of the tool can be recovered by *dropping* (clicking in a reset area or choosing **Modes** > **Drop Current Tool** when a tool is selected, but inactive. If the tool is active, the first drop deactivates the tool, keeping the current settings. The second drop would then reset to the default settings.

You can also choose **Modes** > **Deselect Tool**. This is just like clicking the active tool button, making it inactive.

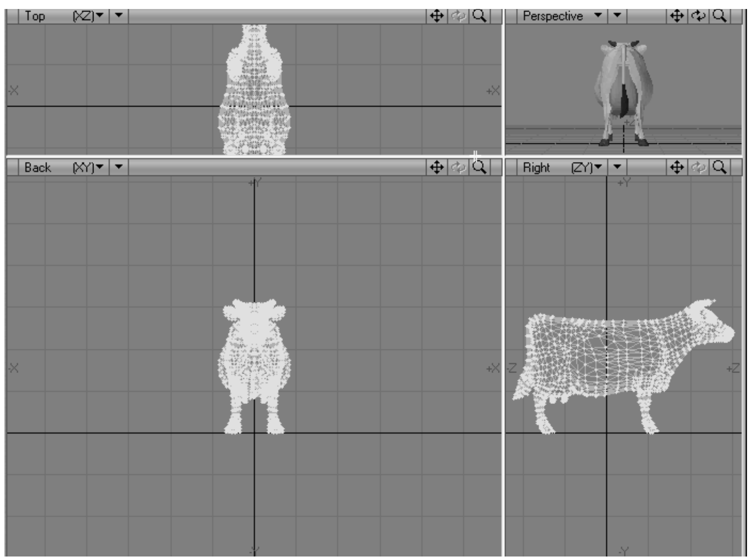
FOUR VIEWPORTS

The largest part of the interface consists (by default) of four viewports. You can edit in any viewport. The default three viewports starting in the top left and moving counter-clockwise are labeled Top, Back, Right, and Perspective. The Top view is an overhead point of view looking down on the object. The Back view looks directly at the object from the rear. The Right view is positioned as if you stood to the object's right. Perspective is a forced perspective view that you can rotate by holding the ALT key and dragging your mouse on it.



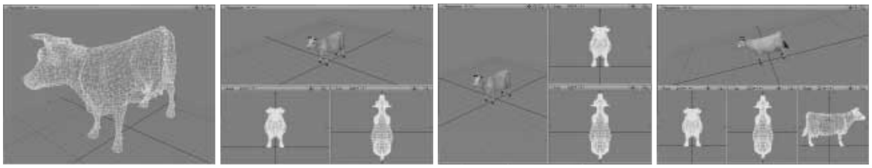
Default four viewports

You can change the relative sizes of the windows by dragging any border between the viewports.



Drag border to change viewport spacing

As you grow familiar and gain experience with Modeler, you will find yourself switching back and forth among these viewports, changing their size, position, and function, and even changing the number, to get the best view from which to use a particular tool. You can adjust the screen to see as much or as little detail as you want, by using various options.



Various viewport setups

NOTE
See Chapter 27 for information on how the viewports operate and options that allow you to change them.

MULTI-LAYER OBJECT STANDARD

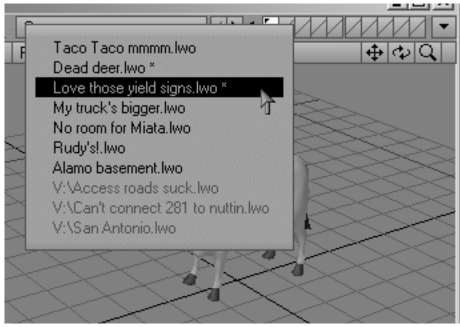
Every object can be a "MultiMesh," that is, consist of an unlimited number of *layers*, similar to layers in many paint programs. The MultiMesh lets you work independently on specific parts of an object. During modeling operations, you can set layers independently to be in the foreground or background, so you can work on a combination of layers as if they were in the same layer. Background layers can be used as a reference, but are often required when using certain modeling tools.

When you save an object, you save all layers with it. When you load an object, all of the individual layers remain intact; however, you may specify that certain layers be hidden when you use the object in Layout. You can thus save things like scratch objects—which you would not want rendered—in the same object file.

NOTE
When loaded into Layout, individual layers of a MultiMesh are treated as individual objects.

Multi-document Environment

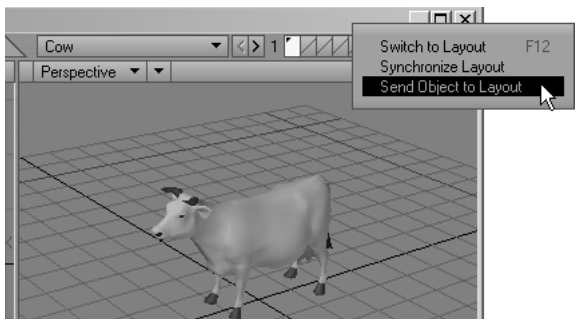
Like files in your wordprocessor, multiple object files can be loaded simultaneously. The Current Object pop-up button to the left of the layer buttons switches from object to object. Object names listed with an asterisk (*) have been changed and may need to be saved. Attempting to close the application triggers a request to save all changed objects. Ghosted objects are ones that have been loaded into Layout, but not into Modeler. Choosing a ghosted object from the pop-up menu will load it.



Current Object pop-up menu

Layout Communication

If the Hub (see Chapter 3) is running, a small pop-up menu button appears in the top-right corner of the interface, just to the right of the layer buttons.



Layout communication commands

Switch to Layout switches to your Layout window if it is open or opens up Layout if you do not already have it open.

Changes to objects in Modeler are reflected automatically in Layout when you select the Layout interface. You can force this synchronization by choosing **Synchronize Layout**.

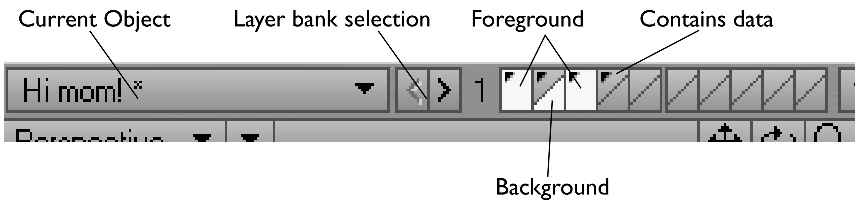
Send Object to Layout *sends* the current object to Layout. Actually, it sends a *pointer* to the current object's file. As such, the object must be saved to your hard drive first.

If you load an object in Layout only, its name is ghosted in the Current Object pop-up menu. Selecting a ghosted item will load it into Modeler and initiate the syncing process.

Layer Navigation

You can use the layer buttons in the top-right corner of the main interface to work with layers in banks of ten. You can apply most modeling functions across multiple layers simultaneously by placing them in the foreground. You do this by clicking the top part of a layer button; all other layers are unselected, whether they are in the foreground or background. You can add foreground layers by holding the SHIFT key as you click. You can unselect a foreground layer by holding the SHIFT key as you click on a selected foreground layer button.

To place a layer in the background, click on the bottom part of a layer button. Selecting a background layer will unselect all other background layers. You can add background layers by holding the SHIFT key as you click. You can unselect a background layer by holding the SHIFT key as you click on a selected background layer button.



NOTE
You can quickly swap the state of foreground and background layers by pressing the apostrophe key (').

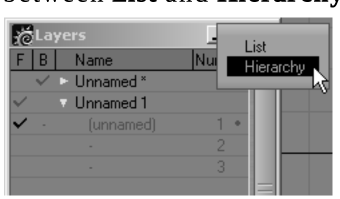
You can change which bank of layers—ten layers per bank—are controlled by these buttons. Just click the < or > buttons. Bank 1 controls layers 1 through 10, bank 2 controls layers 11 through 20, and so forth. The bank selection buttons are mapped to your PAGEUP and PAGEDN keys by default. The highest bank number is 99, but you can access higher layers using the Layer Browser panel.

Layer Browser Panel

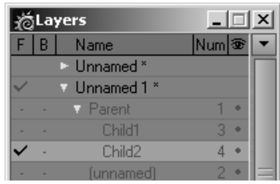
You can also use the Layers Browser window (**Modeler** > **Windows** > **Layer Browser Open/Close**) to do the following:

- Access more than ten layers at a time
- Set foreground/background state, even between objects
- Navigate between objects
- Name layers
- Set layer visibility flag
- Parent object layers

The Layer Browser window has a pop-up menu that lets you switch between **List** and **Hierarchy** views.

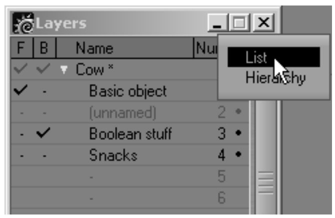


The List view just lists the layers. In the Hierarchy view, the layers of an object are shown in their *parental relationship*, and that can be altered by dragging and dropping layers. Layers cannot be moved between objects and all children at the same level are shown in numeric order.



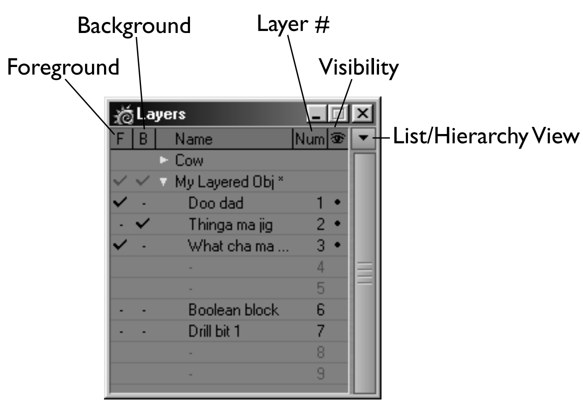
Parenting layers

When in Hierarchy view, the layer list shows non-existent layers at the root level, and lists two more than the current max. This makes it match the list mode behavior more closely.



List/Hierarchy View pop-up menu

The ALT+# (1 through 0) keyboard shortcut will place a layer into the background. Note that if only one layer is active, it cannot be placed into the background.

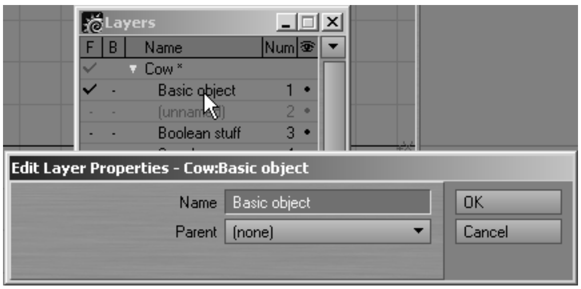


All of the currently loaded objects will be listed. Click on the white triangle to expand or contract the layer listing for a particular object. Layers in the foreground will have a checkmark in the corresponding F column. Clicking in the F column for a different layer will place that layer by itself in the foreground. To add to the current selection, hold the SHIFT key as you click. The B column works similarly on background layers.

The Visibility flag can be toggled on/off for each layer by clicking in the column. A layer is invisible to Layout when the (Visibility flag) dot does not appear. You may want to use this on scratch layers and cutting objects used for booleans. This setting has no effect on Modeler.

NOTE
A few empty layers are always listed after the last occupied layer.

Double-click on a layer name (*unnamed* by default) or choose **Detail** > Layers: **Layer Settings** to bring up the dialog to set the layer name and parent.



Layout Properties dialog

HIDING PANELS

Since your screen can often get cluttered with open panels, you can quickly hide/show these floating windows by pressing the ALT+F1.

LOADING AN EXISTING OBJECT FROM DISK

Objects can be created, loaded into Layout, or loaded directly into Modeler.

To load an object into Modeler:

Select **File** > **Load Object**. The object becomes the current object and all of its layers are loaded.

NOTE
You may select multiple objects, if supported by your platform's file dialog, or select **VBFileRequester** as your **File Dialog** on the Interface tab of the Display Options panel (**Display** > Viewports: **View Options**).

Choose **File** > **Load Object into Layer** to load an object file into the currently selected layer. If the file contains multiple layers, they will all be loaded into the same layer.

Choose **File** > **Revert Current Object** to put the current object in its last-saved state. Essentially, this is like closing and reloading the object.

Encapsulated PostScript Loader

The EPSF_Loader command (**File** > **Import** > **EPSF_Loader**) converts Illustrator (.ai) and encapsulated PostScript (.eps) files into two-dimensional LightWave objects, by converting bezier curves into tessellated polygons.

NOTE
If the EPS uses a gradient fill, you will need to convert it to a solid fill (e.g., using Illustrator) before using this loader.

The **Curve Division Level** option sets the desired level of detail.

The **Convert to** pop-up menu sets the type of object that will be created.

The EPS file is defined by either typing the full path in the **EPSF File** input field or using the requester button to bring up a file requester.

Auto Axis Drill will *attempt* to automatically drill holes in faces, where appropriate. (e.g., the two holes in the number 8).

Auto Centering will center the resulting geometry.

NOTE
Due to the inherent differences between 2D structured drawings and 3D objects, there will usually be some amount of cleanup required.

NOTE
This command has been tested with Illustrator files up to version 8.

STARTING A NEW OBJECT

Use the current object pop-up menu to select which object you are editing.

To start editing a new object from scratch:

Select **File** > **New Object**. A new object called UNNAMED is created with a nice set of fresh empty layers.

SAVING OBJECT FILES

When you save an object, all layers are saved and preserved. You have several options, all available from the **File** menu:

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

Save Object As This option opens a file dialog letting you name (or rename) the object before saving.

Save Layers As Object
Use this to save a foreground layer to its own object file. If multiple layers are selected, they will be saved in a single layer. Make sure you add the .LWO filename extension to your object name when saving.

Save All Objects This option saves all files using their filenames. If any new objects haven't been saved before, you are alerted and given the option of naming them.

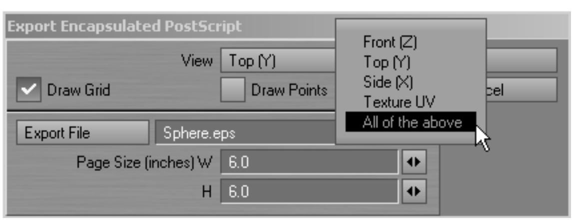
NOTE
An asterisk will appear next to the names (on the current pop-up menu) of objects that have been modified since their last save.

EXPORTING OBJECTS

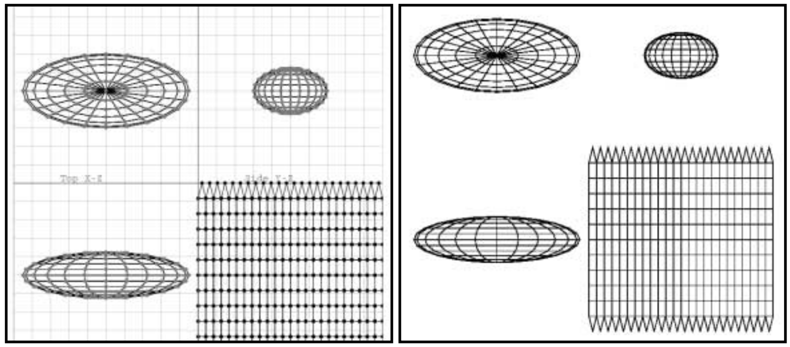
The **File** > **Export** menu has several options to export your LightWave object into different formats. They are discussed in the Appendix under the heading *Foreign Object Support.*

Export Encapsulated Postscript

This plug-in will allow you to save your object as an encapsulated postscript file (.EPS). In the **View** pop-up menu, select which of your viewports you want included in the file. If you want the grid or points included, activate **Draw Grid** or **Draw Points**.



Enter the path and filename in the file input field or click the **Export File** button to bring up a file dialog. Define the size in inches of the resulting drawing using the **W** and **H** field.



With (left) and without (right) grid and points

CLOSING OBJECT FILES

Once you are finished with an object, you can close it by selecting **Close Object** from the **Files** pop-up menu. You can close all object files by selecting **Close All Objects**. If any objects are unsaved, you are alerted and given the option of saving them.

WARNING
Objects revert to their on-disk version in other Hub applications (e.g., Layout) when you close them or exit Modeler and do not save them first. This will cause you to lose unsaved attributes like surface changes.

CUSTOM PREFERENCES

You can use **Modeler** > **Preferences** > **Import Preferences File** and **Modeler** > **Preferences** > **Export Preferences File** to load and save, respectively, preference settings. Use this feature to access special-purpose preference settings you might set up. You can also use **Modeler** > **Preferences** > **Revert to Startup Preferences** to go back to the settings that existed when you first started Modeler.

NOTE
The preference settings that exist when you shutdown Modeler will become the startup settings for your next session.

USER COMMANDS

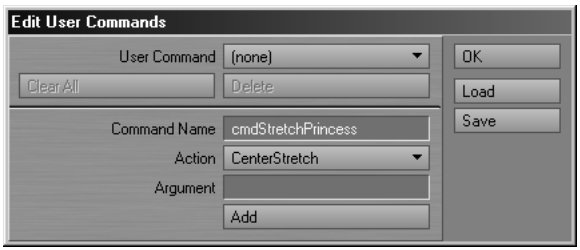
User commands are user-defined names that are associated with a particular plug-in. You can also provide arguments for the plug-in. Once defined, they can be used as standard commands and assigned to a menu and/or key.

NOTE
You can set up multiple user commands using the same plug-in with different arguments.

NOTE
The command arguments can be derived from the LScript documentation on the LightWave CD.

To add a user command:

1 Choose **Modeler** > **Commands** > **Edit User Commands**. This will display the Edit User Commands dialog.



2 Enter the name you want to use. We suggest you also use the prefix CMD (e.g., CMDSTRETCHPRINCESS). This will keep all of your defined user commands together when they appear in a list together with standard commands.

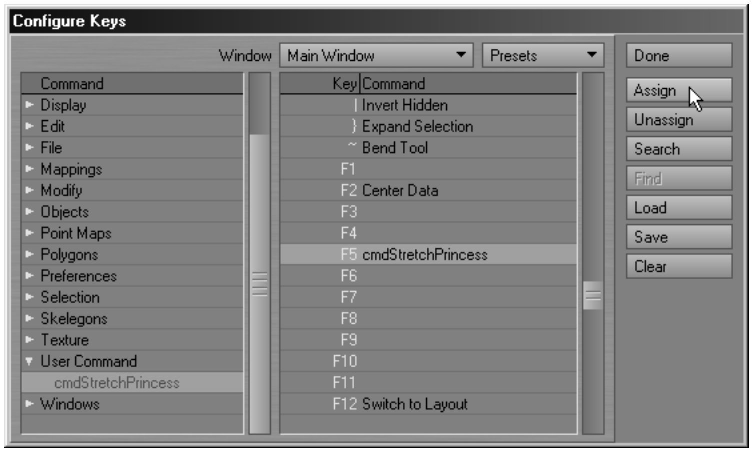
3 Select the command you want to use on the **Action** pop-up menu.

4 If the plug-in requires arguments, enter them in the **Argument** field.

5 Click the **Add** button. Your command will appear on the **User Command** pop-up, along with any other user commands you have created.

Click **Delete** to delete the selected **User Command**. Selecting the **Clear All** button will delete all defined user commands, but don't click this unless you really mean it! There is no implicit undo for this. You will have to either re-add commands manually or use **Modeler** > **Preferences** > **Revert to Startup Preferences**.

6 Click **OK** to close the dialog. The user command will now appear in any command lists.



NOTE
If you need to change the plug-in associated with a user command, you must delete it and then re-add it.

Maintaining User Command Sets

Although the user commands are a part of the main preferences, the **Load** and **Save** buttons let you retrieve and store *user command sets* that you develop.

Execute Command

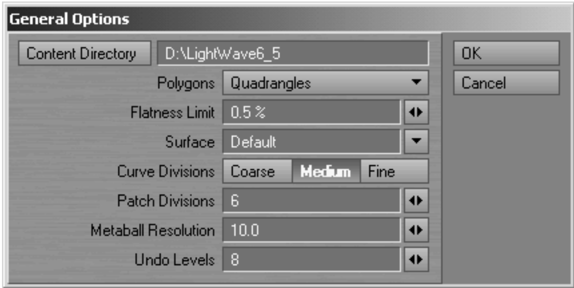
The **XCommand** function (**Construct** > Utility: **LScript**) lets you directly enter a user command to be executed.

Defining a Startup Command

You may define any plug-in (with arguments) to run automatically when you first run Modeler. Simply choose **Modeler** > **Commands** > **Edit Startup Command** and define the command and argument, if applicable.

General Options

You can access the General Options panel by choosing **Modeler** > **Options** > **General Options**.



Modeler General Options panel

Content Directory

The **Content Directory** (see Chapter 6) is a central file path for loading objects, images, and scenes. Note that this setting is shared by all LightWave applications.

Default Polygon Type

Some operations in Modeler create many polygons in one operation. As such, the computer must decide whether to generate triangles (three-point polygons) or quadrangles (four-point polygons), depending on the shape's requirements.

To force the computer to use **Triangles** or **Quadrangles**, select either for the **Polygon** setting. Choose **Automatic** to let the computer create the most appropriate polygon type for each given operation.

HINT
To create optimal objects, your goal is to keep the polygon count down. They will load and render faster than bloated objects, with no visible difference. As such, **Quadrangles** is the suggested setting for most cases. This lets you manually choose the areas that require triangles.

Flatness Limit

The **Flatness Limit** setting determines whether Modeler *regards* a polygon as planar or not. Note that a non-flat polygon within this setting can still cause rendering errors. (See Chapter 21, "Non-Planar Polygons.")

The Default Surface Name

All polygons must have a surface name. When geometry is created, polygons are given the default surface name of DEFAULT. You can change the default name by changing the **Surface** field. This will affect only geometry created from that point on, however. Use the pop-up to the right of the field to select from existing surface names.

NOTE
When you create a surface using the Change Surface dialog (**Detail** > Polygons: **Surface**), the option (**Make Default**) automatically sets the default surface name.

Curve Divisions

The **Curve Divisions** setting determines how smoothly a curve (e.g., spline curves, text, etc.) should be interpolated. The finer the setting, the greater the number of polygons used, and the smoother the resulting curve division.

Patch Divisions

When a SubPatch object is *frozen* with the Freeze command, it is converted into a polygonal object. The **Patch Division** setting determines the level of detail used in the resulting object. The number entered in the **Patch Division** field must be 1 or greater and is restricted to whole numbers. (See Chapter 29, “Patch Level and Conversion to Polygons.”)

The number of polygons per SubPatch surface will be equal to the square of the **Patch Division** number. For example, if set to 4, each SubPatch surface will be converted into 16 polygons arranged in a 4 by 4 array. A setting of 2 would result in 4 polygons arranged in a 2 by 2 array.

The higher the setting, the higher the number of polygons used. Because of LightWave’s surface smoothing capabilities, you can often get away with a setting of 2 and sometimes even 1, which will keep the polygon count of your objects to a minimum.

Metaball Resolution

The **Metaball Resolution** setting determines the amount of detail used to display metaballs. (See Chapter 29, “Metaball Resolution.”)

Undoing Operations

LightWave Modeler offers an undo function that lets you take back the last change you made while working on an object. The **Undo** button (U) is located along the lower edge of the interface. The number of *undos* is determined by the **Undo Levels** setting. The maximum value is 128, but you can also be limited by your computer system RAM. However, minimally-configured systems should be able to go back several steps if necessary. Even power-users should find 10 undo levels sufficient.

Undoing the Undo

It’s possible to go too far unintentionally and undo too many steps. To correct for this error, click the **Redo** button located next to the **Undo** button. As long as you have made no editing changes, you can redo all the way back to your last modeling operation.

The combination of **Undo** and **Redo** is a powerful one, allowing you to step backward and forward through the evolution of a model. This is especially useful when you use some of the more advanced modeling tools, where you may need to make a series of subtle changes to a model as you design it, undoing and redoing as needed, until you achieve the desired shape.

WARNING

If you undo back five steps, then use a new tool, you will no longer be able to redo those steps. Undo is a kind of *history* that unfolds as you model. If you go back and change the model, then you have changed *the course of history* in a fundamental way. The structure of the model is now different, and redo will be unable to restore it. (You aren't Marty McFly.)

chapter 21
Points and Polygons

Chapter 21: Points and Polygons

The process of object editing involves manipulating points and polygons. It will include selecting some portion of an existing object, choosing the operation to perform, adjusting any parameters for that operation, then carrying out the operation.

The concept of *selecting* is very important to Modeler. Selection is the act of specifying either the object itself, or a portion of it, that you intend to edit next. The most basic elements you can modify include points and polygons—the dots and multi-sided shapes that make up all objects. Any number of points, polygons, or objects, in any combination, may be selected for a specific operation.

NOTE
Everything in a layer is considered selected when nothing is selected.

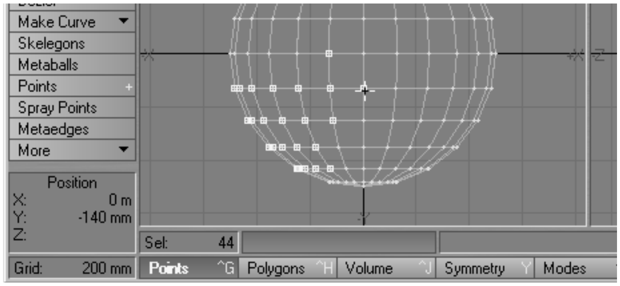
NOTE
When selecting and unselecting, make sure that no modeling tool is currently active. You can usually tell if a tool is active based on the mouse pointer. You can quickly deactivate any tool by pressing the SPACEBAR; however, if no tool was active, you will change the edit mode. You can also choose **Modes** > **Deselect Tool**.

POINT SELECTION

Points generally appear in two ways on screen, selected and unselected. Selected points are highlighted, while unselected points remain small dots.

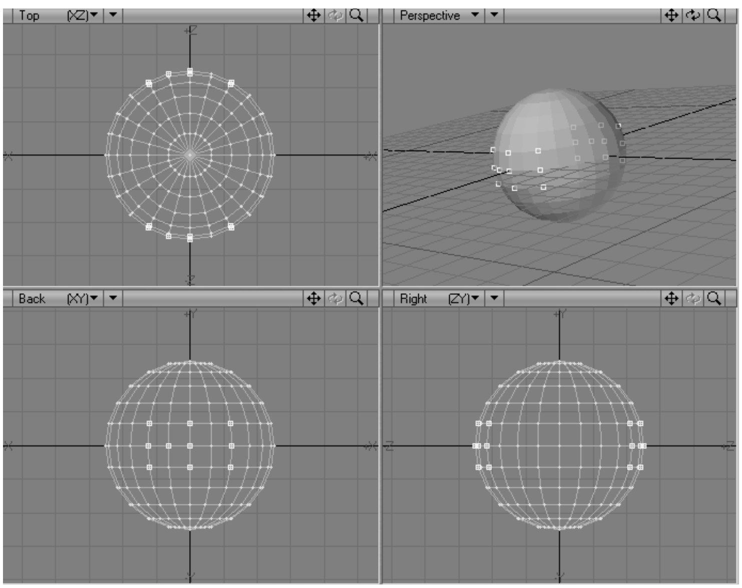
To select points:

1 Use Point Selection mode (**Point** button or CTRL+G). The mode selection buttons are located at the bottom-left of the screen.



Point Selection mode button selected

2 Drag your LMB over points in an object and then release the mouse button to select those points.



NOTE
Once you have selected points and released the mouse button, you cannot select additional points in the same manner.

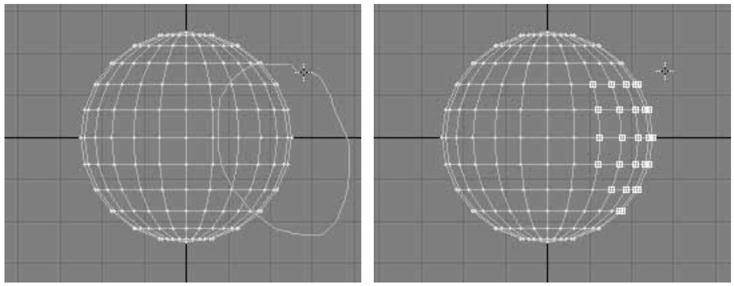
HINT
If you know the number of points you want to select, check the information display above the **Point** (Selection mode) button. It shows you the number of selected items.

To add more points to a selection:

Hold the SHIFT key while you select with your LMB to add points to the existing selection or just click your MMB.

To lasso a group of points:

With the RMB, drag out a circle around points to select a group. As you might expect, you can add more points to a selection by holding the SHIFT key as you drag.



To unselect points:

If selected points exist (and you have released the mouse button), dragging over them with your LMB will unselect them. You can also use the lasso by dragging with your RMB.

To unselect all points:

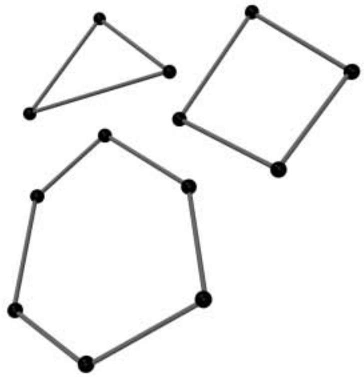
Click in any non-active (i.e., not a button) area on the toolbar to unselect all points. You can also use the Drop Current Tool command (/).

NOTE

See Chapter 27 for additional selection commands.

POLYGONS

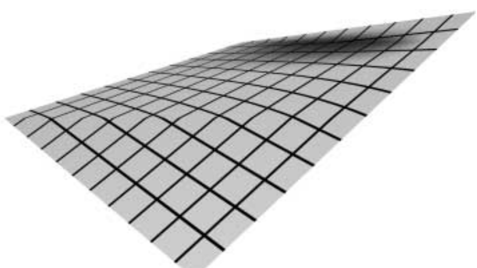
As discussed briefly in the prior chapter, a *polygon* is some number of points joined by lines to form a solid face. A triangular polygon is a three-point polygon. A quadrangular polygon is a four-point polygon. A multi-sided polygon (some programs call these n-gons) is a complex polygon, one with many points. As long as the points that make up a polygon remain in one (flat) plane, the polygon will render properly. If not, polygons can fall victim to unpredictable rendering errors. Even if a polygon is flat when modeled, it can become “unflat” (non-planar) if a displacement map, bones, or other effects that move points non-uniformly are applied to the object during an animation.



Polygon illustrations

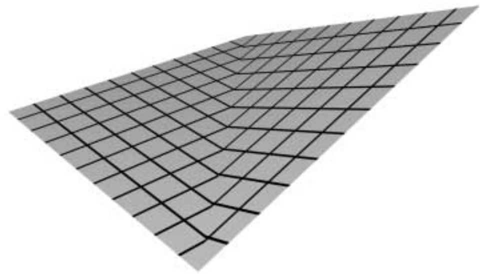
Non-planar Polygons

Non-planar rendering errors can be eliminated by using triangles. Why use triangles? Well, say you had a square piece of glass laying on a table. It's physically impossible to lift just one corner without also lifting at least one of the other corners, right? However, if the glass is triangular, we can easily lift one corner and leave the other two still resting on the table. A non-planar polygon is like our square piece of glass with just one corner lifted in the air—impossible for you and LightWave's renderer. Although you can make objects with non-planar polygons, they will likely cause sporadic rendering errors.



Example rendering error due to non-planar polygon

If we split the glass square in half from one corner to another, we end up with two triangles sharing a single side. (In Modeler speak, this is called tripling polygons.) We can then, in effect, lift one corner without a problem because the shared side would act like a hinge.



Tripled polygons

Flatness

Whether Modeler regards a polygon as planar or not depends on its flatness value and the **Flatness Limit** on the General Options panel (**Modeler** > **Options** > **General Options**) setting. If a polygon's flatness exceeds the limit, then it is considered non-planar. A polygon's flatness can be determined by selecting the polygon and opening the Polygon Info panel (**Display** > Selection: **Info**). It is unlikely that you will ever need to adjust the **Flatness Limit** default setting of .05%.

NOTE
If a polygon is non-planar, but within the Flatness Limit, it may still cause rendering errors.

POLYGON EFFICIENCY

So why not just use triangles exclusively? Well, the more polygons your object has, the longer it takes to render. Polygons with more than three points are still useful when they will not be bent or twisted—bending or twisting these types of polygons usually results in *non-planar* polygons, which cause rendering errors. It is best to use a combination of polygons when you model, using three-, four-, and greater-sided polygons where necessary. Use the type of polygon that best fits the needs of the animation and model.

SPECIAL-USE POLYGONS

A few types of polygons are useful for very specific purposes. Single-point polygons appear to be normal points, except that you can select them and assign them surface qualities like any polygon. However, unless single-point polygons are assigned luminous or motion blur attributes (or are used with HyperVoxels), generally they are not visible when rendered. With luminosity and/or motion blur, they appear as a pinpoints or streaks of lights in images or animations. This is highly useful for emulating rain, stars, fireworks effects, and more.

Fireworks with single-point polygons

Two-point polygons, or lines, are useful for strings, such as on stringed instruments, fields of wheat, and so on. Technically, a line has no measurable thickness, but LightWave lets you render at an adjustable size.

A field of wheat with two-point polygons

NOTE

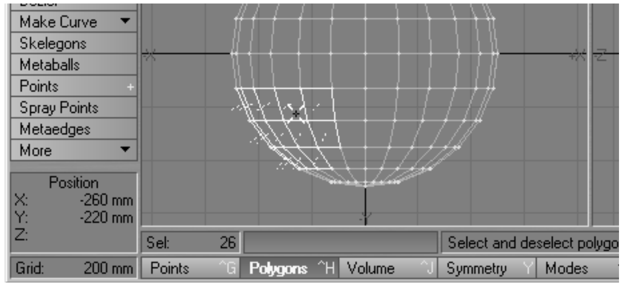
See the discussion on **Particle/Line Thickness**; this feature is on Layout's Rendering tab on the Object Properties panel, see Chapter 9 for more information.

POLYGON SELECTION

Polygons generally appear in two ways on screen, selected and unselected. A selected polygon will appear highlighted with outlines in yellow. You will see that selecting polygons is very similar to selecting points.

To select polygons in a viewport:

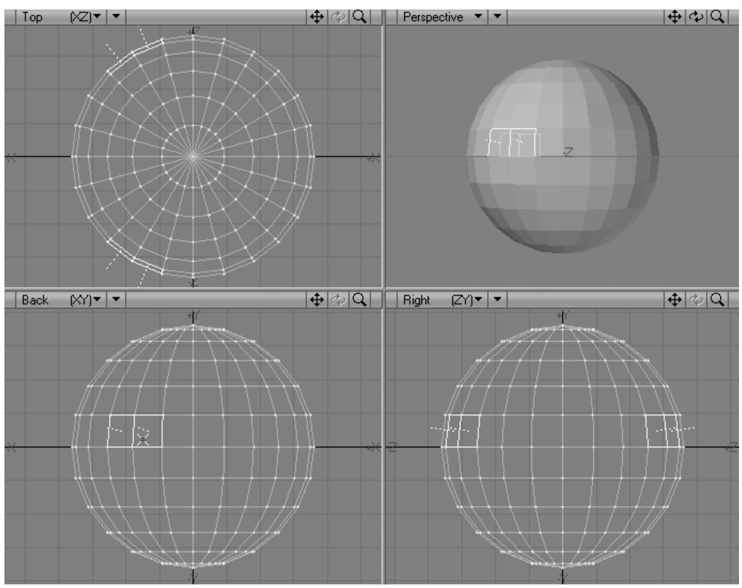
1 Use **Polygon** selection mode (CTRL+H). The mode selection buttons are located at the bottom-left of the screen.



Polygon Selection mode button selected

2 If you are using a viewport set to a non-shaded Rendering Style (Display Options), like wireframe, drag your LMB over the edge of polygons and then release the mouse button. This selects polygons that share that edge. With a shaded Rendering Style like Texture or Smooth Shade, you can click directly on the polygon's face.

As polygons become selected, they will become highlighted and you will see the surface normal indicator (the perpendicular dotted line) appear. (The Normal display option must be active.)



NOTE

Once you have selected polygons and released the mouse button, you cannot select additional polygons in the same manner.

HINT

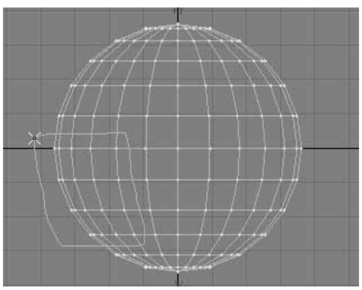
If you know the number of polygons you want to select, check the information display above the **Point** Selection mode button. It shows you the number of selected items.

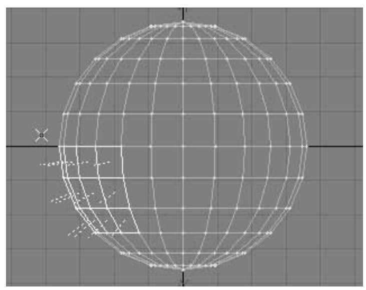
To add more polygons to a selection:

Hold the SHIFT key while you select with LMB to add more polygons to the existing selection or just click your MMB.

To lasso a group of polygons:

With the RMB, drag out a circle around polygons to select a group. Only the polygons completely within the outline are selected. Of course, you can add more polygons to a selection by holding the SHIFT key as you drag.





To unselect polygons:

If selected polygons exist (and you have released the mouse button), dragging over them with your LMB will unselect them. You can also use the lasso by dragging with your RMB.

NOTE

Selecting a group of polygons and then deselecting the ones you don't want is often the only way to select the ones you do want.

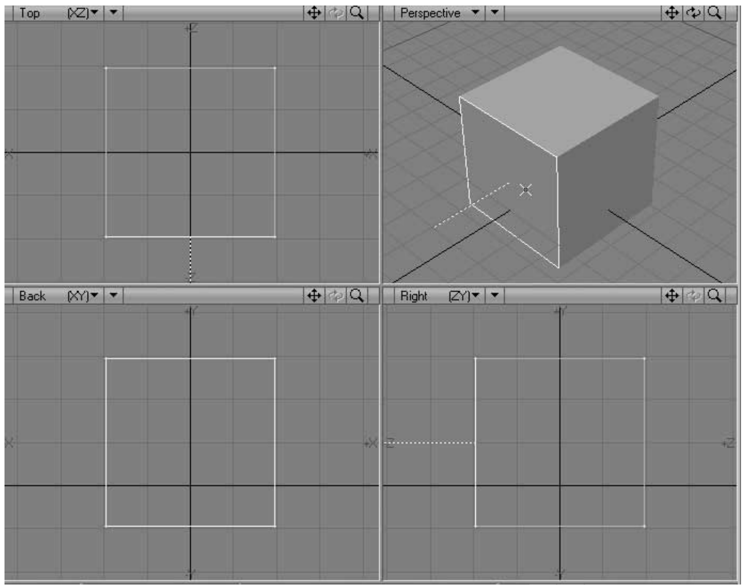
To unselect all polygons:

Click in any non-active (i.e., not a button) area on the toolbar to unselect all polygons. You can also use the Drop Current Tool command (/).

NOTE
See Chapter 27 for additional selection commands.

SELECTION WITH IN-LINE POINTS/POLYGONS

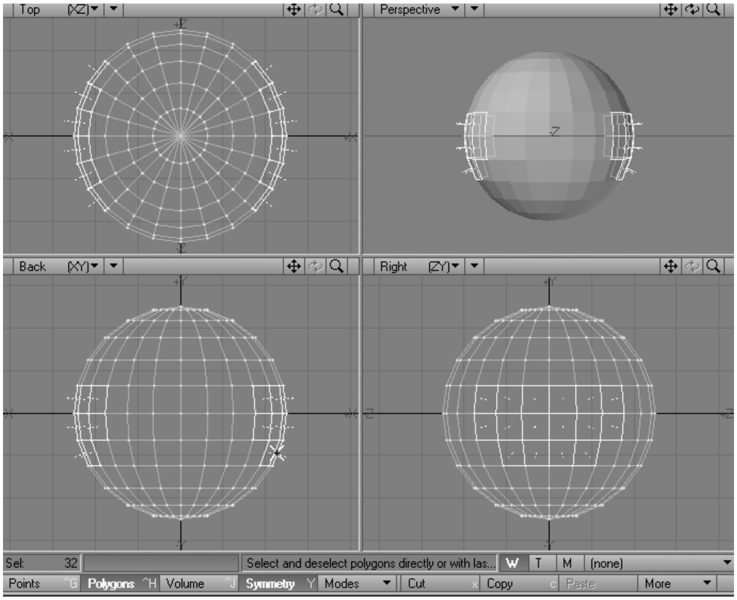
If there are points/polygons *in-line* (on top of each other) in a viewport using a non-shaded Rendering Style (i.e., Wireframe), you select all of the points/polygons when you select with your mouse. However, shaded viewports respect polygons facing away from you and those obscured by other polygons. Thus, you can select by clicking on a polygon's face without worrying about what's behind or on the other side. You can change the Rendering Style from the pop-up menu on the viewport titlebar or on the Viewport tab of the Display Options panel.



Polygon selected in shaded mode

SYMMETRICAL SELECTION

When you select polygons/points on the positive X axis and Symmetry mode (**Symmetry** button) is active, polygons/points on the negative X axis are also selected (or deselected). Polygons/points must be exactly opposite each other on the positive and negative sides of the X axis for this command to work properly.



Selecting polygons in Symmetry mode

NOTE
Generally, you should perform all of your edits on the positive side of the X axis when using Symmetry. Using the negative side may lead to unpredictable results.

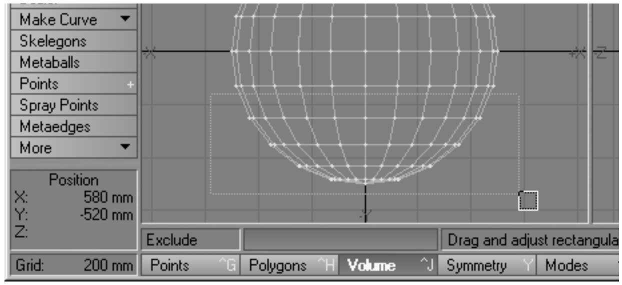
NOTE
Symmetry also affects polygon editing.

VOLUME SELECTION MODE

You may also select a portion of an object using a definable bounding box (i.e., a volume). The **Volume** Selection mode (CTRL + J) button is next to the **Polygon** Selection mode button at the bottom of the interface. Polygons/points selected in this manner do not become highlighted.

To use Volume selection:

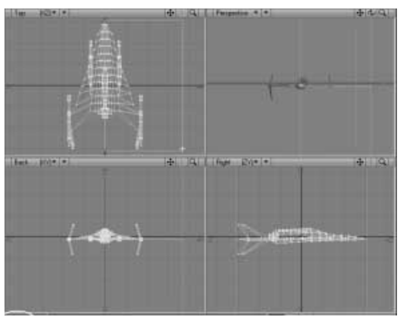
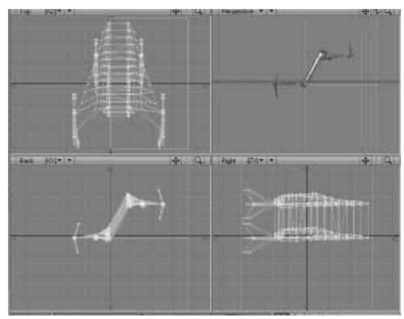
1 Click the **Volume** selection mode button.



Volume Selection mode button selected

2 The initial state of the Volume select is usually Exclude. This means polygons touching the bounding box are excluded. Clicking the **Volume** button again will toggle to the Include state where polygons touching the bounding box are included in the selection. You can also judge the state by looking at the small text field right above the **Point** Selection mode button.

3 Drag out a bounding box in a viewport with your LMB. If one already exists, you can drag it around from the center or drag the edges and corners to re-size. You can reset the bounding box by clicking in an inactive part of the interface.

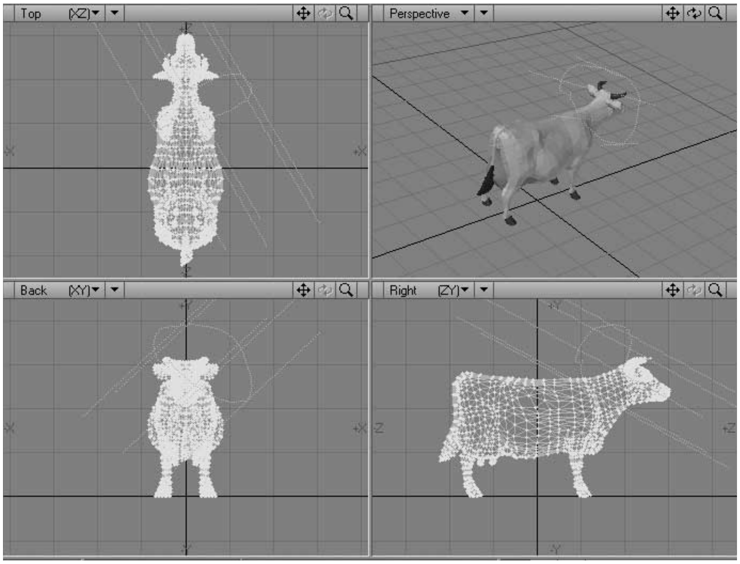



The selection is infinite along the axis perpendicular to the viewport you dragged the bounding box in.

If you open the Volume Statistics panel (**Modeler** > **Windows** > **Statistics Open/Close**), you can also use the **+** and **-** buttons to formally select/unselect the listed points/polygons.

Lasso Volume Select

You can also *lasso-select* a volume by dragging your RMB in a viewport. A big difference between this and normal volume selection is that in a perspective viewport, the *lassoed* area is perpendicular to the viewport. Moreover, the angle will be retained even when the viewport is subsequently rotated.



HINT

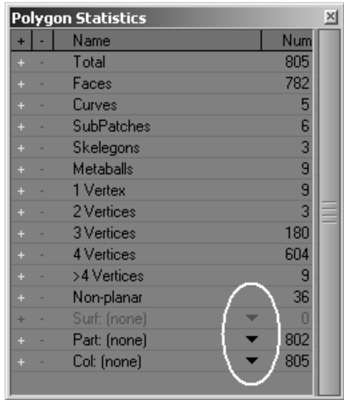
If you are selecting from among many items within a cluttered space, it's probably easier to use the lasso tool for Polygon Selection mode (i.e., RMB).

HINT

Press the SPACEBAR to cycle through the Selection modes (Point, Polygon, and Volume). Note that it will also first drop any modeling tool that may be active.

SELECTION BY CRITERIA

The Polygon Statistics panel (**Modeler** > **Windows** > **Statistics Open/Close**) includes some tools that simplify your life when you select polygons.

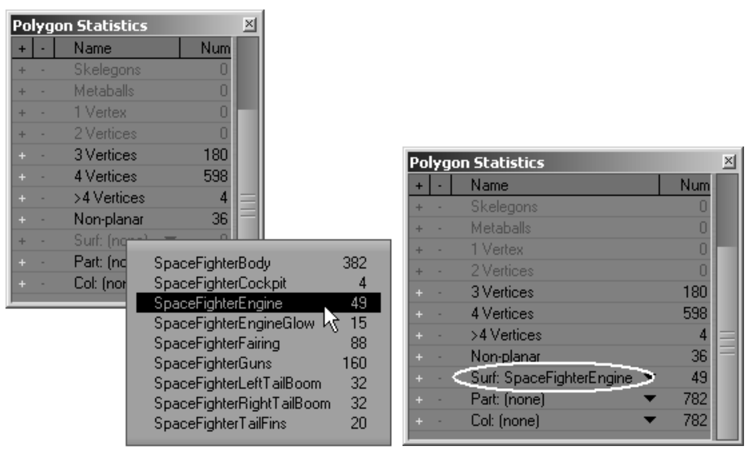


Polygon Statistics panel. Pop-up button for surface, part, and sketch color selection are circled

The Polygon Statistics panel lists polygons by type, including skelegons, metaballs, and SubPatches, as well as the number of each type. **1 Vertex** polygons are those that consist of a single point; **2 Vertices** polygons are those with two points, and so on.

You can select all of the polygons in a particular group by clicking on the corresponding plus (+) symbol. You can unselect the entire group by clicking the minus (-) sign.

You can also select by surface name (Surf), part name (Part), and sketch color (Col). First, select a name by clicking on the pop-up menu (down-arrow button). Then click on the plus or minus signs as needed.



Left: Selecting surface. *Right*: Surface now available for polygon selection

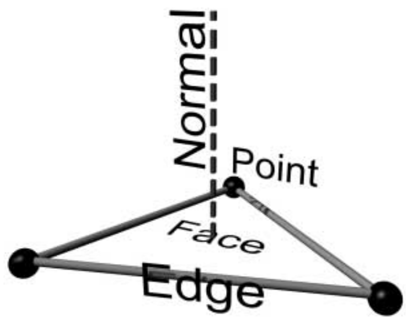
NOTE

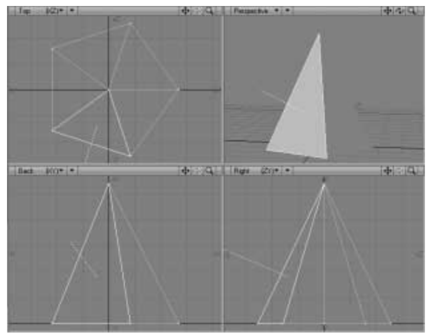
The information presented on the Statistics panel will vary depending on the editing mode (i.e., Points, Polygons, or Volume).

HINT
Leave the Statistics panel open all the time for easy access.

SURFACE NORMAL

The surface normal—admittedly an unusual-sounding name—has great importance in 3D. Here's why: A feature of polygons is that they can be one-sided or double-sided. They have the unique ability to be invisible from one side while visible from the other, or they can be visible from both sides. The surface normal, which appears whenever a polygon is selected, projects away from the visible side of a polygon.





Since LightWave needs to calculate the appearance of all polygons that make up an object, it helps for you to determine beforehand whether the polygons you need for an object will actually be seen from both sides. Why is this important? Because it takes LightWave longer to calculate both sides of the polygon. If LightWave knows that a polygon is only one-sided, it can render images more quickly.

How can you determine whether to use a one-sided or double-sided polygon? Easy. Suppose you're making an apple. Will the scene require that the viewer see the inside of the apple? If not, then you model only the outside of the apple. There's a simple rule of thumb for modeling: build only what your animation requires. The author of LightWave once created a model of San Francisco's famed Golden Gate Bridge. However, since the back side of the bridge would not be seen in the animation, he did not model it. It wasn't needed.

NOTE
Even if you have not modeled double-sided polygons, you can still use the Double Sided surface attribute when rendering.

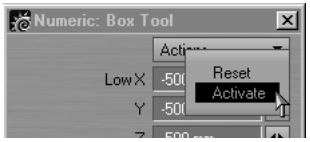
Surface Normal Functions

See Chapter 26 for information on special functions to change the direction of surface normals.

NUMERIC PANEL

The Numeric panel is a multi-functional non-modal window. Its contents and abilities change depending on what tool is selected. When you create primitives, it lets you refine graphically set values. When you use modifying tools, it lets you change various settings that affect how the tool performs. It is displayed by choosing **Modeler** > **Windows** > **Numeric Options** or by just pressing the N key. It may be left open continuously.

Use the **Actions** pop-up menu to **Reset** the fields to their default settings or to **Activate** the tool. (You can also press the N key to activate the tool.) Generally, activating a tool turns on its interactive handles, if any, in the viewports. This can be for primitive shape, influence range, and so on. It will also activate the numeric fields, if they are ghosted.



Activate option

NOTE
You can also reset to defaults by clicking a reset area when a tool is selected, but not activated.

chapter 22

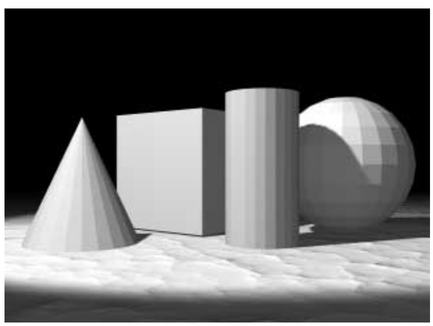
Creating Geometry

Chapter 22: Creating Geometry

As you might expect, no one is going to create a *starship* by individually creating each and every point and polygon. You create most of your objects with primitives, that is, boxes, balls, discs, and cones.

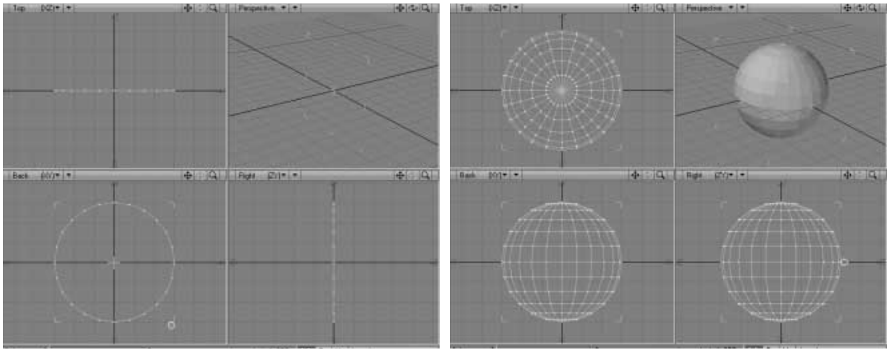
Look around the room. Nearly everything in it can be broken down into one or more of these primitive shapes. Your monitor is basically a box, your desk is made up of several boxes, your trash can is an extruded disc, and so on.

The modeling process normally starts with a primitive and you just modify and add to it. So, the better you get at breaking down objects into primitives, the better modeler you will become.



THE PRIMITIVE TOOLS

On the **Create** menu, there are several basic primitive tools in the Object group in the toolbar. Generally, most of the tools are used in the same way. First, you drag out a two-dimensional shape and then add depth to your object by expanding it in a different viewport. For example, with the Box tool you begin by creating a square and add depth, making it a box. The Ball, Disc, and Cone tools all start off the same, with a circle. It is really the second step that differentiates those tools.



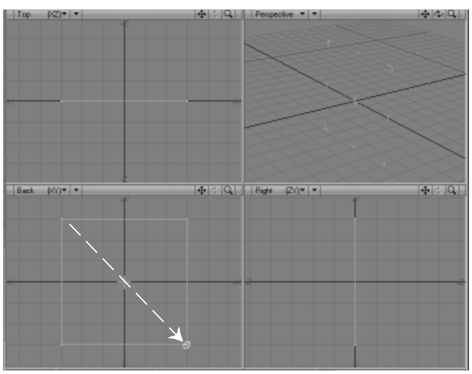
Dragging out a ball

You'll probably use the Box tool most often, because most objects are box-like in general form. The Ball tool is great for creating anything from a tiny pea to a giant planet. The Disc tool is used to form cylinder-shaped objects like coins or tubes. The Cone tool is perfect for making delicious ice cream cones...mmmmmm.

To create a primitive:

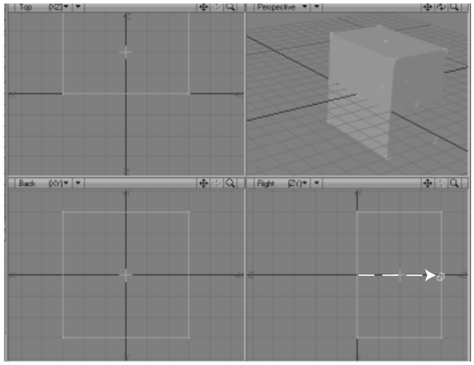
1 Select the primitive tool button on the **Create** menu (i.e., **Box**, **Ball**, **Disc**, or **Cone**).

2 Drag out the initial 2D shape in any viewport with your LMB. This activates the tool. Pressing the CTRL key before you click and holding it while you drag will symmetrically constrain the shape. The dimensions of your outline are shown in the lower left corner.



3 After you release the mouse button, you can adjust the shape by dragging any of the sides or corners. You can reposition it by dragging the center.

4 Go to another viewport and drag to extend the shape into the third dimension. Release the mouse button.



5 You can continue to adjust the primitive's size and position.

NOTE
If you find you need to start over from scratch, just click in an inactive toolbar area or use undo.

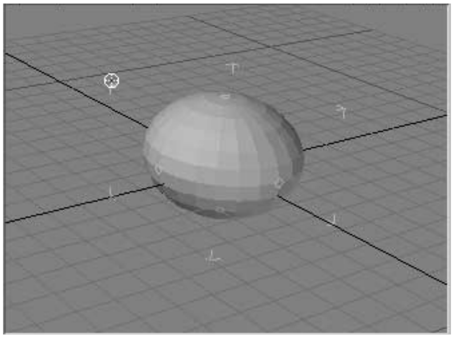
6 When you are satisfied, deselect the primitive tool by clicking on it or choosing **Modes** > **Deselect Tool** to turn your outline into an object. (Alternatively, you can select another modifying tool.)

Tools and the Right Mouse Button

Most (interactive) tools that create geometry (Box, Ball, Capsule, etc.) use the RMB to basically drop the tool—making geometry, if applicable—and then begin a new operation with it.

Creating in Perspective

You can create primitives in a perspective viewport pretty much as you would in a 2-D viewport. You will see highlighted control points at the corners, sides, and center. You will also notice that you can see (fainter) hidden sides and control points. You can also drag any of the control points. Dragging a side control point will move the corresponding side along its perpendicular axis.



Creating in perspective

NOTE
Remember, you can rotate your view by holding the ALT key as you drag.

Using the Arrow Keys

The arrow keys let you quickly adjust certain settings depending on the primitive being created, as follows:

Primitive	Right/Left Arrow	Up/Down Arrow
Box	Incr./decr. horizontal **Segments***	Incr./decr. vertical **Segments***
Ball	Incr./decr. **Segments**	Incr./decr. **Sides**
Disc	Incr./decr. **Segments**	Incr./decr. **Sides**
Cone	Incr./decr. **Segments**	Incr./decr. **Sides**

**based on viewport beneath mouse pointer*

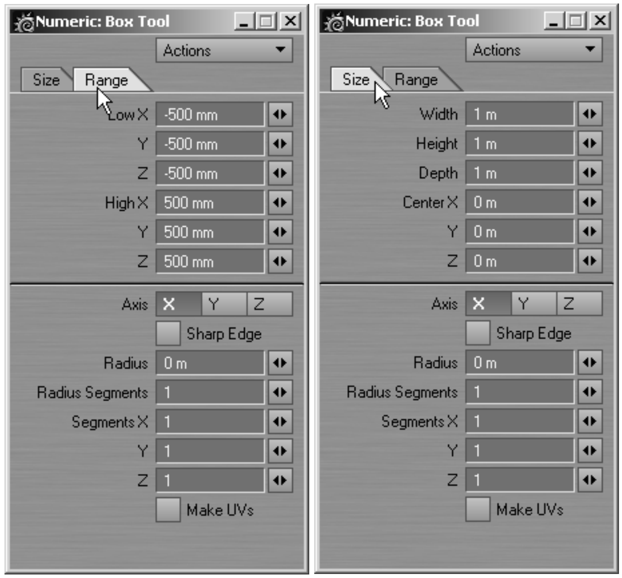
USING THE NUMERIC PANEL

You can also numerically create a primitive or use the numeric panel with the graphical creation tools. The input fields on the panel update in real-time as you graphically adjust your box with the mouse and vice versa. As such, you can drag out a rough version of your box with your mouse and then enter precise values on the numeric panel.

Actions Pop-up Menu

The **Actions** pop-up menu (see Chapter 21), can activate the numerical settings for a primitive tool. To do this, simply select **Activate**. Select **Reset** to return all of the fields to their default value.

Box Tool Fields

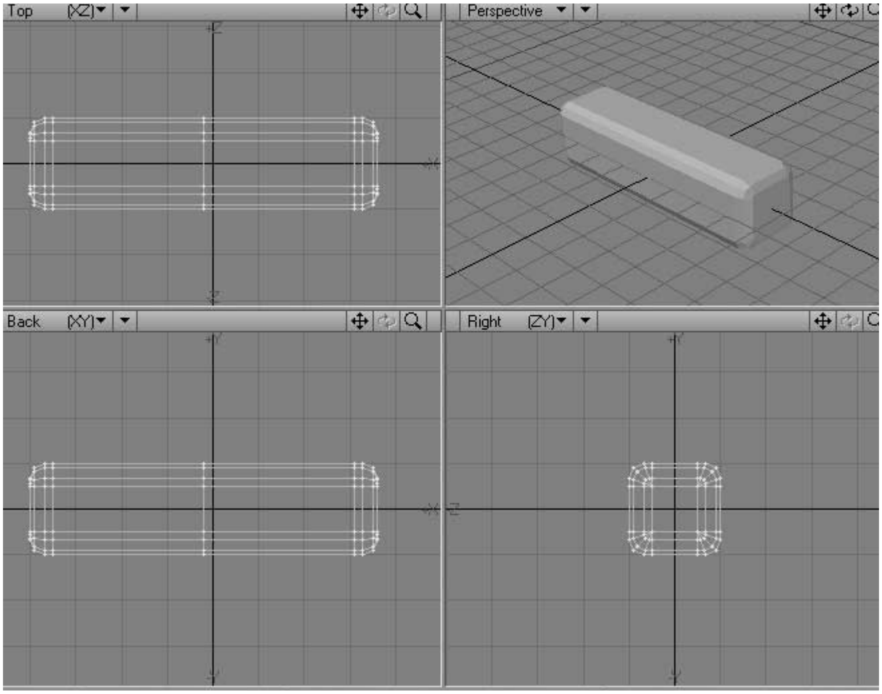


You have two different ways to adjust the shape of your box, the Size and Range modes. With the Size tab selected, you can adjust the **Width**, **Height**, and **Depth** of your box dimensions. The **Center** XYZ settings set the position of the center of the box.

With the Range tab selected, you can adjust the positions of two opposing corners of your box using the **Low** XYZ and **High** XYZ settings.

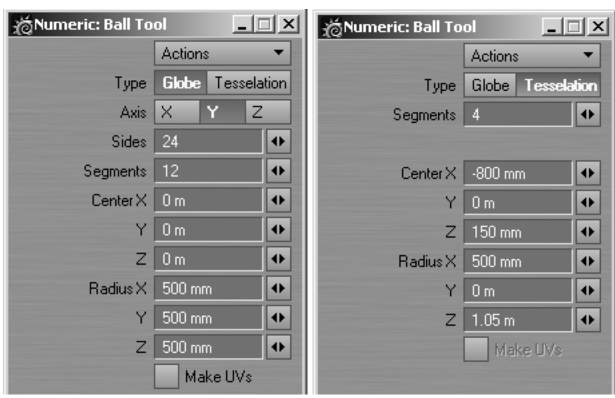
You can round off the edges of the box by setting the **Radius** to a value greater than 0. The **Axis** sets the direction of the radial polygon pattern used to round the edges. Use the **Sharp Edge** option to smooth or unsmooth the edges.

The **Segments** XYZ settings define the general number of segments along those axes, excluding any geometry added to round the edges.



Rounded-edge box

Ball Tool Fields



Type The make-up of a **Globe** is determined by the number of **Sides** and **Segments**. **Tessellation** balls are set using a **Level** parameter; these balls are created entirely of triangles and are, therefore, smoother looking.

HINT
Use a **Tessellation** ball if the camera will be close to the object and the surface is smooth.

Axis (**Globe**) determines the major axis for the ball. This is also set when you click to first create the ball using the axis perpendicular to the (orthogonal) viewport.

Sides (**Globe**) determines how many segments should be used around the ball.

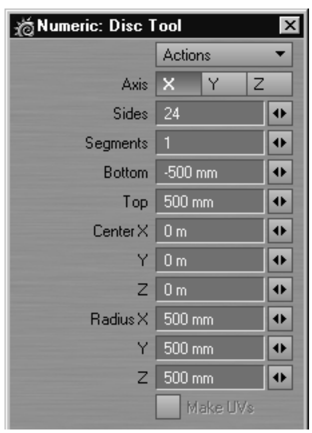
Segments (**Globe**) sets how many vertical segments should be used. Three-sided polygons are always used at the top and bottom. Elsewhere, the polygons used are dependent on the **Polygons** setting on the General Options panel (**Modeler** > **Options** > **General Options**).

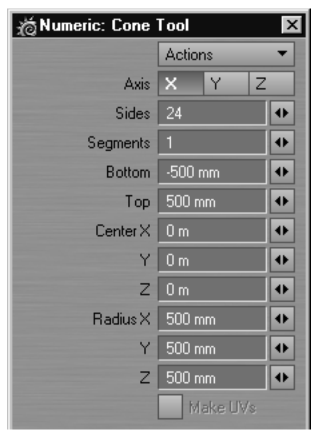
(**Tessellation**) sets the number of segments along the edges between the twelve polyhedral vertices. Higher settings will increase the complexity of the triangle matrix, taking longer to create and requiring more RAM.

Center The XYZ coordinates of the center of the ball.

Radius The radius of the ball along the X, Y, and Z axes.

Disc/Cone Tool Fields





Axis The primitive is aligned with this axis.

Sides Sides determines how many segments should be used around the perimeter.

Segments Segments sets how many vertical segments should be used.

Bottom Bottom is the starting point of the primitive along the selected **Axis**.

Top Top is the ending point of the primitive along the selected **Axis**.

Center Center refers to the coordinates for the center of the primitive.*

Radii Radii refers to the radius of the primitive (wide end for a cone) along the axes.*

The **Center and **Radii XYZ** fields that correspond to the selected **Axis** will have no effect on the shape.*

NOTE
Most tools allow numeric input. Thus, the contents of the numeric dialog will change as you select different tools. You may find it handy to just leave the numeric dialog open.

Make UVs Option

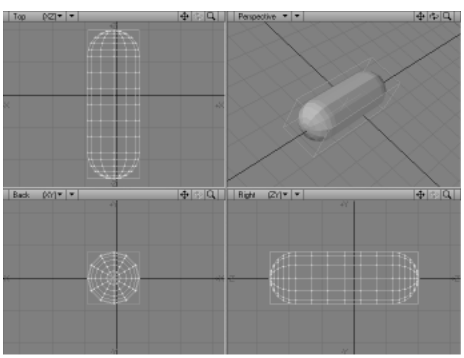
The **Make UVs** option at the bottom of the various primitive numeric panels assigns some default UVs based on the geometry of the object. Note that a UV Texture map must be currently selected or this option will be ghosted. (See Chapter 28 for more information.)

Make UVs option

OTHER PRIMITIVE TOOLS AND FUNCTIONS

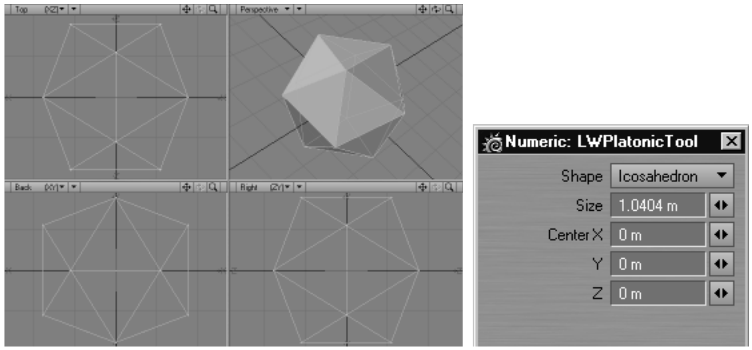
The Capsule Tool

Another primitive tool is the Capsule tool (**Create** > Objects: **Capsule**). Your initial click defines the perpendicular axis of the object. You can drag the center position handle to move the shape, and you can drag the outer bounding box edges to resize it. The numeric settings are similar to those for the standard primitive tools.



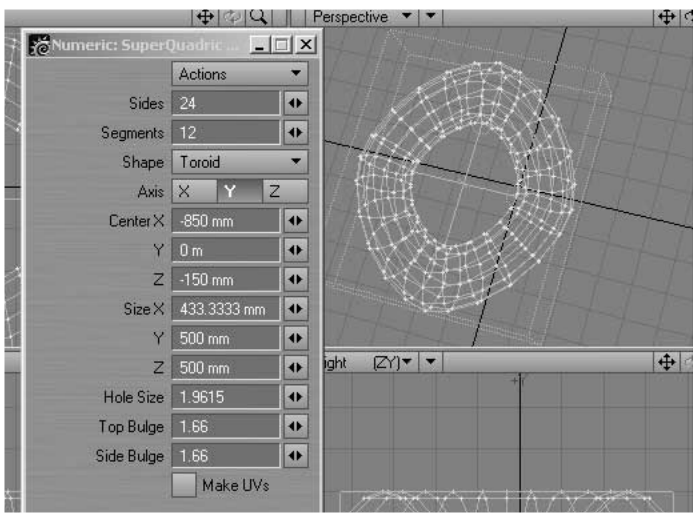
The Platonic Solid Tool

Yet another primitive tool is the Platonic Solid Tool (**Create** > Objects: **Platonic Solid**). The shape of the primitive is controlled by the **Shape** pop-up menu on the numeric panel. You can drag the center position handle to move the shape, and you can drag the outer bounding box edges to resize it. The other numeric settings are similar to those for the standard primitive tools.



The SuperQuadric Tool

Use the SuperQuadric tool (**Create** > Objects: **SuperQuadric**) to create *quadrics* objects interactively. You can choose between two shapes: Ellipsoid (spherical) and Toroid (donut).



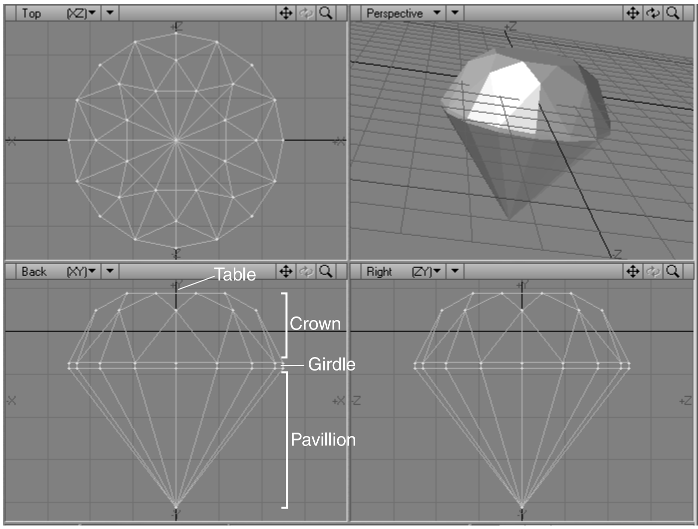
You can use standard drag handles to manipulate the shape and position interactively. The numeric panel provides more detailed control.

A perfect circle—looking along the set **Axi**s—can be made by setting the **Side Bulge** to 2. A value of about 5 yields a rounded square shape.

NOTE
If you must know, a quadric is a shape made of the squares of the coordinates, a generalization of a sphere, which is $x^2 + y^2 + z^2$.

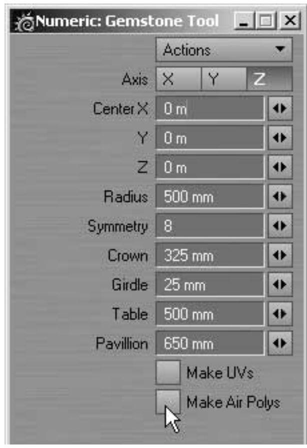
The Gemstone Tool

Diamonds are an animator's best friend or something like that. You can quickly make a nice round-cut diamond with the Gemstone tool (**Create** > Objects: **Gemstone Tool**).



On the numeric panel you can control various aspects of your diamond. **Center** sets the center position and **Radius** the overall radius of the object.

Higher **Symmetry** values increase the number of polygons used on the crown and lower values decrease the number. The **Crown**, **Girdle**, **Table** and **Pavillion** settings control the size of those areas on the diamond. (See above illustration for reference.)



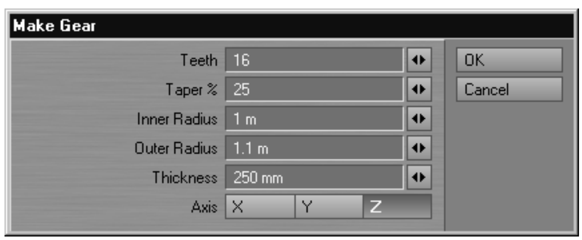
The Gemstone Tool numeric panel

The Equilateral Function

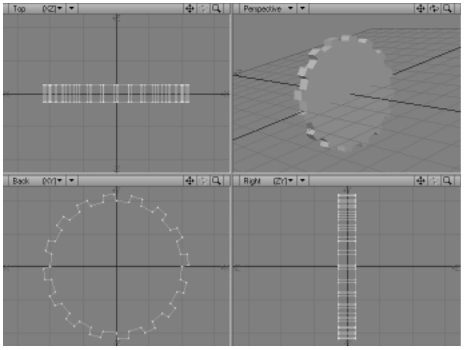
The Equilateral function (**Create** > Objects: **Equilateral**) will create an equilateral triangle of the specified size.

The Gear Function

The Gear function (**Create** > Objects: **Gear**) creates a gear-shaped object.



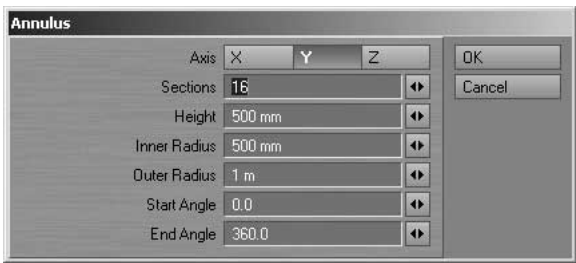
Teeth	Teeth refers to the number of teeth in the gear.
Taper %	Taper sets the “pointiness” of the teeth. A higher setting creates sharper teeth.
Inner Radius	Inner Radius refers to the distance from the center of the gear to the bottom of the teeth.
Outer Radius	Outer Radius is the distance to the outside edge of the teeth.
Thickness	Thickness is the width of the gear.
Axis	Axis is the perpendicular axis of the gear.



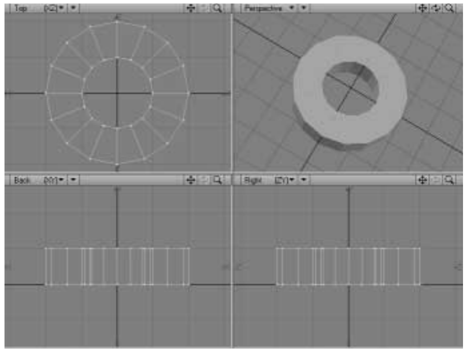
Gear

The Wedge Function

The Wedge function (**Create** > Objects: **Wedge**) will generate an *annulus* (like a flat donut).



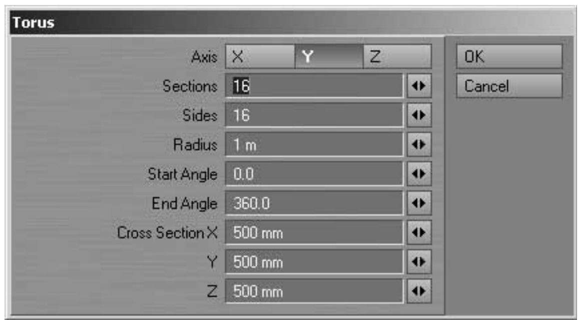
Axis	Axis is the axis perpendicular to the donut hole.
Sections	Sections refers to the number of radial sections to use for the object.
Height	Height specifies the width of the object.
Inner Radius	Inner Radius specifies the radius of the inner hole.
Outer Radius	Outer Radius specifies the radius to the outside perimeter.
Start Angle	Start Angle is the starting angle around the selected **Axis**.
End Angle	End Angle is the ending angle around the selected **Axis**. (360 for a complete donut.).

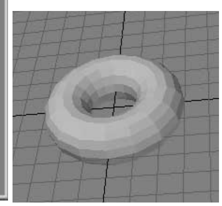


Wedge

The Toroid Function

Choose **Create** > Objects: **Toroid** to generate a torus (donut) or a section thereof.





Axis Axis is the axis perpendicular to the donut hole.

Sections Sections specifies the number of segments used around the donut.

Sides Sides specifies the number of sides each segment should use.

Radius Radius is the distance from the center of the donut to the center of a cross section.

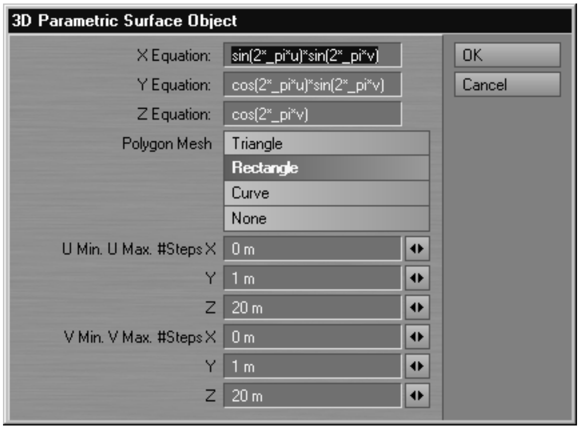
Start Angle Start Angle is the starting angle around the selected **Axis**.

End Angle End Angle is the ending angle around the selected **Axis**. (360 for a complete glazed donut. mmmm).

Cross Section Cross Section is the XYZ size of a cross section. Note that the value corresponding to the selected Axis is not considered.

The Parametric Surface Object Function

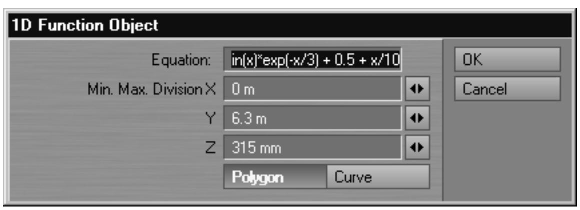
The Parametric Surface Object function (**Create** > Objects: **ParametricObj**) creates parametric surfaces based on the equations entered for X,Y, and Z in terms of UV coordinates. The function lets you create the object using **Triangle**s, **Rectangle**s, **Curve**s, or just points (**None**). The equations may include any of the functions listed in the *Appendix*.



Parametric Surface Object Function

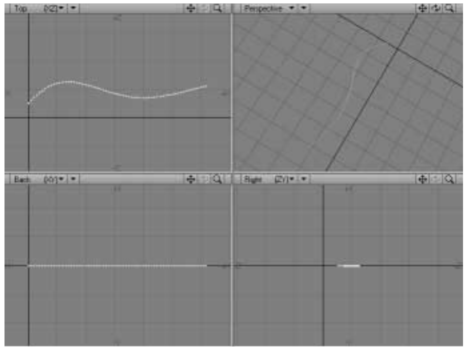
The Plot1D Function

The Plot1D function (**Create** > Objects: **Plot1D**) generates either a curve consisting of two-point **Polygon**s or a spline **Curve**. The curve is generated along the X axis, and its height is in the Z axis.



Plot1D Function

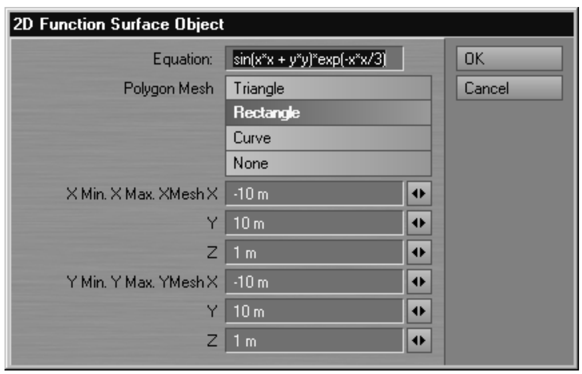
The height is determined by the value of the expression in the **Equation** field at each x. In the fields **Min**, **Max**, and **Division**, enter the x value for the start of the curve, the x value for the end of the curve, and the number of divisions of the curve.



Plot1D Function

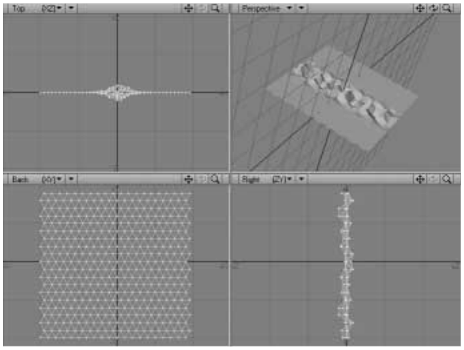
The Plot2D Function

The Plot2D function (**Create** > Objects: **Plot2D**) generates a two-dimensional surface in the XY plane consisting of **Triangle**s, **Rectangle**s, **Curve**s, or points (**None**). The height (in the Z direction) is determined by the value of the expression in the **Equation** field at each division in the X and Y axes.



Plot2D Function

In the fields labeled **X Min**, **X Max**, **Y Min**, and **Y Max**, enter the boundaries for the surface. In the **XMesh** and **YMesh** fields, enter the number of divisions along that axis.



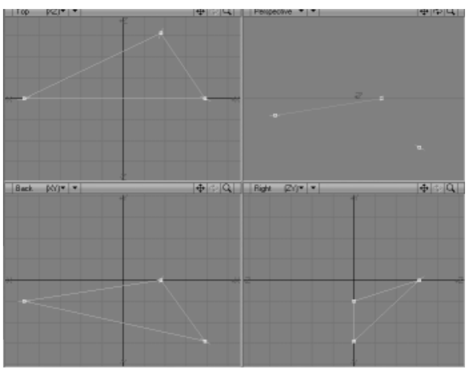
Plot2D Function

USING THE PEN TOOL

The Pen tool (**Create** > Elements: **Pen**) gives a quick way to create polygons on the fly.

To create a polygon with the Pen tool:

Choose **Create** > Elements: **Pen** and click in a viewport with your LMB. A polygon is created using points you define by clicking. If you drag the mouse button, you can refine the point's position before you create it, which happens when you release the mouse button. You do not need to click in the same viewport for each point.



Using the Pen Tool

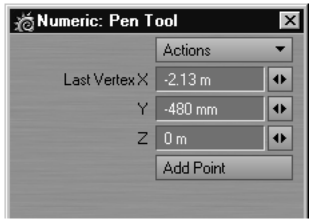
To constrain the position of a new point to 45-degree increments, relative to the last created point, hold the CTRL key down before you click.

To create single point polygons with the Pen tool:

Simply click in a viewport with your RMB. With each click you will create a single-point polygon!

Using the Pen Numeric Panel

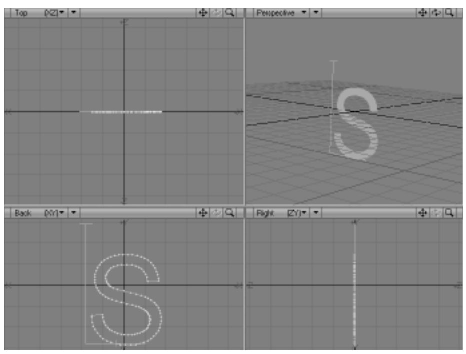
You can also enter specific numerical values in the Pen tool numeric panel. Clicking the **Add Point** button creates the point, essentially mimicking the RMB.



Pen Numeric Panel

THE TEXT TOOL

The Text tool (**Create** > Elements: **Text**) lets you interactively create type objects using True Type and Adobe PostScript Type 1 fonts. Once you create them, these objects can be beveled, extruded, drilled, and more.


Using the Text Tool

NOTE
If the **Text** tool button is ghosted, make sure you have added some fonts, as discussed below.

The Edit Font List Panel

Before you can use the Text tool interactively, you must have previously loaded one or more fonts into the font list. This is done using the Edit Font List panel (**Modeler** > **Options** > **Edit Font List**).

Edit Font List panel

Use the **Font** pop-up menu to select the font for your object. Clicking the **Load Type-1** button brings up a file requester where you can load a PostScript font. If you have PostScript fonts, the files often have a .PFB filename extension. Clicking the **Add True-Type** button brings up a font selection requester you can use to select a font. (Note: the point size is irrelevant.)

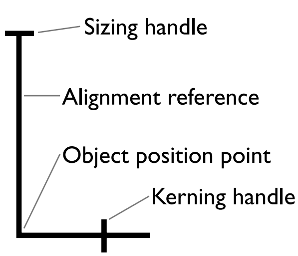
Click **Clear Entry** to remove the selected font from the font list. **Clear List** will purge the entire list. **Load List** brings up a file requester where you can load a previously saved font list file. This can be the LWM.CFG file. **Save List** brings up a file requester for saving the current font list to a file. Note that the font list is automatically saved when you exit Modeler, so it is not mandatory that you save the list. Use this option to create custom file lists for special purposes.

NOTE
The clearing operations do not affect the actual files stored on your hard drive. Also fonts in the list utilize no RAM until used.

Interactively Creating Text

When you first click in a viewport after selecting the Text tool (**Create** > Elements: **Text**), the text insertion pointer appears. You can just start typing to create the Text *template*.

The Text Insertion Pointer



You can use the BACKSPACE or DELETE keys to clear the last character typed. To clear the entire text line, press SHIFT + BACKSPACE or SHIFT + DELETE. If you need to use a Modeler keyboard shortcut, like the N key, while using the Text tool, first press the ESC key to exit the Text tool, then press the desired shortcut key.

Modifying the Template

The position of the text is referenced to the bottom left of the pointer. You can reposition the text in any viewport. Dragging the sizing handle up or down will scale the text larger or smaller. Dragging the kerning handle will adjust the amount of space between characters.

NOTE
The sizing handle cannot be dragged past the bottom.

Text will be aligned with reference to the vertical bar. Pressing the TAB key will cycle through the alignment options.

You can cycle through the font list using the UP and DOWN ARROW keys or you can use the numeric requester. If you have already started a template, the font will change.

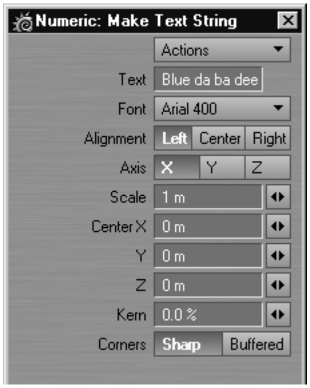
You must deselect the Text tool or select another tool to actually create the object. Alternatively, clicking your RMB will create the object and also move the template to your mousepointer position. You can quickly make unlimited copies of the text in this manner.

The number of points used to approximate curves in characters depends on the setting for **Curve Division** on the General Options panel (**Modeler** > **Options** > **General Options**).

NOTE
The characters that are available within the selected font may be limited. You may or may not have all possible characters allowed by the PostScript format depending on whether those characters were created in the font originally.

The Text Numeric Panel

The related numeric panel provides interactive feedback and also lets you set options numerically. **Axis** is the perpendicular axis used in creating the text; however, it is usually set interactively by clicking the mouse in a view window. The **Text** input field lets you enter the text to generate into a 3D object.



Text Numeric Panel

From the **Font** pop-up menu, you can select any of the fonts defined in the previously discussed Fonts List.

Some fonts, when converted into 3D objects, have additional points near their corner vertices. When these polygons are beveled, these additional points can cause beveling inaccuracies. Use the default setting, **Sharp**, for the majority of your text generation, as this will avoid the creation of such points. The **Buffered** setting allows the additional points to be created, should you decide that you wish it.

The **Alignment** setting determines how the text is aligned around the text insertion pointer. **Center** is the XYZ position of the insertion point; however, it is usually set interactively by clicking the mouse in a view window. The **Scale** value sets the general height of the text; however, the actual size may be less and will vary from font to font. **Kern** sets the spacing between fonts and depends on the font's own characteristics. Negative numbers are allowed to bring characters closer together.

NOTE
The **Scale** setting (numeric panel) cannot be set to zero or a negative number.

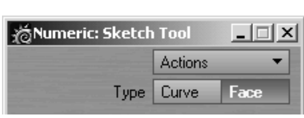
THE SKETCH TOOL

The Sketch tool (**Create** > Elements: **Sketch**) is purely a two dimensional modeling tool. With it, you can draw polygons (and curves, discussed later) in any viewport.

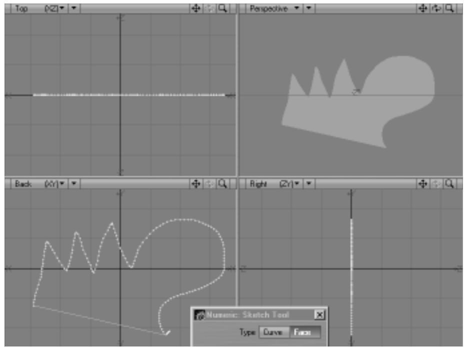
To sketch a polygon:

1 Choose **Create** > Elements: **Sketch**.

2 Open the numeric panel and select **Face** as the **Type**.



3 Drag out the desired shape. (It is not necessary to touch the beginning of the line with the end. The polygon is closed automatically.) When you release your mouse button, a polygon is created using the shape you sketched out. The polygon is always centered in the depth dimension of the viewport.



Sketching Curves

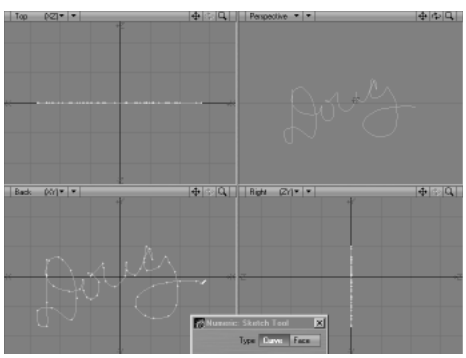
You can use the Curve option on the numeric panel of the Sketch tool to create a spline curve instead of a polygon. The tool operates exactly as previously described, except an open spline curve is created instead of a polygon.

To sketch a curve:

1 Click **Create** > Elements: **Sketch**.

2 Open the numeric panel and select **Curve** as the **Type**.

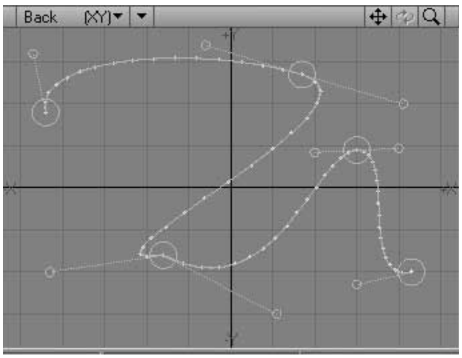
3 Drag out the desired shape. When you release your mouse button, a curve is created using the shape you sketched out. The curve is always *written* on a plane positioned at 0 of the viewport's perpendicular axis.



NOTE
You may be already familiar with curves like these since many desktop drawing programs use them. Spline curves will be discussed in detail later.

THE BEZIER TOOL

The Bezier tool (**Create** > Elements: **Bezier**) allows you to create standard spline curves using tangent controls.



Bezier Tool

To use the Bezier tool:

1. Choose **Create** > Elements: **Bezier**.
2. Place your mousepointer where you would like to begin your curve.
3. Click and hold your LMB—the beginning and ending of your first curve is created. Drag your mouse to where you want the curve to end and release the mouse button.
4. Successive clicks will create additional curves.
5. Drag the tangent handles (circles) to adjust the curve shape. Holding the CTRL key while dragging will allow you to move both sides of the tangent simultaneously.
6. Drop the tool by hitting the SPACEBAR.

NOTE
Once you drop the Bezier tool, a normal Modeler curve is created. As such, you cannot re-edit the curve using the Bezier tool.

On the numeric panel, you can set the subdivision level (for smoother or less smooth curves), create a closed curve, and delete the last tangent point.

Bezier tool numeric panel

SPLINE CURVES

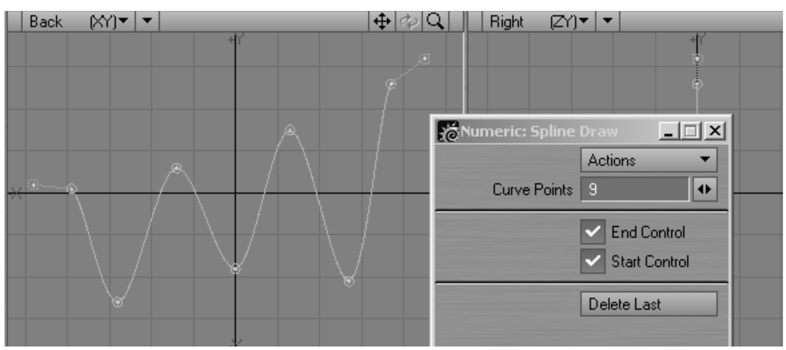
In addition to making polygons from points, you can also create curves. Curves are a powerful modeling tool and help you create smooth edges and flowing organic objects or object details. They are also used for some modeling tools as a directional path.

You can create curves using a minimal number of points, yet achieve a smoothness that would take many times the number of points if you attempted to mimic the curve with a polygon. Moreover, with fewer points, it is easy to subtly or dramatically change the curve's shape. However, by themselves, curves will never render in a finished image. They are essentially a free-form modeling tool used to create polygonal objects.

There are basically two types of curves: open and closed. An open curve has a beginning and an end—essentially a curving line. A closed curve has no beginning or end; it is a closed loop. A circle is a closed curve.

The Spline Draw Tool

The Spline Draw tool (**Create** > Elements: **Spline Draw**) is simply a tool for making (open) spline curves. To use the tool, just click and move points. Clicking off the curve adds new points to the end.



The Spline Draw tool

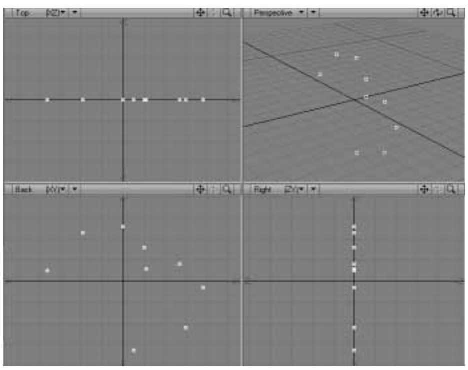
Its numeric panel offers options for making control points, discussed later, and deleting the last point from the curve. Changing the **Curve Points** value will *resample* the curve into more segments. Note that resampling the curve can change its shape, particularly if the number is incremented gradually, like with the mini-slider. If you want a more refined curve that accurately matches the one you have drawn, enter the new number directly into the field to avoid the intermediate curves.

Creating from Points

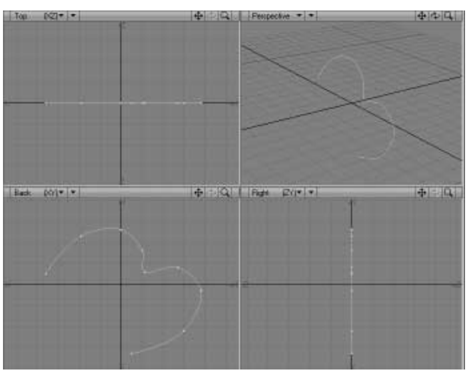
You can also create curves from points in a manner nearly the same as creating polygons.

To create a curve:

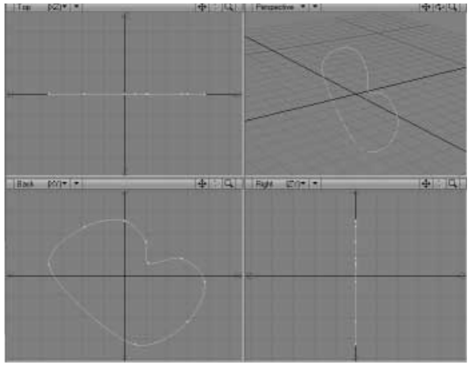
1. Create a few points. Clicking with your RMB using the Points tool (**Create** > Elements: **Points**) will work.
2. Your points must be selected in the order you want the curve to flow. However, if you immediately go to—or were using—the Point Selection mode, the points should already be in the selected state. If not, select them in the desired order.



3. To create an open curve from the points, click **Create** > Elements: **Make Curve** > **Make Open Curve** or press CTRL + P.



To create a closed curve from the points, click **Create** > Elements: **Make Curve** > **Make Closed Curve** or press CTRL + O (the letter O).



You can modify the curve's shape by simply moving the individual points. Notice how the curve flows smoothly through the interior points. It is easiest to see this as you drag points on either side of a point. If you need additional control, you can add additional points to an existing curve.

To add points to a curve:

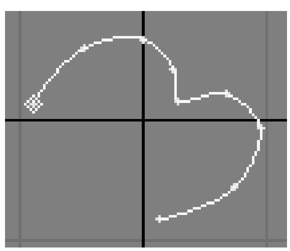
Activate the Add Points tool (**Construct** > Subdivide: **Add Points**) and click anywhere along the curve.

To delete points from a curve:

Select the points and use the **Cut** command (X).

Curve Direction

Curves also have a head and tail. This is important for certain modeling operations discussed elsewhere. When the curve is selected in the Polygon Selection mode, the head is indicated by a small diamond; it is the first point you selected when creating the curve.



You can flip the head and tail around by using the Flip command (**Detail** > Polygons: **Flip**).

Using Control Points

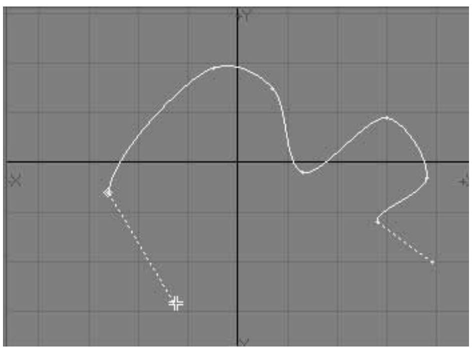
When using a curve as a modeling tool, sometimes you need to adjust the curve to get it to "that perfect angle." Because of the way points control an open curve's shape, it can be difficult to get the desired result at the ends of a curve. Fortunately, Modeler lets you have extra "control" points at each end, which affect the curve shape, but don't actually become part of the curve's modeling characteristics.

To activate control points on a curve:

1 If you have more than one curve, select your desired curve first using the Polygon selection mode.

2 Choose **Detail** > Curves: **Control Points** > **Begin Control Point** to detach the first point from the curve. It is your starting control point. Dragging this point will change the shape of the curve at the new starting point.

3 Choose **Detail** > Curves: **Control Points** > **End Control Point** to detach the last point from the curve. It is your ending control point. Dragging it will change the shape of the curve at the new ending point.



The control points commands will also toggle the control point state off, if selected again.

NOTE
It is perfectly legal to have only a beginning or ending control point on a curve.

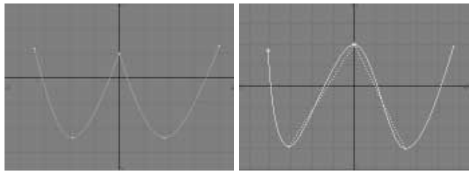
HINT
If you plan to use control points on a curve, you should add an extra point at the beginning and end of a curve for that purpose.

Smoothing Two Overlapping Curves

When two curves share the same end point, because of point merging or some other reason, the curve may not flow smoothly through that point. This can be particularly troubling when designing a complex *spline cage*. You could kill the curve (k), reselect the points and create a new curve, but there is an easier way.

To smooth curves sharing a common end point:

Select the two curves (Polygon selection mode) and click **Detail** > Curves: **Smooth**. This smoothes the *joint* as though the two curves were originally plotted as one continuous curve—even though they remain individual curves.



Before (*left*) and after (*right*) using Smooth

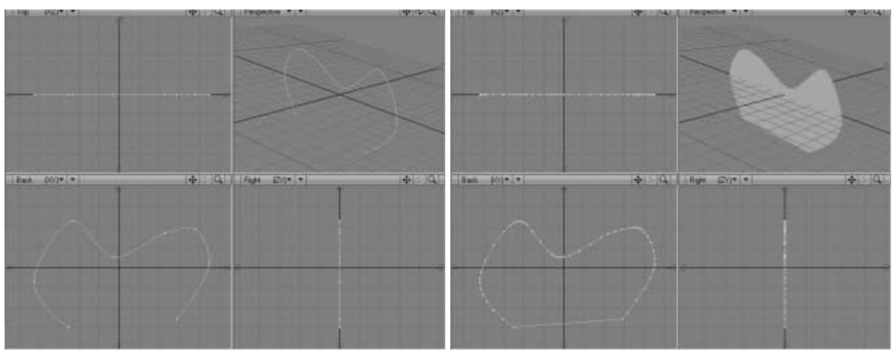
NOTE
Make sure that the point they share is in fact one point. (If not, merge the points into one.)

Turning Curves into Polygons

Since curves will never be seen by themselves, sometimes you need to turn them into polygons.

To convert a curve into a polygon:

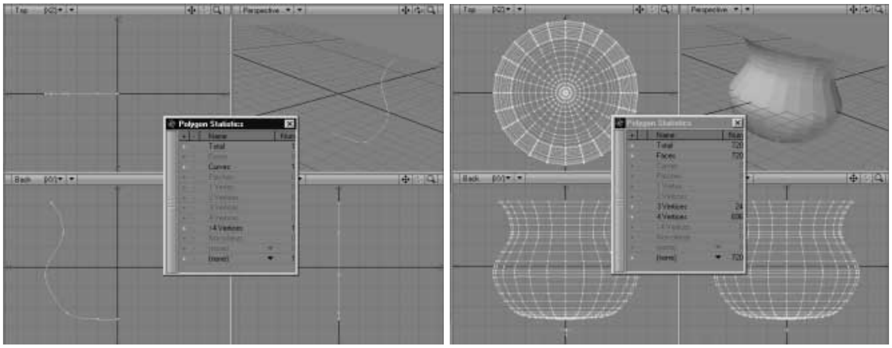
Select the curve(s) and choose **Construct** > Convert: **Freeze**. Each independent curve will become a distinct polygon. An open curve will be closed (between its end points) before it is converted to a polygon.



The number of points that Modeler will use to approximate the curve depends on the setting for **Curve Division** on the General Options panel (**Modeler** > **Options** > **General Options**).

Using Modeling Tools on Curves

Most of the standard modeling tools will work on a curve as well. However, a curve is sometimes converted into a polygon before the tool is applied. This happens with the Extrude and Lathe tools (**Multiply** menu), for example.



Left: Curve. *Right*: Lathed object made of polygons

METABALLS AND SKELEGONS

See Chapter 29 for information on Metaballs and Metaedges. Skelegons are covered in Chapter 10.

THE POINT TOOL

Remember a point needs to have X, Y, and Z coordinates. You can create points in real-time or by positioning first and then creating.

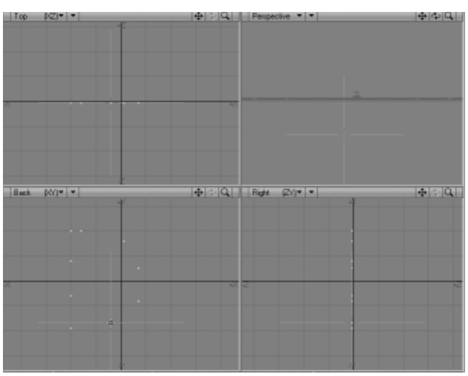
To create a point:

1 Choose **Create** > Elements: **Points** to activate the Points tool.

2 In any viewport, with your LMB drag the large crosshairs to the desired position (you can reposition in any viewport). The information display (bottom-left corner) will give you position feedback as you move your mouse.

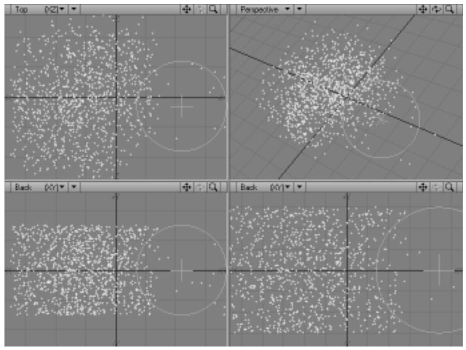
3 Click the RMB to create the point. Be careful not to move your mouse pointer. Deselecting the Points tool will also create the point, if you don't have a steady hand. Deselecting and then reselecting the Points button will create the point and keep you in the create point mode.

4 You can also position the point by dragging with your RMB. However, the points will all be created along the same plane (when you release the RMB).



THE SPRAY POINTS TOOL

Sometimes you'll want to create a random splattering of single-point polygons for stars or perhaps gases when used with volumetrics. The Spray Points tool (**Create** > Elements: **Spray Points**) acts like a points spray gun. Just activate the tool and drag your mouse pointer.



Spraying points

You can control the flow (**Rate**) and **Radius** of the spray on the numeric panel, as well as specify a color vertex map and color.

MAKING A POLYGON

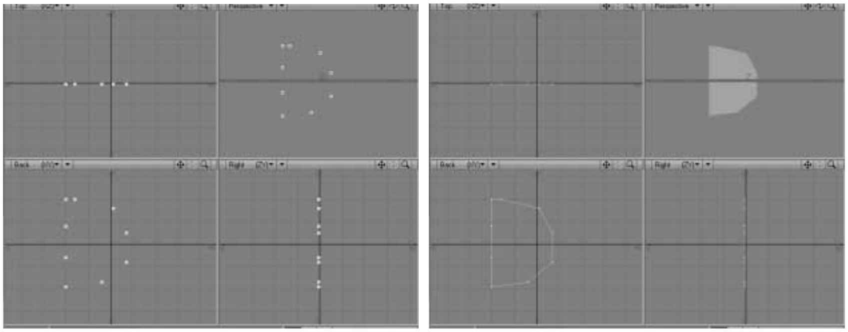
As we have discussed, polygons are made up of points. Excluding the one and two-point specialty polygons, polygons should contain at least three points. You will likely find that your finished objects are constructed of a variety of polygons with different numbers of points.

NOTE

Don't get discouraged if making polygons seems very tedious. Not that it isn't, but most of the time you'll create your objects by starting with primitives. Happily, creating polygons from points is the exception rather than the rule.

To create a polygon from points:

In the Point Selection mode, select your points in a clockwise order. Then, click **Create** > Elements: **Make Pol**.



Left: Points for polygon. *Right*: Resulting polygon

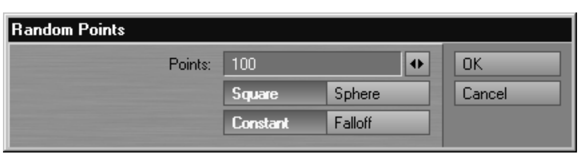
NOTE
Once you create a polygon, you'll want to view it in the Polygon Selection mode to see if its surface is facing the correct direction based on the normal, discussed later.

SINGLE-POINT POLYGONS

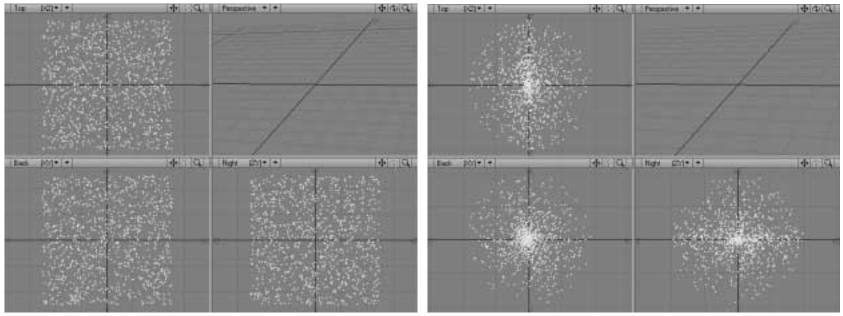
Sometimes you will want to turn single points into polygons. This is necessary for objects like stars in a starfield, since you cannot assign surface attributes to a point. Choose **Create** > Elements: **Points to Polys** to turn selected points into single-point polygons.

RANDOM POINTS

Use the Random Points command (**Create** > Elements: **RandPoints**) to create a defined number of points distributed randomly. You can choose between a **Square** or **Sphere** shape. **Constant** generally confines the points to the selected shape, while **Falloff** tapers off the point distribution along the perimeter.



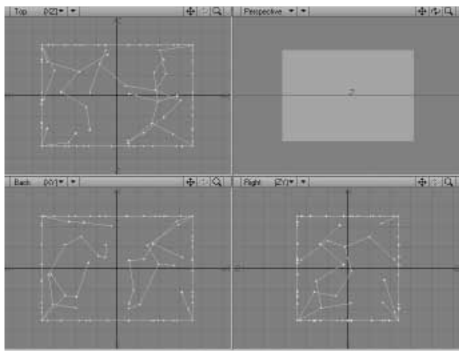
Random Points



Left: Square. *Right*: Sphere

Random Pricks

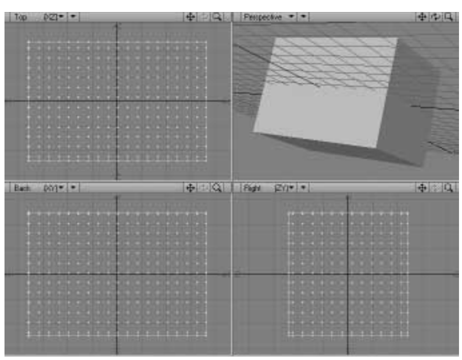
Create > Elements: **RandPricks** creates random surface points on polygons. Use this to add detail to an object where you will later kill all of its polygons.



Random Pricks

Stipple

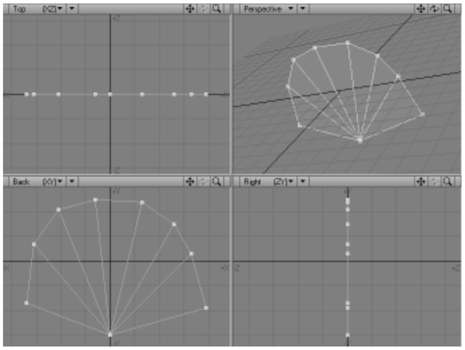
Create > Elements: **Stipple** creates a grid of regularly spaced points over the surface of the object in the active layer. The **Spacing** fields are for separate XYZ point spacing settings.



Stipple

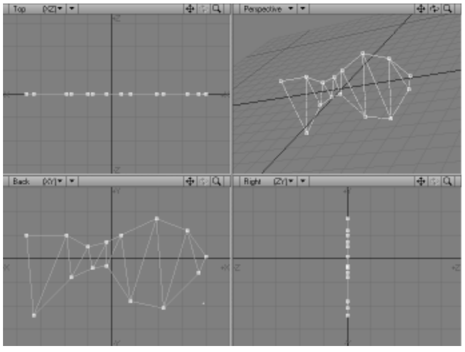
TRIANGLES

You can use the Triangle Fan command (**Create** > Elements: **Make Fan**) to create a subdivided polygon from selected points. (Selecting in a clockwise fashion works best.) The triangles will share the first point selected.



A triangle fan

The Triangle Strip command (**Create** > Elements: **Make Strip**) is similar, but creates a strip of triangle polygons from a set of points. This can be useful for game engines that prefer to have geometry built in strips of triangles. The command basically creates triangles in a strip of first point, second point, third point-poly, and so on.

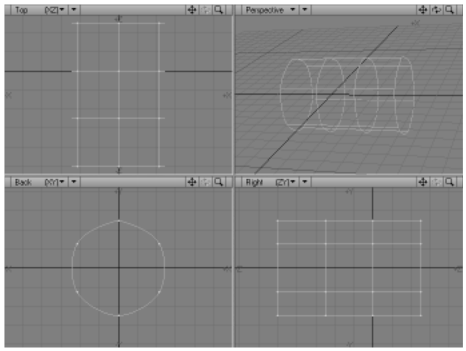


A triangle strip

CAGE

Create > Elements: **Cage** creates a cylindrical spline cage (connected curves) using **Axis**, **Sections** (i.e., segments + 1), **Sides**, **Top**, **Bottom**, and **Radius** settings.

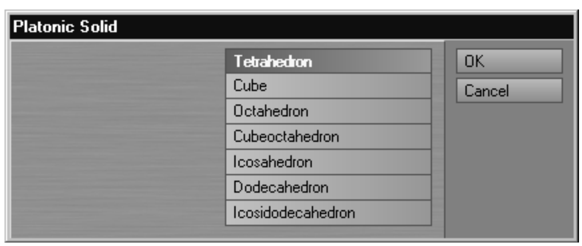
Cage dialog


Cage result

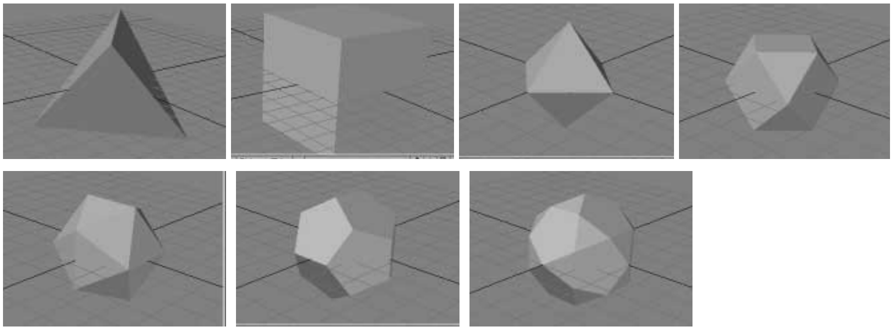
OTHER CREATION FUNCTIONS

Platonic Solid

Choose **Construct** > Utility: **Additional** > **Platonic** to generate any of the seven platonic solids—yes, even including the perennial favorites Cube and Tetrahedron!



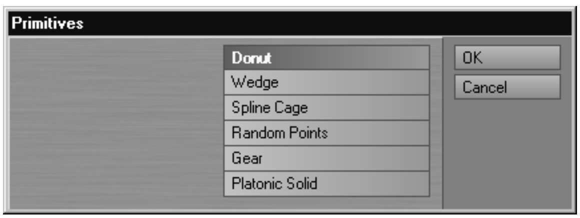
Platonic Solid



Top to bottom, left to right: **Tetrahedron**, **Cube**, **Octahedron**, **Cubeoctahedron**, **Icosahedron**, **Dodecahedron** and **Icosidodecahedron**

Primitives

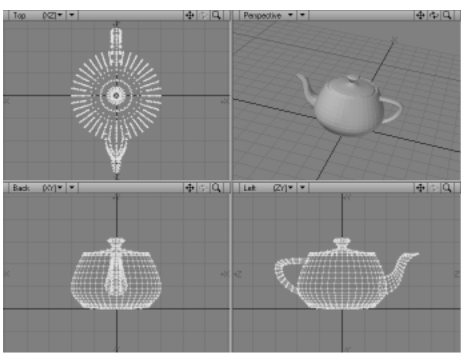
Choose **Construct** > Utility: **Additional** > **Primitives** to display a menu for the following six commands: Toroid (**Donut**), **Wedge**, Cage (**Spline Cage**), RandPricks (**Random Points**), **Gear**, and Platonic (**Platonic Solid**).



Primitives

Teapot

Spot o’ tea anyone? Generate a quick teapot by choosing **Construct** > Utility: **Additional** > **Teapot**.



Teapot

chapter 23
Modifying Geometry

Chapter 23: Modifying Geometry

Moving, rotating, and scaling points and polygons is easy with the tools (and functions) available on the Modify menu. Generally, the tools work on points or polygons, depending on what edit mode you have active.

HINT
Modeler tools can be deselected by clicking the highlighted button again or, if it appears in a menu, by selecting it again. An easier method is to just hit the SPACE key, which is mapped by default to Change Selection Mode.

WARNING
If your object uses polygons with more than three sides, you will likely see the effects of non-planar polygons created as you apply some tools. You may need to triple your polygons as a final step.

The Numeric Panel

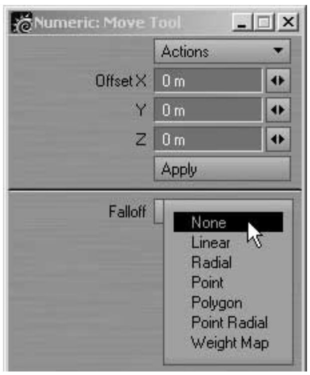
All of Modeler's tools have their own numeric values which will appear on the generic numeric panel, if you have it open (N). It will reflect the numerical values of your actions for the particular tool you are using. Use the numeric panel for precise tweaking. Initially, it will reflect the values for the last interactive use of the selected tool. You may alter the values and click the **Apply** button to use the tool with those settings—this can be done multiple times.

Falloff Mode

As you will notice, many tools are just variations of the same movement and differ only by how the tool's effective range *falls off*. Take the Move tool for example. It simply repositions the selected items equally to a new location will no falloff. With Shear, the range of the effect varies along a linear axis. Magnet is similar, but the effect falls off in a radial pattern. DragNet is much like Magnet except that the center of the radial pattern is at the point you drag. Finally, Drag is a move, but the range is only a single point.

NOTE
The tools on the Modify menu are grouped by movement (Move, Rotate, etc.)

All of these tools have a **Falloff** pop-up menu on their respective numeric panels. By changing the **Falloff** setting, you can effectively make one tool function the same as another tool that uses that **Falloff** by default. In other words, with the Move tool, you could get the same effects as using the Shear, Magnet, DragNet, and Drag tools, by just changing its **Falloff** setting. You might think of these tools as just shortcuts to a defined group of modify settings.



Falloff Meanings

None — **None** means no falloff. Thus, your basic Move, Rotate, Size, and Stretch tools are set to this by default.

Linear — **Linear** makes the effect falloff along a linear axis, so it is the default for Shear, Twist, Taper1, and Taper2.

Point — **Point** means falloff is immediate around a point, so you can move only a single point. This mode is used by Drag.

Point Radial — **Point Radial** is similar to **Radial** except that the center of the falloff is at the point being dragged. DragNet uses this setting.

Polygon — **Polygon** means falloff is immediate around a polygon, so you can move only a single polygon.

Radial — **Radial** means the effect will fall off in a radial pattern (in a cylindrical or spherical shape), so it is the default for Magnet, Vortex, Pole1, and Pole2.

Weight Map — **Weight Map** uses the selected Weight map for the falloff. *Unweighted* points will not move at all, while heavily weighted points will move the most. Negatively weighted points will move inversely. This allows you to use an irregularly shaped falloff. (See Chapter 28.)

NOTE
Interestingly, you can actually rotate a single point using the Rotate tool and **Falloff** set to **Point**; however, it will have no meaningful effect.

Setting Linear Falloff

The linear-falloff tools are Shear, Twist, Taper 1, Taper 2, and Bend. By default, a linear-falloff tool's effect is automatically applied to the object (or selected items) 100 percent at one end and zero at the other, along the axis perpendicular to the editing viewport. This is indicated by the **Range** option at the bottom portion of the numeric panel (N), which has **Automatic** selected.

To use the tool in the **Automatic** mode via the numeric panel (by clicking the **Apply** button), select the perpendicular **Axis**. (If the axis has not been set either manually or by a previous mouse-based modify operation, the **Axis** setting is used to compute the falloff when you click **Apply**.)

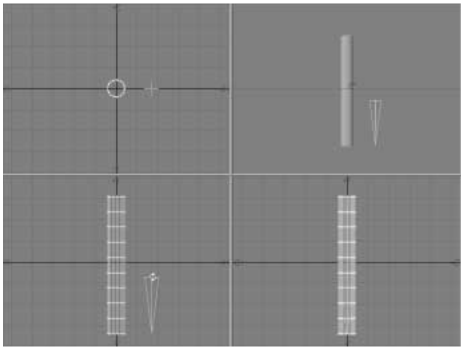
HINT
Using the **Automatic** range is the simplest method. Thus, if possible model your object along one of the three axes.

Defining a Specific Range

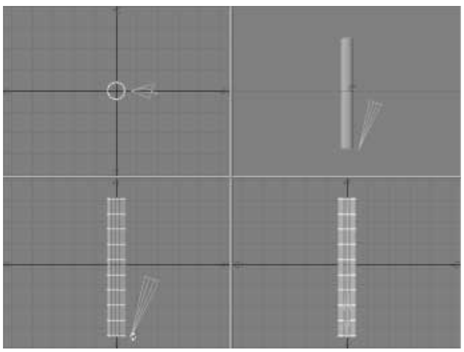
Instead of using an **Automatic Range**, you can define a specific **Fixed Range** using an *axis tree*. This type of **Range** can be placed at any position and at any angle in 3D space. The effect tree looks like a pair of crossed wedges. The tool's effect is applied along this tree with zero at its tip and 100 percent at its base (fat end)—portions of the object beyond the base are still affected 100 percent. The edges of the tree reflect the fall-off curve shape.

To create a fixed range:

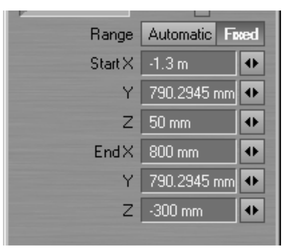
1 Select the desired linear-falloff tool and drag out the axis tree with your RMB. (The **Range** option will be set to **Fixed** on the numeric panel.)



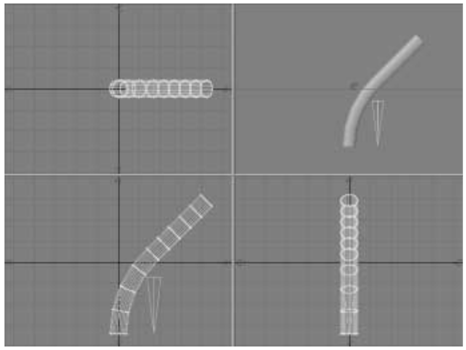
2 Reposition the ends by dragging them with your RMB.



The XYZ positions of the starting and ending points are reflected in the numeric panel. You can edit these values, if necessary.



In the image that follows, we used the Bend tool. Notice the impact of the fixed range. Geometry from the wide end of the axis tree and beyond is 100 percent impacted by the tool. The opposite end is not affected at all, with a falloff in between.

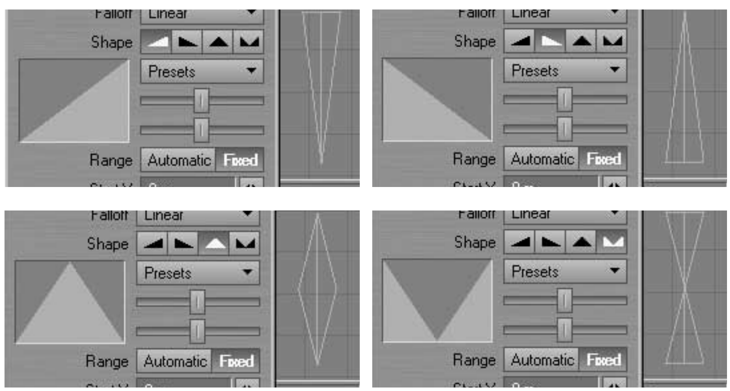


To reset to automatic range:

To reset to automatic range, click your mouse pointer on an inactive interface area. Note this will also switch you back to the **Automatic Range** mode.

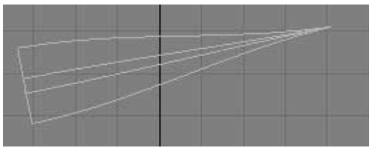
Setting the Falloff Shape

You can quickly select between various basic falloff directions using the **Shape** buttons. (The fixed-range axis tree is shown here for illustration purposes. It will not appear if you use an automatic range.) You can also use the LEFT and RIGHT ARROWKEYS to change the selected **Shape**.



You can adjust how the effect falls off using the two sliders, which act like tension spline controls for the beginning (upper slider) and ending (lower slider). You can quickly set up some common curves by selecting from the **Preset** pop-up menu. The graph on the numeric panel, as well as the outer edges of the axis tree (when using a fixed range) gives you a visual picture of the falloff. You can also use the UP and DOWN ARROWKEYS to cycle through the **Preset** selection.

Adjusting falloff curve shape

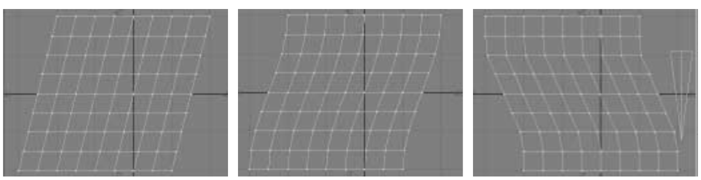


Axis tree

NOTE
The falloff settings apply equally to an automatic range.

NOTE
The Bend tool does not have falloff options.

Here are some falloff examples:

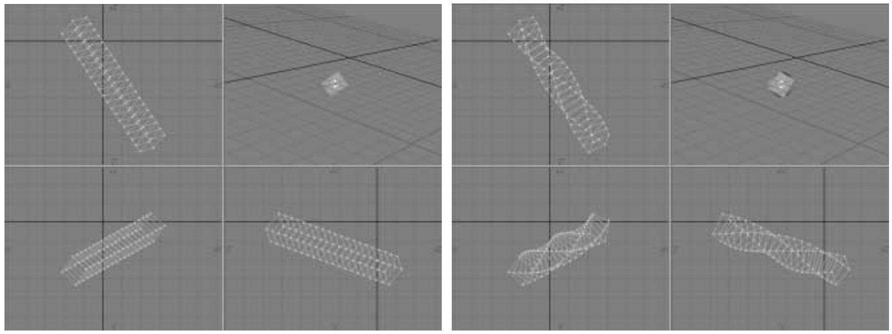


Left to right: Automatic Linear; Automatic Smooth; and Fixed Linear

Flexing an Object on an Arbitrary Axis

Let's say you had an object that was not aligned nicely along the X, Y, or Z axis. If you wanted, you could define a fixed range and align the effect tree with the object. However, this can be a tedious process. A better alternative would be to use a Perspective viewport.

Remember, editing generally occurs along a viewport's arbitrary horizontal and vertical axes. Using a Perspective viewport, this can be something other than directly along the X, Y, or Z axis. Thus, you can align your object so that it is perpendicular in the viewport and just use the **Automatic Range**.



Twisting in perspective viewport

SETTING RADIAL FALLOFF

The radial-falloff tools are Magnet, Vortex, Pole Evenly, and Pole. Their influence area falls off in a radial pattern (a sphere or cylinder). This influence area is set independently of any selected polygons or points, but works in conjunction with them. That is, if points/polygons are selected, only those within the influence area are affected.

By default, the tool's effect is automatically applied to the entire object (or selected items). This is indicated by the **Range** option, at the bottom portion of the numeric panel, which will have **Automatic** selected.

Automatic range

Setting the Falloff Shape

How the effect falls off is determined by the two sliders, which act like tension spline controls for the beginning/end (upper slider) and center (lower slider). You can quickly set up some common curves by selecting from the **Preset** pop-up menu. The graph gives you a visual picture of the falloff.

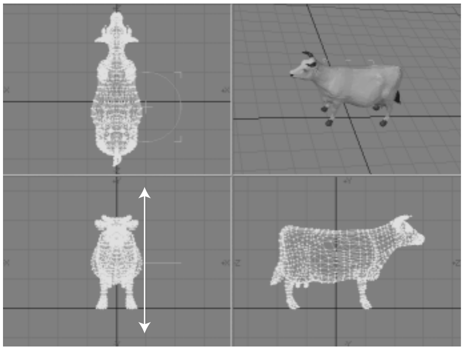
Various falloff shapes

Defining a Specific Range

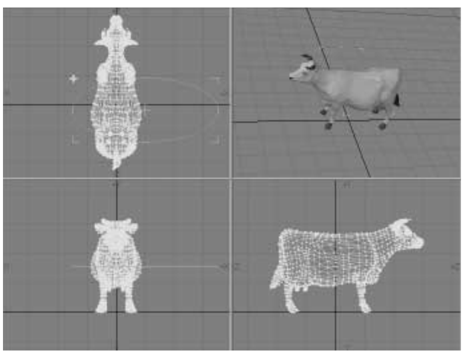
Instead of using an **Automatic Range**, you can define a specific **Fixed Range**, which can be positioned and sized anywhere in 3-D space.

To set the influence area:

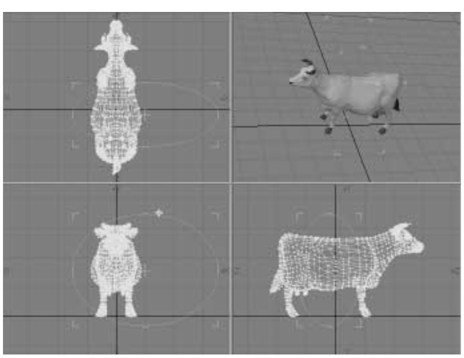
1 Select the desired radial-falloff tool and drag out the influence area tree with your RMB. (The **Range** option will be set to **Fixed** on the numeric panel.) At this point, you have defined a cylindrical area that extends infinitely along an axis perpendicular to the viewport. Use the CTRL key to constrain to a circle.



2 Resize the initial area as needed by dragging the outline edge handles with your RMB or reposition by dragging the center handle.



3 To limit the depth of the area, drag out the outline in a different viewport. The amount of influence is strongest at the center and diminishes towards the edges. Polygons with all points outside of the area will not be affected.



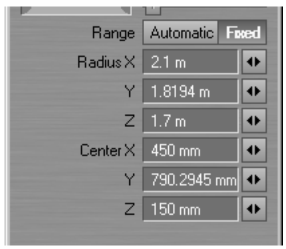
NOTE
Creating the radial-falloff influence area is similar to creating a Box primitive, particularly in a perspective viewport.

To reset the influence area:

To reset the influence area, click your mouse pointer on an inactive interface area. Note this will also switch you back to the **Automatic Range** mode.

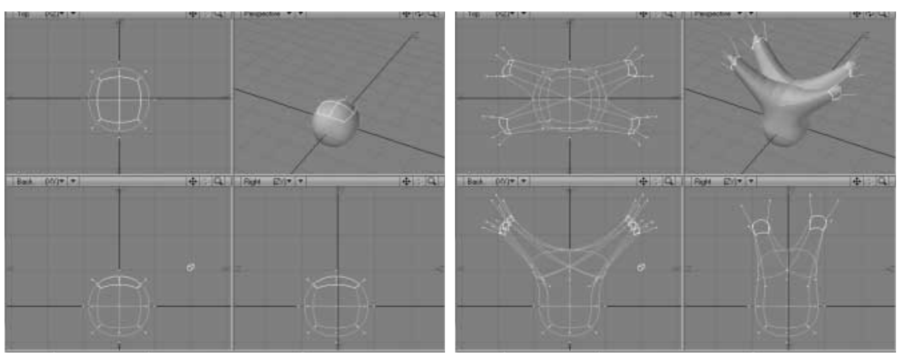
The Numeric Panel

The **Radius X**, **Y**, and **Z** values reflect the size and shape of the influence area. If you have defined a cylindrical (i.e., two-dimensional) area, one axis will be 0. The **Center X**, **Y**, and **Z** values define the center of the area of influence.



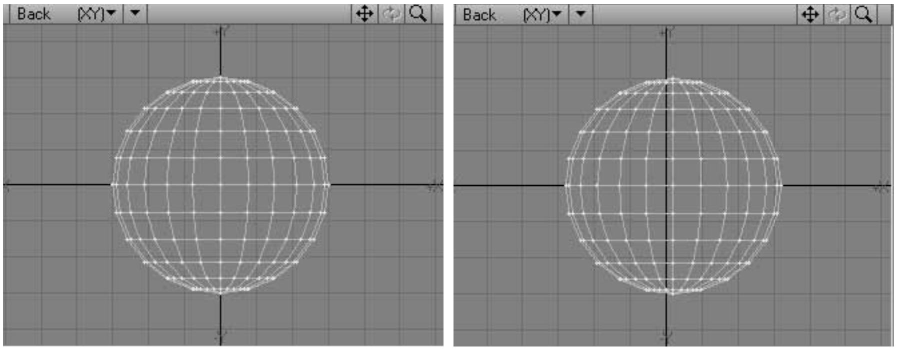
SYMMETRICAL EDITING

The Symmetry mode (**Symmetry** button, located along bottom edge) not only works on selection, but also on editing. Operations on the positive side of the X axis also inversely affect the negative side of the X axis. When this mode is active, your object is theoretically split in half at X=0.



Generally, you should perform all of your edits on the positive side of the X axis when using Symmetry. Using the negative side may lead to unpredictable results.

If you get unpredictable results, make sure the negative-side geometry is *exactly* the mirror of the positive-side geometry with X=0 as the center point. If the negative-side geometry is the slightest bit off, it will not be affected. Generally speaking, this mode was meant to be used where the geometry to the left and right of X=0 mirror each other. (However, all of the geometry does not need to be a mirror image, only the portion you want to work on symmetrically.)

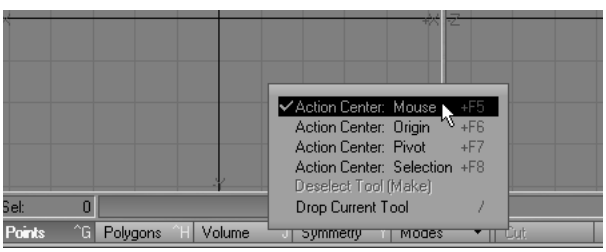


Left: The ball is symmetrical. *Right*: The ball is not symmetrical

> **NOTE**
> The **Fit Selected** command (SHIFT + A) respects symmetry. If both sides of an object are selected and Symmetry is active, then only one-half of the selection is used to compute the *fit*.

ACTION CENTER CONTROL

The rotate and stretch tools (i.e., Rotate, Size, Stretch, Twist, Taper 1, Taper 2, Vortex, Pole 1, and Pole 2) can use different centers. This **Action Center** state is set on the **Mode** pop-up menu. **Mouse** means to use the mouse position. **Origin** is XYZ 0, 0, 0. **Selection** uses the center of a bounding box around what is currently selected. **Pivot** uses the first layer's pivot point.



Modeler Modes menu

THE MOVE GROUP

This group of tools and functions are used to reposition geometry.

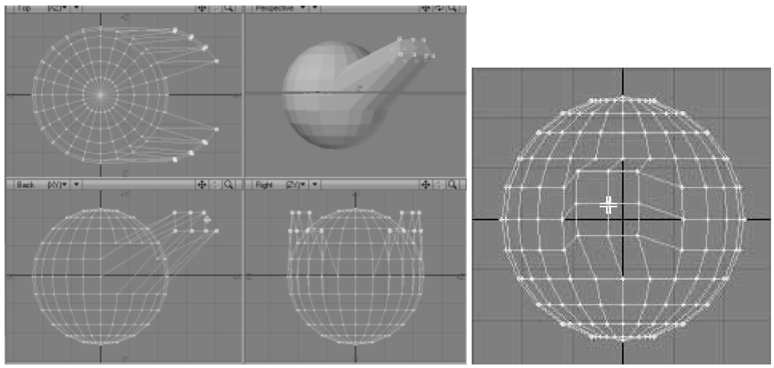
The Move Tool

The Move tool (**Modify >** Move: **Move**) is the basic tool used to reposition geometry.

To move geometry:

1 Select your points or polygons.

2 Choose **Modify** > Move: **Move** and drag your mouse in any view. The selected geometry will follow your movement.



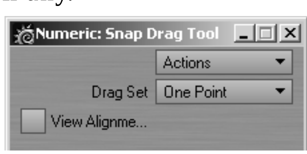
Left: Moving points. *Right*: Moving a polygon

3 Hold the CTRL key before you drag to constrain the movement along one of the axes.

NOTE
To directly set a point's location, see Chapter 27, "Point Information."

The Snap Tool

The Snap tool (**Modify >** Move: **Snap**) allows *points* to be dragged and snapped to align with other points. When you drag a point with the Snap tool, the point is snapped to the grid or to one of the non-moving points, if any.



If you set **Drag Set** to **One Point** on the numeric panel, only the single point will move. If you choose **Connected Points**, points that are connected to the point being dragged will also move. If you choose **All**

Points, all points will move. As with other tools, you can use selection to limit which points are affected. Snap makes it easy to manipulate *unconnected* parts without having to make explicit selections.

With **View Alignment**, points will be *snapped* to approximately align with the other points along an axis perpendicular to the viewport, but not fully in three dimensions.

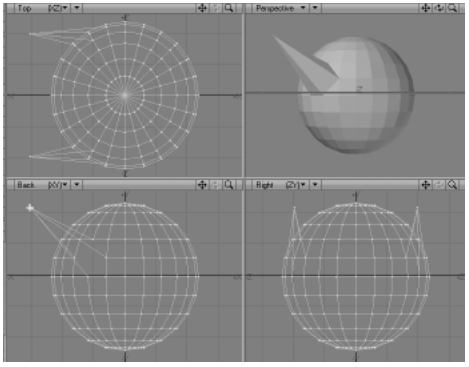
NOTE
Although this tool works on individual points, selecting polygons and working in the Polygon Edit mode will limit its effect to points that share a vertex with the selected polygons.

The Drag Tool

To freely drag points use the Drag tool.

To drag points:

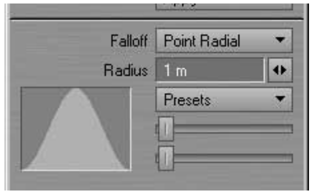
1. To limit the edit to certain points, select them first. The Drag tool will not affect unselected points unless none are selected. Thus, if you have points in-line with each other in a viewport and don't want to drag all of them, select the target points first. (Note that all of the selected points do not move as you drag, only the ones under your mouse pointer.)
2. Choose **Modify** > Move: **Drag** to access the Drag tool.
3. Position your mouse pointer directly over a point and drag.



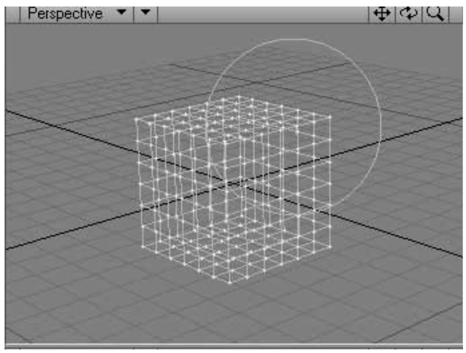
The DragNet Tool

Dum Da Dum Dum. The story you are about to hear is true... The DragNet tool (**Modify** > Move: **DragNet**) combines the Drag and Magnet tools. While you might use it somewhat interchangeably with Magnet, they operate quite differently. Magnet's radial-falloff influence area works independently from the actual use of the tool. With DragNet, the spherical range is always centered around the initial point of dragging. Moreover, you must actually drag a point, although other points in the influence area will also move.

DragNet uses the same falloff controls as the other radial-falloff tools. However, there is no automatic range setting.



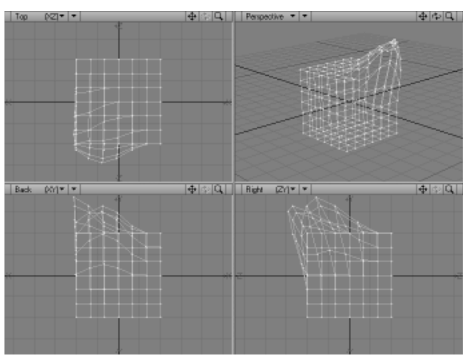
The influence area is defined by the **Radius** around the initial pointer position. You can graphically set this by dragging out a circle with your RMB.



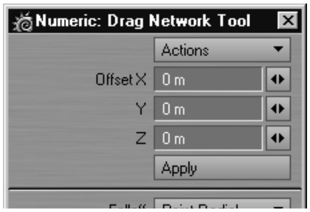
To use the DragNet tool:

Select **Modify** > Move: **DragNet** and drag a point. Other points in the influence area will also move, though not as much.

The information display tells you the **Offset** value, that is, how far you dragged the mouse. If you set the influence area interactively with the RMB, the information display shows the **Radius** value.



The **Offset X**, **Y**, and **Z** values on the numeric panel reflect the amount of movement along those axes for the tool's last operation. You can edit the values and then click the **Apply** button to apply them, which you can do multiple times.



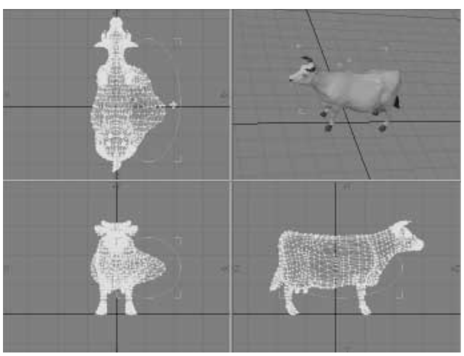
The Magnet Tool

The Magnet tool (**Modify** > Move: **Magnet**) is a nifty tool for pushing in or pulling out sections of objects. The effect is smoothly applied, so it tends to create soft-edged bulges or dips in a surface. You might use the Magnet tool to create an egg shape from a sphere. (Simply enclose the upper hemisphere in the influence range, center the area near the top of the sphere, and drag upward with the LMB.)

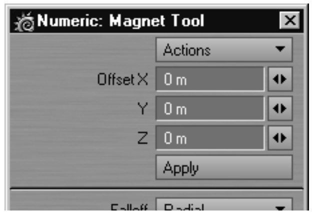
To use the Magnet tool:

Select **Modify** > Move: **Magnet** and drag your LMB in a viewport. The initial pointer location defines the center of the move, although this may not be obvious if it is located outside of a defined range.

The information display tells you the **Offset** amount for the affected axes, that is, the distance you have moved your mouse.



The **Offset X**, **Y**, and **Z** values on the numeric panel will reflect the amount of movement along those axes for the tool's last operation. You can edit the values and then click the **Apply** button to apply them, which you can do multiple times.

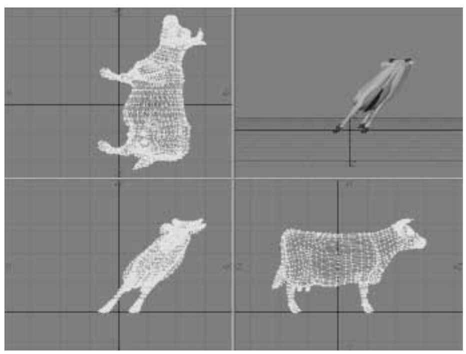


The Shear Tool

No, this doesn't have anything to do with sheep. (Good guess though.) Shearing is making an object slant to one side. A good example is italicized text. Think of Shear as Move with axial falloff. That is, the strength of the moving influence is not equal across the object, but gradually falls off along a selected axis.

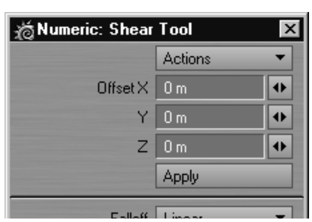
To use the Shear tool:

Select **Modify** > Move: **Shear** and drag your LMB in a viewport in the direction you want the slant.



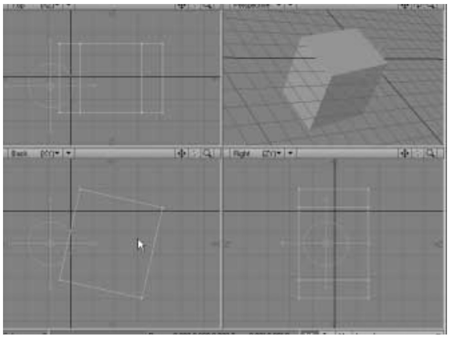
The information display tells you how much **Offset** you applied along two of the three axes, depending on the view you edit in. Hold the CTRL key down while dragging to constrain the shear to your initial dragging axis.

The **Offset X**, **Y**, and **Z** values on the numeric panel will reflect the amount of movement along those axes for the tool's last operation.



The Rove Tool

The Rove tool (**Modify** > Move: **Rove**) allows you to move and rotate with a single tool! To move an object, simply click and drag. Dragging on an axis constrains movement to that axis. Rotate the object about the center of the tool by clicking and dragging the rotation ring. The center of rotation can be repositioned by dragging the center of the tool.



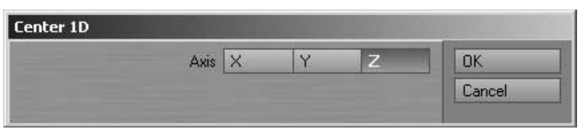
The Rove tool

Center Data

The Center Data command (**Modify >** Move: **Center**) centers the contents of the foreground layer(s) around the Origin (i.e., 0, 0, 0).

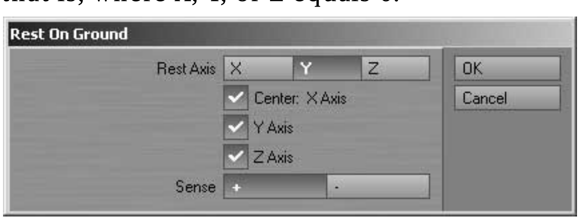
Center1D

The Center1D command (**Modify >** Move: **Center1D**) is similar to Center, except a requester will appear that lets you center along a single axis. The center of the object is placed at the 0 position for the selected **Axis**.



Rest_On_Ground

The Rest_On_Ground command (**Modify** > Move: **Rest_On_Ground**) will move selected polygons so that they rest on a defined ground plane, that is, where X, Y, or Z equals 0.

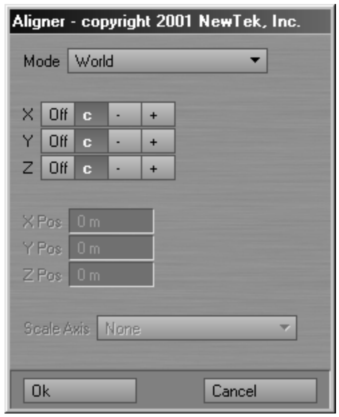


The **Rest Axis** is the axis perpendicular to the plane you want the object to rest on. You may also **Center** the object on any of the axes. The

Sense option rests the object on the positive (**+**) side of the plane or the negative (**-**) side. The default setting would be ideal for, say, centering a vehicle around the Origin and placing it so it sits flat on the Y axis plane.

Aligner

Aligner will align geometry not only with axes, but also other geometry. The Mode setting determines what type of alignment you want.



Aligner in World mode

The **World** mode will align geometry in the foreground to the X, Y and Z axes. The **c** setting will center on the axis. The minus (**-**) or plus (**+**) setting will place the geometry on the negative or positive side of the axis. **Off** disables that axis. So if you had a “floor” at Y = 0 and you wanted the object centered, but resting on this floor, set the Z and X axes to **c** and the Y axis to **+**.

In the **F.G.->B.G.** mode, the foreground is aligned to the background. This mode uses a bounding box around all of the geometry in the reference layer (the background). You can center or position on the negative or positive side of the reference. The **-+** setting will align the right edge of the geometry with the right edge of the reference layer. The **+-** setting will align the left edge of the geometry with the right edge of the reference layer. You may also scale the geometry to the reference layer. Select the axis or axes to scale using the **Scale Axis** pop-up menu.

The **B.G.->F.G.** mode works just like **F.G.->B.G.**, but uses the foreground as the reference for the background.

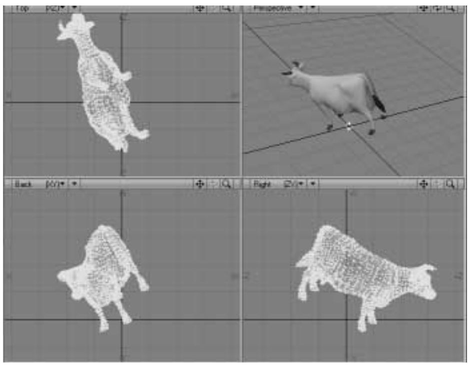
The **Absolute** mode works like the **World** mode except you can numerically define where the center is. In other words, you can make it something other than 0, 0, 0.

THE ROTATE GROUP

This group of tools and functions are used to rotate geometry. Rotating an entire object in Modeler is much the same as doing it in Layout, except you rotate around the perpendicular axis of the viewport. As such, you will often have to use various viewports to rotate an object.

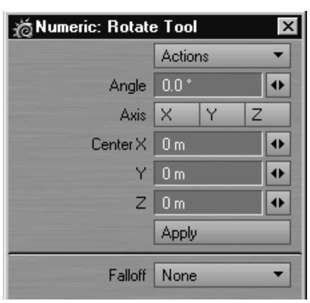
The Rotate Tool

Select **Modify** > Rotate: **Rotate** and drag your LMB left for counter-clockwise rotation or right for clockwise rotation. The center of the effect is determined by the **Action Center** setting, discussed previously. Hold the CTRL key down as you drag to rotate in 15-degree increments. The information display (lower-left corner of the interface) will tell you the angle of rotation, that is, the number of degrees you have rotated.



The Rotate Numeric Panel

The **Angle** defines the degree of rotation. The **Axis** setting defines the axis of rotation. When you click in a viewport, the axis is an imaginary line running perpendicular to the view you are editing in (i.e., coming straight out towards you). For the Back view, it would be the Z axis; in the Top view, it would be the Y axis; and in the Right view, it would be the X axis. If arbitrary axes were used, no **Axis** will be selected.



The **Center XYZ** values define the center of rotation coordinates.

HINT

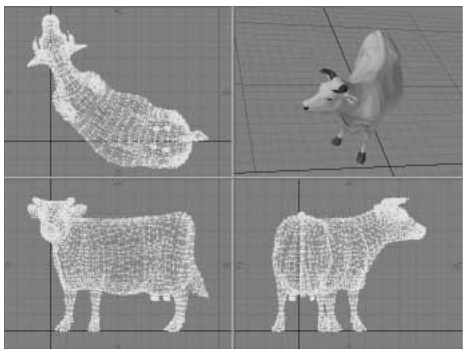
In any viewport, pressing the R key will rotate an object clockwise 90 degrees around a perpendicular axis located at the current mouse pointer location.

The Bend Tool

The Bend tool (**Modify** > Rotate: **Bend**) is a combination of Rotate and Shear. It will move one side of an object and rotate it at the same time, causing the entire object to bend.

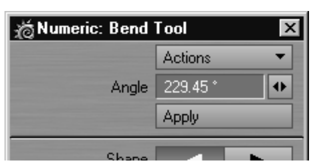
To use the Bend tool:

Select the tool and drag your LMB in the direction you want the bend to occur. The initial pointer location defines the center of the rotation.



The information display tells you the Angle of bending, that is, the number of degrees you bent the object. Hold the CTRL key while dragging to constrain movement to increments of 15 degrees—this makes it much easier to bend along a single axis.

The **Angle** value on the numeric panel will reflect the degrees of rotation.

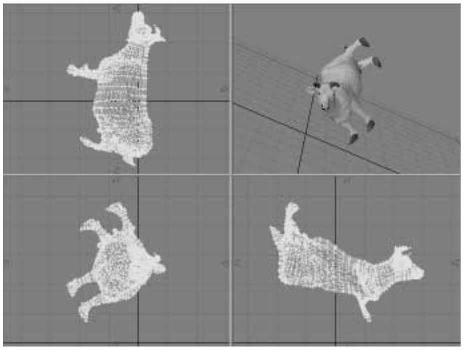


The Twist Tool

The Twist tool (**Modify** > Rotate: **Twist**) is similar to Shear, although the results are quite different. With Shear, the sides of the object are moved one way or the other. With Twist, one end of the object is rotated while the other remains stationary. Think of Twist as Rotate with axial falloff. That is, the strength of the rotational influence is not equal across the object, but gradually falls off along a selected axis.

To use the Twist tool:

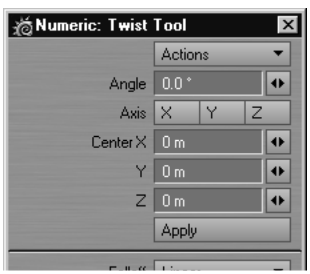
Select the tool and drag your LMB left for counterclockwise twisting or right for clockwise. The center of the effect is determined by the **Action Center** setting, discussed previously.



The cow says "ouch!"

The information display tells you the **Angle** of twisting, that is, how many degrees you twisted. Hold the CTRL key down while dragging to twist in increments of 15 degrees.

The **Angle** value on the numeric panel reflects the degrees of rotation. The **Axis** buttons relate to the rotational axis. If you use an arbitrary axis (i.e., twist in a perspective viewport), no **Axis** buttons are selected. The **Center X**, **Y**, and **Z** values indicate the center point of rotation, through which the axis runs.



The Vortex Tool

The Vortex tool (**Modify** > Rotate: **Vortex**) will smoothly rotate a selected area of an object.

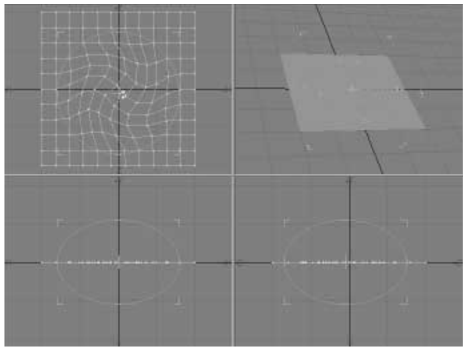
HINT

You might use **Vortex** to create a swirl on an ice cream cone, a whirlpool in an ocean, or a cosmic whorl of stars.

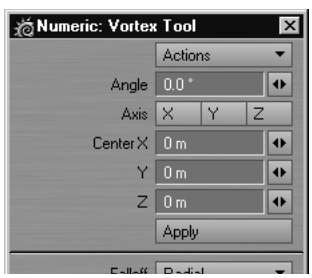
To use the Vortex tool:

Select the tool and drag your LMB in a viewport. Dragging right will rotate the influence area clockwise, while dragging left will rotate counterclockwise. The center of the effect is determined by the **Action Center** setting, discussed previously.

The information display will tell you the amount of **Angle** of rotation.

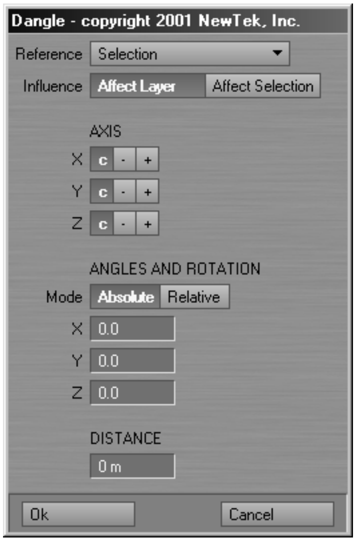


The **Angle** value on the numeric panel reflects the degrees of rotation. The **Axis** buttons relate to the rotational axis. If you use an arbitrary axis (i.e., rotated in a perspective viewport), no **Axis** buttons are selected. The **Center X**, **Y**, and **Z** values indicate the center point of rotation, through which the axis runs. You can edit the values and then click the **Apply** button to apply them, which you can do multiple times.



Dangle

Dangle will rotate a layer or selection relative to a rotation center. The **Reference** is what determines the axes used for rotation. **Selection** is a bounding box around the selected geometry, **World** is the Origin, and **B.G. Layer** is a bounding box around the background layer geometry.



Dangle

If **Influence** is set to **Affect Layer**, the rotation is applied to all of the geometry in the layer. With **Affect Selection**, only the selected geometry is rotated. Note that the rotation center can still be based on the **Selection**, even if you are using **Affect Layer**.

When the **c** setting is used for **Axis**, the rotation center is at the center of the **Reference**. Use the minus (**-**) or plus (**+**) options to move the rotation center to the negative or positive side of the bounding area, along the axis—these obviously have no effect if **World** is used.

When the bounding area is computed, an angle for the top “polygon” is measured based on its normal. If you use **Absolute** mode, it will change the rotation angle to be the exact amount entered. **Relative** will add the rotation to the geometry.

Enter the desired rotational values in the XYZ fields. These relate to the axis around which the rotation is performed.

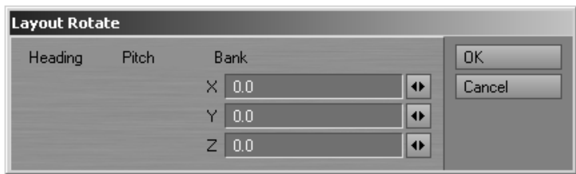
To use the **Distance** function, select two points prior to running Dangle. The distance between those two points will be shown in the **Distance** field. Enter a new value and set **Influence** to **Affect Selection**. Click **Ok**. The two points will now be apart the entered distance. Essentially, the points are scaled, using the rotation center defined by Dangle, until the points are at the defined distance. If you use **Affect Layer**, the entire layer will be scaled up proportionately. (If only one point is selected, the Origin is used as the other point.)

RotateAnyAxis

The RotateAnyAxis command (**Modify** > Rotate: **RotateAnyAxis**) lets you rotate the contents of a layer around an arbitrary axis defined by a two-point polygon in the background layer. The rotation is centered on the axis point nearest to the Origin.

RotateHPB

The RotateHPB command (**Modify** > Rotate: **RotateHPB**) lets you rotate the contents of a layer using heading, pitch, and bank values, like in Layout.

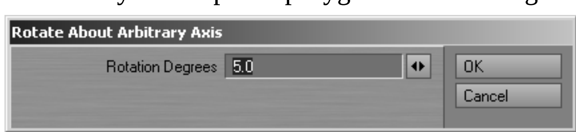


Rotate-About-Normal

The Rotate-About-Normal command (**Modify** > Rotate: **Rotate-About-Normal**) will rotate selected polygons using the surface normal of the first selected polygon as the perpendicular axis of rotation.

Rotate-Arbitrary-Axis

The Rotate-Arbitrary-Axis command (**Modify** > Rotate: **Rotate-Arbitrary-Axis**) will rotate selected polygons around an arbitrary axis defined by a two-point polygon in the background layer.



Rotate-To-Ground

The Rotate-To-Ground command (**Modify** > Rotate: **Rotate-To-Ground**) will rotate and move selected polygons to a defined ground plane based on the first polygon selected. The **Rest Axis** is the axis perpendicular to the plane you want the object to rest on. The **Sense** option rests the object on the positive (+) side of the plane or the negative (-) side.

Rotate-To-Object

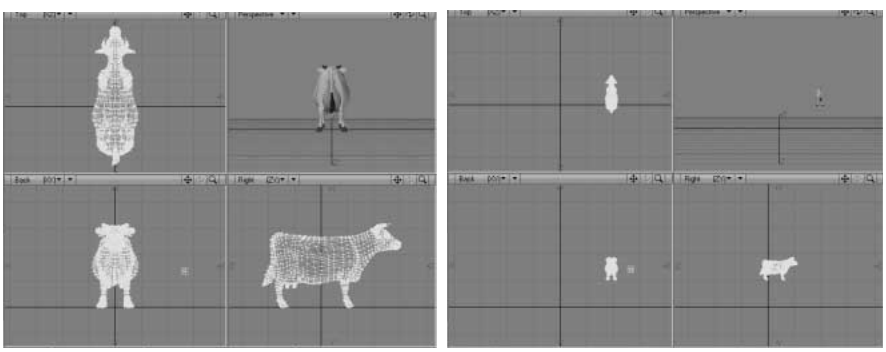
The Rotate-To-Object command (**Modify** > Rotate: **Rotate-To-Object**) will rotate relative to a polygon. The first selected polygon defines the target orientation. The second and any other selected polygons will rotate and align to the first selected polygon.

THE STRETCH GROUP

The Stretch group of tools is used to scale geometry. Many of the tools are a brother-sister team, with one that scales the geometry equally on all axes and the other that lets you scale the object independently along each axis. These tools are similar to Layout's tools with the same names. However, you use the different viewports to change the XYZ values.

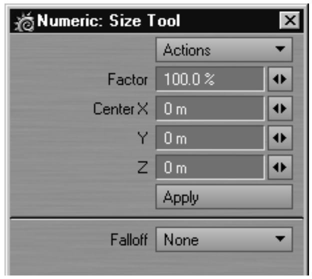
The Size Tool

Select **Modify** > Stretch: **Size** and drag your LMB right to increase the size and left to decrease it. The center of the effect is determined by the **Action Center** setting, discussed previously. The information display will tell you the scaling factor you have applied.



The Size Numeric Panel

The Size numeric panel has only two basic settings: the **Center XYZ** values, defining the coordinates for the center of sizing; and the size **Factor**, which defines amount of sizing, as a percent.

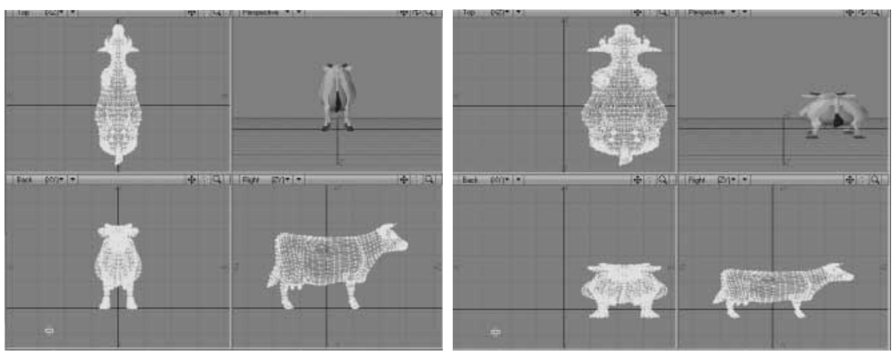


The Stretch Tool

The Stretch tool (**Modify** > Stretch: **Stretch**) is similar to the Size tool, except you can scale the object independently along each axis.

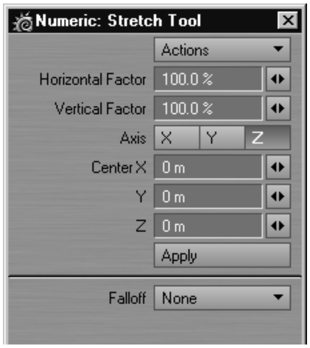
To stretch an object:

Select the Stretch tool (**Modify** > Stretch: **Stretch**) and drag your mouse right or left to increase or decrease the size along the horizontal axis. Moving up or down will increase or decrease the size along the vertical axis. The center of the effect is determined by the **Action Center** setting, discussed previously. The information display will tell you the scaling factor you have applied. Hold the CTRL key while dragging to constrain movement along the initial dragging axis.



The Stretch Numeric Panel

The **Axis** setting defines the axis of scale. When you click in a viewport, the axis is an imaginary line running perpendicular to the view you are editing in. The **Horizontal Factor** and the **Vertical Factor** represent the amount of scaling on those axes for the viewport. The **Center XYZ** values define the coordinates for the center of scaling.



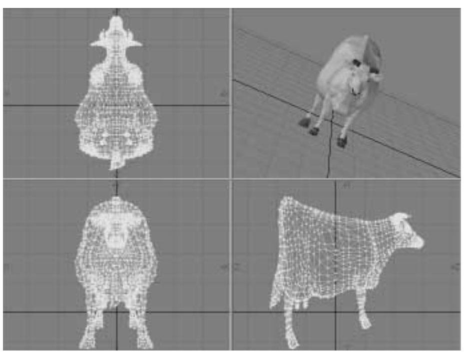
Tapering Objects

The Taper Evenly tool (**Modify** > Stretch: **Taper 1**) will resize an object at one end. You could use this tool to turn a box into a pyramid. Think of Taper Evenly as Size with axial falloff. That is, the strength of the sizing influence is not equal across the object, but gradually falls off along a selected axis.

To use the Taper Evenly tool:

Select **Modify** > Stretch: **Taper 1** and drag your LMB left/right to apply the scaling. The center of the effect is determined by the **Action Center** setting, discussed previously.

The information display tells you the **Scale** factor you applied.



The **Factor** value on the numeric panel reflects the amount of scaling. The **Center X**, **Y**, and **Z** values indicate the center point of scaling.

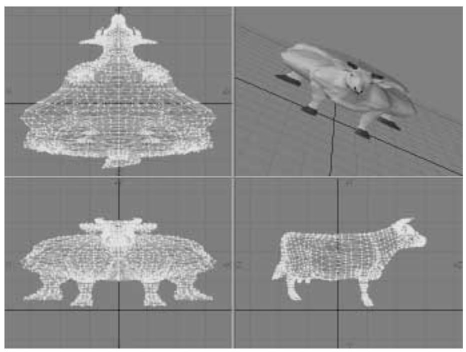
Taper Take 2

The Taper tool (**Modify** > Stretch: **Taper 2**) is very similar to the Taper Evenly tool. The difference is you can scale the affected end independently on the two axes. Basically, Taper is to Taper Evenly, what Scale is to Size. Think of Taper as Stretch with axial falloff. That is, the strength of the moving influence is not equal across the object, but gradually falls off along a selected axis.

To use the Taper tool:

Select **Modify** > Stretch: **Taper 2** and drag your LMB left/right to apply the scaling horizontally. Drag your LMB down to apply it vertically. The center of the effect is determined by the **Action Center** setting, discussed previously.

The information display tells you the **Scale** factor you applied along the two affected axes. Hold the CTRL key down while dragging to constrain the tapering to your initial dragging axis.



The **Horizontal Factor** and **Vertical Factor** values on the numeric panel reflect the amount of scaling in those directions relative to the viewport used. The **Axis** buttons relate to the axis along which the effect occurs. If you use an arbitrary axis (i.e., taper in a perspective viewport), no **Axis** buttons are selected. The **Center X**, **Y**, and **Z** values indicate the center point of tapering, through which the axis runs.

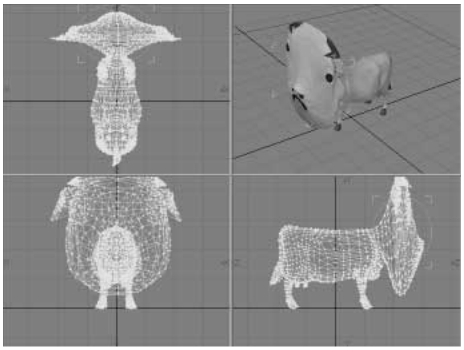
The Pole Evenly Tool

The Pole Evenly tool (**Modify** > Stretch: **Pole 1**) scales an area of an object along the vertical and horizontal axes.

To use the Pole Evenly tool:

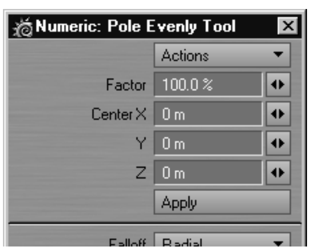
Select **Modify** > Stretch: **Pole 1** and drag your LMB in a viewport. The center of the effect is determined by the **Action Center** setting, discussed previously.

The information display tells you the **Scale** factor you applied.



Nutting wrong with this cow, eh?

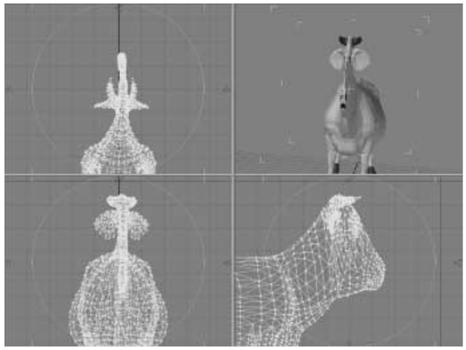
The **Factor** value on the numeric panel reflects the amount of scaling. The **Center X**, **Y**, and **Z** values indicate the center point of scaling. You can edit the values and then click the **Apply** button to apply them, which you can do multiple times.



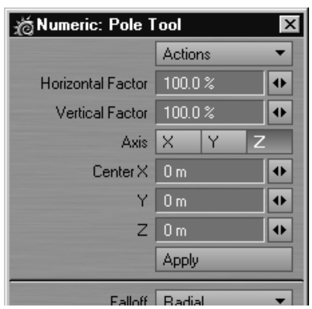
The Pole Tool

The Pole tool (**Modify** > Stretch: **Pole 2**) is identical to Pole Evenly, except you can scale an area independently on the two axes—dragging your LMB left/right to affect horizontally and up/down to affect vertically.

The information display tells you the **Scale** factor you applied along the two relevant axes.



The **Horizontal Factor** and **Vertical Factor** values on the numeric panel reflect the amount of scaling in those directions relative to the viewport used. The **Axis** buttons relate to the axis along which the effect occurs. If you use an arbitrary axis (i.e., use in a perspective viewport), no **Axis** buttons are selected. The **Center X**, **Y**, and **Z** values indicate the center point of scaling, through which the axis runs. You can edit the values and then click the **Apply** button to apply them, which you can do multiple times.



Absolute Size

Absolute Size allows you to scale geometry with great flexibility and precision. When the **Values** setting is **Independent**, the object can be scaled independently on each axis. If **Locked** is used, the aspect of the geometry is maintained—only change one axis value in this mode.

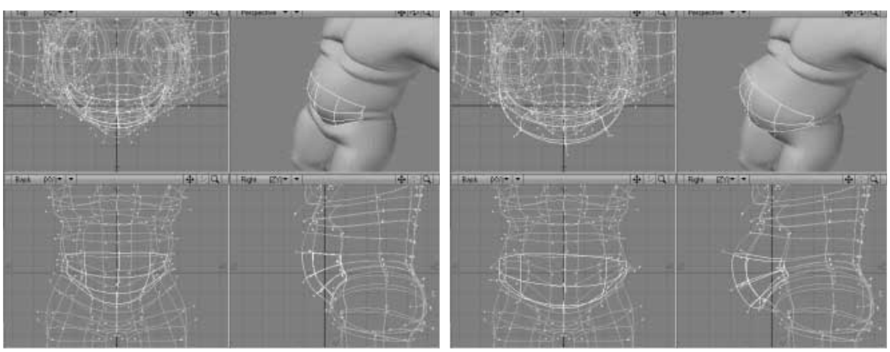
Absolute Size

The **Reference** setting determines a reference for the selected Action Center. **Bounding Box** uses a bounding box around all of the geometry. **World** uses the Origin. **B.G. Layer** uses a bounding box around geometry in the background. **Absolute** lets you numerically set the center position.

The **c** setting will scale using the center of the reference. The minus (**-**) or plus (**+**) setting will scale using the negative or positive side of the reference.

Smooth Scaling

A variant of the Stretch tool is the Smooth Scale command (**Modify** > Stretch: **Sm Scale**). This command will scale selected polygons, but attempt to smooth things out as it scales. No new geometry is added, so the smoothing is done using existing points and polygons. The **Offset** value, editable on the numeric panel, is the exact amount that the object will increase in size overall; however, due to the smoothing effect, not all parts will increase this much. Negative numbers are also allowed, which reverse the effect.



HINT

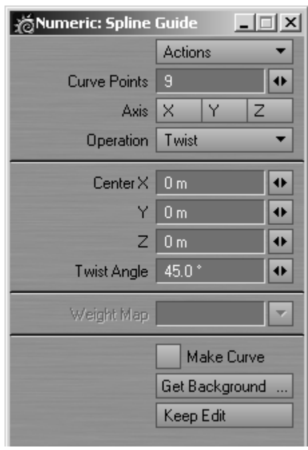
Smooth Scale is great for use on SubPatch objects. Use it on the stomach of a figure to give it a beer belly.

The Spline Guide Tool

Generally, the Spline Guide tool (**Modify** > Stretch: **Spline Guide**) performs interactive deforming operations based on a *guide* curve. Clicking in the window creates the initial guide curve and center axis. Initially, the guide curve will be aligned along the longest distance of the selected geometry, extending from its minimum to maximum point. The center axis is centered within the geometry.

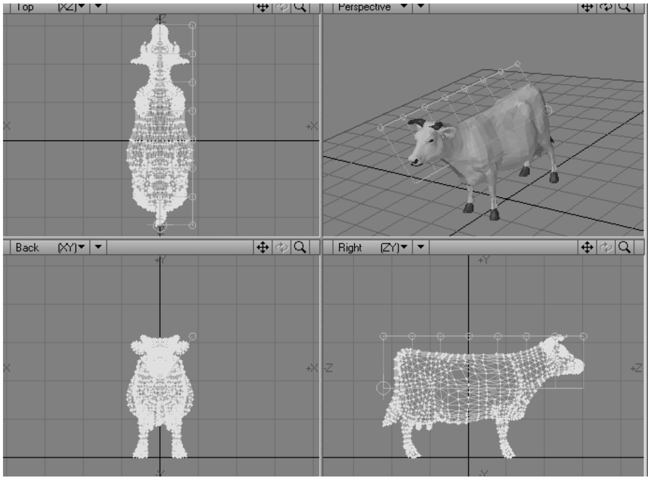
Initially, the guide curve points have no effect on the geometry—like bones in their rest position. However, moving the points will vary the strength of the effect. Moving the center axis will change the influence of all the points on the curve.

What effect is applied is determined by the **Operation** setting on the numeric panel.

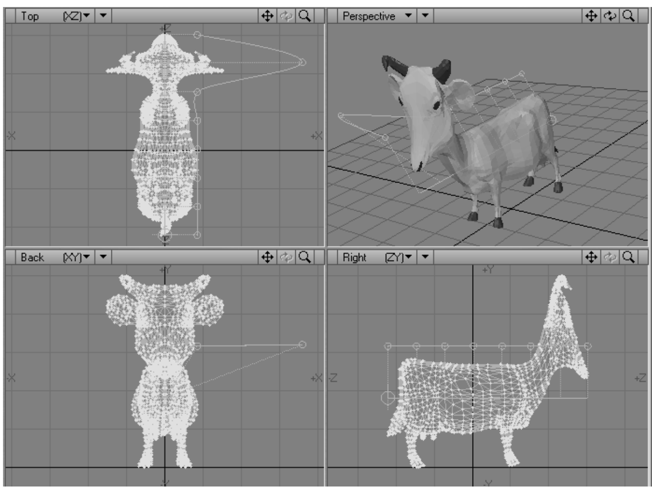


Spline Guide numeric panel

The **Scale** operation scales the geometry in the plane perpendicular to the axis around the center axis. Moving the guide points closer to the center shrinks the geometry, moving them farther away expands it.



Initial guide curve



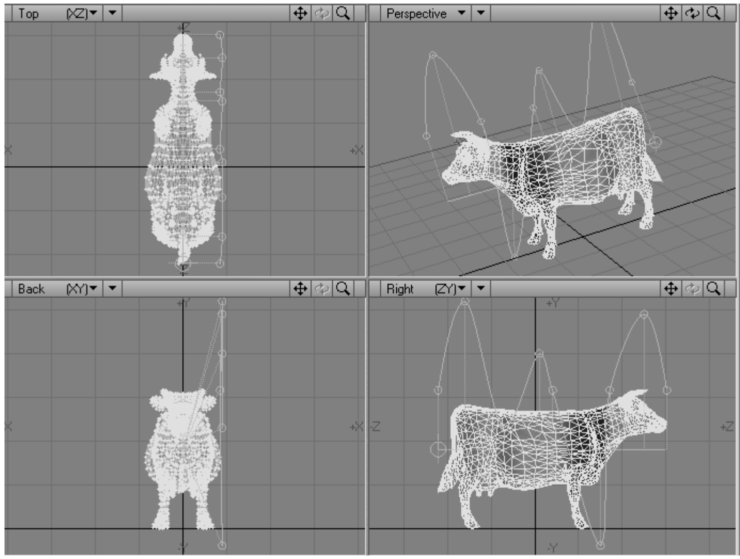
After scaling one control point

The **Stretch** operation scales the model in the plane perpendicular to the axis around the center like the **Scale** operation, but uses different amounts of scaling in the plane (call it X-Y) depending on the X and Y distance of the guide points from their initial positions. Thus, the geometry can be expanded in one direction and reduced or left unchanged in the other.

The **Twist** operation rotates the geometry around the center axis. The **Twist Angle** parameter acts as an angle scaling factor.

The **Bend** operation bends the model to conform to the curve.

The **Weight Map** operation works a little differently than the other operations in that it doesn't deform the geometry, but affects the selected weight map instead. Essentially, it sets the values based on the reshaped curve. Move them closer to the center axis for negative weights, farther for positive.



Using Weight Map

If you activate **Make Curve**, a curve polygon will be created from the guide curve when you deselect the tool. Use this to save the guide curve shape. This curve can be placed in a background layer and *loaded* into the guide curve by clicking **Get Background Curve**.

Click **Keep Edit** to accept the current changes without resetting the curve. This is useful if you want a certain curve to be applied in subsequent operations.

Changing the **Curve Points** value will *resample* the curve into more segments. Note that resampling the curve can change its shape, particularly if the number is incremented gradually, like with the mini-slider. If you want a more refined curve that accurately matches the one you have drawn, enter the new number directly into the field to avoid the intermediate curves.

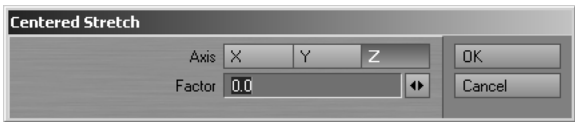
OTHER STRETCHING COMMANDS

CenterScale

The CenterScale command (**Construct** > Utility: **Additional** > **CenterScale**) will scale objects in the current layers about the object's implied center by a given **Factor**. The factor can be specified as a User Command argument, in which case, no requester will appear.

CenterStretch

The CenterStretch command (**Construct** > Utility: **Additional** > **CenterStretch**) will stretch objects in the current layers along one of the axes about the object's implied center by a given **Factor**. The factor and axis can be specified as a User Command argument (e.g., Y 0.3333), in which case no requester will appear.



THE DEFORM GROUP

The Jitter Command

The Smooth command's ugly sister is Jitter (**Modify** > Deform: **Jitter**). This command will add a roughness to the polygons of an object by randomly moving points within a certain radius of their current position. Jitter has several **Type** settings, each applying the effect in a different manner.

Uniform Uniform performs a uniform jitter on every point within a rectangular region. A point may be shifted by as much as the value specified for the **Radius** parameter.

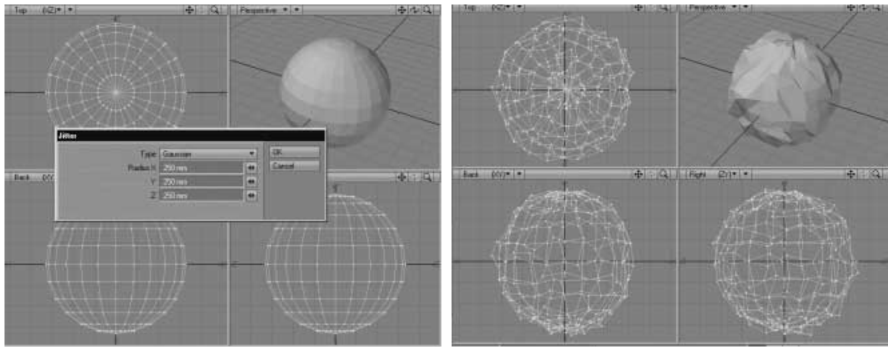
Gaussian Gaussian results in a slightly less ragged jitter than the **Uniform** option, performing a normal distribution around the starting point's location in an ellipsoid of the given **Radius**. A point may be shifted by as much as the value specified for the **Radius** parameter.

Normal Normal uses the local surface normal and randomly moves points plus or minus within the specified **Range** along that surface normal.

Radial Radial randomly distributes points inward or outward along a radial vector from a single point defined by the **Center** coordinates. **Range** is the value used to set the range within which a point may move from its current location.

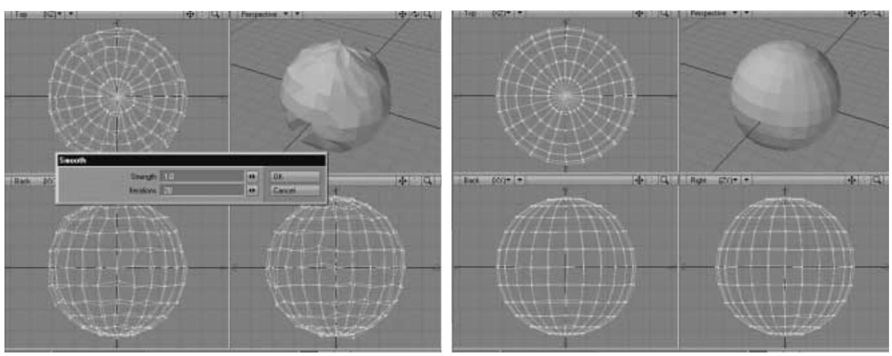
Scaling Scaling is a variation of the Radial option. Points are jittered outward from a center point defined by the **Center** coordinates. **Factor** is the value used to set the range, as a percentage, within which a point may move from its current location.

A jitter value of 0 applies no jittering, meaning no change will occur. Values above or below 0 will affect the object.



The Smooth Command

The Smooth command (**Modify** > Deform: **Smooth**) causes Modeler to try to smooth any jagged surfaces. There are no *correct* values for these settings; however, even relatively low **Strength** values can overly distort an object. **Iterations** specifies how many times the smoothing algorithm should be reapplied.

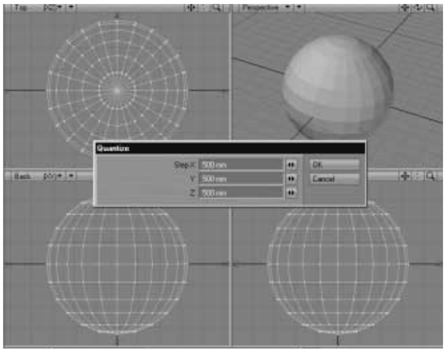


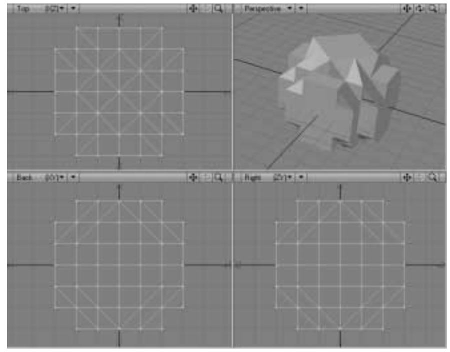
NOTE
Smooth is most effective when **Iterations** is set to a high value.

HINT
For more drastic smoothing, you might use Metaform (**Construct** > Subdivide: **Subdivide**) instead; however, this tool increases the number of points and polygons.

The Quantize Command

The Quantize command (**Modify** > Deform: **Quantize**) will snap points to specific coordinate spacing intervals (like a grid). This is very useful for snapping a non-planar polygon into a specific plane or flattening one side of an organic or curved object.





chapter 24

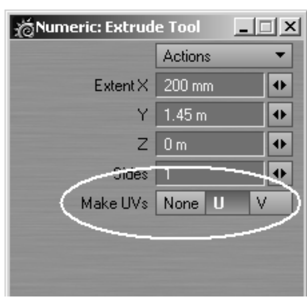
Multiplying Geometry

Chapter 24: Multiplying Geometry

Modeler features some great tools for multiplying your geometry and automatically creating additional points and polygons. There are tools to add a bevel, extrude existing geometry, clone polygons, model using background geometry, create mirror images, and more.

MAKE UVS OPTION

The tools/commands that perform an extruding operation (Extrude, Lathe, Rail Extrude, etc.) have *Make UVs* options. You can have the U or V value assigned to a range from 0 to 1. Note that a UV Texture must be selected or this option will be ghosted. (See Chapter 28 for more information.)

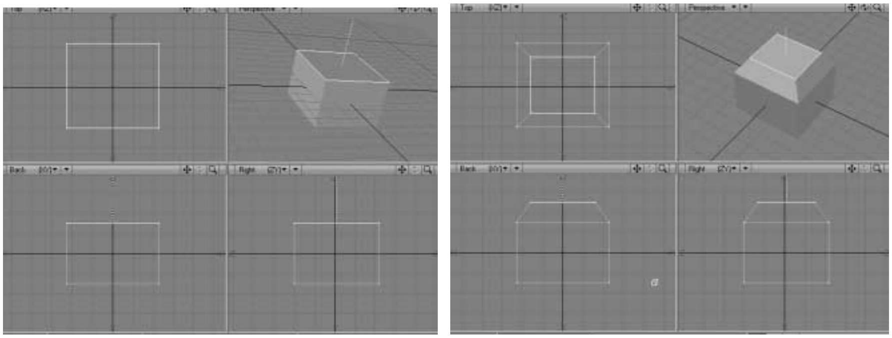


THE EXTEND GROUP

This group of tools multiplies geometry by extending it in some fashion, adding additional geometry along the way.

The Bevel Tool

The Bevel tool (**Multiply** > Extend: **Bevel**) is likely one of the most commonly used modeling functions. Basically, it takes a polygon and extrudes it away from the original along the surface normal. It is similar to using the Extrude tool, but there will be no polygon at the position of the original polygon. When applied to multiple polygons, each polygon receives its own bevel—even on a double-sided polygon.



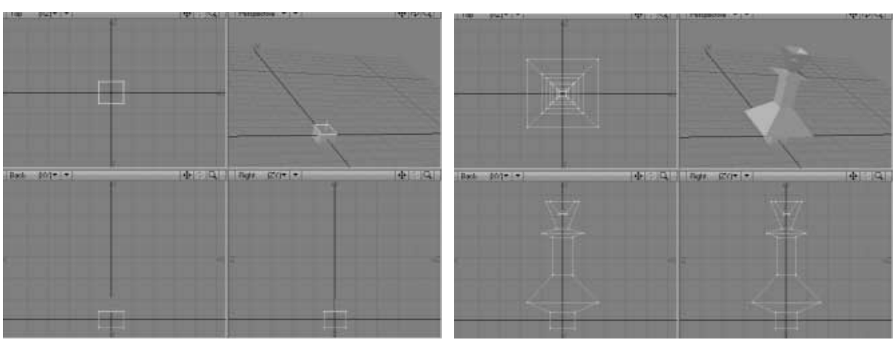
Dragging your pointer after activating the tool lets you graphically place the bevel. Deactivating Bevel or selecting another tool makes the bevel. Click on an open area of the interface to cancel the bevel before it is made or use Undo.

Dragging up/down lets you interactively change the **Shift** amount, the distance that the selected polygon will move along its surface normal. On the numeric panel, positive values make the bevel move in the same direction as the normal. Negative values make it move the opposite direction.

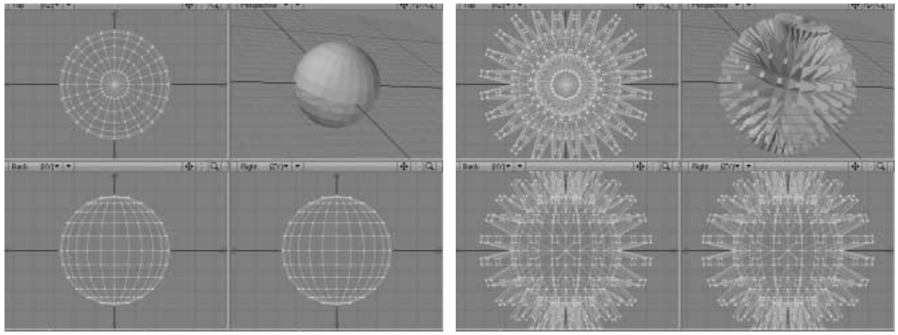
Dragging left/right lets you interactively change the **Inset** amount, the distance that the edges of the polygon will move in the same plane as the polygon. On the numeric panel, positive values move inward and negative values move outward.

Drag at an angle to simultaneously change **Shift** and **Inset**. Hold the CTRL key to constrain to **Shift** or **Inset**, depending on which direction you drag initially.

If you select polygon(s), the top polygon will remain selected so you can perform bevel operations one after another. Click your RMB to accept the current bevel and begin a new bevel operation. You can also press the B key twice to make and start a new bevel.

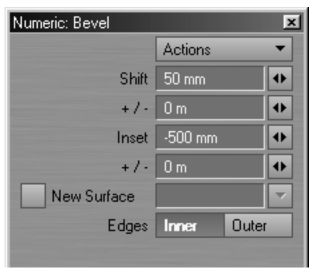


You can bevel multiple polygons simultaneously too!



Other Numeric Options

The **+/-** fields on the numeric panel let you set randomization for the **Inset** and **Shift** values. The actual value applied will be varied by a random number between plus or minus the **+/-** value. For example, if **Inset** was 20m and its **+/-** was 5m, then the actual **Inset** would be between 15 and 25 meters. Generally, you want to use this feature only when beveling many polygons simultaneously.



HINT
To avoid negative random values, make sure the **+/-** value is always less than or equal to the related **Inset** or **Shift** value.

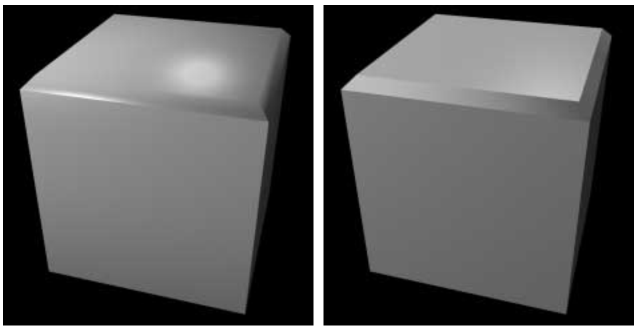
Inner is the default **Edges** setting and bevels the polygon inward and forward along the surface normal. **Outer** will bevel outward and backward; however, the original polygon does not move and the new beveled edges extend away from the surface normal. This option reverses the mouse movement effects. In most cases, you use **Inner**.

By default, the surface name of the source polygon is used for the new geometry. If you select **New Surface**, you can enter a surface name (even an existing one) in the input field. You can change this setting anytime before *accepting* the bevel.

NOTE
Note that the selected polygon retains its surface name; only the new edge polygons receive the defined name.

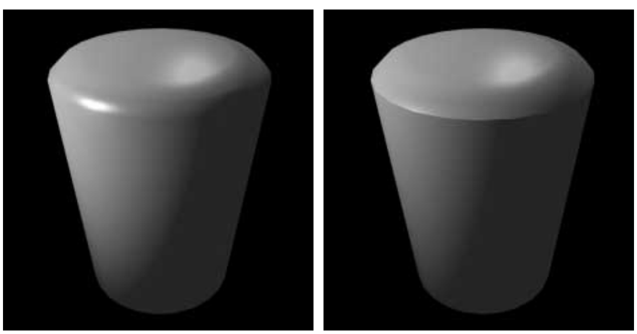
Beveling Tips

- After performing a bevel, the selected polygon remains selected. To accentuate the beveled edge when the surface uses the Smoothing surface option, perform a quick cut and paste to separate the selected (top) polygon from the added sides. This action precludes LightWave from smoothing over the beveled edge. You may also want to cut and paste the original polygon (before performing the bevel). Make sure you do not merge points after doing this!



Left: Polygons not separated. Right: Starting and top polygons separated

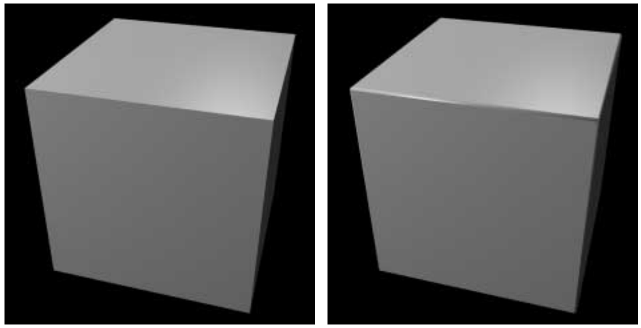
- If you use the same surface, which uses Smoothing, and the geometry goes from straight to curved, you may want to separate the polygons (with a cut-and-paste) at the transition point to get a clean edge.



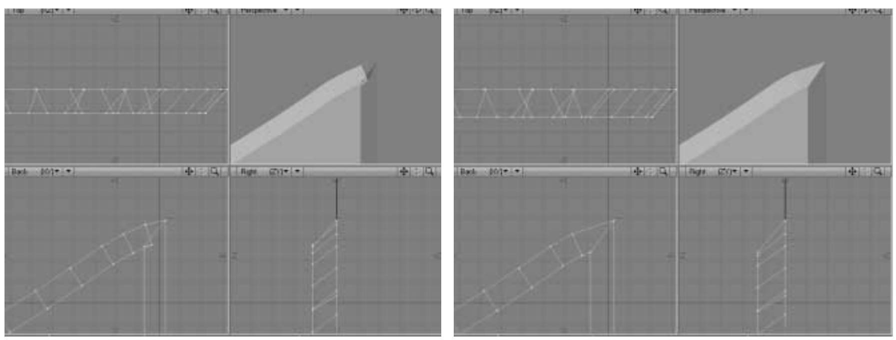
Left: Polygons not separated. Right: Starting (top) polygon separated

- If you want to create an enclosed solid object when beveling a 2D polygon, copy it to another layer first and choose **Detail** > Polygons: **Flip**. Go back to the original layer and perform the bevel normally. Then, paste the flipped polygon back into the original layer and choose **Construct** > Reduce: **Merge Polygons** to merge points.

- Cut and paste a polygon (like one side of a box) away from its host and bevel it a very small amount. Then, cut and paste the top polygon as well. This makes a nice clean edge that catches specular glints.

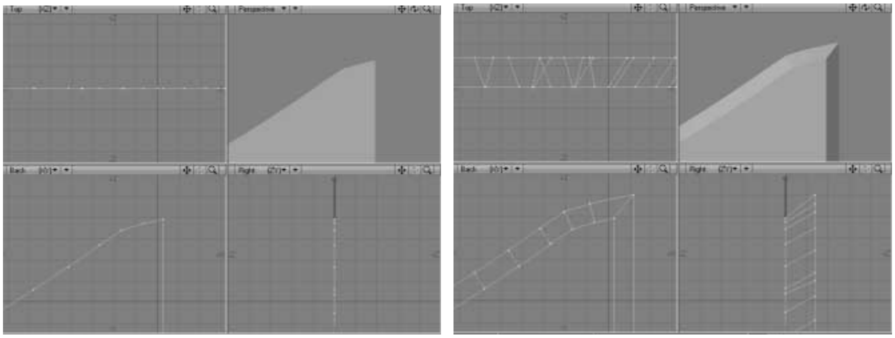


- When you bevel polygons with sharp edges, you can often run into the problem of points being too close at the corners—very common when you bevel text characters. One solution is to delete unnecessary points near the corners prior to beveling. Don't be too scared about deleting too many points; the change is usually so slight that no one will notice, particularly if the text is animated.



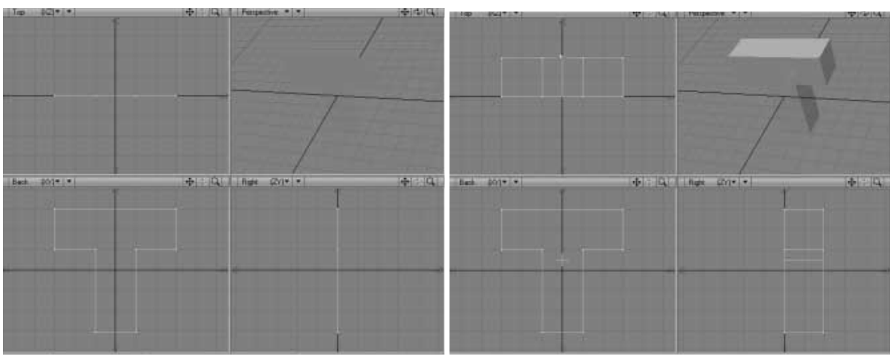
NOTE
Sometimes you also need to manually drag points on the new polygon away from the corner.

- Another corner-fixing trick is to bevel out instead of in. This has the benefit of not requiring you to mess with any points, but it affects the spacing between characters, if you are working with text. For this, use the **Outer** setting for **Edges** in the Bevel numeric panel.



The Extrude Tool

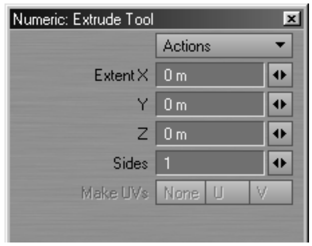
Use the Extrude tool (**Multiply** > Extend: **Extrude**) to give depth to a two-dimensional shape. You can extrude any two-dimensional polygon or curve. The distance and angle from the initial dragging point sets the extrusion length and direction. As you continue to drag your mouse, you see the extrusion length and angle change accordingly.



The initial dragging point is not significant, only the distance and direction matter. In fact, when you release the mouse button, the axis will reposition itself starting at the object. You can further manipulate the axis by dragging its end in any viewport.

When you are satisfied, simply click **Multiply** > Extend: **Extrude** again (or select another tool). To reset, click a non-active part of the interface or click **Undo**.

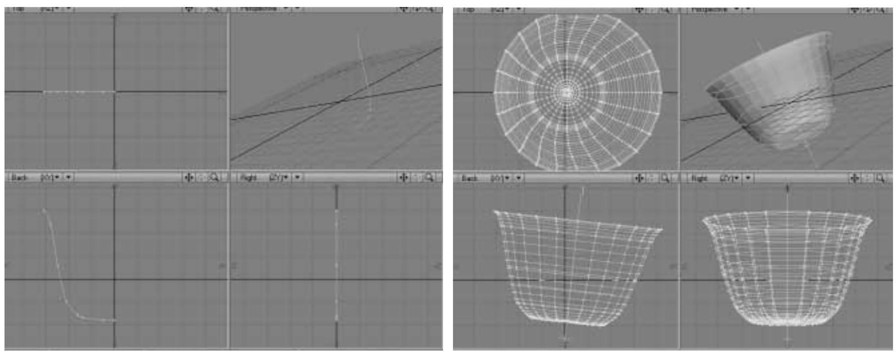
The numeric panel for Extrude defines the relative XYZ coordinates for the end of the axis in the **Extent** fields. You can change the number of **Sides** created on the panel, which can be useful for multi-segmented items like snakes, or objects you wish to bend or twist.



Extrude Tool numeric panel

The Lathe Tool

The Lathe tool (**Multiply >** Extend: **Lathe**) will *sweep* an object around an axis. Essentially, Lathe creates a cylindrical object with a profile of the original form. For example, if you lathe a circular-shaped polygon, you create a donut. You can sweep any two-dimensional object or curve. The classic use of this tool is to create a vase or cup.

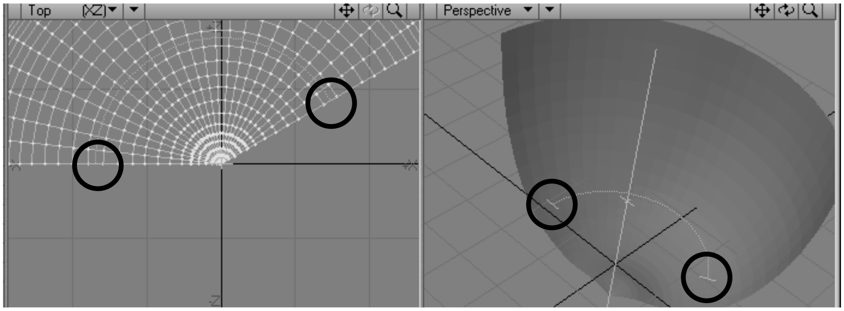


Dragging your mouse in a viewport determines the initial axis. The lathe axis has a crosshair at its **Center** (the initial dragging point), which is used to drag the entire axis around. To change the angle, drag either end of the axis. You can also drag out the axis, which gives you more control over the effect, but the length of the axis does not in itself influence the impact of the tool.

When you are satisfied, deselect the tool or select a different one to execute the edit. To cancel, simply click in an open area on the interface or click **Undo**.

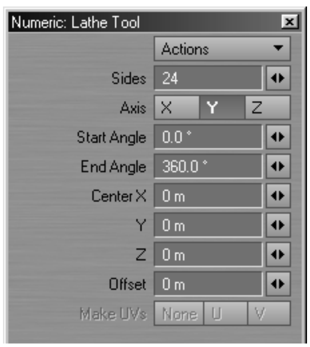
NOTE
With this tool, polygons may end up flipped the wrong way. To correct them, simply hit the F key.

If you wish to change the number of sides in the lathed object (like the number of slices in a pie), you can use the LEFT and RIGHT ARROWKEYS. If you'd prefer to lathe less than a full 360 degrees, you can drag the handles as indicated in the following illustration. (Note: If the two handles are overlapped, you can still drag to separate them.)



Lathe interactive handles

You can also open the Lathe tool's numeric requester to numerically change settings. The **Start Angle** and **End Angle** fields let you enter the degree of lathing, **Sides** sets the number of sides, and **Offset** will shift the shape along the lathe axis. The **Center** values define the center of rotation coordinates.



Lathe Tool numeric panel

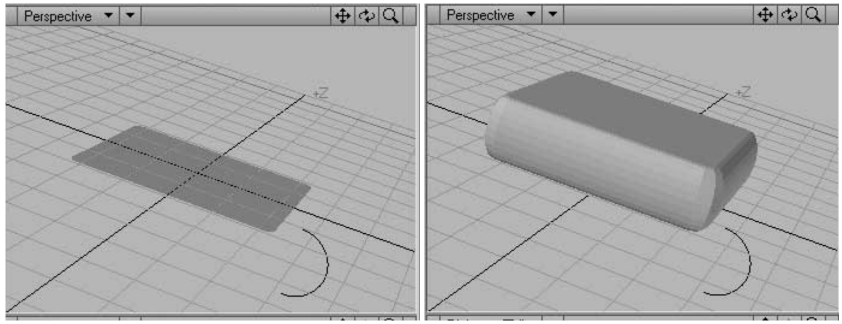
HINT

To create a coiled spring: create a disc with, say, a one meter diameter. Select Lathe and place the axis about one meter to one side of the disc. Then, open the numeric requester and set the **End Angle** to 720 degrees (two full rotations), **Sides** to 48, and **Offset** to 2m.

The Rail Bevel Tool

Rail Bevel (**Multiply** > Extend: **Rail Bevel**) is a very fancy beveling tool that works in a manner similar to the normal Bevel tool. Dragging your mouse left/right adjusts the new bevel geometry's distance from the original polygon. Dragging up/down adjusts the *inset* direction (towards or away from the polygon's center). The CTRL key constrains the change to a given degree of freedom—just like the normal Bevel tool.

The difference between Rail Bevel and Bevel is that Rail Bevel uses a *profile* from a background layer to generate several bevel operations in a single step.



Example Rail Bevel

The profile is a series of points you create. They can be unconnected, or part of a polygon or curve. The only important factor is point order, which is normally determined by their order of creation.

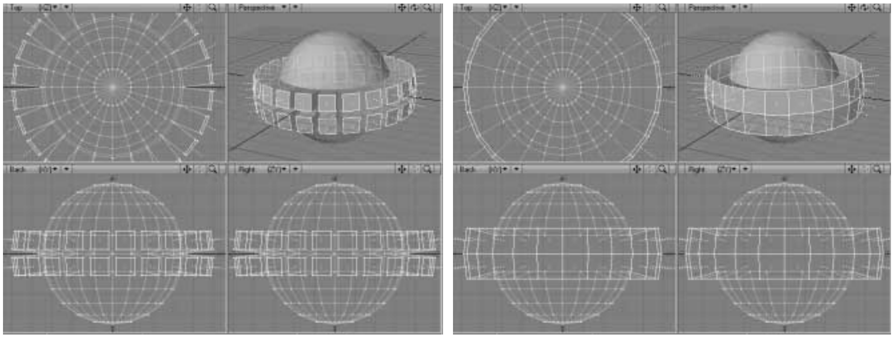
The profile is treated as if it were oriented based on the polygon's plane and extrudes along the direction of the polygon's normal. Each point in the profile is treated as a *bevel*, where the X and Y axes represent the relative inset and offset positioning, respectively.

New geometry inherits VMap and face/curve properties of the original polygon.

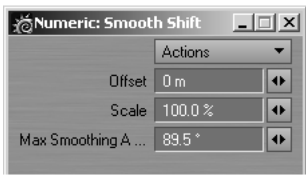
HINT
If you have multiple profiles in different layers, you can apply them one after another by selecting them (one at a time) as the background layer while using Rail Bevel.

The Smooth Shift Tool

The Smooth Shift tool (**Multiply** > Extend: **Smooth Shift**) is a beguiling tool that may not seem useful at first; however, you will find a variety of uses for it, like creating thin grooves in an object. Smooth Shift moves selected polygons along their *smoothed vertex* normals. The smoothed vertex normal is computed using the average of the normals for the polygons that share that vertex. This tool is similar to Bevel—however, Bevel applies effects individually to each selected polygon, whereas Smooth Shift treats all selected polygons as a group.



Left: Bevel. *Right:* Smooth Shift



Smooth Shift numeric panel

Polygons on a convex surface tend to expand since the normals at their vertices point slightly apart, while polygons on a concave surface tend to shrink. Polygons that are smoothed together stay together since they share normals at their shared vertices. Where polygons pull apart, either because their smoothed vertex normals diverge and exceed the **Max Smoothing Angle** or because some are selected and some are not, the gaps are filled with bands of new polygons.

The **Offset** is the distance of shifting along the vertex normal. Positive values make the shift occur in the same direction as the normal. Negative values make it shift in the opposite direction.

The smoothing method in Smooth Shift requires a cutoff angle. The **Max Smoothing Angle** on the numeric panel determines whether polygons will be smoothed over. Any two polygons with adjoining edges at angles greater than this value will be smoothed across.

Dragging your mouse horizontally will change the **Offset** amount on the numeric panel.

The **Scale** setting, accessible from the numeric panel only, expands or shrinks the new geometry around the center of its bounding (box) area.

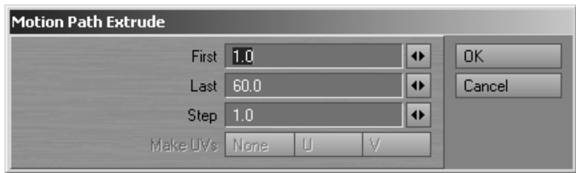
HINT

To achieve what amounts to a group-bevel,—where the selected polygons are treated as one instead of individually—use **Smooth Shift** with an **Offset** of 0. Then, manually move the polygons (which will still be selected).

The Path Extrude Command

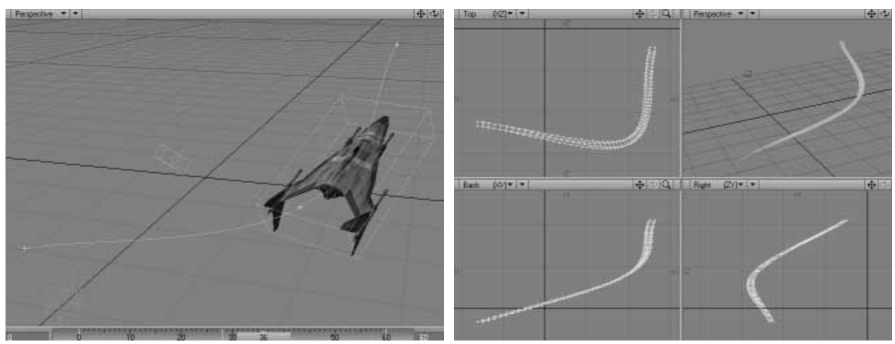
The Path Extrude command (**Multiply >** Extend: **Path Extrude**) will extrude a point, polygon, or object along a motion path. It is nearly identical to the Path Clone command, discussed later. The difference is that Path Extrude forms a continuous skin for the object instead of making individual copies of the object.

When you select the Path Extrude, a file requester asks for a motion file. Then, a dialog will appear.



Path Extrude dialog

Enter the **First** and **Last** frames of the motion path you wish to use (the values default to the entire length of the path). **Step** indicates how many segments to clone in terms of frames (a **Step** value of one will create one segment for each frame). So a motion path that takes 30 frames with a Step of 1 will have 30 segments. The same motion path with a **Step** of 2 will have 15 segments.



Left: Object motion path in Layout. *Right*: Flat box extruded along same path

NOTE
The motion path for an item in Layout can be saved by selecting it and then choosing **File** > **Save** > **Save Motion File**.

If you a UV Texture is currently selected, the **Make UVs** options are available. You can create the UV map perpendicular to the U or V axis.

The Rail Extrude Command

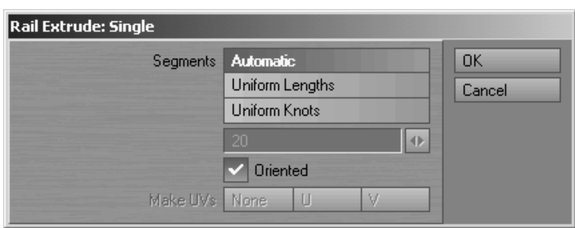
The Rail Extrude command (**Multiply >** Extend: **Rail Extrude**) will extrude a point, polygon, or object along a curved path (single rail clone) or multiple paths (multiple rail clone). It is nearly identical to the Rail Clone command, discussed later. The difference is that Rail Extrude forms a continuous skin for the object instead of making individual copies of the object.

The curve(s) must be in the background layer and the polygons to be extruded in the foreground layer. The orientation and position of the foreground polygons will affect the result. Generally, they should be at the very beginning of the curve and oriented perpendicular to it—as if the curve was a thread you were threading through the polygons.

The direction that the curve faces (which end is considered the beginning) will have bearing on both the shape and direction of the cloning. The direction affects the direction of the extruding. If you don't get the results you expect, try flipping the curve's starting point with **Detail** > Polygons: **Flip**. If you are using multiple curves, generally, they should run in the same direction.

If you a UV Texture is currently selected, the **Make UVs** options are available. You can create the UV map perpendicular to the U or V axis.

Single Curve



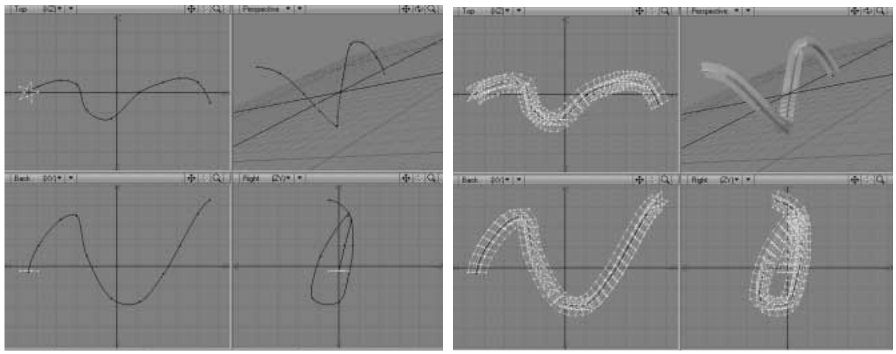
The Rail Extrude: Single dialog will appear when a single curve is in the background.

Segments — **Segments** is the number of segments in the resulting object. Select **Automatic** to distribute a number of segments along the rail curve (based on the **Curve Divisions** setting on the General Options panel) according to the curve's knot (point) spacing.

Select **Uniform Lengths** to distribute some number of segments (which you specify) evenly along the length of the entire curve. No matter what the knot spacing is, the segments will remain evenly spaced.

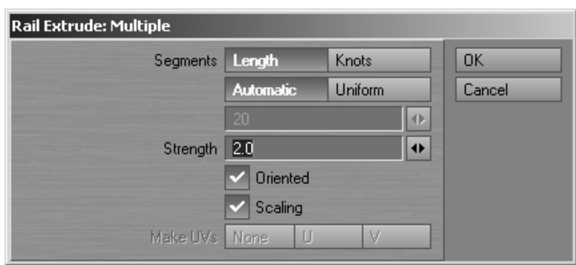
Select **Uniform Knots** to distribute some number of segments (which you specify) evenly between the knots that make up the curve. Whether the knots are close together or far apart, there will be the same number of segments between each.

Oriented **Oriented** determines the orientation of the polygon template as it follows the curved path. When active, it causes all segments to angle themselves automatically so that they are aligned with the rail curve as it turns. When inactive, it causes all segments to remain in the same orientation as the template so that they face exactly the same way.



A one-rail example

Multiple Curves



The Rail Extrude: Multiple dialog will appear when multiple curves are in the background. In practice, one curve will act as the main cloning curve and the other(s) act as shaping curves. The effect of the shaping curves is determined by the distance from the main curve. However, you may change the effect by moving the polygons closer or farther away from any of the curves.

Segments **Segments** is the number of segments in the resulting object. Select **Automatic** to distribute a number of segments along the rail curve, based on the **Curve Divisions** setting on the General Options panel. If you also select **Knot**, segments are distributed with regard to the curve's knot (point) spacing. If you use **Length**, knot spacing is disregarded.

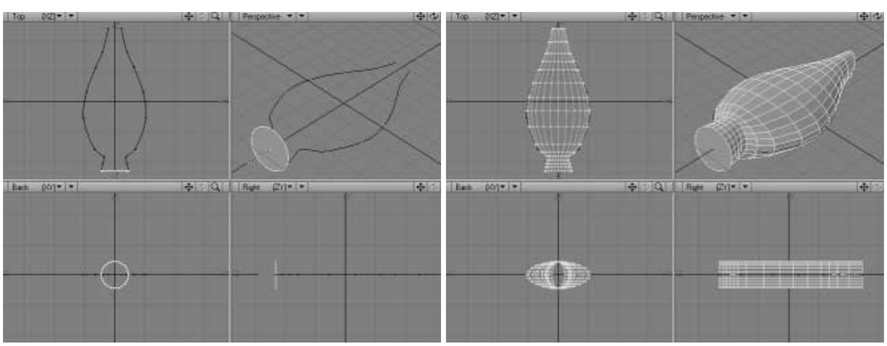
To specify the number of segments, choose **Uniform** and enter the number in the input field. If **Length** is also active, the segments will remain evenly spaced no matter what the knot spacing is. If you use **Knot** instead, there will be the same number of segments between each knot, no matter how close together or far apart they are.

Strength — **Strength** determines how strongly the rails vie for control of the point locations. You will not notice much of a difference unless you have **Scaling** (below) deselected. The higher the **Strength** value, the closer the object will *hug* the rails.

Oriented — **Oriented** determines the orientation of the polygon template as it follows the curved path. When active, it causes all segments to angle themselves automatically so that they are aligned with the rail curve as it turns. When inactive, it causes all segments to remain in the same orientation as the template so that they face exactly the same way.

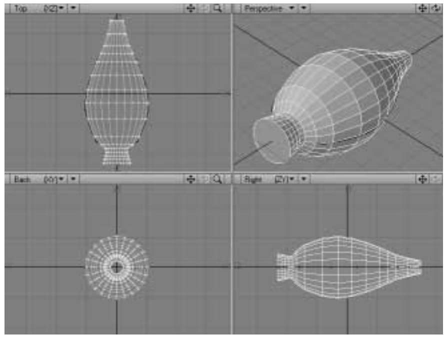
Scaling — **Scaling** determines how segments along an axis are sized. If the rails in the background layer spread apart along any axis or axes, then the segments will automatically stretch along those axes as well. Turn **Scaling** on to scale segments equally, rather than along those specific axes only.

For example, with **Scaling** off, if you Rail Extrude a polygon along the Z axis using two rails that spread further apart in the X axis, the segment's copies will be stretched on the X axis to maintain their relationship to the guide rails.



Left: Disc and two curves in background. Right: Result with Scaling off

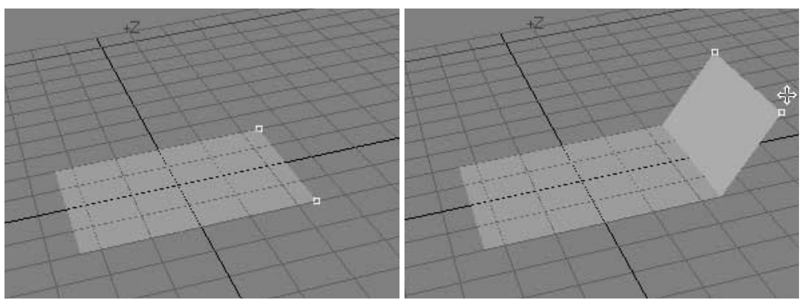
Turning **Scaling** on causes this expansion to occur equally on both X and Y axes.



Result with Scaling on

The Extender Command

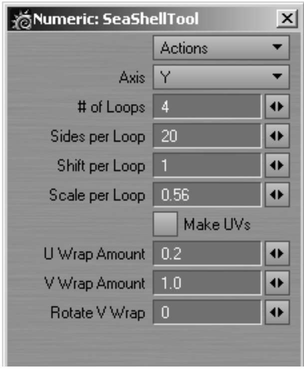
The Extender command (**Multiply** > Extend: **Extender**) clones selected points and creates new geometry connecting the original points and the clones. Essentially, Extender is like using Smooth Shift with 0 Offset, except with points instead of polygons.



Left: Extender executed on selected points. *Right*: Cloned points moved away from original positions

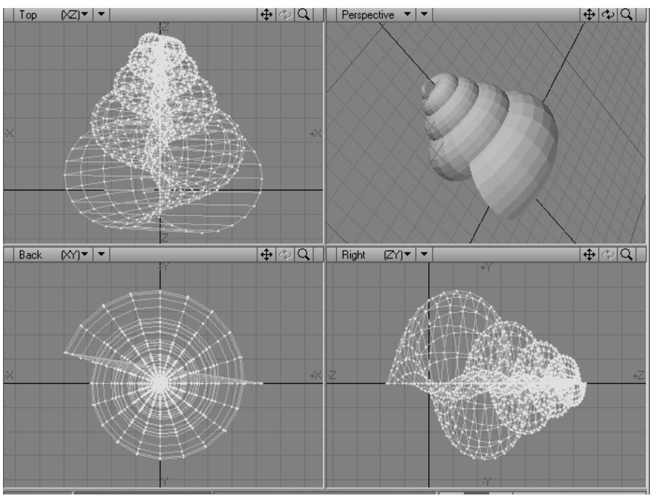
The SeaShellTool

The SeaShell tool (**Multiply** > Extend: **SeaShellTool**) lets you interactively make seashell shapes. Essentially, this is like using the Lathe with a scaling factor on an object.



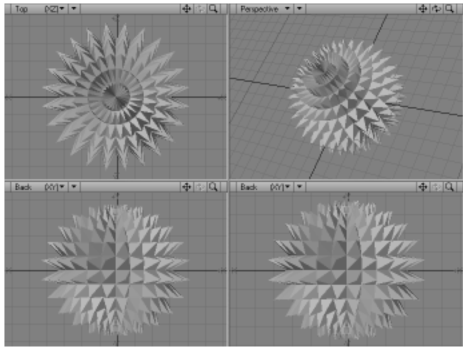
SeaShell numeric panel

Open the numeric panel to set options. **Axis** is the perpendicular axis around which the shell is twisted. **# of Loops** is the number of times the source polygon is twisted fully. **Sides per Loop** is the number of segments to use per loop. **Shift per Loop** controls the vertical shifting of the loop. **Scale per Loop** is the scaling factor achieved after each loop. SeaShell also has automatic UV creation options. Like any tool, you must deselect it to *make* your object.



The Spikey Tool

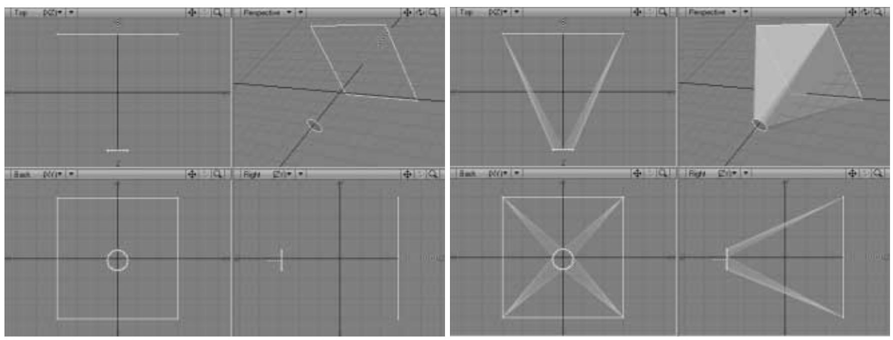
The Spikey tool (**Multiply** > Extend: **Spikey**) subdivides the selected polygons and moves the center point out in the direction of the surface normal giving a spiked appearance. Simply select the tool and drag your mousepointer in a viewport.



The Spikey tool applied to a basic ball

The Create Skin Command

With the Create Skin command (**Multiply** > Extend: **Skin**), you can cover a series of polygons or curves with a polygon *skin*. The shapes do not need to have the same number of points in common. This is also called *lofting* or *loafing* (as if covering the slices of a loaf of bread).



The polygons or curves must be selected (or created) in the order that you want them connected.

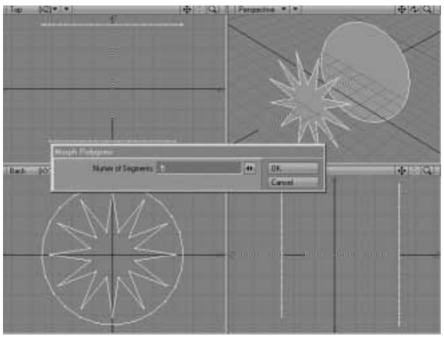
Create Skin works best when you use single-sided polygons for the framework. Double-sided polygons can result in strangely skinned objects.

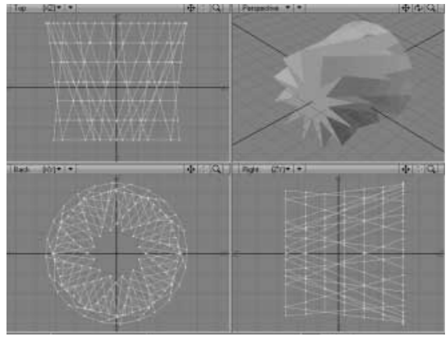
If you skin using polygon curves, the curves will remain in the Modeler workspace following the operation. However, they do not become part of the resulting skinned object.

Although Create Skin is similar to the Make Spline Patch command (**Multiply** > Combine: **Patches** > **Make Spline Patch**), which uses curves, the difference between the two commands is significant. The Make Spline Patch command tends to follow smooth contours, which maintains a rounded surface. Create Skin tends to connect curves with straight-line segments, which creates less smooth profiles.

The Morph Polygons Command

The Morph Polygons command (**Multiply** > Extend: **Morph**) will create a number of connected intermediate polygons between two *selected* polygons or curves that have the same number of points. They will have an outer surface, as if you also used the Skin command on them. Morph Polygons works best when you use it between single-sided polygons.





HINT
To get a twist effect, rotate the source or target polygon a little.

If you morph using curves, the curves are treated as straight-edged polygons, and will morph in a linear fashion. The curves also remain in the Modeler workspace following the operation, although they will not be part of the morphed object.

The **Number of Segments** field defines the number of morphed steps you desire between the two selected polygons.

NOTE
As with the morphing function in Layout, the Morph Polygons command is dependent on point order. As such, you should create the source and target from the same polygon. You will probably need to flip some polygons after using this command.

THE COMBINE GROUP

This group of tools multiplies geometry by using other geometry as carving or set-up tools.

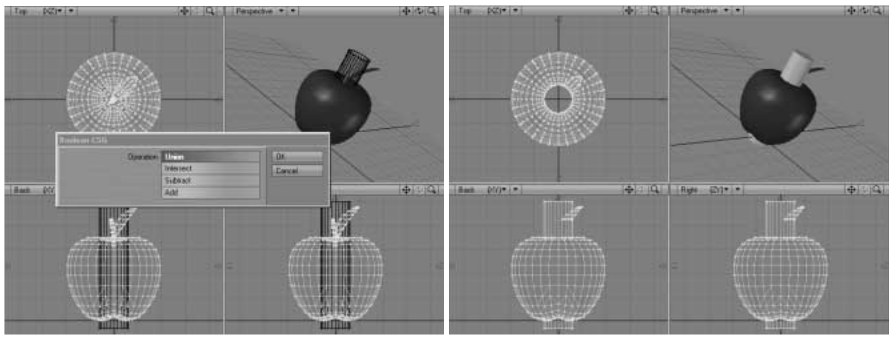
The Boolean Command

The Boolean command (**Multiply** > Combine: **Boolean**) is similar in some respects to the Solid Drill command, discussed later. However, Boolean operations treat objects as solids with volume, instead of hollow 3D objects. They will merge, split, carve, and join objects in a variety of ways. Like the drilling tools, you must have an object in the background layer. The major distinction of Boolean is that it treats the foreground object as a solid mass and thus creates inside surfaces along cut edges.

With Boolean, some portion of the foreground and background objects must physically overlap in 3D space.

Union

Union joins the background layer with the foreground layer, taking the solid objects and combining them into a single object. Interior faces are removed and no new polygons or surfaces are created; therefore, all surfaces will retain their original names.To work properly, both objects must be closed 3D solids.

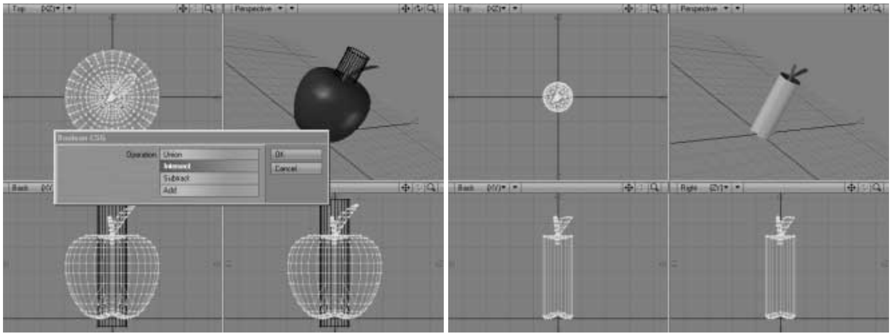


Union mode before (*left*) and after (*right*)

From the rendering perspective, you could get a similar result by just pasting both objects into one layer; however, Boolean results in the fewest polygons and smaller object files, which will use fewer system resources and render faster. There may be times, however, that due to the complexity of objects, Boolean will not produce acceptable results.

Intersect

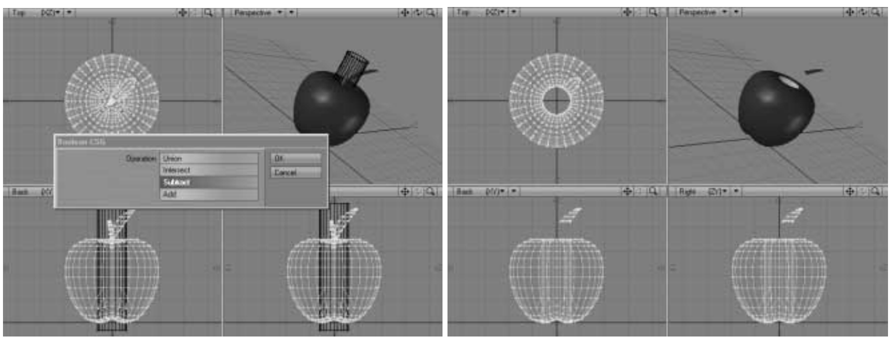
Intersect takes solid objects and leaves behind whatever portion was common between them. It leaves behind only those portions of the background layer and foreground layer that were overlapping. (This corresponds to the portion that the **Union** operation discards.) No new polygons or surfaces are created; therefore all surfaces will retain their original names. To work properly, both objects must be closed 3D solids.



Intersect mode before (*left*) and after (*right*)

Subtract

Subtract removes the background drilling template object from the foreground object, leaving a copy of itself embedded there. Essentially, it carves the background layer out of the foreground layer, leaving new inside surfaces in its wake. New surfaces will take on the names of surfaces that carved them. To work properly, both objects must be closed 3D solids.

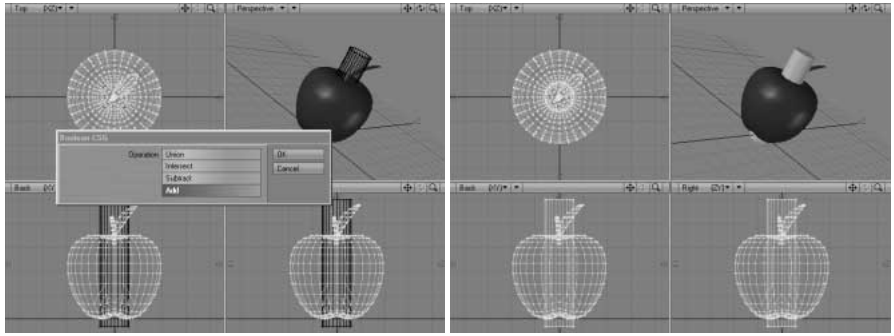


Subtract mode before (*left*) and after (*right*)

The results of this operation will vary depending whether the polygons being drilled consist of single-sided or double-sided polygons (see subsequent discussion).

Add

Add joins the background layer with the foreground layer so that the overlapping shapes are fully combined as one. You can use combinations of 2D and 3D objects. This is different from simply saving two overlapping objects from two different layers, since the polygons that physically overlap are merged in a form of mutual drill operation. No new surfaces are created, although some polygons may be subdivided. Therefore all surfaces will retain their original names.



Add mode before (*left*) and after (*right*)

The result is similar to just cutting and pasting the tube object into the apple layer. The difference is that where surface polygons from the two objects intersect they are sliced by each other to form a common edge. All interior polygons and surface names are retained.

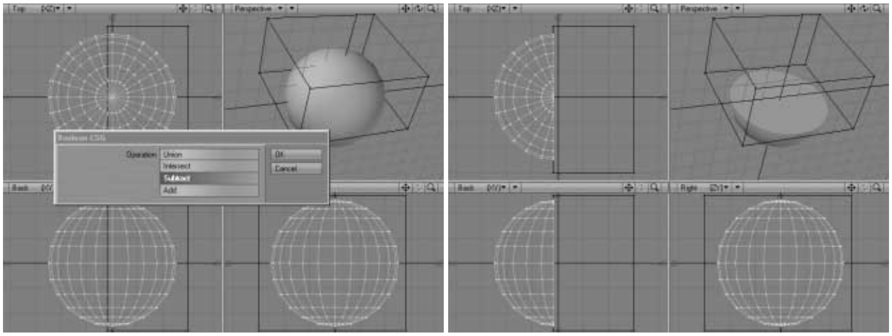
You can use more than two objects at a time, but it's often better to work with a pair at a time. The math involved in Boolean operations is complex, and using multiple objects in a single operation is quite time-consuming.

To use the Boolean command:

1. Place the target object in a layer.
2. Place the other object in another layer and put the object from step 1 in the background.
3. Position the object in the foreground as desired. If you plan to use it to cut away parts of the target object, it must align accordingly in 3D space.
4. Swap foreground and background layers using the apostrophe key (') shortcut.
5. Click **Multiply** > Combine: **Boolean**.
6. Select the type of boolean **Operation**. Click **OK**.

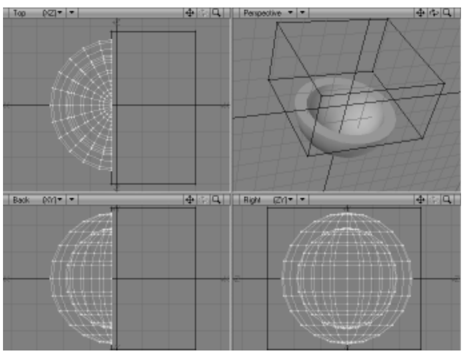
Single vs. Double-sided Polygons

Single-sided and double-sided polygons play a unique role in the Boolean **Subtract** operation. Imagine that you are a knife blade, slicing through a 3D sphere made of single-sided polygons. Every time you cut through a polygon layer, you pass from the outside of the sphere to the inside. If you passed through only one polygon, you are now inside the 3D sphere. Accordingly, a Boolean knife blade slicing through a one-sided sphere would leave solid halves.



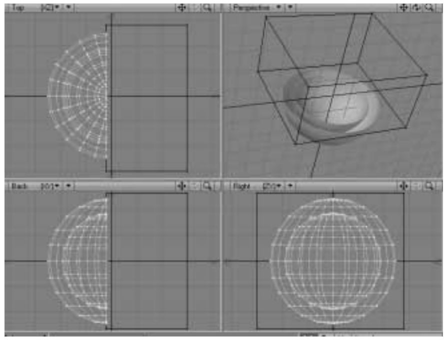
Normal boolean Subtract on two solid objects

Now say there are two spheres made of single-sided polygons, one sphere being smaller than the other, with the smaller one inside the other (normals facing in). You pass through the first polygon—from air to solid—then through a second polygon—from solid back to air. You would now be inside a hollow core inside the inner sphere. A Boolean knife slicing through this sphere would leave a hollow core.



Boolean Subtract on (single-sided interior) hollow ball

Now, if our two spheres were double-sided instead, Modeler would treat each sphere as if it were constructed with paper-thin walls (double-sided polygons are just back-to-back polygons). Remember, normally the sphere would be treated as a solid ball. Upon encountering a double-sided polygon, Modeler will think that it has passed through two polygons. In the paper-thin space between the front-facing and back-facing polygon, the Boolean knife blade sliced through air, then solid, then air again.



Boolean Subtract on (double-sided interior) hollow ball

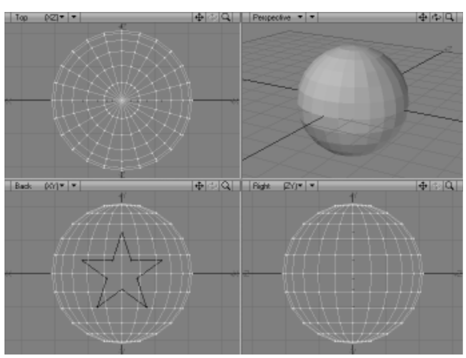
NOTE
Boolean will usually not operate as expected if the foreground and background objects are both double-sided.

The Drill Tools

There are two *drill* commands: Drill (**Multiply** > Combine: **Drill**) and Solid Drill (**Multiply** > Combine: **Solid Drill**). The difference is that with Drill you use a two-dimensional object as your drill bit. The bit extends infinitely along the drilling axis, so it has an imaginary third dimension. Think of it as looking straight down along the length of a drill bit. Forgetting the fact that you know it has length, it would look like a disc, right? With the Drill command, you just need a disc-shaped polygon and the length is assumed.

Solid Drill is essentially identical to Drill except—as you might have guessed—it uses an actual three-dimensional object as your drill bit.

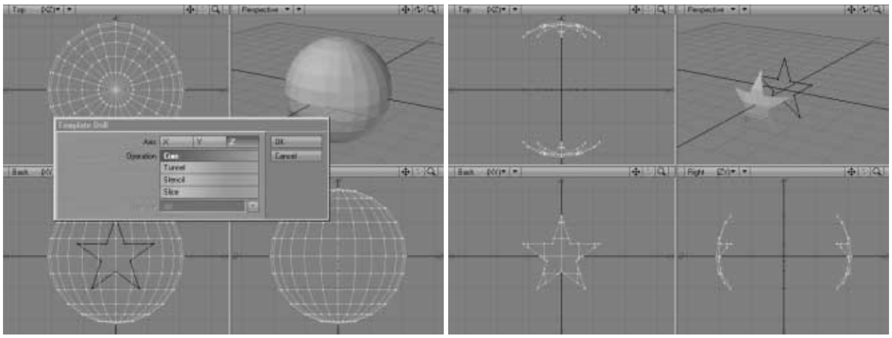
When you use a drill in the real world, you must first place a drill bit in the drill. This same concept applies to Modeler. Modeler uses an item in the background layer as the drill bit that will cut the item in the foreground layer(s).



> **NOTE**
> If your drill bit is an open curve, Modeler will treat it as a closed curve with its endpoints joined.

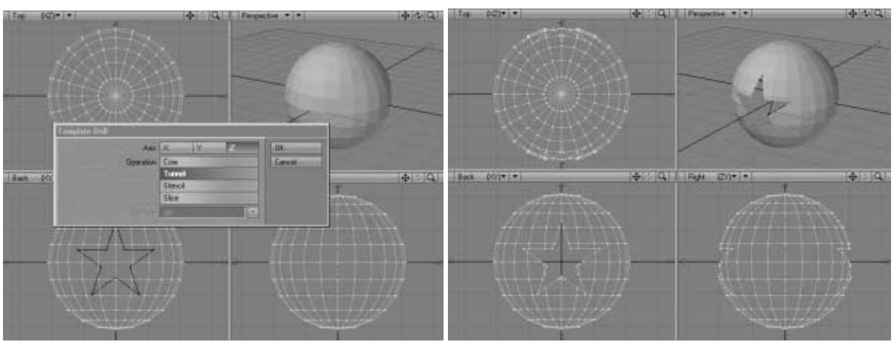
Both drill tools have the same four options:

Core The **Core** operation will include all polygons that reside inside the 3D space of the drill bit, plus portions of the polygons that straddle the borders.



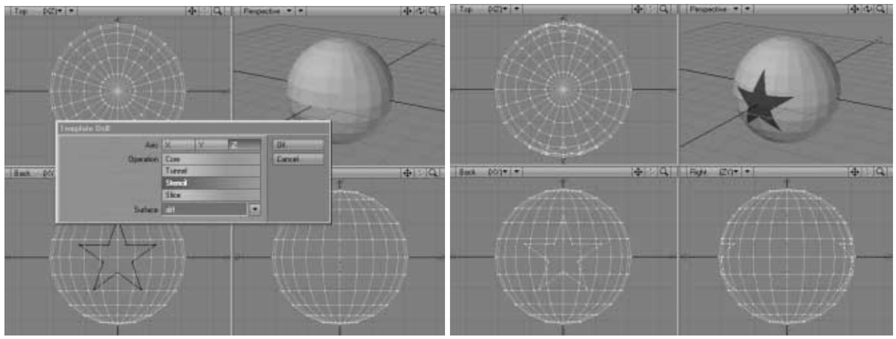
Core mode before (*left*) and after (*right*)

Tunnel **Tunnel** is the opposite of **Core**. It creates a *tunnel* in your object. All polygons that reside outside the 3D space of the drill bit will be included, plus the trimmed polygons that straddle the borders.



Tunnel mode before (left) and after (right)

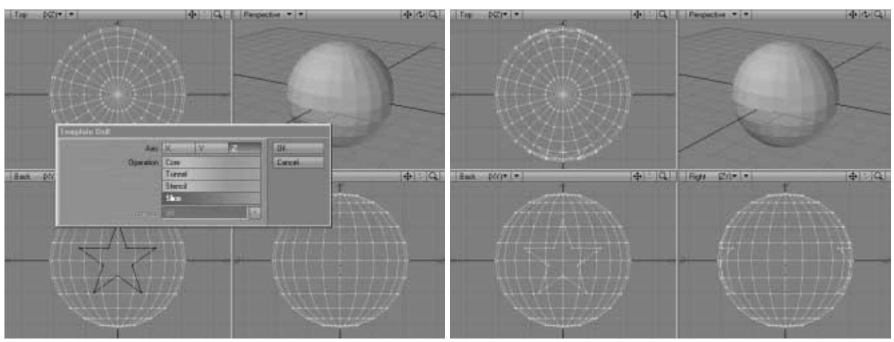
Stencil The **Stencil** option stencils the drill bit's shape onto any intersecting polygons. It also names all of the polygons that fall within the bit's shape. You can either enter a surface name in the field or select an existing name from the pop-up menu.



Stencil mode before (*left*) and after (*right*)

This option is great for stenciling text or logos onto objects—an alternative to using surface color textures. To help draw attention to a stenciled area, you might wish to use the Bevel tool (**Multiply** > Extend: **Bevel**) or Smooth Shift tool (**Multiply** > Extend: **Smooth Shift**) on the stencil.

Slice **Slice** is the same as **Stencil**, except it does not rename any surfaces.



Slice mode before (*left*) and after (*right*)

To drill an object:

1 Place the target object in a layer.

2 Place your bit object in another layer and put the target object in the background.

3 Position the bit object as desired. Use the viewport whose perpendicular axis (i.e., the axis you can't affect) is the one you want to drill along. For example, if you wanted to drill along the Z axis, position the bit object in the Back type viewport.

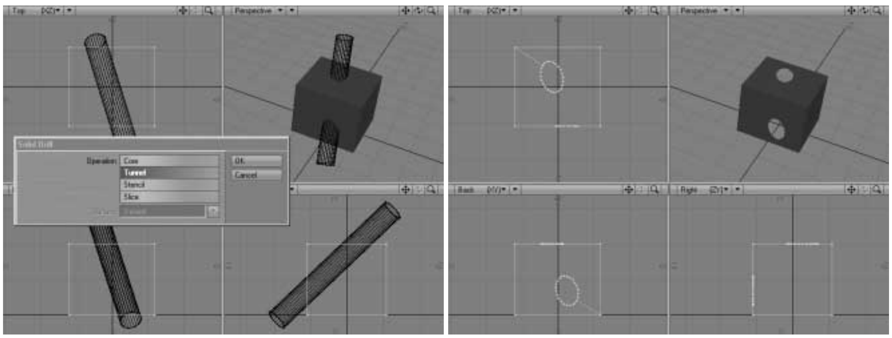
4 Swap foreground and background layers using the apostrophe key (‘) shortcut.

5 Choose **Multiply** > Combine: **Drill**.

6 Select the drilling **Axis** and the type of drilling **Operation**. Click **OK**.

The Solid Drill Tool

The Solid Drill command (**Multiply** > Combine: **Solid Drill**) will work just like the Drill command, but you have control over the effect along the drilling axis. There is no Axis option. It is unnecessary since you use a 3D object that has no *implied* third dimension.



NOTE
The objects in the foreground and background layers must physically overlap for the tool to work properly. They cannot be just lined up on an axis.

NOTE
Both the drill bit and the item to be drilled must be closed solids. A hemisphere with one open side, or a hollow tube, would not qualify, but a closed hemisphere or a tube with sealed ends would.

Faster and Better Drills and Booleans

Complex Drill and Boolean operations require a lot of computing. Selecting the affected polygons in the foreground layer before using either command will greatly speed things up. This lets Modeler concentrate on only those polygons that need to be affected. Also, if your initial result is not satisfactory—particularly with complex objects—try it again after nudging one of the objects on an axis a little.

Patches Menu

The commands in the **Multiply** > Combine: **Patches** menu are covered in Chapter 29.

THE DUPLICATE GROUP

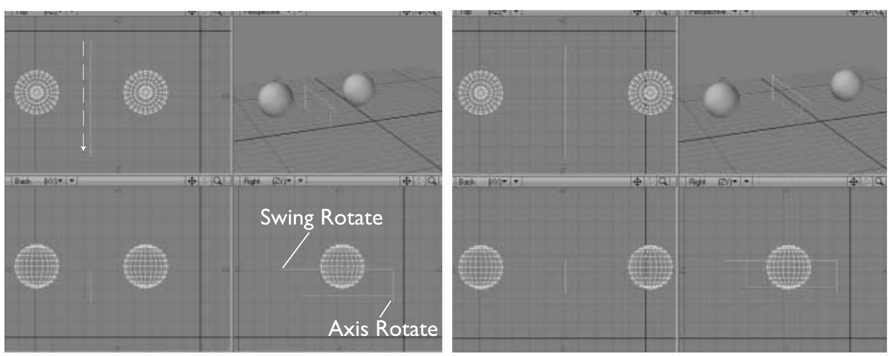
This group of tools multiplies geometry by replicating existing geometry.

The Mirror Tool

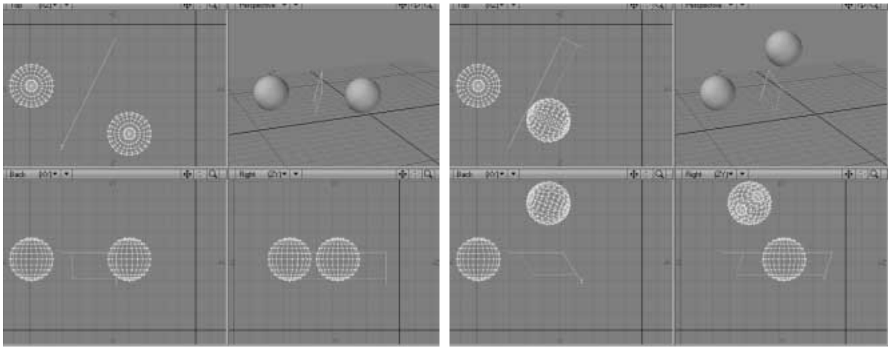
Use the Mirror tool (**Multiply >** Duplicate: **Mirror**) to create an exact mirror image of an item. You can mirror the whole object, selected points, or selected polygons. Many objects you create will be symmetrical in nature. So it often makes sense to build only half of an object, create a mirror image, and join them together. Other times your object may include certain parts that are identical, and an automated way to duplicate them is very helpful.

This interactive tool works using a mirroring *fence*. You drag out the fence with your mouse. This forms the main axis for the fence's length. Once you release the mouse button you can reposition by simply clicking somewhere off the fence—you are essentially repositioning the initial drag point. You can rotate the fence using the rotation handles. The swing rotation handle *swings* the fence like a gate around one end of the fence. The axis rotation handle rotates the fence around the axis along the length of the fence.

If **Free Rotation** is inactive on the numeric panel, the default setting, you will only be able to draw and rotate the fence across planes aligned with an axis. If you hold the CTRL key down, you will be able to use 15-degree increments. If **Free Rotation** is active, you are free to draw and rotate the fence at any angle.



Left: Initial drag. *Right:* Repositioning using perspective viewport

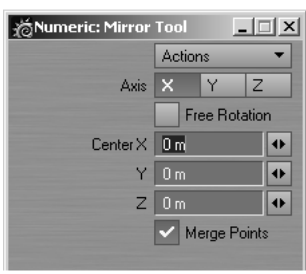


Left: Swinging the fence. Right: Rotating around the axis

When you are satisfied, simply click **Multiply** > Duplicate: **Mirror** again (or select another tool). To reset, click a non-active part of the interface or click **Undo**.

NOTE
If adjusting the axis rotation handle only resizes the fence, try adjusting the handle in the perspective view.

The numeric panel defines the XYZ coordinates of the **Center** point (where the position handle is). The **Axis**, if any, is the axis perpendicular to the fence. If the **Merge Points** option is active, any overlapping points are merged automatically. Use this if you are mirroring two halves of an object together.

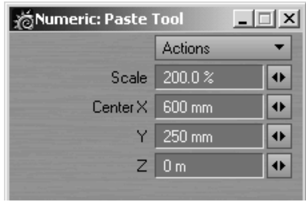


Mirror numeric panel

Paste Tool

The Paste tool (**Multiply** > Duplicate: **Paste**) lets you interactively paste whatever is in Modeler's copy buffer (i.e., something you previously cut or copied). Once you click in a viewport, you can position the geometry by dragging your mouse. Like most tools, drop it or select another to make the geometry, or click your RMB to create the geometry without dropping the tool.

Using the numeric panel, you can even scale the size of the geometry.



NOTE
The Paste tool has no effect on the actual contents of the copy buffer.

The Array Command

The Array command (**Multiply** > Duplicate: **Array**) duplicates the selected item any number of times using specific increments of distance. You can create a grid out of this object, like a matrix. (Unfortunately, no one can be told what the matrix is.) The shape of the array can either be **Rectangular** (box-shaped) or **Radial** (circular). Use the **Merge Points** option to automatically merge points at the end of the operation.

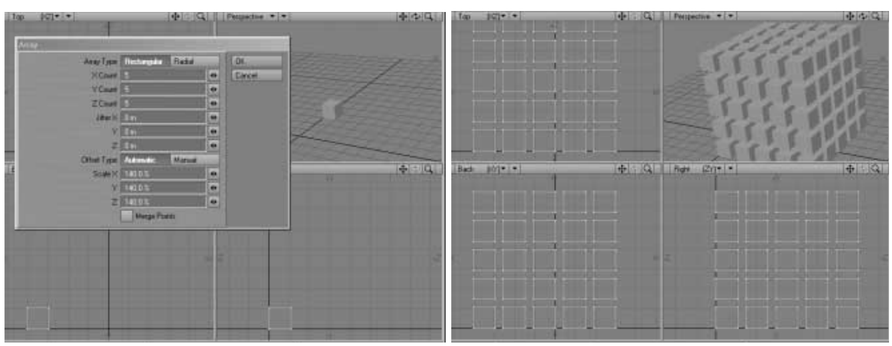
Array Type: Rectangular

When **Rectangular** is selected, you will have the following settings available:

Count Count is the number of repetitions of the original object along each of the axes.

Jitter Jitter adds randomization. A distance up to the value entered is added or subtracted to what the position would have otherwise been.

Offset **Automatic** uses the object's own dimensions as the spacing value. The **Scale** fields define a percentage of the dimensions to use. A value of 100 percent in the input fields will place the objects side by side. **Manual** lets you enter specific spacing values for each axis.

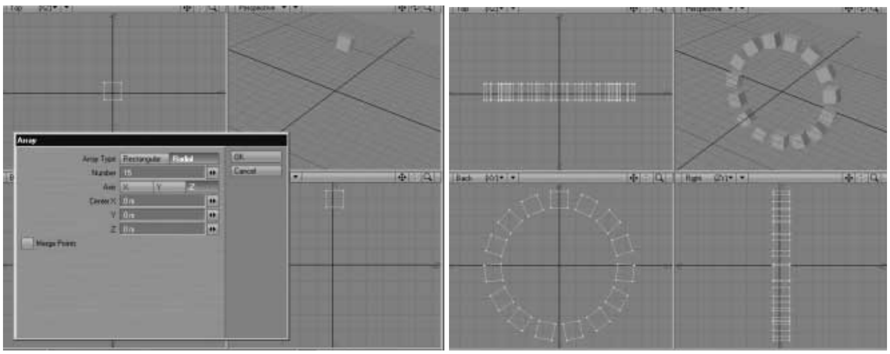


NOTE
Using **Automatic** on a 2-D object will cause the copies to be on top of each other along the missing dimension axis.

Array Type: Radial

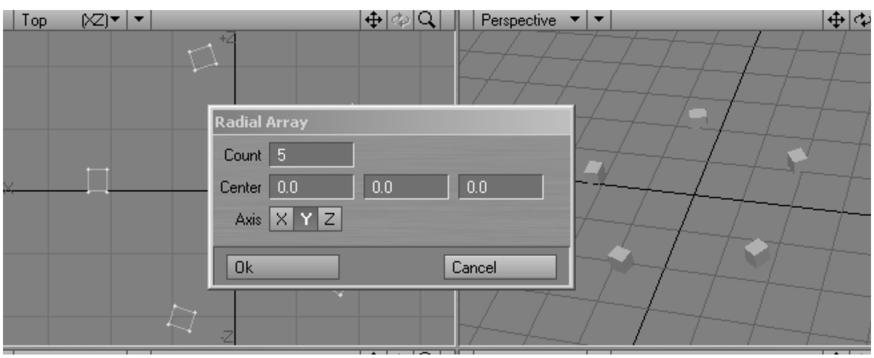
When **Radial** is selected, you have the following settings available:

Number	The number of repetitions of the original object.
Axis	The axis perpendicular to the radial shape.
Center	The coordinates of the center of rotation through which the axis extends.



Radial Array

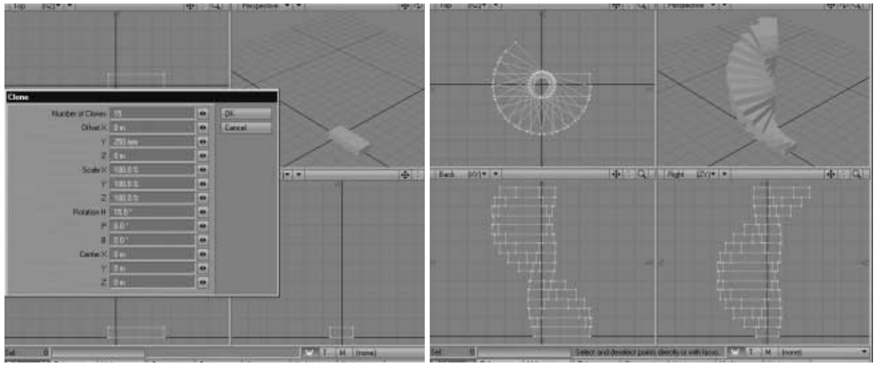
Radial Array creates an array of geometry around the **Center** position using the specified **Axis**. The **Count** setting sets the number in the final array.



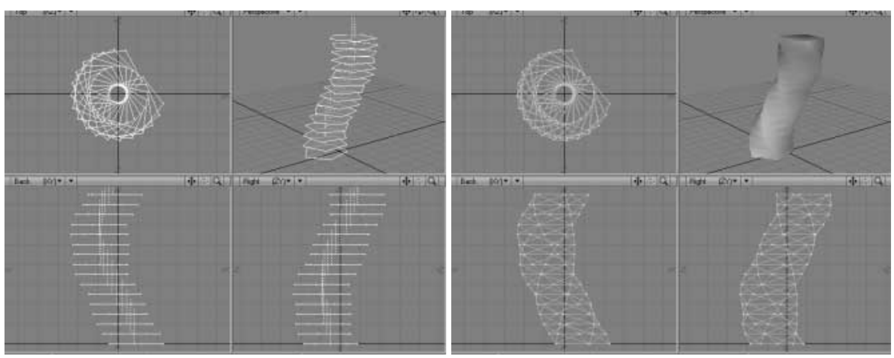
Radial Array

The Clone Command

The Clone command (**Multiply** > Duplicate: **Clone**) creates copies of an item that you can spread out in an even layout or array. You can create a series of copies along an axis at specified increments. Clone duplicates a selected item one or more times using specific increments of distance, rotation, and scaling. You can create a spiral staircase out of a single step, or a wood screw out of a triangular polygon outline, for example.



Immediately after a Clone operation involving a template (i.e., 2D) polygon, you can use the lasso to encircle and select the entire group of newly cloned polygons. They are automatically selected in the order they were created. This means you can use the Create Skin command (**Multiply** > Extend: **Skin**) to place an outer surface around the new framework. After using Create Skin, the framework polygons are no longer needed (since they are inside the skin and will not render anyway). They are still selected, so you can delete them (except maybe the two ends), and leave only the skinned object remaining. This technique is useful when you create objects like a wood screw, where you no longer need the internal (unseen) polygons after you complete the outer surface.



Left: Clones. *Right*: After skinning

Enter the number of copies of the selection to create in the **Number of Clones** fields.

The **Offset** fields determine the incremental distance between one clone and the next along the **X**, **Y**, and **Z** axes. The **Scale** fields set the amount of incremental scaling for each copy. The **Rotation** fields set the amount of incremental rotating.

NOTE

The effect is progressive, so the change is applied to each clone based on the previous clone. So, for example, with scaling, the copies will get progressively larger or smaller.

The **Center** fields define the coordinates to use as the center point, around which the operation takes place.

HINT

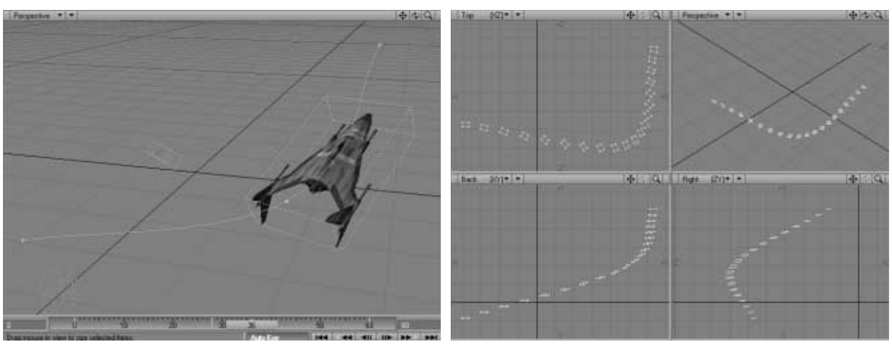
If you plan to clone with rotation, try to build your base object so that the rotation can happen around the Origin whenever possible.

HINT

To evenly space objects around an axis, take the number of total objects you want to end up with and divide it into 360. (You can even enter the formula into the input field.) This gives you the incremental rotation. Then set the **Number of Clones** to the total objects minus one.

The Path Clone Command

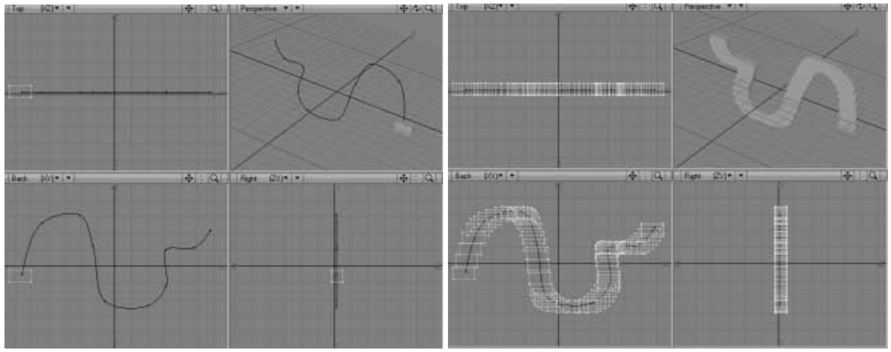
The Path Clone command (**Multiply >** Duplicate: **Path Clone**) works just like the Path Extrude command, discussed earlier. However, instead of resulting in an object with a smooth skin, it creates *skinless* segment clones along the motion path.



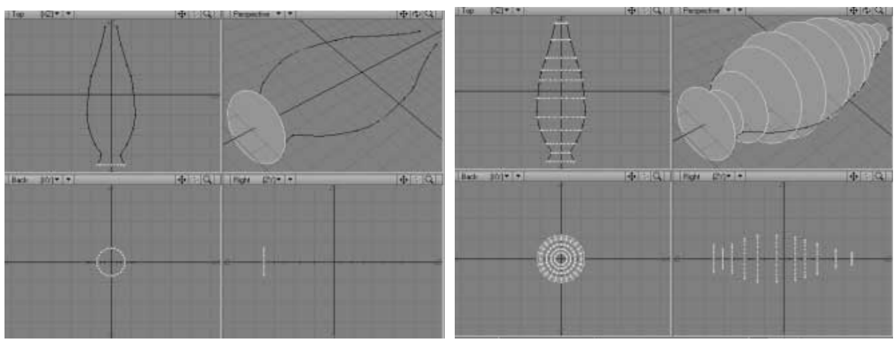
Left: Object motion path in Layout. Right: Flat box cloned along same path

The Rail Clone Command

The Rail Clone command (**Multiply >** Duplicate: **Rail Clone**) works just like the Rail Extrude command, discussed earlier. However, instead of resulting in an object with a smooth skin, it creates *skinless* segment clones along the background curve(s).



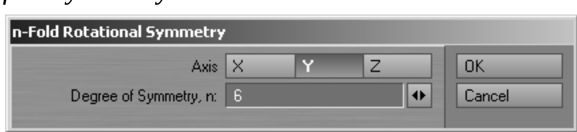
Left: Starting object in foreground and curve in background. Right: Result with Automatic mode

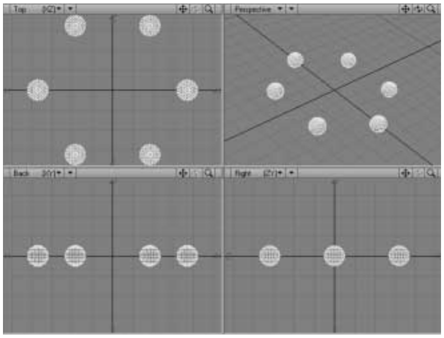


Left: Disc and two curves in background. Right: Result with Scaling on

The Symmetrize Command

The Symmetrize command (**Multiply** > Duplicate: **Symmetrize**) clones an object *n* times with a rotation each time so that the result displays *n-fold symmetry* about the selected **Axis**.

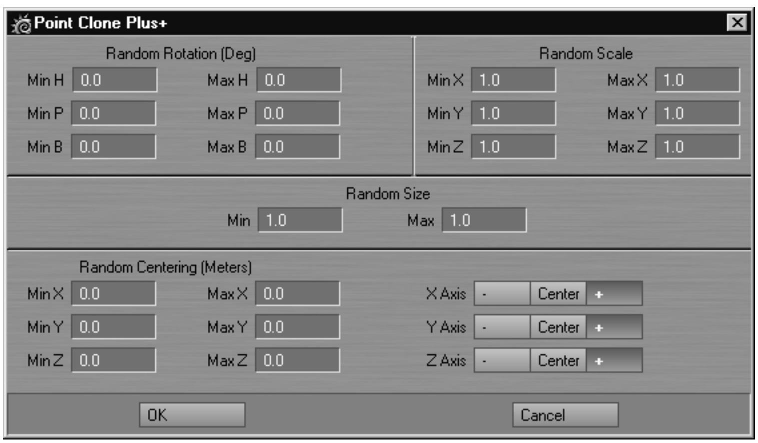




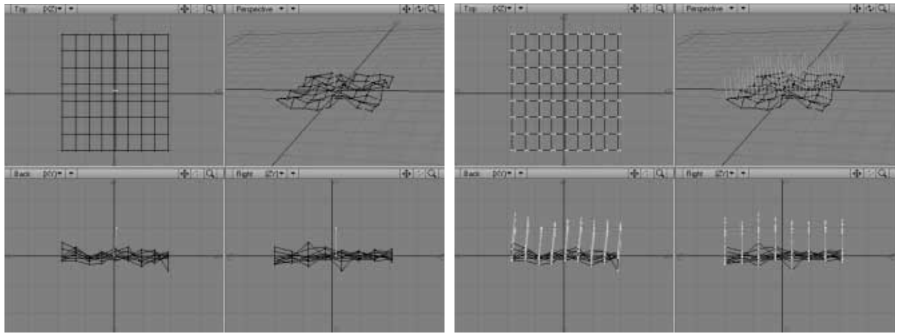
Symmetrize

Point-Clone-Plus

The Point-Clone-Plus command (**Multiply** > Duplicate: **Point-Clone-Plus+**) will clone objects in the foreground to the locations of points in the background. You can randomize the rotation, scale, and centering independently for every axis between two values. **Random Size** will randomly change the overall scaling of objects. The **XYZ Axis** options specify whether the objects should be centered on the points or flush on one side or another. To specify static values, enter the same number in the **Min** and **Max** fields.



This command is great for cloning and placing, say, tree objects on a landscape or putting hair on a head.



chapter 25
Constructing Geometry

Chapter 25: Constructing Geometry

So far, we've covered modeling tools that create an object by adding, rearranging and duplicating points/polygons. In this chapter we will cover some basic point and polygon maintenance tools as well as some time-saving ways to add and remove points and polygons. We'll also discuss some tools where you can use other objects and curves as modeling tools.

BASIC POINT AND POLYGON MAINTENANCE

The basic steps to copy and paste points and polygons are essentially the same. First, you select the points or polygons. Then you either click the **Copy** button (c) to copy the selection into the memory buffer, retaining the original, or click the **Cut** (X) button to delete the selection. You can also use the DEL key to run the Delete command; it is similar to Cut, but does not copy the data to memory for future pasting.

Once something is in the memory buffer, you can click the **Paste** button (V) to paste the contents back into the layer. The memory buffer is not reset after pasting. Thus, you can paste the contents multiple times. The buffer contents change only when they are replaced by a subsequent cut or copy.

NOTE
Geometry cut (or copied) to the memory buffer loses its association with geometry it is cut away from, even when it is pasted back in.

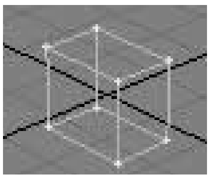
NOTE
For interactive pasting, use the Paste tool (**Multiply** > Duplicate: **Paste**) discussed in Chapter 24.

THE REMOVE POLYGONS COMMAND

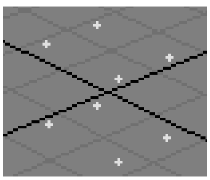
Frequently, you want to use some of the points of an object (often primitives) in creating your objects. A good example is using part of a disc's points to model a rounded corner. The way to do it is to use the Remove Polygons command (**Construct** > Reduce: **Rem Polygons**).

To remove polygons but not points:

1 Select the target polygons.



2 Choose **Construct** > Reduce: **Rem Polygons** or press the K key.



HINT
An easy way to remember this keyboard shortcut is to think of it as "killing" the polygons.

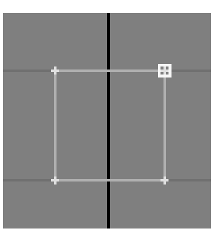
THE REMOVE POINTS COMMAND

The Remove Points command (**Construct** > Reduce: **Rem Points**) *disassociates* the selected point(s) from the selected polygon, but does not delete the point. Non-selected polygons that used that point remain unchanged.

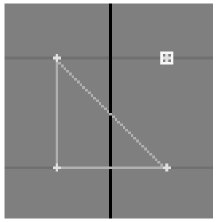
To remove a point from a polygon:

1 Select the polygon(s) containing the point to be removed (Polygon Selection mode).

2 Select the points to be removed (Point Selection mode).

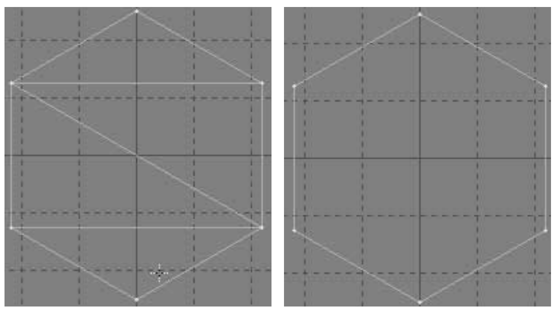


3 Choose **Construct** > Reduce: **Rem Points**. To actually delete the points, use the **Cut** function.



THE MERGE POLYGONS COMMAND

Construct > Reduce: **Merge Polygons** joins two selected polygons that share at least one common edge. A common edge is one where two or more polygons share all of the points along that edge.



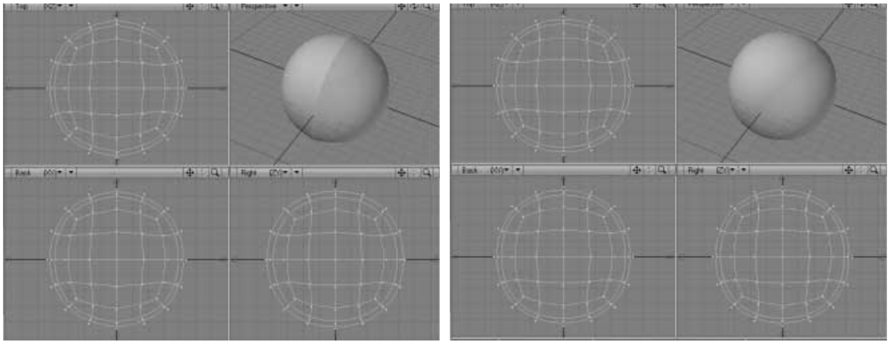
The Merge Polygons command also works on curves, if they share a common terminating point. It doesn't matter if you use the start or end of the curve. They will be merged into one continuous curve.

Common problems are edges that don't share all points. Try merging points first, to eliminate overlapping points. After using this command, check for non-planar polygons, which often result.

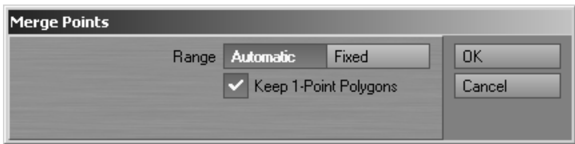
THE MERGE POINTS COMMAND

Use the Merge Points command (**Construct** > Reduce: **Merge Points**) to remove points that occupy the same or very nearly the same space. Remember, if all of the points along a polygon's edge are shared, LightWave's surface smoothing can smooth over the two polygons. Many circumstances exist where you may create unnecessary duplicate points, but here is a list of some common ones:

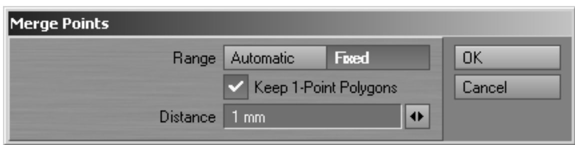
- After you cut polygons to a separate layer and then paste them back.
- After you use the Mirror tool and drill operations, because the complex mathematics involved often generates extra points.



Left: Overlapping points cause visible seam. *Right*: After merging points there is no visible seam or panty line



Merge Points Automatic Range



Merge Points Fixed Range

If the **Keep 1-Point Polygons** option is disabled, polygons are automatically deleted if they become single points (due to their vertices merging together). The default is for these polygons to be retained.

The Merge Points command has two **Range** modes: **Automatic** and **Fixed**. These basically determine the criteria for when points should be considered *mergeable*.

Automatic merges the points that share the same space. For objects or portions of objects that you have cut from the main object with the intention of merging later (back into their original positions) use **Automatic**.

Absolute merges any points within the specified **Distance** of one another. When merged, one point will move to the location of another—Merge will <u>not</u> average the distance between points to create a new point midway between them. As such, the related polygon may be stretched as necessary.

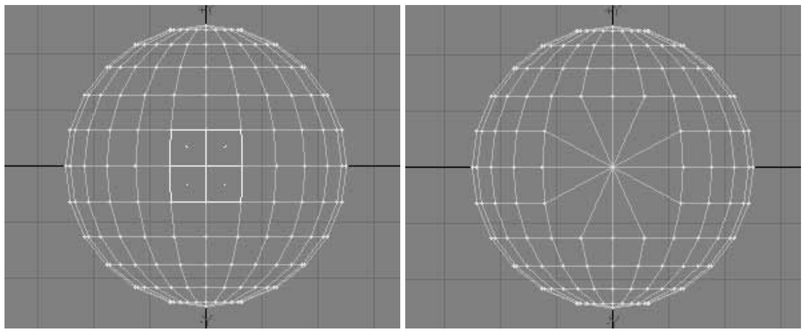
HINT

To speed the operation, select the points you wish to merge before using this command. (It's OK if you select more than you need.)

NOTE
See also the Weld command, in Chapter 26.

COLLAPSE POLYGONS

The Collapse Polygons command (**Construct** > Reduce: **Collapse Pols**) replaces selected polygons with a single point at their average position. It is like moving all of the points for the selected polygons to their center and then merging the points.



Left: Four polygons selected. *Right*: After using Collapse Polygons

THE UNIFY POLYGONS COMMAND

The Unify Polygons command (**Construct** > Reduce: **Unify Polygons**) converts selected polygons sharing the same points (such as double-sided polygons) into a single polygon. It creates single-sided polygons with regard to the direction of their surface normals. However, unusual object shapes can occasionally fool its complex algorithms, resulting in single-sided polygons facing the incorrect direction. If this happens, you may need to use the Align Polygons or Flip Polygons commands to correct the resulting direction of the normals.

THE QEMLOSS2 FUNCTION

The qemLOSS2 function (**Construct** > Reduce: **qemLOSS2**) uses a surface simplification algorithm in an attempt to reduce the number of polygons in an object. The plug-in lets you rapidly produce good quality approximations of excessively detailed polygonal models. qemLOSS2 is an excellent way to create low resolution *stand-in* objects for scene layout or for multi-resolution models. You can use the object replacement capability in Layout to substitute a low resolution object in place of an object with a high polygon count when it is far away from the camera.

Left: standard 5,502-polygon cow. *Right*: slim 2,750-polygon version.

This plug-in uses routines adapted from Michael Garland's public domain QSlim Simplification Software. The algorithms used in this software are described in the papers written by Michael Garland and Paul S. Heckbert, *Surface Simplification Using Quadric Error Metrics*, SIGGRAPH 97, and *Simplifying Surfaces with Color and Texture using Quadric Error Metrics*, IEEE Visualization 98.

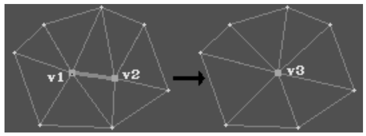
Introduction

Many 3D models contain a large number of polygons, especially algorithmically created models (such as those created by implicit surface construction techniques) and models created with 3D scanners and digitizers. Polygon reduction in these types of models is currently a very active research area in computer graphics. Obviously, rendering 3D scenes is much faster when the models contain a minimum number of polygons. Also with the current trend towards sharing 3D worlds over the Internet using VRML, level of detail models (LOD) are becoming absolutely necessary for creating worlds in which the user can browse and interact in a reasonable manner.

This plug-in provides polygon reduction on objects within Modeler. Only one parameter, the reduction **Goal**, must be set by the user, the remaining default values should provide good reduction for many objects with a high polygon count. However, to get the best results, the user must understand some of the basic concepts behind the algorithm.

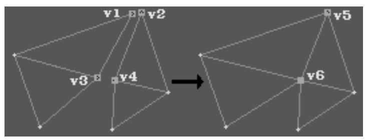
Terminology

The simplification algorithm is based on contractions of vertex pairs. It supports two types of contractions: *edge* and *non-edge contractions*. An edge contraction occurs when the vertex pair shares an edge. This is the primary type of contraction that occurs during the reduction stage (in fact, non-edge contractions are turned off by the default parameters). The following figure shows an example of an edge contraction where vertex v1 and vertex v2 are joined to form a new vertex v3. Since v1 and v2 share an edge (highlighted below), one or more triangles will always be removed during this contraction. In this example, two triangles are eliminated from the mesh.



Edge Contraction

Non-edge contractions let the algorithm join unconnected areas of the object together. The next figure shows an example of a non-edge contraction (also called *aggregation*).



Non-edge Contractions

In fact, there are two non-edge contractions taking place: v1 and v2 contract to form v5; v3 and v4 contract to form v6. Since by definition the vertices in the contraction pairs do not share an edge (there is no edge between v1 and v2, or the v3, v4 pair), there is no actual reduction in the polygon count of the object. However, since this feature joins previously unconnected areas of the object together, the potential exists for future reductions. Many times this can also provide a better low resolution approximation of an object that has many disconnected regions.

As the algorithm proceeds through each iteration, an approximation of geometric error accumulates at each vertex of the object. If this geometric error is less than the user defined **Maximum Error Tolerance**, the vertex is marked as a viable candidate for another contraction. Once the geometric error for a vertex becomes greater than the maximum threshold value, it will no longer be considered in any more contractions. During each iteration, the vertex pair with the smallest combined geometric error is chosen for the current contraction.

The algorithm proceeds until the simplified object is reduced to the user's targeted **Goal** (number of polygons) or until all the vertex errors have become greater than the **Maximum Error Tolerance**. These two parameters control how much reduction will take place, the remainder of the parameters control various aspects of the vertex contractions.

Two types of vertices receive special consideration by this plug-in: surface border and boundary vertices. Parameters are provided for you to weight the geometric error for these two special types of vertices:

- A *surface border vertex* is a point that is shared by two or more surfaces.
- An edge that exists in only one triangle is a *boundary edge*, and determines two endpoints that are called *boundary vertices*.

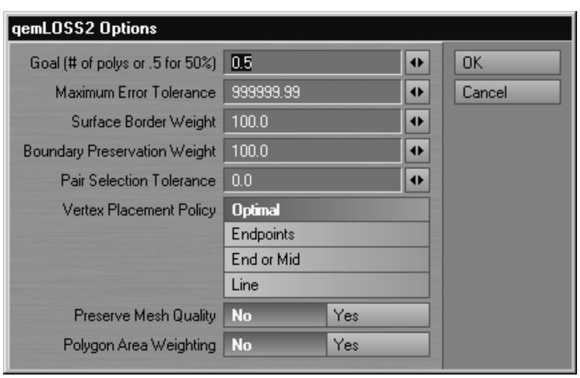
The **Surface Border Weight** and **Boundary Preservation Weight** parameters let you weigh the geometric error at these points. The higher the weight, the less likely the vertex will be replaced. Clever use of these parameters (along with some equally clever surfacing) can provide quite a bit of control over the contraction process. The **Polygon Area Weighting** parameter, causes every vertex's geometric error to be weighted by the area of the polygons that contain the vertex. Once again causing larger values (triangles with larger areas) to be less likely to be removed.

To help preserve the shape of these shared borders, weighting of the surface border vertices, using **Surface Border Weight**, has been added to the algorithm to help restrict movement/replacement of those vertices.

WARNING
Always save your objects before you run qemLOSS2!

Using qemLOSS2

Make sure you have an object in the current foreground layer(s) of Modeler. You also need at least one empty layer, because the existing object remains unchanged, and the reduced object is placed in the first available empty layer. Choose **Construct** > Reduce: **qemLOSS2** and you are presented with the following panel.



NOTE
qemLOSS2 ignores any polygon selections and works only on the entire object in the foreground layer, including any hidden polygons.

Goal lets you set the final number of polygons you would like in the simplified object. You may enter either a desired polygon count (an integer such as 1000), or a percent based on the number of polygons found in the original object (a real number with a percent sign at the end, such as 65.2%). If you enter a percentage, it will simply calculate the polygon goal by multiplying that percentage by the total polygon count in the original object. So 100% will mean no reduction takes place, and 0% means the object will disappear completely.

The first major step qemLOSS2 takes is to triple your polygons, which will increase the polygon count, if your model contains non-triangular polygons. A **Goal** of 100%, however, will reduce the number of polygons back to the original number, but now they will all be triangles. If that reduction doesn't preserve your model's shape well enough, don't hesitate to try percentages over 100%, those models will still be smaller than the tripled original.

NOTE
Other parameter settings may cause the algorithm to fail to reach the reduction goal.

Maximum Error Tolerance sets the geometric error threshold. A large value (such as the default 1,000,000) almost guarantees that you will achieve the reduction goal in the previous parameter field. Smaller values may preserve the shape of the original object better, however you may not achieve the desired reduction goal entered above.

Surface Border Weight gives you some control over how often and far surface border vertices are relocated during the reduction process. A large value restricts the relocation of points that lie along surface borders, thereby preserving the shape of the border fairly well. Smaller values let the points move farther from their original location, possibly causing the border to change its shape. A value of 0 will not constrain the surface borders at all, and vertices along a border may not end up where you expect when the simplification is finished.

Boundary Preservation Weight allows you some control over how boundary vertices are relocated during the contraction process. A small value lets the boundary points move farther than larger values. A value of 0 will not constrain the boundary at all, and the object may not look at all like you expect when the simplification is finished.

If you end up with unexpected gaps between non-contiguous polygon surfaces, you can try to preserve those boundaries by increasing the **Boundary Preservation Weight**. However, you might get better results by merging the edge points together before running qemLOSS2. You can always just cut and paste the polygons afterward, if the separation was essential.

Pair Selection Tolerance determines whether non-edge contractions are performed during the simplification process. If this value is 0, non-edge contractions are turned off, and only edge contractions will take place during the simplification. Any value greater than 0 will cause non-edge contractions to be possible during the reduction. If you enter a negative number, qemLOSS2 will automatically use 5% of the radius of the object's bounding sphere. This is the value you should use if you want to start experimenting with this parameter.

Take care when changing this value to anything but 0! It is strongly recommended that you leave this at 0 for all complex models with lots of polygons. If you use this, first reduce the model to a fairly small polygon count with it turned off (=0), then reduce the reduction again with a carefully chosen **Pair Selection Tolerance**, or better yet, just use a negative number. It is a very memory intensive operation.

Vertex Placement Policy is best left at **Optimal**. When a pair of vertices is contracted, the algorithm must decide where to locate the new vertex. The algorithm can use any one of the three final options listed in this selection of buttons. **Optimal** calculates the new placement based on the location with the least amount of geometric error. Doesn't hurt to experiment with these options though.

Sometimes vertex pair contractions do not preserve the orientation of the faces in certain areas of the contraction, If **Preserve Mesh Quality** is changed to **Yes**, the normal of each neighboring face is compared before and after the contraction. If the normal flips, the contraction is penalized greatly by making the geometric error for that contraction very large (so that contraction will probably never take place). In most cases, this will not be a problem, and the plug-in will work faster if this parameter stays set to **No**. But it's not terribly slow, so it certainly doesn't hurt to experiment with it.

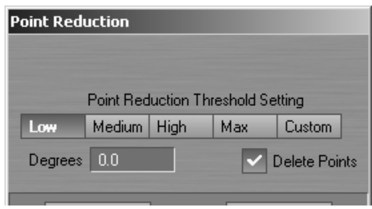
If you change **Polygon Area Weighting** to **Yes**, the area of the triangle containing the vertex is used to weight the geometric error. This will cause larger triangles to increase the geometric errors of its vertices, so it is less likely to be chosen for simplification.

The qemLOSS2 function displays a progress monitor while it creates the reduced object in the first available empty layer. Here is what happens: first, the object in the foreground layer(s) is copied to the first available empty layer, then all its polygons are converted to triangles using Modeler's Triple command. Next, all vertices and polygons are converted into the necessary data structures needed for the simplification routines, and the copied object is subsequently removed. Once the simplification routine finishes, the reduced polygon object is placed in the previously empty layer.

NOTE
For additional information, check out http://amber.rc.arizona.edu/lw.

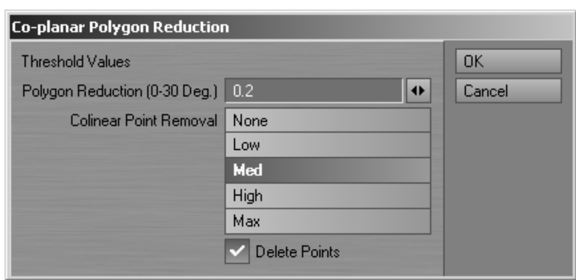
THE REDUCE-POINTS COMMAND

The Reduce-Points command (**Construct** > Reduce: **Reduce-Points**) traverses polygon edges and removes any points (from the polygons) that form an angle with its adjacent points that is less than the **Degrees** setting. This plug-in is especially useful to reduce the number of points and polygons generated by extruding or beveling text objects. Simply select a **Point Reduction Threshold Setting** and the **Degrees** will be set automatically. You can also enter your own **Degrees** setting. Activate **Delete Points** to delete the removed points.



THE REDUCE-POLYGONS COMMAND

The Reduce-Polygons command (**Construct** > Reduce: **Reduce-Polygons**) merges co-planar polygons with three or more vertices into a single polygon and can optionally remove any stray points resulting from the polygon removal. Polygons are merged if the relative angle between two polygons is less than the **Polygon Reduction** setting, they share an edge, and have the same surface name.



Colinear Point Removal deletes points that are still a vertex of a new polygon. You may set the level of this removal computation. To delete all points that are no longer a vertex of a polygon, check the **Delete Points** box.

ADDING POLYGON POINTS

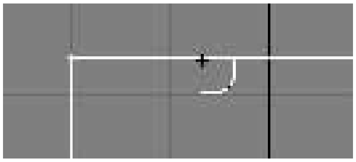
Sometimes you will need to add a point to a polygon and then move the point to change the shape of the polygon. You might also want to add multiple points so you can manually subdivide a polygon into smaller polys.

To add a point to a polygon:

1 First, select the target polygon. If your selected edge is shared with another polygon, select that one as well. If you do not, the edge will split and the polygons will no longer be joined.

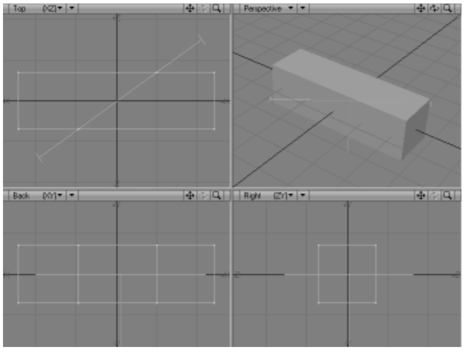
2 Choose **Construct** > Subdivide: **Add Points**.

3 Click the black plus sign at your desired insertion point along the edge.

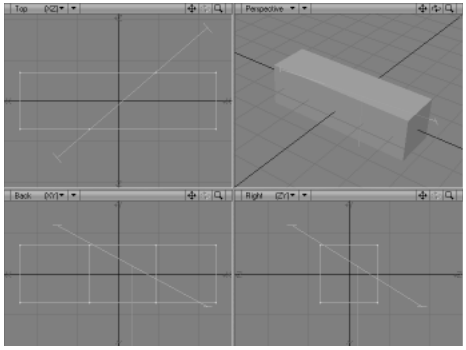


THE KNIFE TOOL

The Knife tool (**Construct** > Subdivide: **Knife**) lets you interactively get results similar to the Drill function's Slice option; however, it requires no background object. You simply drag out a *slicing line* in any viewport—nothing need be in a background layer. The line is infinite along the axis perpendicular to the view you initially drag in.



The other viewports will let you see the line from a different angle and a small dotted line will be visible that shows you the direction of the slice. After you have released the mouse button, you can drag the *T-handles* at the ends of the line (in any viewport) to reposition them or drag the center of the line to move the entire line. The tip of the dotted line can also be dragged to change the direction in increments of 90 degrees. This is an interactive tool so you will see the new geometry results as you position the line.



Like most tools, drop it or select another to make the geometry, or click your RMB to create the geometry without dropping the tool. Click on an open area of the interface to cancel the operation before it is made or use **Undo**.

The Knife tool's numeric requester indicates the **Start** and **End** XYZ coordinates, defining the end points of the line. The **Axis** settings define a target point, relative to the base of the dotted direction line, that the line goes through.

THE SUBDIVIDE POLYGONS PANEL

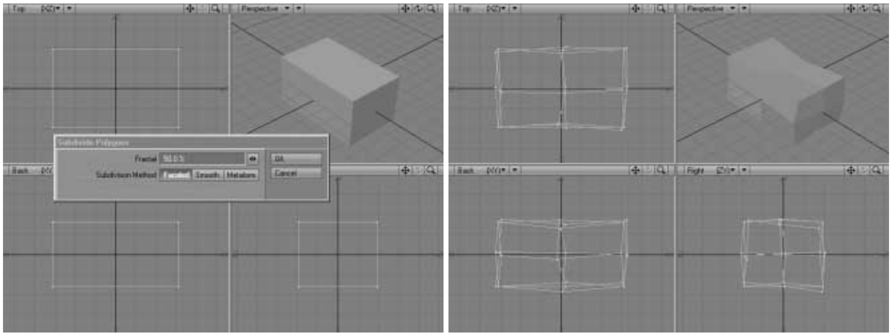
The Subdivide Polygons panel appears when you choose **Construct** > Subdivide: **Subdivide**. This panel provides several commands that let you automatically increase the detail of existing polygons. To use this panel, you must select the target polygons first.

NOTE
You cannot use Subdivide Polygons on polygons with more than four sides.

Randomizing Options

All Subdivide options let you enter a **Fractal** factor that will randomly jitter newly created points. The result is a subdivided object with its points randomly jittered. A factor of 0 will not jitter points and higher values will jitter points more.

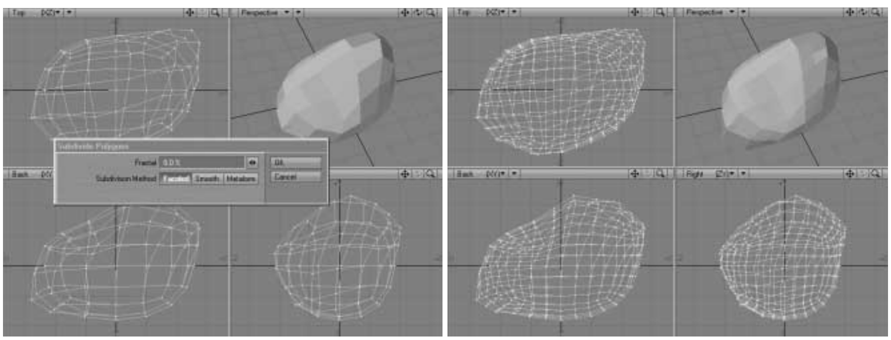
The jitter radius is a fraction of the edge length, times the **Fractal** factor. A factor of 1 jitters the new points by at most 50 percent of the length of the edge that they will subdivide. A factor of 0.1 is five percent jitter and a factor of 2 will move points 100 percent of the edge length. Points added to the center of four-point polygons are jittered a comparable amount, although the formula for them is more complicated.



Using Fractal

Faceted Subdivide

The **Faceted** option adds more segments to existing polygons, but doesn't change the overall shape. Let's say you made a box, but just wanted one side to have multiple segments. In this case you could use **Faceted**.



Faceted Subdivide

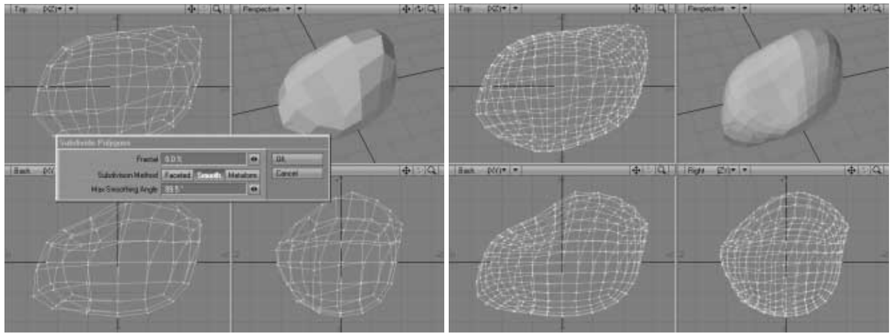
NOTE

Faceted Subdivide works on triangles, like after the Triple command is used.

Smooth Subdivide

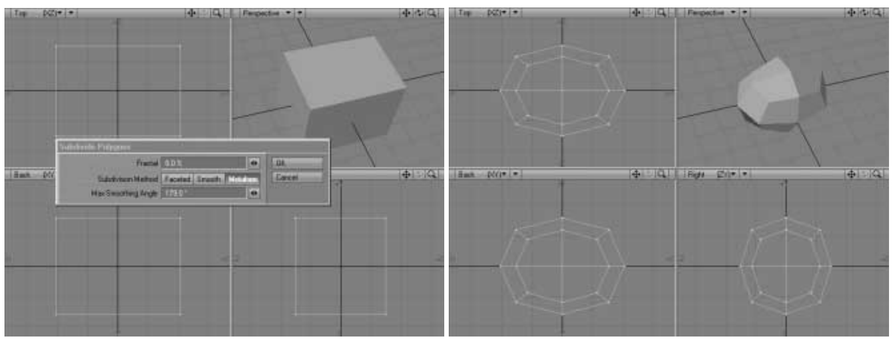
Smooth will subdivide objects that have curved, smoothed, or rounded areas. Modeler assumes any two polygons are part of a smooth surface and will extrapolate where subdivided detail should be placed to maintain (even enhance) the curvature of the original form.

The value entered in the **Max Smoothing Angle** field is used to determine whether adjoining polygons should be smoothed or not. Adjacent polygons whose normals form an angle above this value will not be smoothly subdivided. The default is 89.5 degrees, so polygons at right angles or sharper will not be affected.


Smooth Subdivide

Smooth Subdivide

Metaform is another method of subdividing. The object will be smoothed dramatically, with the original object acting as kind of a bounding box template for a slightly smaller, more rounded form. **Metaform** is very useful for creating smooth organic-looking objects out of simple geometric structures.


Smooth Subdivide

Metaform is an adaptive process, meaning that locations that contain greater detail (i.e., more points/polygons) will have more detail in the smoothing process. It is easy to create rounded forms with **Metaform**. Moreover, if you use **Metaform** on part of an object by selecting polygons, you can create specific areas of greater detail on an object without subdividing the entire object again and again.

Adjoining polygons with surface normal angles greater than the **Max Smoothing Angle** will not be subdivided in a smooth manner. The default angle is 179 degrees, so only polygons that are nearly parallel with each other are not affected.

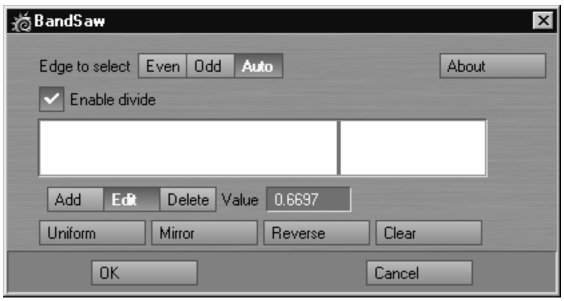
Metaform is often used one or more times on a box to create a rounded starting point for SubPatch cages. It is superior to using, say, a ball primitive, since all polygons are quads.

NOTE

Smooth and **Metaform** can achieve similar results using the same maximum smoothing angle; however, **Smooth** will tend to smooth outward, and make a slightly larger object. **Metaform**, on the other hand, smoothes inward, and make a more compact object.

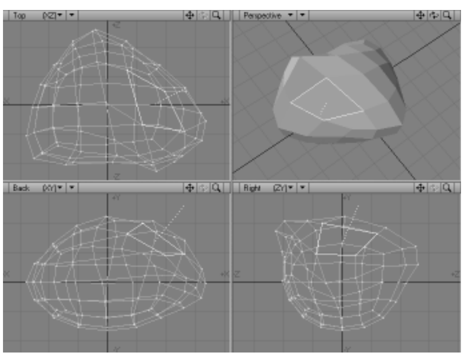
THE BANDSAW COMMAND

Bzzzzzzzzzzzzzzzzzzzzz...super knife! BandSaw will slice through a contiguous band of four-point polygons (or patches). To use the Bandsaw, just select a polygon and activate the BandSaw tool (**Construct** > Subdivide: **Bandsaw**). (If you select more than one polygon, the first selected polygon controls the operation.) This tool is tailored for use on SubPatch objects in areas where you need more control points.

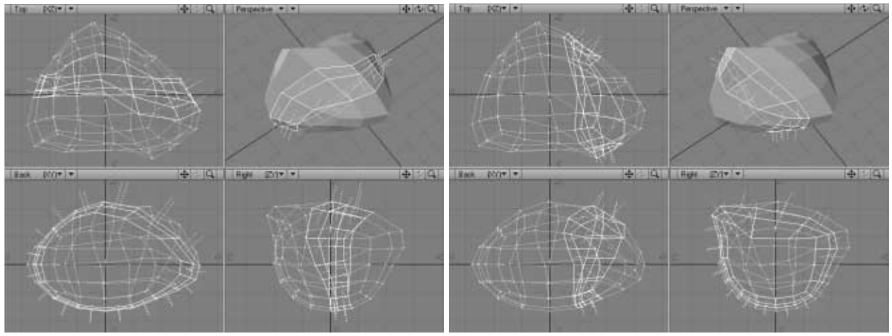


Bandsaw

The operation continues all the way around until it hits itself or a non-four-point polygon. The **Edge to select** option determines the direction. With a quad, there are only two ways and these are described as **Even** or **Odd**. Select **Auto** for BandSaw to pick the direction for you.



Single polygon selected



Some results using Bandsaw

By default, there will be a single default slice right in the center. This is the positioning of the slice that will be made on the polygons. You can move it by selecting the **Edit** mode and dragging it with your mouse. You can also add more slices in the **Add** mode or remove them in the **Delete** mode.

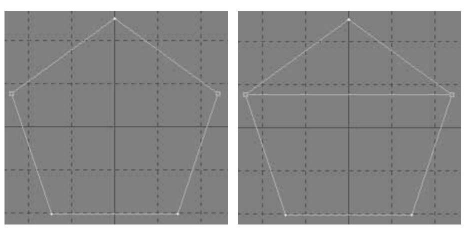
Uniform equalizes the spacing between slices. **Mirror** will mirror the selected slice across the center. **Reverse** inverts the order—handy if you placed the slice(s) on the wrong side.

To actually perform the slice, you need to activate the **Enable divide** option. However, you can also use BandSaw to just select the band of polygons by not activating this option.

THE SPLIT POLYGONS COMMAND

Choosing **Construct** > Subdivide: **Split** will divide a polygon into two smaller polygons using its existing points. If used on a curve, the curve will be split into two distinct curves.

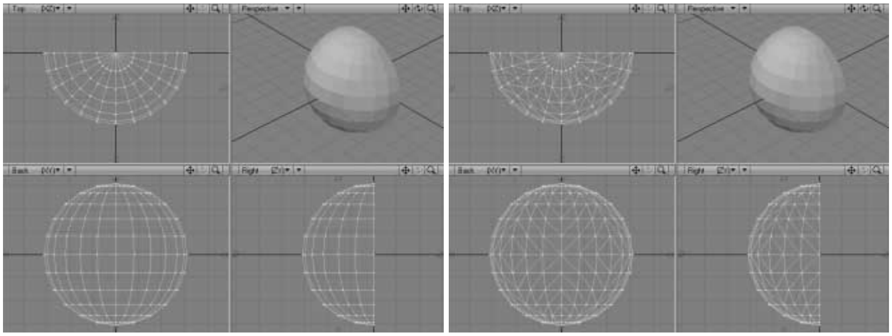
Select a polygon and switch to the Point edit mode. Select two points that do not share any edges. Clicking **Split** will subdivide the polygon along a line between those two points.



Splitting polygons

THE TRIPLE COMMAND

The Triple command (**Construct** > Subdivide: **Triple**) converts the selected polygon(s) into triangular polygons. For example, a square polygon becomes two triangular polygons. Triangles cannot by definition become non-planar; therefore, this tool is useful for objects you wish to animate using bones, or objects that will have displacement maps. Tripling non-planar polygons makes them planar. Note that tripling does not change the general shape of the object/polygons. Also, once a polygon is a triangle, tripling it again has no effect.



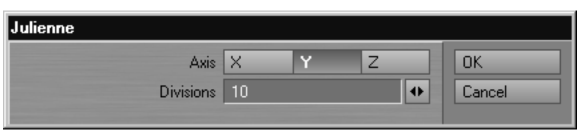
Before (*left*) and after (*right*) tripling polygons

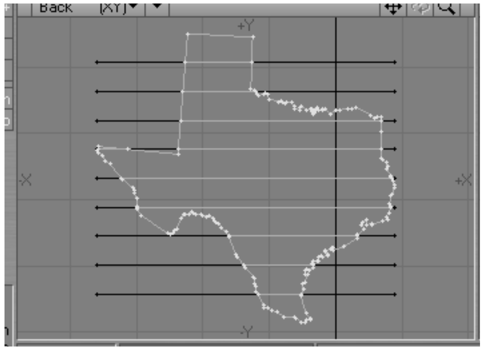
HINT
To select only non-planar polygons, use the Polygon Statistics panel (**Modeler** > **Windows** > **Statistics**).

NOTE
Also see the Fast-Triple-Fan and Fast-Triple-Traverse commands discussed later.

THE JULIENNE COMMAND

The Julienne command (**Construct** > Subdivide: **Julienne**) slices an object into sections that are regularly spaced along a given axis. Julienne is very useful for subdividing objects that will be bent or used with a displacement map.





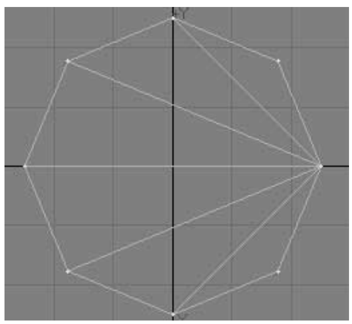
After using Julienne

THE SPLIT SKELEGON COMMAND

See Chapter 10.

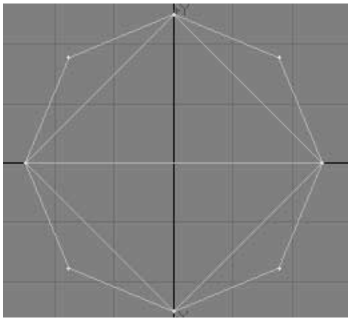
THE TRIPLE FAN COMMAND

The Triple Fan command (**Construct** > Subdivide: **Triple Fan**) triples polygons, with new polygons sharing a common point in a fan shape.


Triple Fan

THE TRIPLE TRAVERSE COMMAND

The Triple Traverse command (**Construct** > Subdivide: **Triple-Traverse**) triples polygons, with new polygons being created along the outside edge, moving inward.


Triple Traverse

SubPatches

See Chapter 29.

The Freeze Command

The Freeze command (**Construct** > Convert: **Freeze**) is used to convert curves (see Chapter 22), SubPatch objects (see Chapter 29) and Meta-primitives (see Chapter 29) into polygons.

Make... Commands

The *Make...* commands in the **Construct** > Convert: menu group are used to create skelegons (see Chapter 10) and Meta-primitives (see Chapter 29) from normal geometry.

The Toggle Metamesh Command

This command toggles the display of Meta-primitives on/off. See Chapter 29.

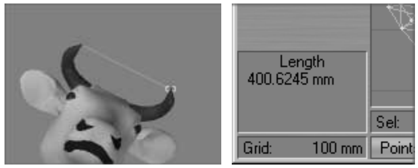
Judging Distance and Angles

You will often need to estimate distances and angles in the modeling process. The grid size can yield reasonably accurate figures, but if the two points are not aligned somewhat vertically or horizontally, estimating can be more difficult.

Fortunately, Modeler provides what is best described as a virtual tape measure. With it, you can measure the distance between any two points in any view.

To use the Measure tool:

Choose **Construct** > Utility: **Measure** > **Measure Tool** and then drag between the two positions you want to measure between. The information display in the bottom left corner will tell you the exact distance. You can reposition either end by dragging it. To reset, click in a non-active part of the interface.



Measuring distance between horns

In a similar manner you can also measure angles.

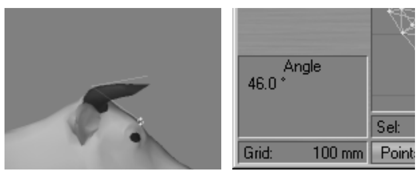
To use the Angle Tool:

1 Choose **Construct** > Utility: **Measure** > **Angle Tool**.

2 Place your pointer at the vertex of the angle you wish to measure and drag the LMB out to form one side of the angle. A line will extend from the drag point.

3 Release the mouse button. You can reposition the end of the line by dragging it; however, you cannot move the starting position at this point.

4 Place your mouse pointer near the starting point (but not on top of it) and drag out the other side of the angle. The information display will now tell you the angle between the two lines.



Measuring horn angle

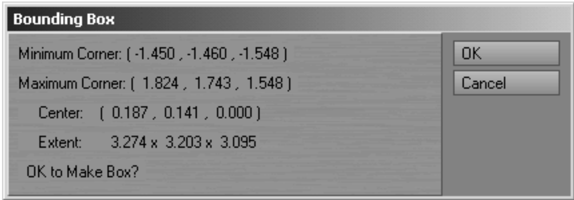
5 Once you release the mouse button, you can reposition the vertex or either end by dragging them. To reset, click in a non-active part of the interface.

NOTE
If you place your mouse pointer on top of the starting point in step 4, the starting point will become the other end to the angle and you will drag out the vertex.

Both of these tools can be used in any viewport to measure any angle in 3D space.

The BoundingBox Command

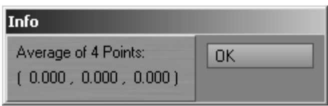
The BoundingBox command (**Construct** > Utility: **Measure** > **BoundingBox**) displays size and center-position information for foreground layers. *Clicking **OK** will replace the layer(s) with a corresponding bounding box*. Use this command to create stand-ins for complex objects, which can be replaced for the final rendered animation.



BoundingBox dialog

The PointCenter Command

The PointCenter command (**Construct** > Utility: **Measure** > **PointCenter**) will display a requester that gives you the XYZ coordinates of the center of the object in the foreground.



PointCenter dialog

LSCRIPT

To learn how to create and edit LScripts, see the documentation included on the CD. They are treated as plug-ins and can be installed in the same manner plug-ins are. You can run them just like any other plug-in command.

ADDITIONAL

Plug-in and LScript commands that do not have a default menu location will appear in the catchall **Construct** > Utility: **Additional** pop-up menu.

chapter 26

Detailing Geometry

Chapter 26: Detailing Geometry

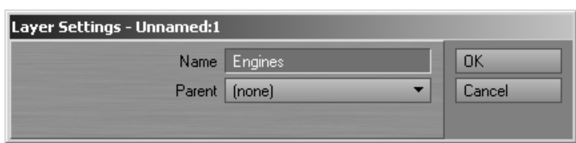
The Detail menu contains a variety of commands. Many are covered elsewhere in the manual.

The Pivot Point Tool

Generally, the pivot point is the positional focal point of the layer's geometry when animated in Layout. By default, it is located at the Origin. Use the Pivot Point tool (**Detail** > Layers: **Pivot**) to move the pivot point. (See Chapter 7 for more information.)

Layer Settings

Choose **Detail** > Layers: **Layer Settings** to bring up a dialog to set the current layer name and parent. (See Chapter 20 for more information.)



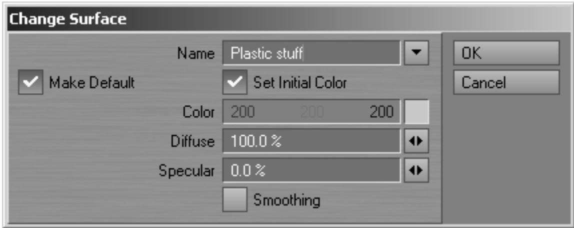
Layer Settings

Flatten Layers

Choose **Detail** > Layers: **Flatten Layers** to flatten multiple foreground layers into a single layer in one step.

Change Surface

Choose **Detail** > Polygons: **Surface** to bring up the Change Surface dialog to set the surface name and basic attributes for the selected polygons. (See Chapter 31 for more information.)



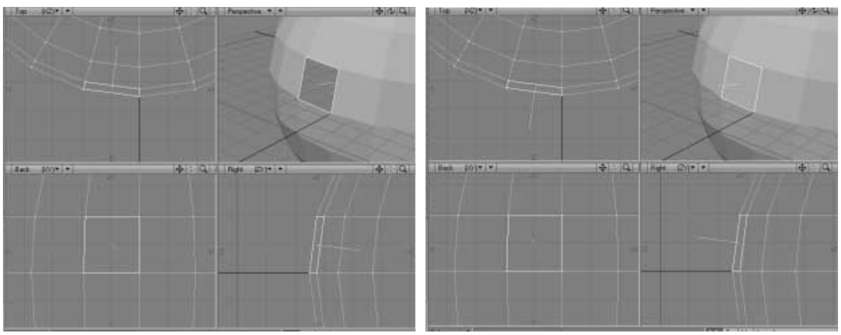
Change Surface dialog

FLIP POLYGONS

Often as you create polygons or use various tools with single-sided polygons, their surface normals will point in the wrong direction. Modeler provides some tools to help you change their direction.

To flip polygons:

Simply select the polygon(s) and choose **Detail** > Polygons: **Flip**. This action reverses the direction of the surface normal for the selected polygons.

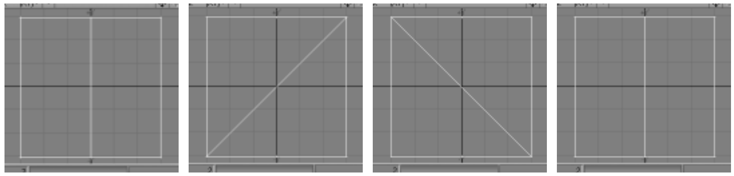


NOTE
Polygons facing the wrong direction is a common problem and one you should get in the habit of checking for regularly.

This command also reverses the direction of selected curves.

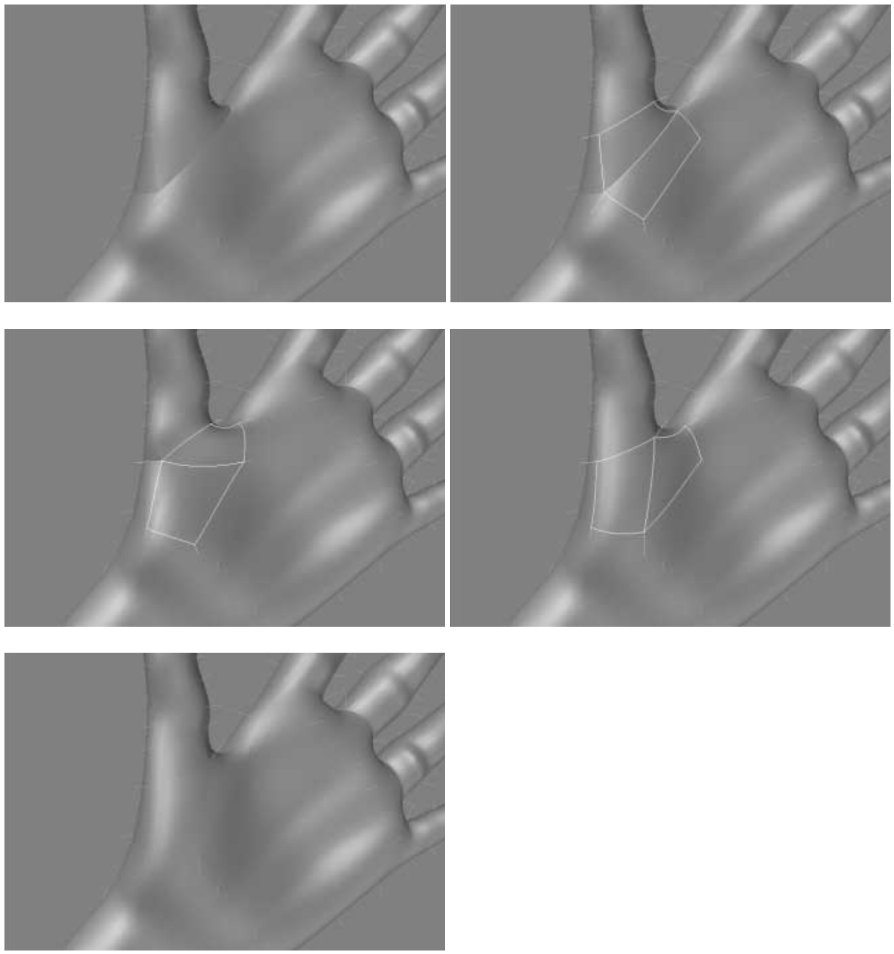
SPIN QUADS

The Spin Quads command (**Detail** > Polygons: **Spin Quads**) can be applied to adjacent four-point polygons (i.e., quads) that share an edge. It merges the two polygons together and then splits them using a different set of opposing polygons. If you apply Spin Quads three times, you'll be right back where you started.



Applying Spin Quads three times to a pair of polygons

So what good is this? Well, it can be quite handy fixing areas on a SubPatch object where a seam just doesn't look right.

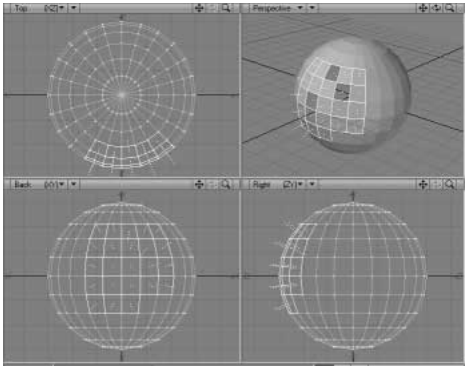


ALIGNING POLYGONS

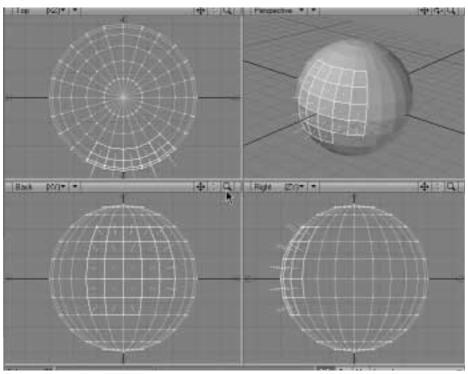
Modeler has another command called Align Polygons (**Detail** > Polygons: **Align**) that *attempts* to automatically make all polygons face the correct direction. After using, you may still need to use the Flip Polygons command (**Detail** > Polygons: **Flip**) before you are done.

To align polygons:

1 Select an area that includes the polygons needing to be flipped, as well as a large number of polygons facing the correct direction. Selecting polygons that share points and edges helps.



2 Choose **Detail** > Polygons: **Align**. Verify that the polygons were flipped as desired. By its very nature, this command sometimes requires some trial and error.



MAKE 2 SIDED

Sometimes you'll actually want double-sided polygons, like when you need to go inside a hollow object. Most of the time, you can get away with simply using the **Double-sided** surface attribute on the Basic tab of the Surface Editor. However, if you need to actually model the double-sided polygons, it is pretty simple. Just choose **Detail** > Polygons: **Make-2 Sided**.

It is also possible to make double-sided polygons manually using the following procedure:

1 Select the target polygons (even if it is the entire object).

2 **Copy** (c) the selection.

3 **Paste** (v) the selection.

4 Choose **Detail** > Polygons: **Flip** to flip the (still) selected polygons.

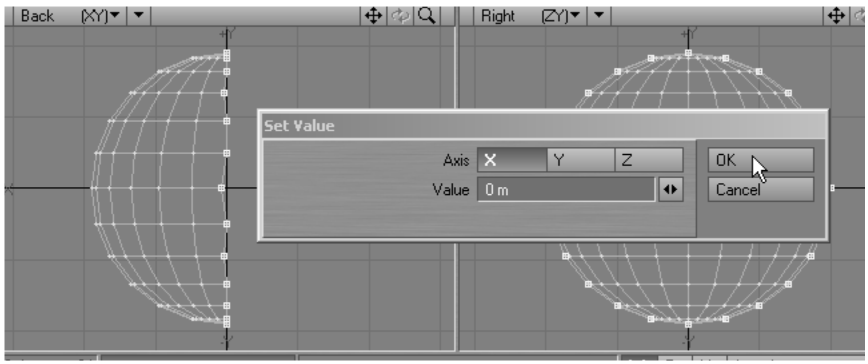
5 Choose **Construct** > Reduce: **Merge Points** to merge points.

CHANGE SKETCH COLOR

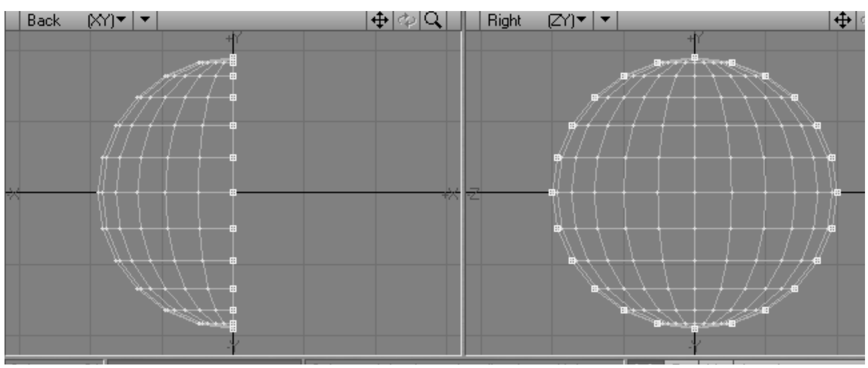
When using the **Sketch** mode **Rendering Style** in a viewport (see Chapter 27), you can change the polygon shading color by choosing **Detail** > Polygons: **Sketch Color**.

SET VALUE

The Set Value command (**Detail** > Points: **Set Value**) sets the X, Y, or Z position values for the selected points or polygons. This is a great tool for aligning all of the points along a strict axis. You may find a need for this when you are trying to merge to halves of a mirrored object together and are having trouble merging all points along the mirroring axis.



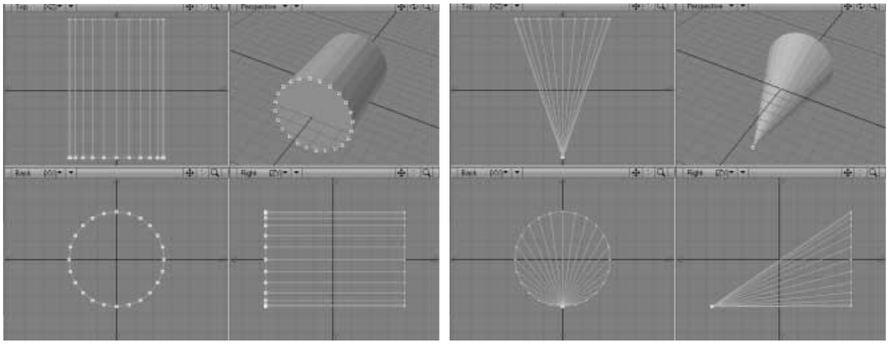
Before using Set Value



After using Set Value

WELDING COMMANDS

The Weld command (**Detail** > Points: **Weld**) is similar to the Merge Points command (see Chapter 25) except the selected points are welded to the last point selected.



Welding one end of a cylinder to a single point

The Weld Points To Average command (**Detail** > Points: **Weld Average**) is similar, but averages the distance between points before welding.

Unwelding Points

The Unweld command (**Detail** > Points: **Unweld**) lets you create multiple copies of the selected points so that none are shared by two polygons at once—effectively reversing an executed Weld command.

CURVE COMMANDS

Curves and their special editing tools are discussed in Chapter 22.

SKELEGON AND METABALL COMMANDS

Skelegon commands are discussed in Chapter 10. Metaballs are discussed in Chapter 29.

chapter 27
Display Menu

Chapter 27: Display Menu

The Display menu contains functions to help you in adjusting viewports, selecting geometry, and changing visibility options.

VIEWPORTS MENU GROUP

The following options help you change and configure your viewports.

Zooming

Modeler provides many ways to zoom around your viewports. You're gonna zoom, zoom, zooma zoom. 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 **Display** > Viewports: **Zoom In** / **Zoom Out**.) Holding the SHIFT key while pressing either key (i.e., the > and < keys) will double the zoom amount.

Another way to adjust magnification is to use the Zoom tool (**Display** > Viewports: **Zoom**). You can drag out a rectangular box using the mouse and the view will automatically zoom in so the rectangular box fills the view window.

The Magnify tool (**Display** > Viewports: **Magnify**) provides yet another way to interactively zoom in and out. This feature lets you smoothly zoom in or out on your pointer position by dragging your mouse left and right.

If you have the titlebars visible, you can drag the zoom drag button  to zoom.

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.)

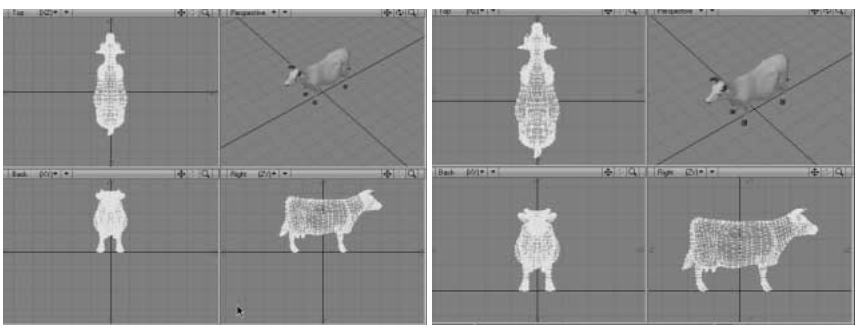
Panning

Each viewport is a window to a larger world. You can pan the viewport beneath your mouse by using the ARROW keys—one grid square for each press. Holding the SHIFT key will move the view four grid squares. Holding the ALT key will *nudge* by one-tenth of a grid square or the (**Fixed**) **Snap Value** amount on the Units tab of the Display Options panel (**Display** > Viewports: **View Options**), if that option is active. ALT + SHIFT will nudge ten times this amount. (It may help you to think of this as moving a window around as opposed to moving the object.)

You may also pan a viewport by holding the ALT key down and dragging a viewport. (This feature is also activated with the Pan tool (**Display** > Viewports: **Pan**).) If you have the titlebars visible, you can drag the move drag button to pan.

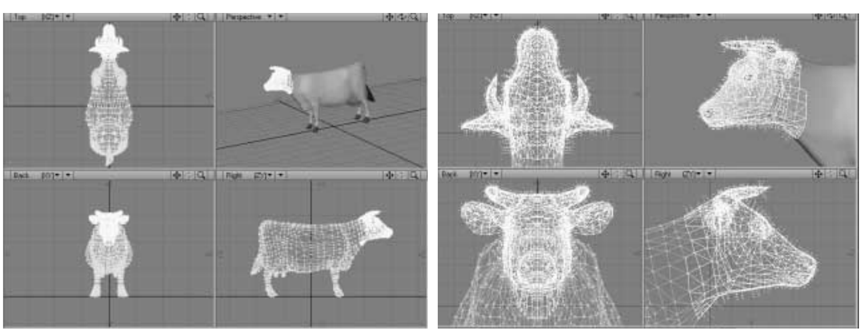
Automatic Pan and Zoom

Certain commands will automatically fit objects into the viewport. To fit the object in all viewports, use the A key, a shortcut for **Display** > Viewports: **Fit All**. You'll often want to use this to *center-up* an object when it is first loaded.



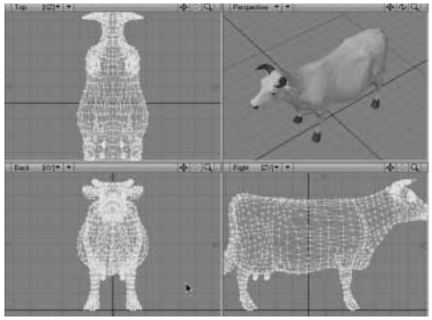
Before and after using Fit All

Press SHIFT + A (or choose **Display** > Viewports: **Fit Selected**) to fit only the *selected* parts of an object into all viewports.



Before and after using Fit Selected

The Fit One View command (CTRL + A) will fit the object in the viewport under the mouse pointer into that viewport.



After using Fit One View. Mousepointer in Back viewport

Pressing the G key will center your view around the mouse pointer. This is a very quick way to maneuver around. *Remember this shortcut!*

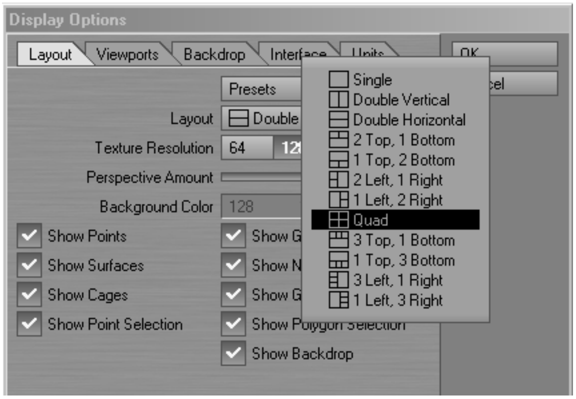
NOTE
Other viewports will also change to stay in sync with the window you are changing (unless they are Center/Zoom independent, discussed later).

VIEW OPTIONS

Modeler's interface is highly configurable, so you can set things up to be most efficient for different users or different uses. You can change the arrangement of viewports as well as change the display characteristics for each viewport independently. Choose **Display** > Viewports: **View Options** to bring up the Display Options panel.

Display Options Panel, Layout Tab

Layout options control the global arrangement of the viewports, the number of viewports and some other global display settings. Use the **Presets** pop-up menu to quickly set common layout configurations (**Quad** is the default setting). Selecting a Preset will affect the **Layout** setting, as well as settings on the Viewports tab.



Display Options Presets

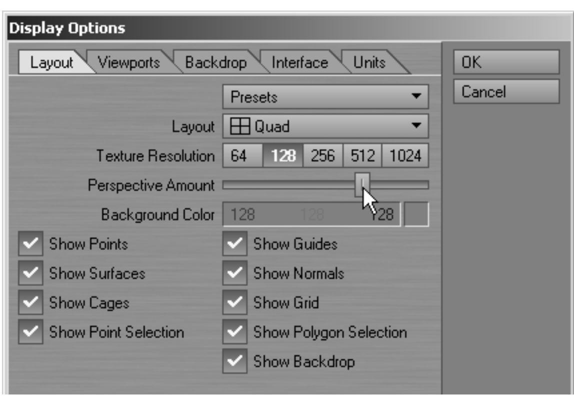
If you want to set up your own custom layout, make a selection from the **Layout** pop-up menu. The icon to the left of the description will give you an idea of how the viewports will be arranged. Later, you will see that you have total control over what is in each viewport. In fact, each **Layout** option will remember the Viewport settings as they were the last time *that* **Layout** option was used.

Texture Resolution

The **Texture Resolution** setting determines the resolution to use for displaying textures in viewports that have their **Rendering Style** set to **Texture**. Higher settings increase the detail of image textures displayed in viewports, but increase memory usage and display refreshing time.

Perspective Amount

The amount of perspective in the perspective view can be varied by changing the **Perspective Amount** slider on the Layout tab. You can go from a very wide-angle to a flat nearly orthogonal view. The setting is global and affects all the perspective views the same way.



Adjusting Perspective Amount

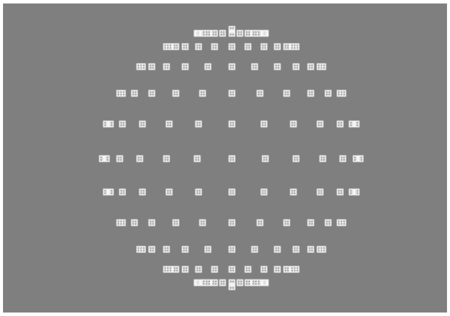
Background Color

Use the **Background Color** selector to change the color that appears in the background for viewports using a shaded display. (See Chapter 3 for information on using color selectors.)

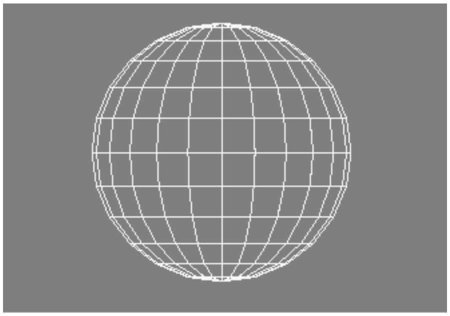
Show Options

The various *Show* visibility options let you independently set what you want to see globally in your viewports.

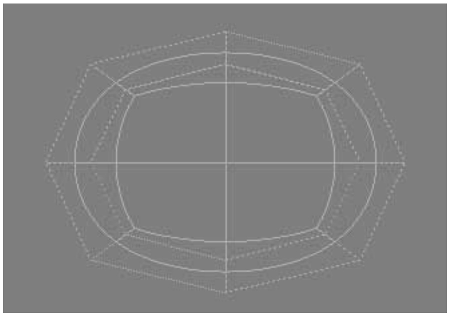
Points Points appear as small dots.



Surfaces Surfaces are the polygon (or SubPatch) surfaces.

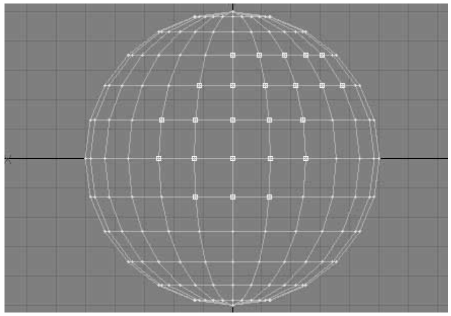


Cages A cage is the outline that connects all of the control points when you edit a SubPatch object.

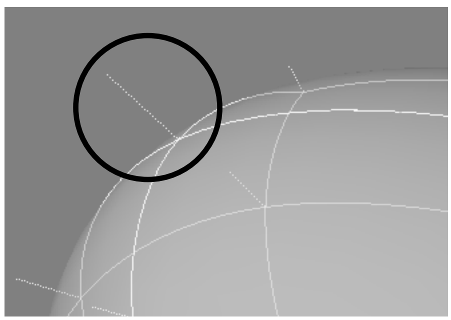


Point Selection

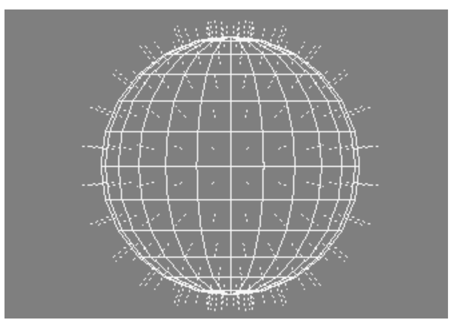
Points appear highlighted when this option is active, if you select points.



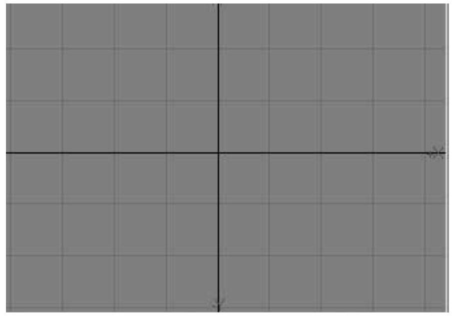
Guides Guides are the dotted-lines that extend from the surface (patch vertex) to the control points on the cage (if visible) when you edit a SubPatch object.



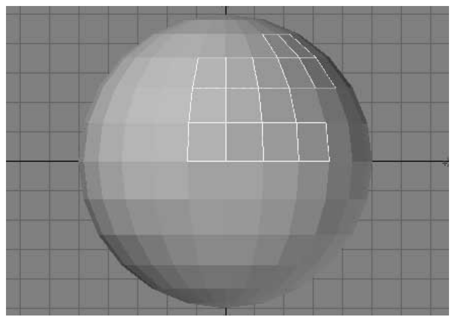
Normals Normals are dotted lines that extend perpendicularly from selected (planar) polygons.



Grid The grid is the background reference grid.



Polygon Selection Polygons appear highlighted when this option is active, if you select polygons.



Normals are off here

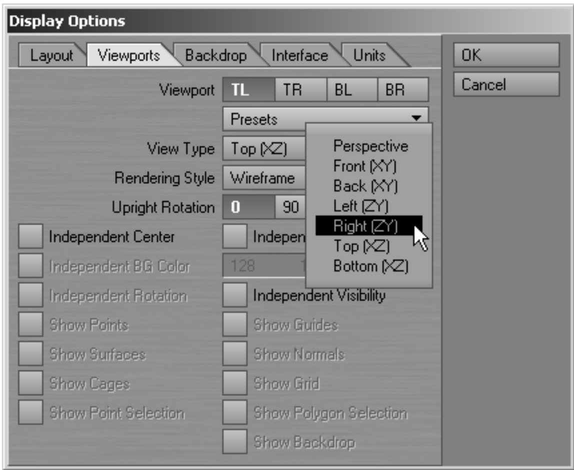
Backdrop A Backdrop is an image loaded for viewports on the Backdrop tab.

NOTE
These visibility options can be overridden independently for each viewport on the Viewports tab.

Display Options Panel, Viewports Tab

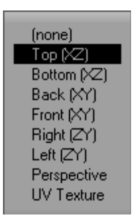
Viewport options let you change how objects are displayed in each viewport. To change the settings for a viewport, first select it using the **Viewport** buttons. For a quad layout, the viewports are top-left (TL), top-right (TR), bottom-left (BL), and bottom-right (BR).

Use the **Presets** pop-up menu to quickly set the options on this tab for the selected viewport region. The selections are listed by view types, but will affect all of the settings on this tab.

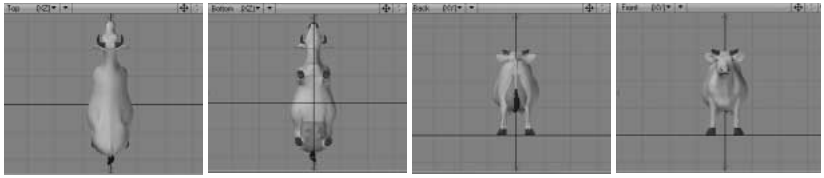


Presets pop-up menu

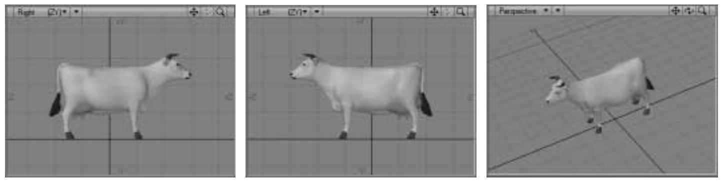
The **View Type** pop-up menu determines the editing axes you want to use for the selected region. For the orthogonal settings, the names generally indicate the viewing perspective. **Back (XY)**, for example, lets you edit along the X and Y axes. This means you are looking along the Z axis. Since it is called *Back*, that means your perspective is from the back (i.e., negative side) of the Z axis, looking toward the positive side. **UV Texture** is an entirely different animal (that is, not a cow). (See Chapter 28 for information on UV Texture maps.)



View Type pop-up menu



Left to right: Top, Bottom, Back and Front



Left to right: Right, Left and Perspective

There are four Display (group) commands assigned by default to certain keyboard shortcuts which affect the viewport beneath your mousepointer. SHIFT+F1 toggles the view between Back and Front, SHIFT+F2 toggles between Top and Bottom, SHIFT+F3 toggles between Right and Left, and SHIFT+F4 toggles between Perspective and UV.

Perspective

You may edit in a Perspective viewport, just as you would in any other, but you may also rotate your view perspective. This is Modeler's *virtual trackball,* which enables you to rotate the object without affecting its orientation in the other three edit windows. It effectively provides you with three axes of rotation. While holding down the ALT key, you can perform these actions:

- Rotate around the X axis (pitch) by dragging up or down directly across the circle.
- Rotate around the Y axis (heading) by dragging left or right directly across the circle.
- Rotate around the Z axis (bank) by dragging to the left or right around the perimeter of the circle.

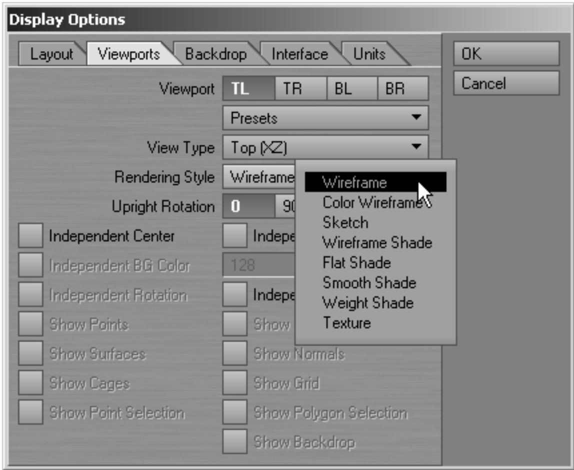
If the titlebars are visible, you can drag the rotate drag button  with your LMB to rotate. If you use the RMB, the viewport will rotate around its perpendicular axis. Holding the CTRL key (or using your MMB) will cause rotations to snap to 15-degree increments.

HINT

Imagine you are manipulating an imaginary ball with your mouse when you manipulate the Preview window.

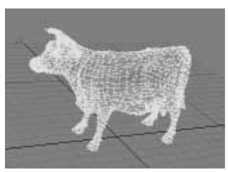
Rendering Style

The **Rendering Style** pop-up menu determines the style of display you want to use for the selected region.

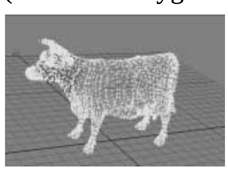


Rendering Style menu

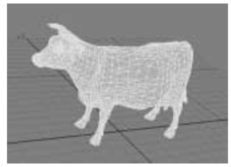
Wireframe Although the **Wireframe** is arguably the most limited display mode, it is the most commonly used **Rendering Style** because of its ease in viewing, selecting, and unselecting points and polygons.



Color Wireframe This is nearly the same as **Wireframe**, except the polygon edges are drawn using their Sketch color (**Detail** > Polygons: **Sketch Color**).

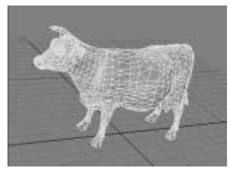


Sketch **Sketch** shows an object in a combined wireframe and flat-shaded view. All polygon edges are drawn, but faces are also visible. This mode does not account for surface settings, however. The polygon edges are always drawn in white and faces are gray. Background layers are visible.



You can change the polygon shading color in this mode by choosing **Detail** > Polygons: **Sketch Color**.

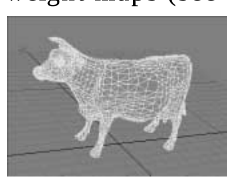
Wireframe Shade **Wireframe Shade** is a smooth-shaded mode that overlays the wireframe lines.



Flat Shade In **Flat Shade** mode, the object is shown as a flat-shaded solid. This mode supports some surface settings, but not smoothing.

Smooth Shade **Smooth Shade** is a smooth-shaded mode that supports some surface settings, such as Color, Diffusion, Specularity, Glossiness, Smoothing, and Double-sided.

Weight Shade **Weight Shade** provides visual feedback for editing weight maps (see Chapter 28).



Texture — **Texture** is similar to **Smooth Shade**, but it also shows images mapped to surfaces.

Upright Rotation

The **Upright Rotation** pop-up menu lets you rotate a viewport clockwise by 0, 90, 180, and 270 degrees.

Titlebar Shortcuts

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 titlebar. The left-most pop-up shows the current **View Type**, clicking it allows you to choose another. The pop-up menu just to its right—it has only a down arrowhead—lets you choose another **Rendering Style**.

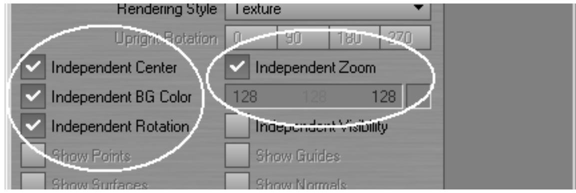
View Type and Rendering Style pop-up menus

Note that if you have **UV Texture** selected as the **View Type**, the **Rendering Style** pop-up menu will instead list all loaded images. Select one to load it into the backdrop of that viewport.

List of loaded images on pop-up menu when using UV Texture view

Independent Options

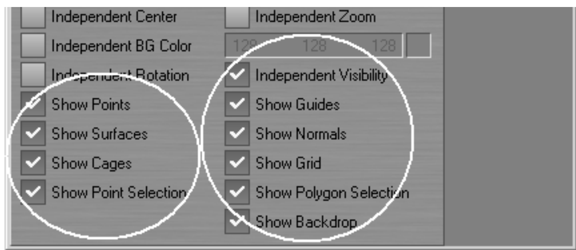
The Independent options let you make certain viewport characteristics independent from other viewports. Changes to the selected characteristic do not affect that characteristic in other viewports and vice versa. For example, if a viewport uses **Independent Zoom**, zooming in it does not affect the zoom of other viewports. You must have your pointer over the viewport to affect an independent viewport. **Independent Rotation** is only applicable to perspective views. Normally, it is enabled. If you have multiple perspective views with this option *disabled*, they will move in unison when you rotate a view.



Independent options

Independent Visibility

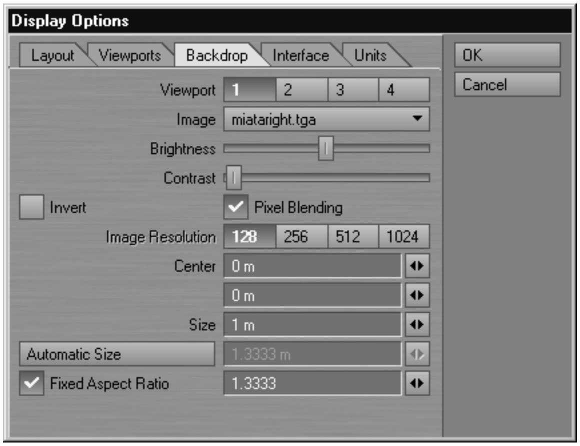
The lower portion of the panel contains the independent visibility options. You can individually select which visibility options are independent. To use them, you must also activate the **Independent Visibility** option. This option lets you quickly enable/disable this feature without losing the state of each setting.



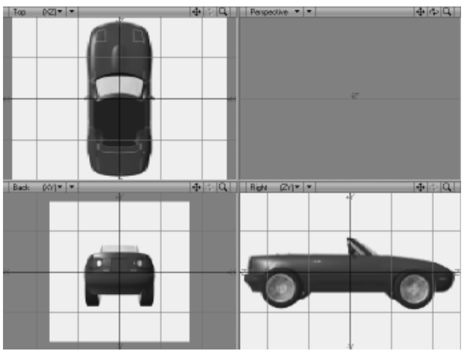
Independent visibility options

Display Options Panel, Backdrop Tab

On the Backdrop tab, you can add full-color backdrop images that act as reference guides when you build objects. They let you model much like tracing a picture using tracing paper. To use backdrops, select the **Viewport** and the desired image from the **Image** pop-up menu.



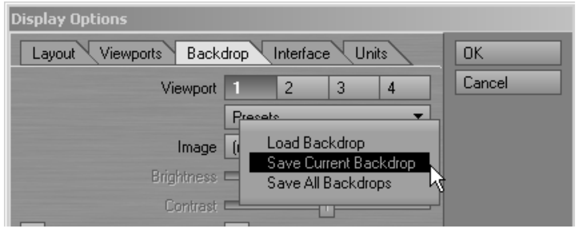
Display Options Panel, Backdrop Tab


Backdrops added to orthogonal viewports

NOTE

If you are using a UV Texture viewport, you can choose a backdrop using any loaded image from what is normally the Rendering Style pop-up menu on the viewport's titlebar.

From the **Presets** pop-up menu, you can save the settings for the selected **Viewport** to a file by choosing **Save Current Backdrop**. You can load this file later using **Load Backdrop** for any selected **Viewport**. You can also save the settings for all viewports by choosing **Save All Backdrops**. When this file is loaded later—using **Load Backdrop—**, the individual viewport settings are restored.



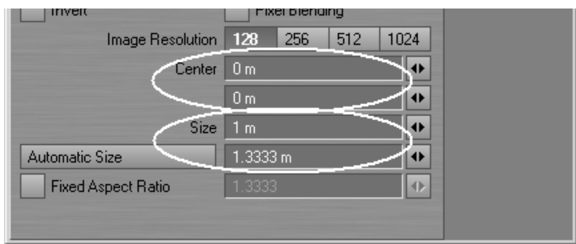
Saving and loading backdrops

NOTE
The backdrop images for the current default layout for each view configuration are cached as long as possible. Also, backdrops are not saved in config files or view presets.

You can also adjust **Brightness** and **Contrast**. You can invert the colors by activating the **Invert** option. If you want to blend pixels for closeup work, activate the **Pixel Blend** option. **Image Resolution** determines the accuracy of the displayed image.

Aspect and Video images

The two **Center** and **Size** input fields are horizontal (top field) and vertical (lower field). Clicking **Automatic Size** will enter values that fit the defined image into an implied bounding box that surrounds visible geometry.



Backdrop Center and Size fields

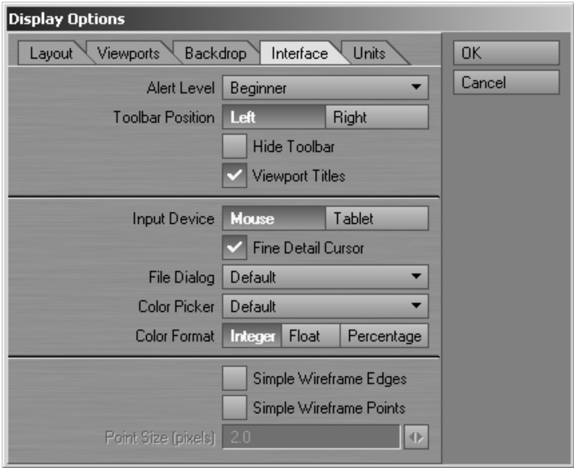
If you activate the **Fixed Aspect Ratio** option, you can enter a <u>frame</u> aspect ratio in the input field. Then, you need only specify a horizontal size—the vertical size is computed automatically.

If your images originated from video, you need to account for the pixel aspect ratio, since video pixels are not square like computer pixels. For example, a D1 NTSC image is 720 x 486 pixels; however, the frame aspect ratio is not 1.48 (720/486) as you might expect. The width of a D1 NTSC pixel is 90 percent of the width, that is, a <u>pixel</u> aspect ratio of .9. As such, the true frame aspect ratio is 1.333 (720/486*.9).

NOTE
The frame aspect ratio of a D1 PAL image is also 1.333 (720/576*1.067).

If you do not account for this difference, your models may appear slightly stretched when ultimately viewed on videotape. A way to fix this is to scale the object after the fact.

Display Options Panel, Interface Tab



Display Options Panel, Interface Tab

The **Alert Level** affects how error, warning and informational messages are displayed. (See also Chapter 6.)

The **Toolbar Position** setting determines if the main toolbar appears on the **Left** or **Right** side. Activate **Hide Toolbar** to make it hidden and maximize your screen real estate. You can also choose **Modeler** > **Interface** > **Hide Toolbar On/Off** to toggle this setting.

The **Viewport Titles** option turns the titlebars, which appear above each viewport, on or off.

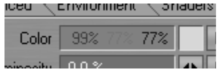
Viewport titlebars

Use the **Input Device** buttons to select the type of input device you are using.

The **Fine Detail Cursor** option, when active, makes your mousepointer use the main crosshair pointer at all times instead of changing when the various tools and selection functions are used.

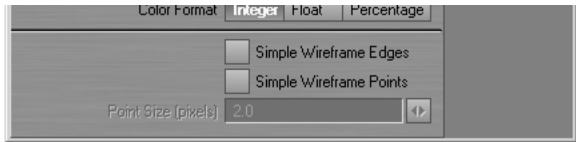
The **File Dialog** and **Color Picker** pop-up menus let you use custom LightWave dialogs for file loading/saving and picking colors. (See Chapter 3 for more information.) Selecting **Default** will use your standard system dialogs. The custom dialogs provide many additional features not available with the standard system dialogs.

The **Color Format** setting determines the scale used where the color selector appears. **Integer** uses values 000 to 255, **Float** uses .00 to 1.00 and **Percentage** uses 0% to 100%.



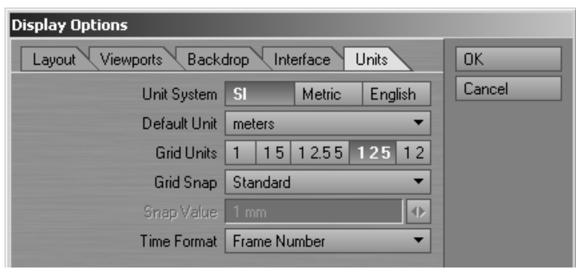
Example color selection setting

The **Simple Wireframe Edges** option turns off polygon offsetting for the *sketch-like* display modes. This may fix display problems on some videocards. The **Simple Wireframe Points** setting uses OpenGL's own points for wireframe points and selected points at a user-specified size. These points can draw much faster than the standard points.



Simple Wireframe Edges and Simple Wireframe Points options

Display Options Panel, Units Tab



Display Options Panel, Units Tab

The **Unit System** determines the units of measurement that are used and displayed by the Modeler screen.

SI SI is the International System of Units (SI is the abbreviation of the French “Le Système International d’Unites.”) Unit measurements in Modeler will now use a base system of meters. Grid sizes and distances can be measured in megameters, kilometers, meters, millimeters, micrometers, and nanometers.

Metric The Metric System is the same as **SI** with the addition of centimeters.

English The English System uses miles, feet, and inches.

NOTE
We recommend you use the SI or Metric systems to model objects, since a system based on 10 is often much easier to use for purposes such as applying textures and adjusting object movement.

If you input a value that uses a different unit of measurement than that the default, LightWave will convert it on the fly. For example, you may be using meters, but typing in “5 ft” will convert to “1.524 m.”

Default Unit

Use the **Default Unit** pop-up menu to set an assumed measure when none is given for distance values.

NOTE
If you use a metric **Unit System**, you should set the Default Unit to **meters**.

For information on units of measurement, please see the Appendix.

Grid Units

Grid Units affects the zoom *step amount* but has no effect on the actual size of the object. You will likely find that **1 2.5 5** and **1 2 5** are your most commonly used grid unit settings.

1 The grid resizes in values that begin with 1, as in 10m, 1m, 100mm, 10mm, etc.

1 5 The grid resizes in values that begin with 1 or 5, as in 1m, 500mm, 100mm, 50mm, 10mm, etc.

1 2.5 5 The grid resizes in values that begin with 1, 25, or 5, as in 1m, 500mm, 250mm, 100mm, 50mm, etc.

1 2 5 The grid resizes in values that begin with 1, 2, or 5, as in 1m, 500mm, 200mm, 100mm, 50mm, etc.

1 2 The grid resizes in values that begin with 1 or 2, as in 1m, 200mm, 100mm, 20mm, 10mm, etc.

Grid Snap

The **Grid Snap** setting forces point creation and item movement to be limited to a specific increment.

None **None** deactivates grid snap, so that items move freely and are not constrained by any grid intersections.

Standard **Standard** sets the grid snap to one-tenth of the current grid size.

Fine **Fine** makes the grid snap as small as possible for the current zoom level. This will be typically two- to five-times smaller than using **Standard**.

Fixed **Fixed** lets you specify the increment of movement in the **Snap Value** input field. The grid will resize itself to use the value you specify. This also fixes the grid’s graticule on screen to be displayed at that increment.

Time Format

The **Time Format** setting allows you to set the display format for time values, like the **Frame Slider Label** option on Layout's General Options tab of the Preferences panel. If the Hub is active, Layout and Modeler settings will be kept in sync.

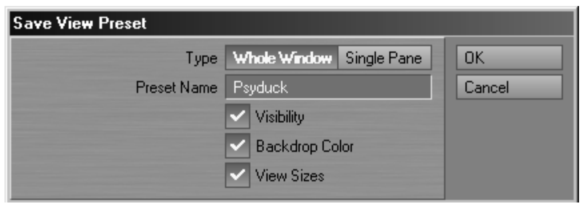
PRESETTING VIEWS

As you work with Modeler you will no doubt find that you use the viewports in particular view configurations. Luckily, you can customize Modeler's display to fit your work habits by assigning workspace view settings to the keys on your numeric keypad.

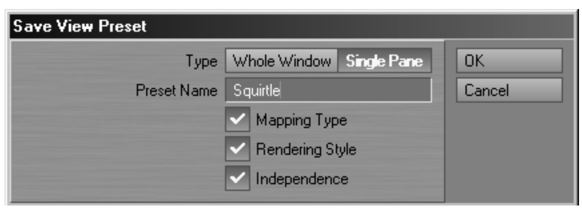
To assign views to keys:

1 Turn on your NUMLOCK key.

2 Hold the CTRL key and press one of the numbers 1 through 9 on the numeric keypad.



Whole Window options



Single Pane options

You can choose to save either the parameters for a **Single Pane** (the viewport beneath your mouse pointer when you activated the function) or the **Whole Window** (all viewports). The available options differ depending on whether you select **Single Pane** or **Whole Window**. The options relate to the corresponding settings on the Display Options panel. (Note: **Mapping Type** (**Single Pane** mode) is called **View Type** on the Display Options panel.)

Keys set using **Single Pane** will apply those changes to the viewport under your mouse pointer. <u>If your mouse is not over a viewport, all viewports are affected.</u>

If you name your assignment and select **Single Pane**, the name appears in the **Presets** pop-up menu on the Viewport tab of the Display Options panel, discussed previously. If you select **Whole Window**, the name appears on the **Presets** pop-up menu on the Layout tab.

Full-screen Toggle Feature

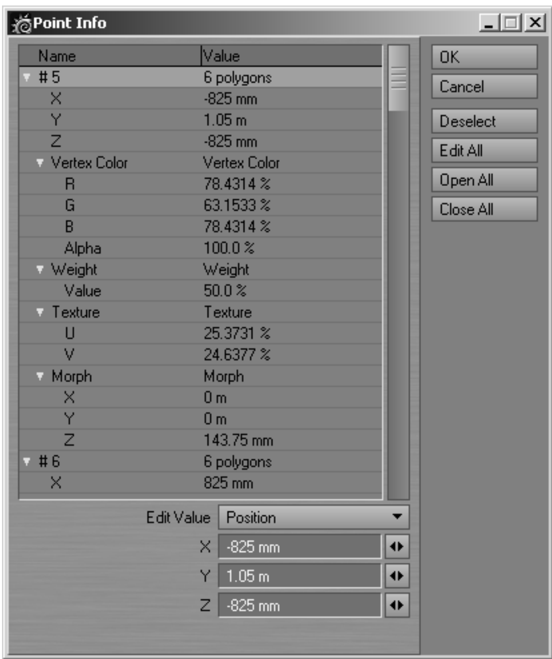
By default the NUMKEY 0 key toggles the viewport under your mouse pointer between full screen and the current viewport settings.

SELECTION MENU GROUP

The following options help you with selection and other related tasks.

Point Information

Select one or more points and choose **Display** > Selection: **Info** to bring up the Point Info panel. From here you can get detailed information on individual points and even edit the data on a point-by-point basis.



Point Info panel

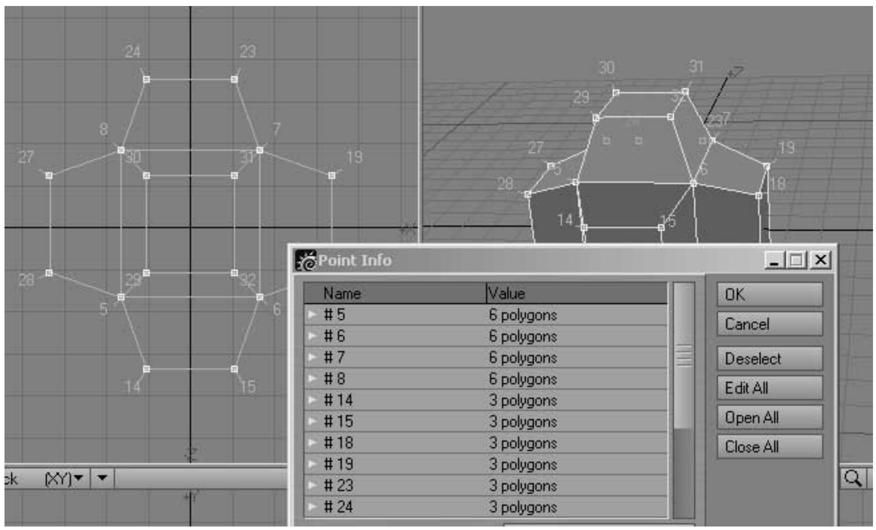
Displayed is the number of polygons the points is associated with, and its XYZ position. If the point has any vertex maps, those are listed as well. To help you locate the related points, point numbers will become visible in your viewports corresponding to entries in the list.

You can change any of the selected point(s) settings, including vertex map data, using the settings at the bottom of the panel. For Vertex Color, you can simply set the new color and Alpha percentage. For Weight maps, adjust the weighting percentage. For UV Textures, you must enter the specific U and V values. For Morph maps, you must enter the amount of XYZ position offset.

To change what type of data is being edited for the selected entries, use the **Edit Value** pop-up menu. Alternatively, you can select the type of data with your mouse in the list.

You can have multiple selections, which will show “(mixed)” when the settings differ for the selected points. Changing settings will affect all selected points.

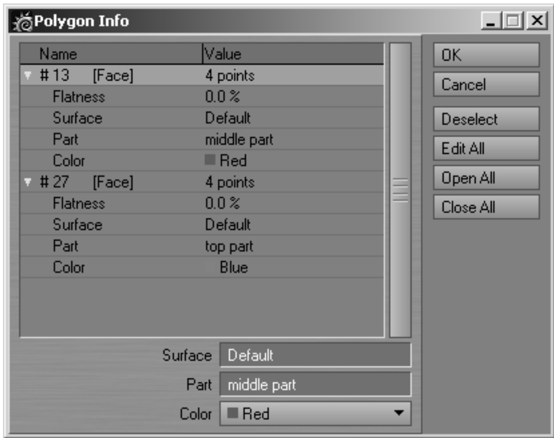
The **OK** button will accept any changes and close the panel. Click **Cancel** to forget any changes (even though they are reflected in your viewports) and close the panel. **Deselect** will unselect any points selected on the panel and remove them from the list. **Edit All** will select all entries. (You can also select and unselect entries with your mouse.) **Open All** will uncollapse all of the entries and show their details. **Close All** does the opposite.



Point numbers visible for selected polygons

Polygon Information

Select one or more polygons and choose **Display** > Selection: **Info** to bring up the Polygon Info panel. From here you can get specific information about each selected polygon and even edit the data.

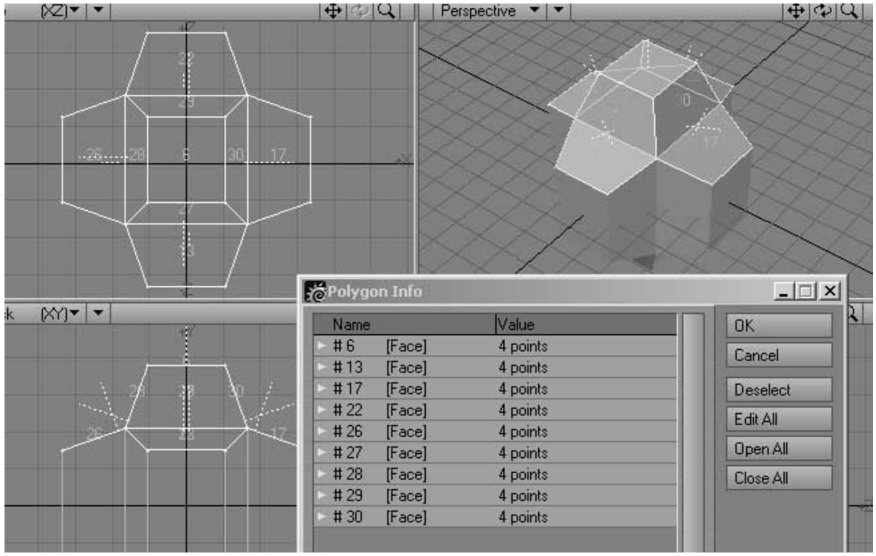


Polygon Information panel

Displayed for each polygon is the type of polygon (e.g., Face, Skelegon, SubPatch, Metaball, etc.), number of points, the flatness percentage (see Chapter 21, "Non-planar Polygons"), surface name, part name, and sketch color (see Chapter 26). To help you locate the related polygon, polygon numbers will become visible in your viewports corresponding to entries in the list.

You can change the surface name, part name and sketch color for the selected polygon(s) using the settings at the bottom of the panel. You can have multiple selections, which will show "(mixed)" when the settings differ for the selected polygons. Changing settings will affect all selected polygons.

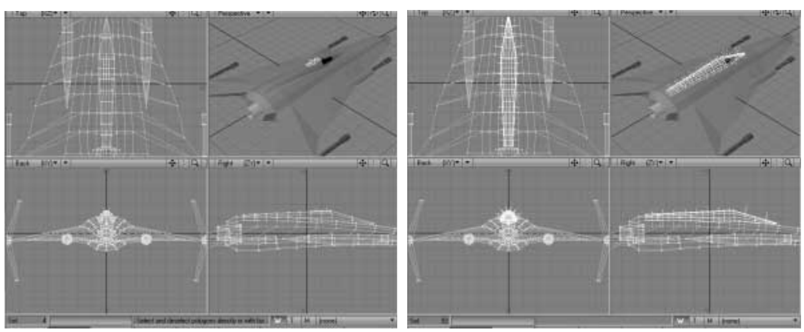
The **OK** button will accept any changes and close the panel. Click **Cancel** to forget any changes (even though they are reflected in your viewports) and close the panel. **Deselect** will unselect any polygons selected on the panel and remove them from the list. **Edit All** will select all entries. (You can also select and unselect entries with your mouse.) **Open All** will uncollapse all of the entries and show their details. **Close All** does the opposite.



Polygon numbers visible for selected polygons

Expanding and Shrinking Selection

The Select Connected command (**Display** > Selection: **Sel Connect**) automatically selects all points or polygons that are connected to the currently highlighted points or polygons. This is a powerful tool and one you should add to your memory banks. An object that is surrounded by others may not be easily selected using a volume, and may have several surface names (making name selection tedious).



NOTE
Polygons that have overlapping points, but don't share any of those points, will not be affected. Use the Merge Points command (**Construct** > Reduce: **Merge Points**) to make the edge shared.

The Expand command (**Display** > Selection: **Expand**) is a single-step version of the Select Connected command. All points/polygons that are adjacent to selected elements are themselves selected. The Contract command (**Display** > Selection: **Contract**) is similar, except that any elements that are adjacent to unselected elements are deselected.

To reverse the state of selected and unselected points or polygons, simply choose **Display** > Selection: **Sel Invert**.

GROUPINGS

Polygon Parts

Parts are the polygon version of point selection sets. (One distinction, however, is that a polygon can be assigned to only one part name, while a point can be a member of multiple selection sets.) You can quickly select a user-defined set of polygons by name in much the same way you can with surface names. This feature operates independently and in parallel with surface naming.

To name polygon parts:

1. Select a group of polygons.
2. Click **Display** > Selection: **Grouping** > **Change Part Name**. Enter the desired name in the **Name** field and click **OK**.

You can add to an existing part group by selecting the additional polygons and then selecting the group's name from the pop-up menu.

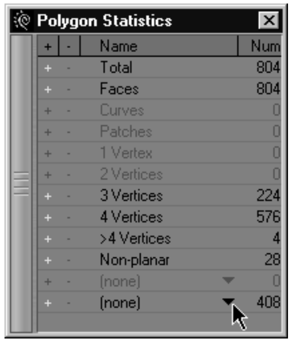
Updating Old Objects

Before the parts feature existed, some users would use surface names to help select and modify geometry, when they really needed only a single surface name for rendering purposes. The Surfaces To Parts command (**Display** > Selection: **Grouping** > **Surfaces to Parts**) can help to convert such objects by automatically converting the surfaces of selected polygons to part names. The selected polygons are renamed using a defined **Surface Name**.

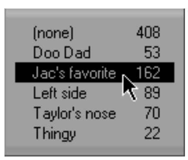
To select parts:

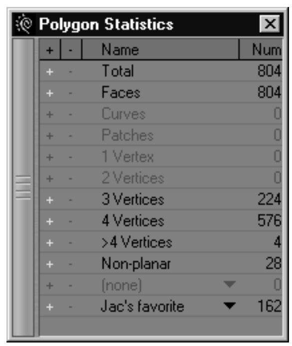
1. In Polygon Selection mode, choose **Modeler** > **Windows** > **Statistics Open/Close** to open the Polygon Statistics panel.

2 Click on the pop-up menu triangle on the last line in the panel. (If it isn't visible, drag the bottom of the panel down.) Select the desired part name.



3 The selected part's name will appear on the line. Click the plus sign (+) to select all of the polygons defined by the part.





Like surfaces, any one polygon can have only one part name. Therefore, if you create a new part that uses polygons from an existing part, those polygons will become associated with the new part only.

Point Selection Sets

You can save groups of points that can be reselected by a user-defined name. Note that a point can be a member of multiple sets.

To create a point selection set:

1 Select a group of points.

2 Click **Display** > Selection: **Grouping** > **Point Selection Sets**. Enter the desired name in the **Name** field and click **OK**.

You can add to an existing set by selecting the additional points and then selecting the set's name from the pop-up menu.

To select points in a selection set:

1 In Points Selection mode, open the Point Statistics panel (**Modeler** > **Windows** > **Statistics Open/Close**).

2 Click on the pop-up menu triangle on the last line in the panel. (If it isn't visible, drag the bottom of the panel down.) Select the desired selection set.

3 The selected set's name will appear on the line. Click the plus sign (+) to select all of the points in the set.

To remove points from a selection set:

1 Select the points you wish to remove.

2 Click **Display** > Selection: **Grouping** > **Point Selection Sets**. Select the set from the pop-up menu.

3 Activate the **Remove Points** option and click **OK**.

You can delete the entire set reference by selecting it and then clicking the **Delete** button. Note that neither deleting operation will physically delete the points.

SELECTION BY VMAPS

Here are some VMap commands that deal with selection.

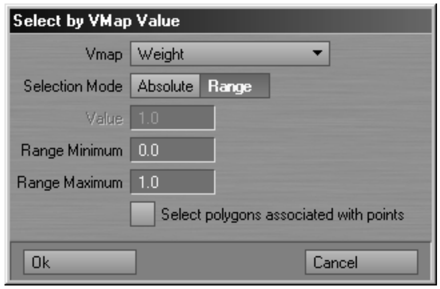
Select by Map

The Select by Map command (**Display** > Selection: **Maps** > **Select by Map**) selects or deselects points that have entries in the chosen **Vertex Map**.

Select By Map

Select By Map Influence

This LScript will allow you to select points in a weight map based on their weight values. To use, first select the target weight map on the **Vmap** pop-up menu. If you want to select a specific value only, enable **Absolute**. If you want to define a range, enable **Range**. You may also select polygons associated with the selected points. This will include any polygon that uses at least one selected point.



Select By Map Influence

Select Polygons from Selection Set

The Select Polygons from Selection Set command (**Display** > Selection: **Maps** > **Select Polygons from Selection Set**) lets you select polygons based on VMap names.

Select Polygons from Selection Set

Use the command while in Polygon edit mode by simply selecting the VMap name in the pop-up menu and clicking **OK**. Any polygon that includes a point with that VMap is selected. If you select **Exclusive**, all of the polygon's points must use the VMap to be selected.

Normally, the new selection replaces the current selection. To add the new selection to your current selection, activate **Keep Current Selection**.

Per-Polygon UV Map Selection Commands

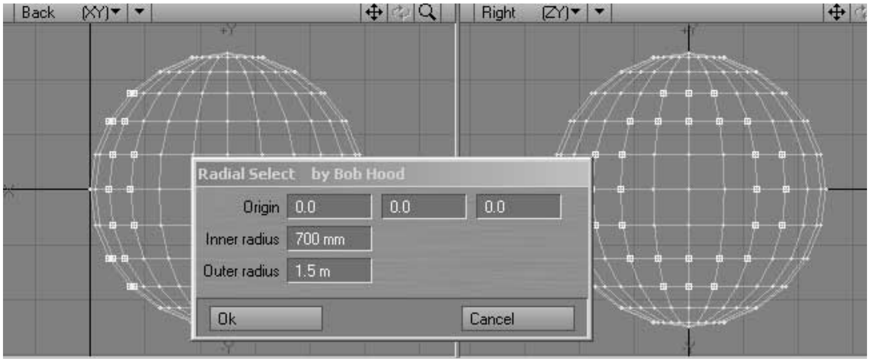
See Chapter 28, "Per-Polygon UV Mapping" for information on the **Select by Polygon Map** and **Select UV Seam** commands in the **Display** > Selection: **Maps** > **Select by Map** menu.

SKELEGON SELECTION COMMANDS

Two commands can help you select skelegons: Select Child Skelegon (**Display** > Selection: **Select Child**), which selects the child of the currently selected skelegon, and Select Parent Skelegon (**Display** > Selection: **Select Parent**), which selects the parent of the currently selected skelegon.

RADIAL SELECT

Radial selection will select the points that fall within a defined radius. **Origin** is the XYZ center of the selection area. Points will be selected if they fall within the **Inner radius** and **Outer radius**.



Radial Select

VISIBILITY MENU GROUP

In some cases, you want to hide the polygons that you don't want edited. This makes it easier to work on parts of a complex object without splitting it up between layers. The polygons are still there, but any editing operations will not directly affect the hidden polygons.

To hide a polygon:

Select some polygons (using any of the previously discussed methods) and then click **Display** > Visibility: **Hide Sel** to hide selected polygons.

You can also hide the unselected polygons by clicking **Display** > Visibility: **Hide Unsel**. Clicking **Display** > Visibility: **Invert** will reverse the hidden states. To reveal hidden polygons, click **Display** > Visibility: **Unhide**.

NOTE
Although you cannot directly affect hidden polygons, you can affect them if you modify visible polygons that share points.

SWAP LAYER STATES

Choose **Display** > Visibility: **Swap Layers** to swap the foreground and background layers. This is a handy command to use with tools that require background objects like Boolean and Drill.

chapter 28
Vertex Maps

Chapter 28: Vertex Maps

As you might expect, every point in an object has independent position information. Points even have rotational information, although you normally don't rotate individual points. A *vertex map* (VMap) is additional data that a user can add onto each point in an object. Any point in an object can have a unique entry (value) for a particular VMap or no entry at all. The VMap data is all stored in the object file.

Basically, VMaps are to points, what surfaces are to polygons. Like surfaces, you name your VMaps, which allows you to access them to perform certain operations. Some VMaps *types* are designed for special built-in functions. For example, UV Maps hold texture placement information. Endomorphs hold offset information for point position. Weight maps are used more generally, like for bone influence; however, the SubPatch weight map is used specifically for control-point tension.

If a point has no entry for a particular VMap, the VMap is not assigned to that point—this map will have *holes* in it. Note that an assigned value of 0 is not the same as not having the VMap assigned to that point. However, some functions that use VMaps consider these two states to be equal—usually by necessity—essentially filling the holes with zeros.

VMaps are normally created using the buttons in the lower-right corner of Modeler. Once the map is created, you can set or edit the values in the map. Many LightWave commands, however, also have the ability to create VMaps.

NOTE
VMaps are extra data and will increase your object's file size. Thus, depending on the situation, you may be able to minimize the file size by judiciously assigning VMaps to points.

Because you can use vertex maps (*VMaps*) throughout LightWave, you should have a solid understanding of how they work. Here are some important basic points:

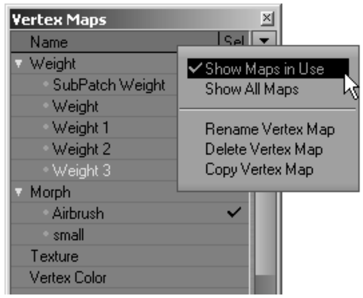
- *Weight* maps have a general range of -100% to +100%.
- Use the Selection Info command (**Display** > Selection: **Info**) to determine what VMaps and values are assigned to individual points.
- Internally, there is a single list of VMaps, which includes the maps from all the objects that are loaded. However, editing a map in one object will not affect other objects (using the same VMap name) because the data in the map is attached to the geometry of that object. There is no *cross-talk* between objects, except in the case of renaming. If you rename a VMap, all VMaps (for loaded objects) with the same name are renamed.
- Generally, the only VMaps listed in the main VMap pop-up menu are those that contain actual data in the current object. Other VMaps are filtered out. Not all VMap pop-up menus will do this filtering, however, particularly those on plug-ins.

  New VMaps are an exception to the above standard. Initially, they will appear in pop-up menus even if they contain no data. However, if you deselect a new VMap before using it, it will disappear from current VMap listings. To possibly save you some typing time, Modeler still places the name in the pop-up menu that appears beside the naming field when you are creating a new VMap.
- There can be one selected VMap of each general category (weight, UV texture, etc.), and they can be selected in different ways. The pop-up menu in the bottom-right corner of Modeler shows the current selection for the weight, UV texture, and morph categories, and can be used to change them. The VMap List window, discussed later, also shows the selected map in each category. Note that all new maps start out selected.

VMAP LIST WINDOW

You can display a non-modal window listing all VMaps in the current object, grouped by type, by choosing **Modeler** > **Windows** > **Vertex Maps Open/Close**.

Right-clicking on a map will display a pop-up menu where you can select various maintenance operations. This menu is also available for the VMap selected in the list using the pop-up menu button (down arrowhead) in the upper-right corner.



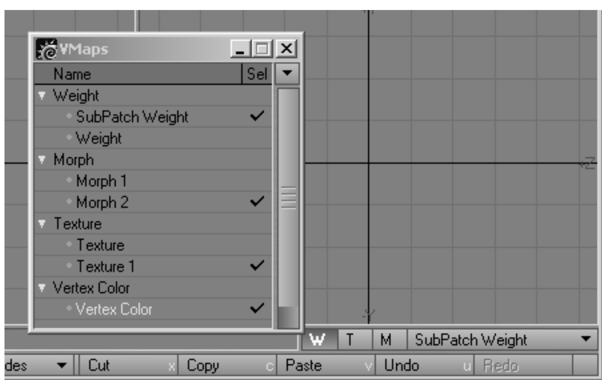
List window pop-up menu

The pop-up menu also has options to show all vertex maps or to only show those that are in use in the current object, the default.

VMap Selection

There are two levels of selection. First, one main VMap, regardless of type, can be selected by clicking on its name in the window (or selecting it in the pop-up menu in the lower-right corner of the Modeler interface). Its name will become highlighted. This is used by various commands that work on any type of Vmap (e.g., Copy VMap and Delete VMap).

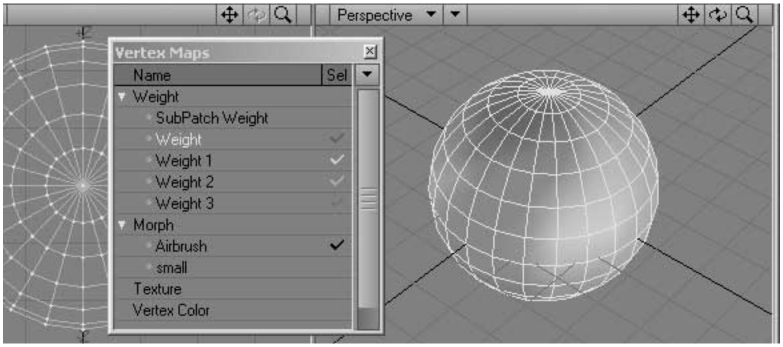
Generally, one VMap from each type (e.g., Weight and Morph) may be selected by clicking in the Sel column. A checkmark appears for these. (Note that the main selected VMap is always the selected VMap for its type.) This selection state is used by tools that only deal with certain VMaps, like the Airbrush tool.



Selecting VMaps

Changing the VMap selected in the pop-up menu on the (main) Modeler interface will change the main selected VMap and the VMap selected for its type. However, remember that not all VMap types (e.g., color VMaps) are accessible from this pop-up menu.

For display purposes, you can select up to four weight maps. They will be visible in Weight Shade mode using different colors.



Multiple weight maps selected

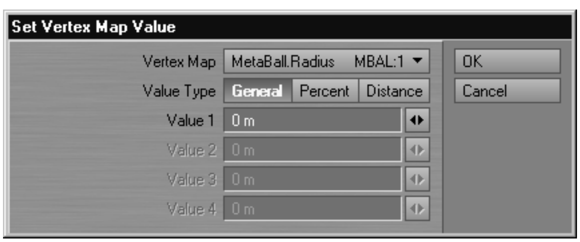
GENERAL VMAP COMMANDS

Here are some commands you can use to perform general maintenance on VMaps. Remember, even the special purpose VMaps, like endomorphs, UV maps, and so on, are all just vertex maps.

Generally, the following commands operate on the *currently selected* vertex map. This is the one highlighted in the Vertex Maps window, discussed previously. *This may be different than the one showing in the lower-right corner of the Modeler interface!*

Setting Map Values

The Set Vertex Map Value command (**Map** > General: **Set Map Value**) sets the current VMap value to whatever you enter. To use the command, set the **Vertex Map** from the pop-up menu on the panel and enter the desired value for the (entire) map.



Different **Value Types** are available so that you can, if necessary, enter the appropriate value depending on the type of VMap you select. This is important for plug-ins because Modeler doesn't know what type of units you need.

The extra **Value** fields are for VMaps that require more than one value. For example, a morph target actually has three values per point. The point needs to know its X, Y, and Z delta to move the vertex appropriately. Other VMaps might need only one value and some may need up to four.

Copy, Delete, and Rename Commands

Copy Vertex Map (**Map** > General: **Copy Map**), Delete Vertex Map (**Map** > General: **Delete Map**), and Rename Vertex Map (**Map** > General: **Rename Map**) are commands that will affect the currently selected VMap.

Clearing Maps from Points

You can clear the selected VMap from selected points using the **Clear Map from Vertices** function (**Map** > General: **Clear Map**). If you don't select any points, it clears the map from all points. The map name will still appear—as if you were creating a new map—, but if you do not edit the map, it will eventually disappear.

NOTE
Notwithstanding the Delete Map command, the Clear Map From Vertices command is still useful since you can clear a map from some of the points (using the map), without deleting the entire map.

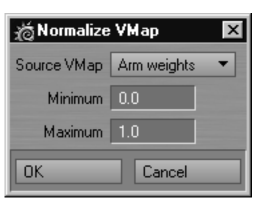
CullMap

The CullMap function (**Map** > General: **Cull Map**) selects or deselects points that have entries in the selected vertex map, if the magnitude of the map at that point is below the **Threshold Magnitude**. Activate **Clear Map** to remove these points from the VMap.

The magnitude in *UV space* is given by the square root of U squared plus V squared. Since UVs are on a scale from 0 to 1, the greatest magnitude possible would be the square root of 2, or 1.414. Thus, to select or clear all points from the UV map that are in a certain region of your texture, you would use this function.

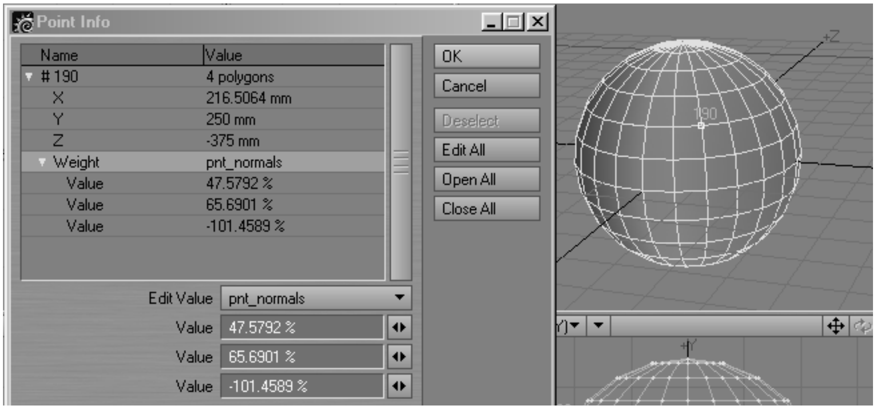
Normalizing Values

You can also normalize a vertex map with the Scale VMap Values function (**Map** > General: **Normalize Map**). The selected VMap values are scaled to fit between the **Minimum** and **Maximum** range. This does not clip the values.



Normal Baker

Normal Baker will store the (normalized) point normal for each vertex as three numbers into a VMap called PNT_NORMALS. The values, starting with the first, represent the X, Y and Z directions of the vertex's normal.



Details of single point after using Normal Baker

WEIGHT MAPS

Here's a question: how much does a point weigh? Well, you might say nothing, but you might be wrong. *Weights* are arbitrary values associated with object points. These values can be used by built-in functions and plug-ins. Weights can control the falloff of modeling tools, change the affect of bones, influence the control point bias in a SubPatch cage, and much more.

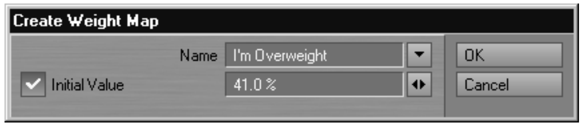
There are two types of weight maps: SubPatch and general purpose. The SubPatch Weight map lets you create SubPatch objects (see Chapter 29) with much greater control and accuracy without adding complexity to the cage. General purpose weight maps are used for a variety of functions, like bone weighting, modeling tool falloff, and so on.

To create a general weight map:

1 Click the Weight map button in the lower-right portion of the interface, which is marked with the letter **W**. (Alternatively, you can choose **Map** > Weight & Color: **New Weight Map** and skip to step 3.)

2 In the pop-up menu to the right, select **(new)**.

3 Enter a map name in the **Name** field. If you wish to set the initial weight value for all points, activate **Initial Value** and enter the desired initial value in the input field. Click **OK**.



Once you create a weight map, you can assign/modify point weight values. Understand that if you don't use the initial value option, merely creating the weight map does not assign any information to your points.

Instead of typing in a map name, you can use the pop-up menu button to the right of the **Name** field. This lists all maps used in the current session, even if no points use the map. Do not confuse it with the vertex map selection pop-up in the lower-right corner of the interface. That pop-up selects the map you are editing. It will also list all of the VMaps that have points with values.

To change a point weight:

1 Select the Weight map **W** button and the previously defined weight map name in the pop-up.

2 Change the viewport style to **Weight Shade** (use the pop-up menu on the viewport titlebar.). You need to change it only for the viewport you will be working in. It's kind of neat to do this in a perspective viewport.

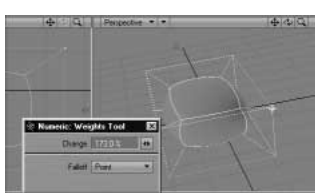
NOTE
The Weight Shade style gives you a visual indication of a map's influence in the form of color shading. This can be very important, particularly when weights are not visibly influencing object geometry.

3 Activate the Weights tool (**Map** > Weight & Color: **Weights**).

4 Open the numeric panel (N).

5 Put your mouse pointer over the point you want to edit.

6 Drag right to increase the weight of the point and left to decrease it (weight can even be negative). You will receive feedback in the form of color in the viewport. Red indicates positive weight and blue indicates negative weight. You will also see the percentage of change in the numeric panel.



NOTE
The percentage of change is a relative change. If you release the mouse button, the change is set. Dragging again applies another change starting at 0%.

NOTE
Using the Weights tool with no points selected, will add the current weight map to all points, not just the one being edited. Non-edited points are assigned a 0% value. If you do not want this to happen, select the points to be affected before using the Weight tool.

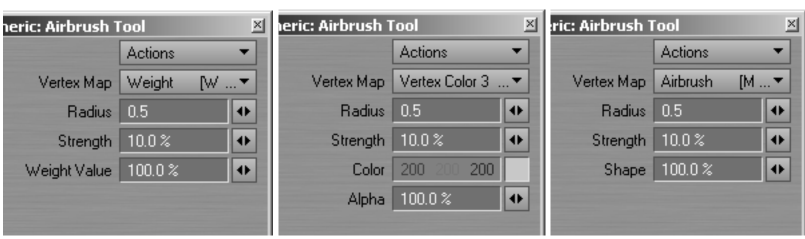
Weight Tool Falloff

The Weight tool defaults to **Point** as the **Falloff** type. However, on the numeric panel, you can change this to any of the falloff settings (see Chapter 23).

The Airbrush Tool

The Airbrush tool (**Map** > Weight & Color: **Airbrush**) interactively adjusts point weights, vertex colors, or morph maps to mimic the effect of an airbrush. Remember, however, that the airbrush is limited by the options allowed by the VMap for any one polygon's vertices—you can't paint an S on a square polygon. You can achieve greater detail when there are more points.

The Airbrush has a pop-up menu, on the numeric panel, to choose the map that it affects. Only those VMaps present in the current object are listed. Selecting a weight map makes it the current map. Also, selecting a new weight map in the main interface (or VMap window) will change the airbrush selection.



Left: to *Right*: Weight, Vertex Color, and Morph map settings

To use the Airbrush:

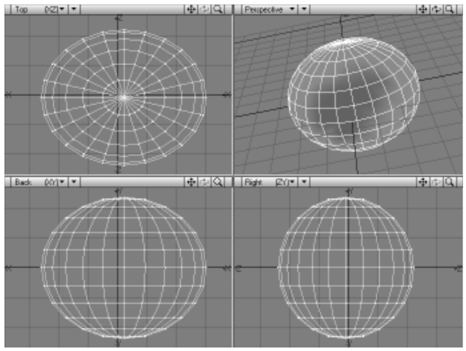
1 Set up the same way as if you plan to use the Weights tool.

2 Select **Map** > Weight & Color: **Airbrush**. On the numeric panel, the **Radius** value sets a relative size for the falloff. You can interactively adjust this value with the RMB. The **Strength** value sets intensity of the tool. Small values (around 10%) will usually work best. Negative values will reduce the amount of weight, even below zero.

For weight maps, the **Weight Value** can be adjust higher or lower from the default 100%.

For color vertex maps, you can adjust the **Color**. If you are using a RGBA map, you can set the **Alpha** amount.

For Morph maps, the airbrush basically paints a selected morph map onto the current morph map. The **Vertex Map** setting is the morph map that acts as the "paint." The **Shape** value adjusts the strength of the morphing.

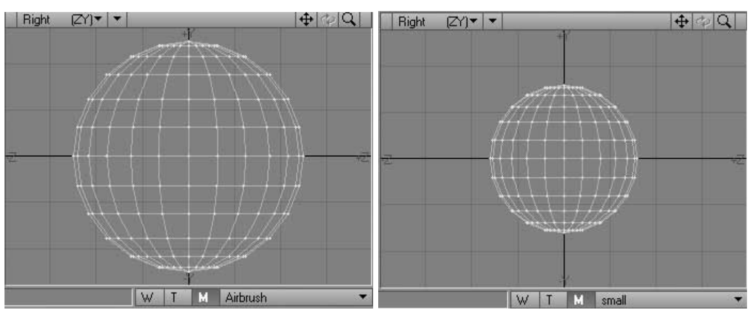


Airbrushing a weight map

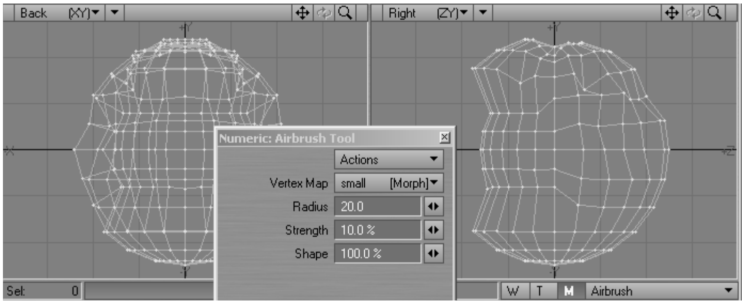
Hold the SHIFT key as you drag to subtract weight or the CTRL key to adjust toward 0. On a vertex color map with alpha, the CTRL modifier will reduce the alpha.

NOTE

The radius area must actually touch a point for this tool to have an effect since it must add value to a point. Also, if you have points selected, only those will be affected.



Left: Before airbrushing the morph map. Right: The morph map used as the paint



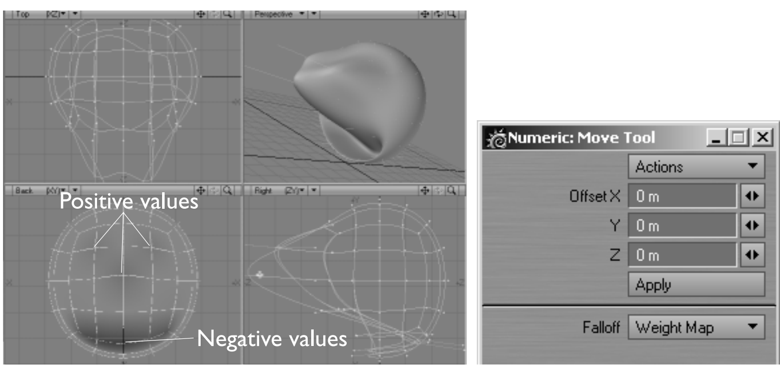
Morph after airbrushing some areas

HINT

The Airbrush can act sluggish when used on a complex object. As an alternative, try the Weights tool and use **Point Radial** as the **Falloff** on the numeric panel. You must be directly over a point to use this method, but you will get faster updates and you can decrease weighting by dragging left. (For more information on Falloff, see Chapter 23.)

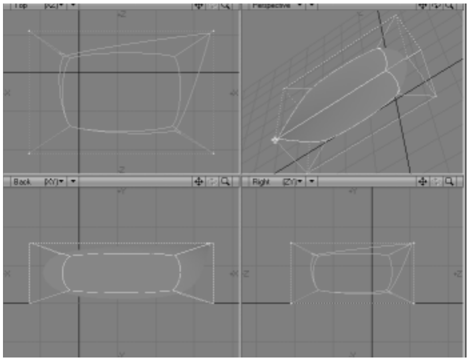
Using Weight Maps with Modeling Tools

Instead of using one of the standard **Falloff** settings (on the numeric panel), many tools let you use a weight map to control their effect. Points with 100% weighting are affected normally, but 0% weighted points are not affected at all. Points with weighting in between are affected proportionately. Negative weighting has inverse effects. Using a weight map gives you point-level control on how a tool is applied.



Using the Move tool with a Weight Map Falloff. Note the effect of positive and negative weighting.

Changing the weight of a point in a SubPatch weight map makes the corner much sharper without adding more control points. (See Chapter 29 for details.)



Effect of a SubPatch weight map

VERTEX COLOR MAPS

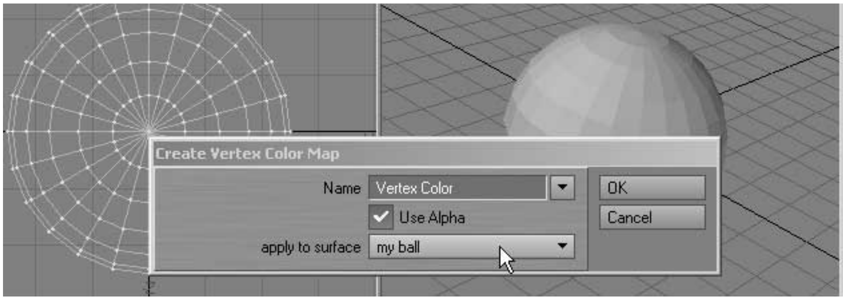
A vertex color map is VMap type that lets you color the vertices (i.e., points) in your object mesh. The color for areas in between the vertices is averaged. (The Baker surface shader also uses vertex color maps.) You can see the effects of vertex color maps in Flat shade, Smooth shade, and Texture viewport modes. Vertex coloring is *multiplied* on top of the base surface color with its texture.

NOTE
Basically, vertex color maps work like a (color surface) texture layer using **Multiply** as the **Blending Mode**. The **Vertex Coloring** percentage operates like layer opacity.

The reason the color is multiplied, instead of added, is that multiplying provides better illumination maps, which is the most useful application of vertex color maps. The effect may surprise you in some situations, however. For example, if you have a green surface (RGB 0,255,0) and you apply a red vertex color map (RGB 255,0,0), the result is is black (RGB 0,0,0). Each color component is multiplied.

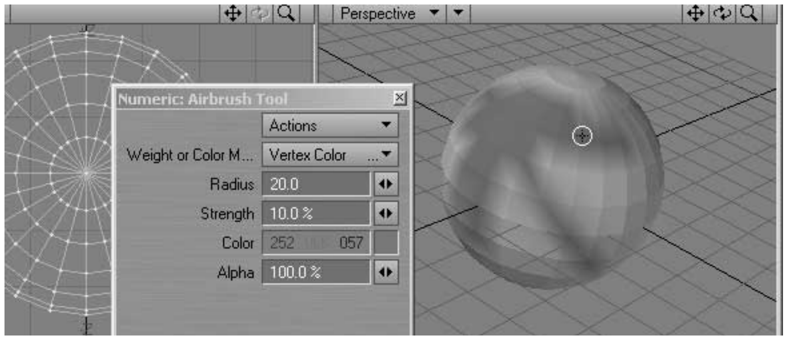
To Use Vertex Color Maps:

1 Choose **Map** > Weight & Color: **New Color Map**. Enter a name in the input field and choose the surface that will use this vertex color map on the **apply to surface** menu. Generally, you should keep **Use Alpha** checked, but see the following discussion.



2 Open the Advanced tab of the Surface Editor and select the same surface used above. You will see the map you just created already selected on the **Vertex Color Map** pop-up menu. (The **Vertex Coloring** is a blending amount. You can leave it at 100%.) If in the future you need to change the color map or remove it from the surface, you would do it here.

3 Choose **Map** > Weight & Color: **Airbrush**. On the numeric panel, pick your map on the **Weight or Color Map** pop-up menu. Choose an appropriate **Radius**, **Strength**, **Color**, and **Alpha**. Start painting.



NOTE
The density of points greatly affects how much detail you can paint.

Vertex Color Alpha

Vertex color maps come in two flavors: RGB and RGBA. (The *A* stands for alpha.) The **Use Alpha** option, when active, selects RGBA.

The RGB maps have only a red, green, and blue value at each vertex, or no color at all, if the point is not in the map. The color from an RGB map is blended according to the **Vertex Coloring** percentage with the surface. If that value is 100%, for example, then the color at a point is the RGB in the map, or the normal surface color if the vertex is unmapped.

NOTE
Remember, a vertex with a value of 0 is not the same as a vertex with no value in the VMap.

The RGBA map adds an alpha value for each vertex (to the RGB information), and the alpha is also used to blend in the vertex color. So, if the **Vertex Coloring** percentage is set to 100%, as above, it's still possible for individual vertices to blend their value with the normal surface color. This allows for better feathering effects at the edges of maps.

The difference between RGB and RGBA is best illustrated with the Airbrush tool. If you have an empty RGB map and start painting, you see that as you add vertices to the map, they first receive a black color. This results from mapped and unmapped vertices sharing the same polygon. Then, as you paint more, the Airbrush color gradually fades in.

With an RGBA map, you see better feathering and none of the odd effects along the edges of the map.

In general RGBA maps are better and more flexible than RGB maps. However the alpha value takes up space, and is unnecessary for maps that will be defined for all vertices and that can uniformly blend with (or replace) the base color.

HINT
When in doubt, activate the **Use Alpha** mode.

Vertex Paint

Vertex Paint is covered at the end of the chapter.

ColorMap Adjust Tool

You can use the ColorMap Adjust Tool (**Map** > Weight & Color: **ColorMap Adjust Tool**) to interactively adjust an existing color map. The tool appears as three vertical sliders in all viewports. You can reposition the group of sliders by dragging the small circle handle in the lower-left corner. You will need to display the numeric panel to get at the controls.

Since this is a tool, you can reset your changes by clicking in a reset area. Also, closing the tool, or selecting another, accepts your changes.

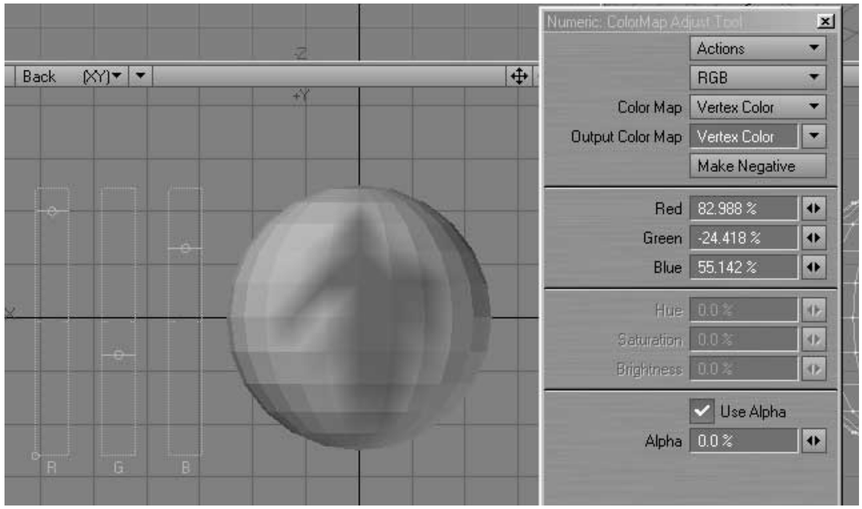
The first option lets you control the adjustment using red, green, and blue (**RGB**), or hue, saturation, and brightness (**HSB**). These settings operate independently. That is, when you switch between settings, the values are not converted to the other standard.

Select the source vertex color map on the **Color Map** pop-up menu. The changes will affect the **Output Color Map**, which may be different if desired. (Note that you can only see the color map in a viewport, if it is assigned to the visible surface.)

To invert the color map, click the **Make Negative** button. This will also reset the sliders to 0.

You can change the color values by dragging the sliders or editing the value on the numeric panel.

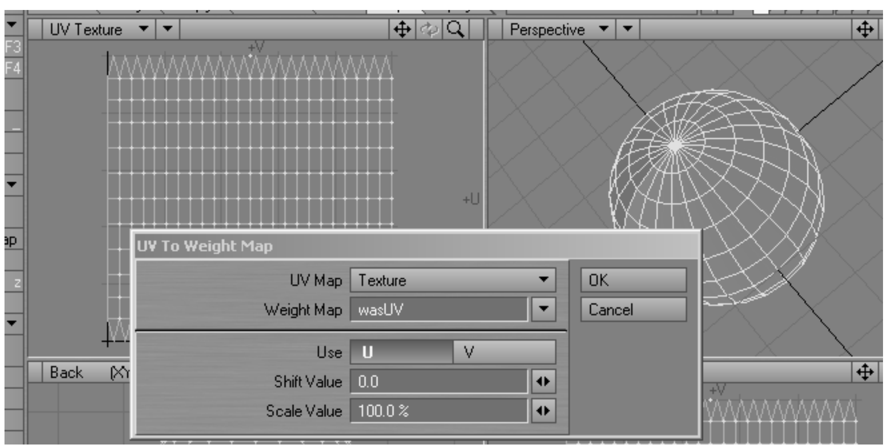
The **Use Alpha** option makes the output map an RGBA vmap instead of just RGB. Even if no adjustments are made, this can be used to copy an RGB map to an RGBA. The percentage setting is the amount of alpha used.



ColorMap Adjust Tool

UV to Weight

The UV to Weight command (**Map** > Weight & Color: **UV to Weight**) converts the U or V values (0 to 1) from a UV Texture map to weight map values (0% to 100%). This can be handy if you (really) want to use a UV as a gradient input parameter (which isn't allowed).



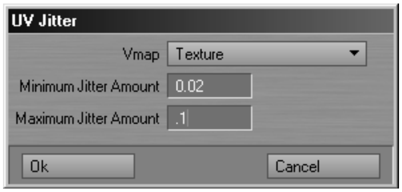
The **Shift Value** setting will adjust the U or V value by the entered amount. Then, **Scale Value** adds the entered U or V value. (Note: Values are not clipped and there is no constraining during the conversion.)

HINT

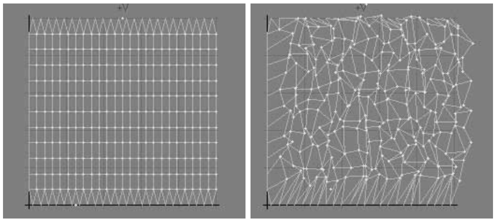
If you want to turn a typical 0-to-1 UV map into a (minus)-1-to-1 weight map, set **Scale Value** to 200% and **Shift Value** to -1.

UV Map Jitter

Use UV Map Jitter to add some randomness to a UV map. Simply select the UV map from the **Vmap** pop-up menu and enter the minimum and maximum amount of jitter. Remember that UV values normally go from 0 to 1, so small settings go a long way.



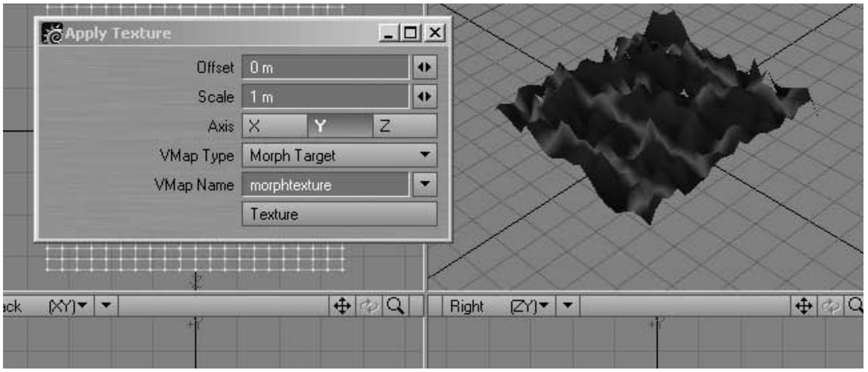
UV Map Jitter



Before (*left*) and after (*right*) jittering

Texture VMap

The TextureVMap function (**Map** > Weight & Color: **Texture VMap**) can apply the luminosity values of a standard texture to a weight or morph map, or the color values to a color map.



A morph map and vertex color map combined

Set the **VMap Type** to either **Weight Map**, **Morph Target**, or **Vertex Color** (map). Entering a new **VMap Name** will create it, if it doesn't already exist.

Click the **Texture** button to bring up the Texture editor. Create a texture as you would normally for, say, a bump map.

The values along the selected **Axis** are affected by the brightness values of the texture. Leave the panel open as you experiment with different textures—updates are in real-time. To *freeze* the texture you want, simply close the panel.

The **Offset** value will move the 3D texture through the surface. The **Scale** value sets the overall size of the texture in a manner not much different than the XYZ Scale settings for a normal surface texture.

HINT
You can get an interesting effect by creating a weight map with this plug-in and then using the weight map as the Falloff for a tool like Size.

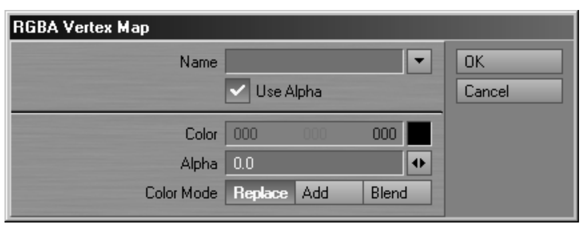
HINT
Use Texture VMap to quickly establish a starting point for vertex color maps. This saves you from painting all of the points individually. See the discussion on "Vertex Color Maps" later in this chapter.

Bone Weights

See Chapter 10, "Bone Weight Help."

Point Color Command

The Point Color command (**Map** > Weight & Color: **Point Color**) lets you set the vertex color for selected points. Enter an existing or new name for the vertex color map in the **Name** field. You can choose whether to include an Alpha value with the **Use Alpha** option, as well as define what that value is.



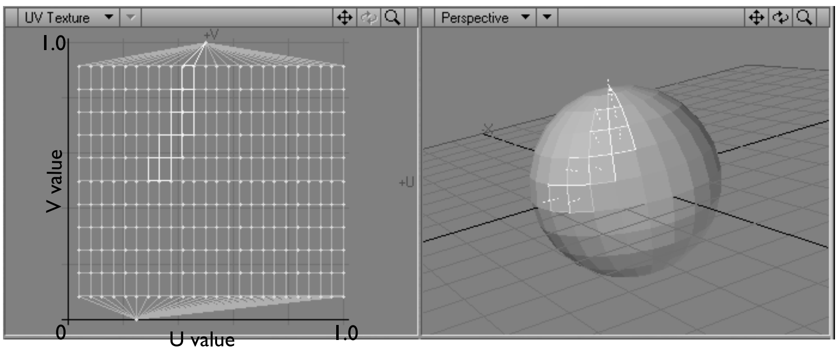
The **Color Mode** determines how the color value is applied. **Replace** simply overwrites the color value. The **Add** mode adds the value to the existing color, if any. Note that the result can be greater than (RGB) 255. The **Blend** mode averages the defined and existing colors.

UV TEXTURE MAPS

Sometimes, standard image mapping tools (i.e., planar, cylindrical, and spherical mapping) may be somewhat limiting where the surface is irregular in shape. These techniques usually work well only where you can globally map the entire texture image using a linear interpolation along two axes. The object geometry essentially has no influence on how the texture is applied.

However, what if you could assign areas of a texture image to points on the surface, essentially *tacking* it down at key points? Well, you can with *UV mapping* in Modeler. Between the tacks, the image is stretched smoothly.

The U and V refer to texture map coordinates and are really not much different than the XYZ coordinates you are familiar with. In fact, UV mapping is the process of setting up a relationship between the two dimensions of an image, U and V, with the three dimensions of an object surface, XYZ.



UV map (with overlayed graph labels) for a sphere. Note selected polygons in both views.

Once this relationship is set up, changing any parameter (i.e., U, V, X, Y, or Z) will also relatively change the appearance of the texture mapping. With UV mapping, the object provides additional information for texture mapping, which can be different for any given point on the surface. The texture is more or less stuck to points on the surface using a relationship that you define.

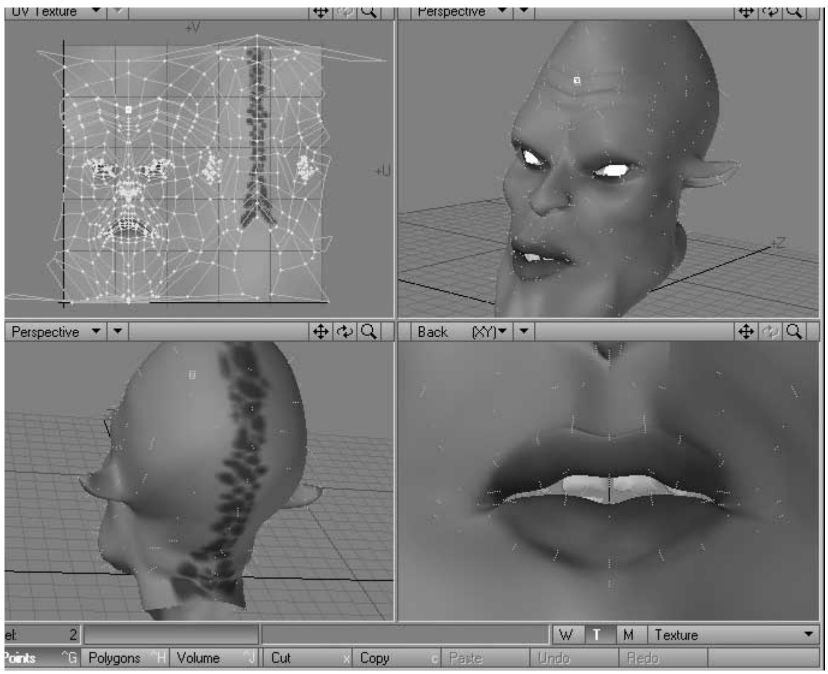
UVs and Projection

UVs have to come from somewhere. For existing polygonal models, the choices are limited to setting the UV coordinates for each point in the object manually, or applying some *projection*, which automatically generates the 2D texture coordinates from the given 3D point positions. In LightWave, you can create UVs by using projections, which also happen to be the same as the standard projections for texture mapping (i.e., planar, cylindrical, and spherical).

Usually, the projection for the UV map is not perfect for the model everywhere. The projected UV map must be tweaked—eyes and nostrils moved over the right parts of a face, or texture features matched to geometry features.

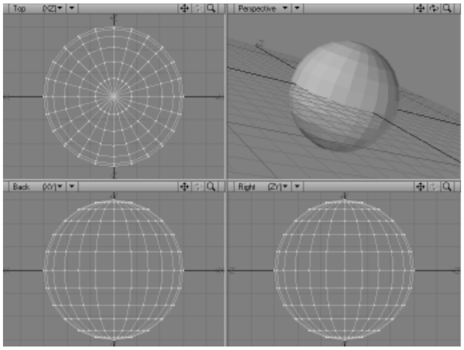
NOTE
Keep in mind that standard projection mapping is more accurate because it has some exact, continuous value over the entire surface. UV mapping, on the other hand, is technically accurate only at small sample points. The surface for the large areas in between the sample points are interpolated. Adjusting the sample points so that the interpolated areas look right is difficult and the reason why UVs are more difficult to use.

For illustration purposes, let's say you had your texture image printed on a piece of very flexible rubber and wanted to fit it on a toy car made of wood. You could conform the rubber material to contours of the car by tacking it down with thumbtacks. That is more or less what UV mapping does. However, it is a little bit reversed: what you do is *tack* the UV points down onto the image.

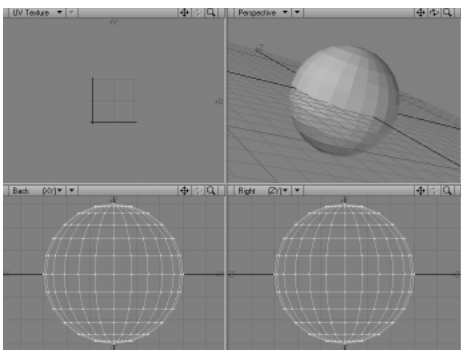


To create a UV map:

1 Load or create an object in Modeler.



2 For one of your viewports (not your perspective view though), go to the titlebar and set the **Type** to **UV Texture**. Also, change your perspective viewport's **Style** setting to **Texture**, so you can see the texture maps.

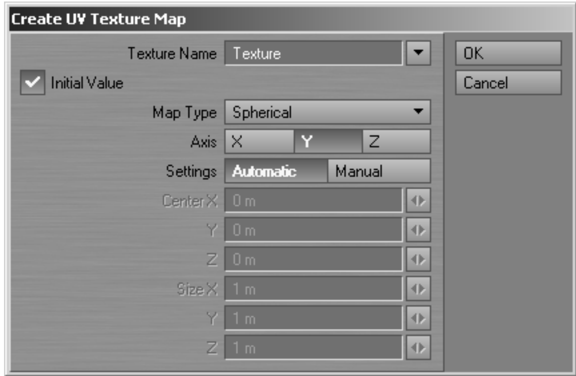


3 Select the **T** button (texture) at the bottom of the Modeler window and select **(new)** from the pop-up menu next to it. (Alternatively, you can choose **Map** > Texture: **New UV Map**. Both execute the Create UV Texture Map command) This lets you define your new UV texture map and give it some initial settings.

4 Edit the **Texture Name** field if you want to use something other than the default.

5 Normally, you check the **Initial Value** option and define initial UV mapping coordinates. This is done similarly to the way you set normal texture mapping.

Automatic will automatically size the texture or you can use the **Manual** mode and define specific **Center** and **Size** parameters.

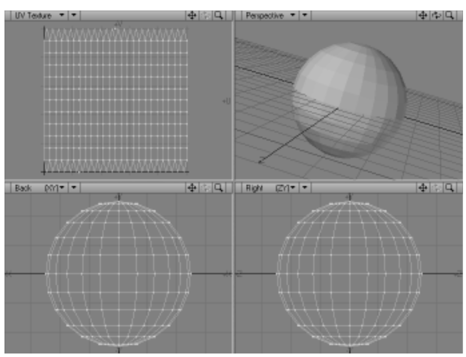


You may want **Initial Value** off if you will later create the UV map automatically, for example, using one of the primitive tools.

6 Click **OK** when done. The name should now be displayed on the pop-up button in the lower-right corner of the Modeler interface.

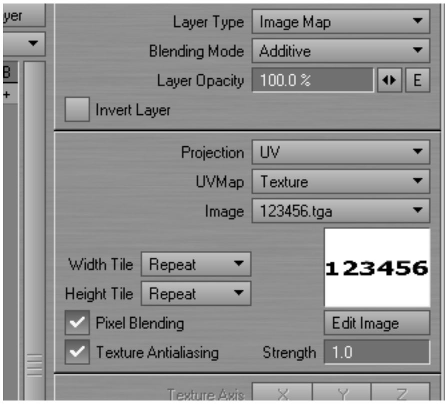
NOTE
You can set up multiple UV maps, each with its own independent set of assigned points. Whichever map name appears on the button is the active map.

7 Press the A key to autofit the object into the viewport.



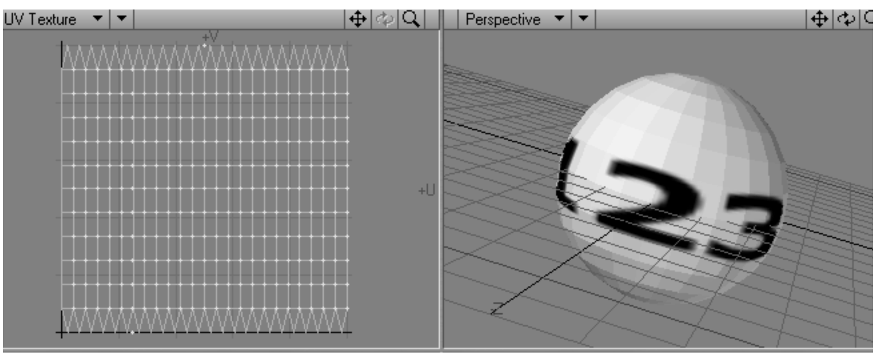
8 Open the Surface editor and select one of the surfaces on your object.

9 Click the **Color** parameter's **T**exture button to open the Texture editor and set **Projection** to **UV**. In the **UVMap** pop-up menu, select the UV Texture map you created earlier. Select **(load image)** from the **Image** pop-up menu and load the image you wish to map.



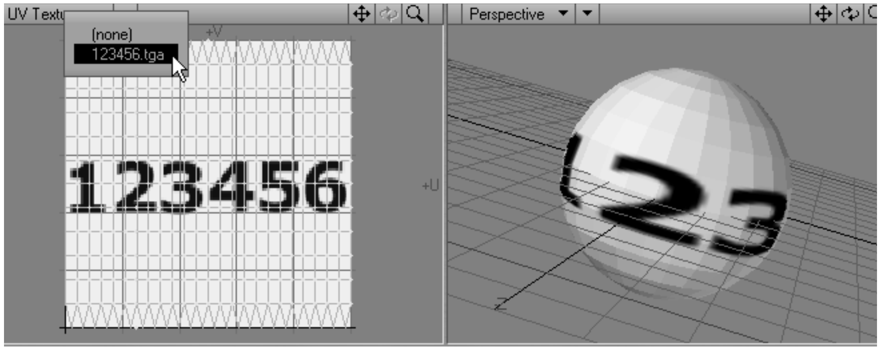
NOTE
You can actually use a UV map on any surface texture.

10 You should see your image UV mapped on the object in your Perspective viewport. Click **Use Texture**.



In the UV Texture viewport, you see the points and polygons of your 3D object surface in a 2D mapping grid format.

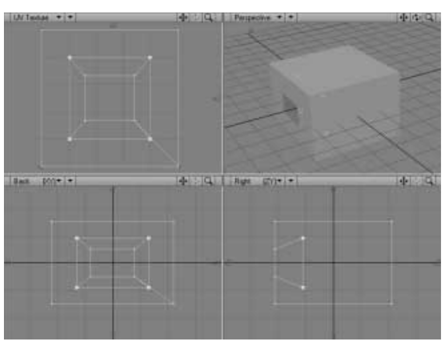
11 Choose the image from the UV Texture viewport's toolbar. (This is quicker than setting it as the backdrop on the Display Options panel.) This lets you see how the points and polygons line up with the image in UV space.

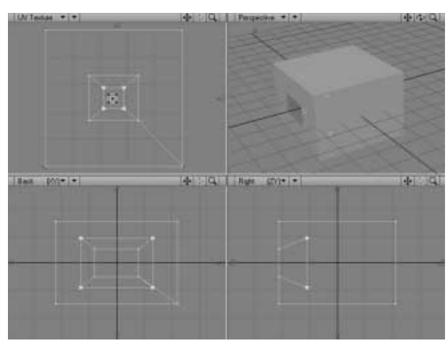


12 To edit the UV mapping, use a modify tool (like Move) in the UV Texture viewport to move the points/polygons around and thereby redefine the relationship between the UV and the XYZ coordinates. Note that editing in this view has no effect on the geometry of the object.

NOTE
There are two ways to match image to surface with UV mapping. You can either take an image and move the UVs on the image (displayed as background), or you can take a screenshot of your existing UVs, and use it as a template image to paint textures on.

Depending on your object, interior points can end up in weird areas. Here is a simple illustration below on the left. Notice how the interior points are outside of the points at the opening. On the right, the UV Map points for those interior points have been moved manually to allow easier matchup with an image.





NOTE
UV Points that are off the image will always wrap to the opposite side (as if the image was tiled).

Guess Viewport UV Image Command

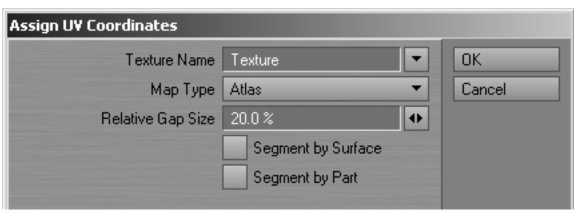
If the current UV map is part of the surface currently selected in the Surface Editor, the Guess Viewport UV Image command (SHIFT+F9) will attempt to guess the backdrop image for a UV Texture viewport. If the viewport under the mouse is a UV viewport then only that one will be changed, otherwise all UV viewports will be affected, if there are more than one.

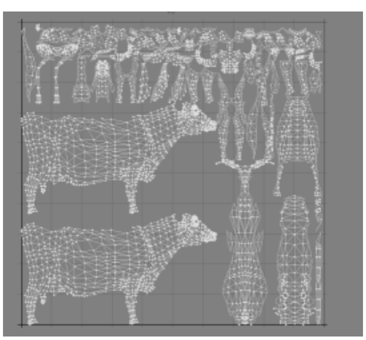
Atlas Map Type

You can generate a *texture atlas* by selecting **Atlas** from the **Map Type** pop-up menu. The selected polygons are spread out in UV texture space so that they do not overlap, and have a minimum of distortion. (*Atlas* is a standard name for this type of mapping and has obvious similarities to books with the same name.)

The **Relative Gap Size** value sets the spacing between groups. The higher this percentage is set, the larger the gaps between disjointed groups. Additional spacing may help with interaction between adjacent areas in a UV map, but it will reduce the resolution of the image for each part. There is also a limit to how far apart they can get, since they all must fit into a finite area.

You can also specify where the atlas has *discontinuities* between polygons with different surfaces or parts.





Recognize this ol' girl?

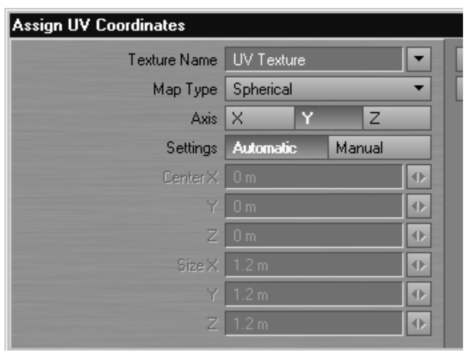
Atlas is great for quick UV mapping situations, particularly where you don't need precise texture details (e.g., hair, clothes, and so on). It's also good for objects that have large flat planar surface areas that don't all lie at nice 90-degree angles. If you've got a totally symmetrical object, cut it in half and Atlas map it. Then, mirror the missing half back. This will give you a nice symmetrical UV map.

Texture Guide

For more interactive adjustment of projection textures and UVs, or to create UVs from an existing projection, the Texture Guide tool (**Map** > Texture: **Texture Guide** see Chapter 31), is indispensible. To use it, you should have a textured object already loaded, and its texture layer open in the Surface Editor. When the Texture Guide tool is launched, it can determine the texture parameters from the texture editor, and modify these. If a UV map for the object is available, the Texture Guide can also sample the projection into the UVs interactively.

The Make UVs Command

You can also choose **Map** > Texture: **Make UVs** to assign values to an existing UV map or new one. You might use this if you did not use the **Initial Value** option when you created the UV map. To create a new UV map, just enter a unique name in the **Texture Name** field. To edit an existing UV map, choose it from the pop-up menu.



To assign UV map information to only part of an object, select the desired points/polygons before you execute the Make UVs command. You can even use a different **Map Type** for each selection.

The difference between the Make UVs and the Create UV Texture Map commands is subtle. The latter begins with a texture name that is guaranteed to be unique (i.e., unused) and *optionally* lets you set initial values. The Make UVs command, on the other hand, begins with the current UV map name and always applies values.

Creating from Command Panels

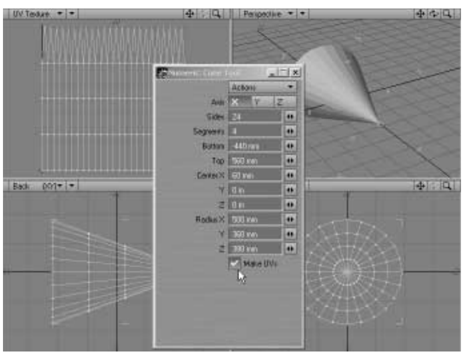
Many command panels have VMap input fields (with a pop-up menu to the right). Entering a non-existent map name will create that map. The appropriate type of map (e.g., weight, UV, etc.) will be created, if the panel does not let you specify the type.

Automatic UV Making

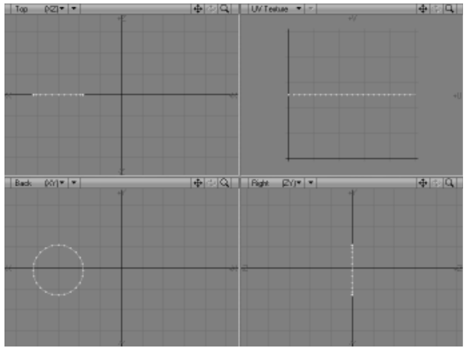
You can also automatically create UVs when you create primitives by activating the **Make UVs** option on the related numeric panel. You must already have pre-defined and selected a texture name. A default UV is created for the currently selected texture map based on the geometry of the object.

To use Make UVs:

1. Create a UV texture map and open a texture viewport.
2. Make a primitive with **Make UVs** active on the numeric panel. You should see a grid in the UV view that is the default UV assignment for the primitive shape.



With the *sweep* operations, like **Extrude**, **Lathe**, **Path Extr**, and so on, you can assign one of the values, U or V, to a range from 0 to 1 along the segments of the sweep. For example, if you create a flat disc with **Make UVs** active, you will see a line in the UV Texture viewport.

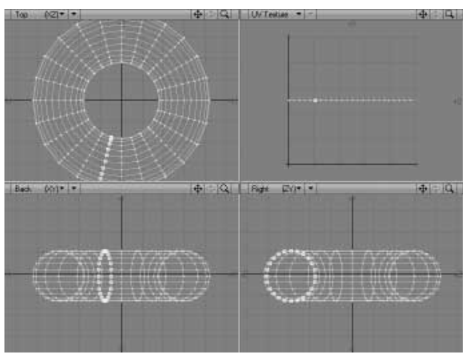


A disc and its default UV

The line represents the UV coordinates for the points around the disc. They go from 0 to 1 on the U axis and are all at 0.5 on the V axis. If you sweep this disc, say with a **Lathe**, then you can see different results in the UV Texture view depending on the **Make UVs** setting for the tool.

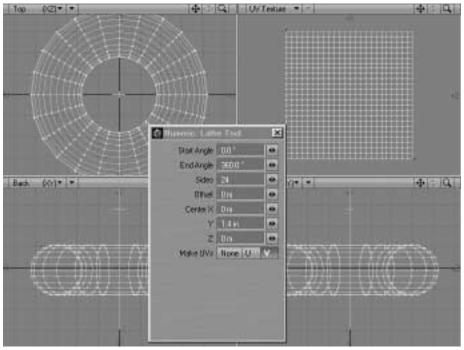
With **None**, you'll see the same line as before. This is the same set of 24 points that make up the original ring.

If you set the value to **U**, then you'd see a line again, but this is all the points of the torus, not just the original ring. In this case the 0.5 V value is duplicated and the U value ranges from 0 to 1 along the lathe. That's the wrong choice in this case, although it would be right if the original line had been vertical (along V) instead of horizontal.



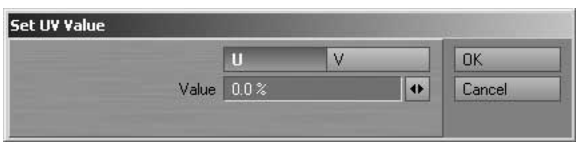
If you set the value to **V**, then you see the grid that you would expect, which is the proper UV parameterization of a torus. The values on U go around the small radius, and the values on V go around the large radius.

The original U values from the ring are duplicated on all the slices and the V value is set to 0 to 1 as we go along the lathe. The original 0.5 V value is discarded.



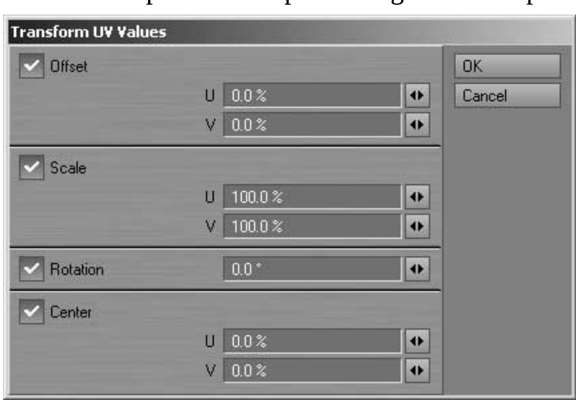
Set UV Value

The Set UV Value command (**Map** > Texture: **Set UV**) can be used to set either the U or V value for the selected point(s). This is analogous to the **Set Value** command, but works on the current UV map instead of the points' XYZ values.



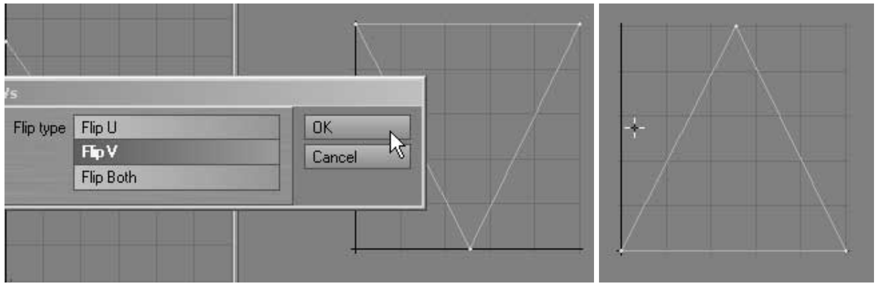
Transform UV Values Command

The Transform UV Values command (**Map** > Texture: **Transform UV**) allows numeric adjustments to the current UV map values by doing an offset, scale, and/or rotation around an optional center. Coordinate values are specified as percentages in UV space.



Flip UVs

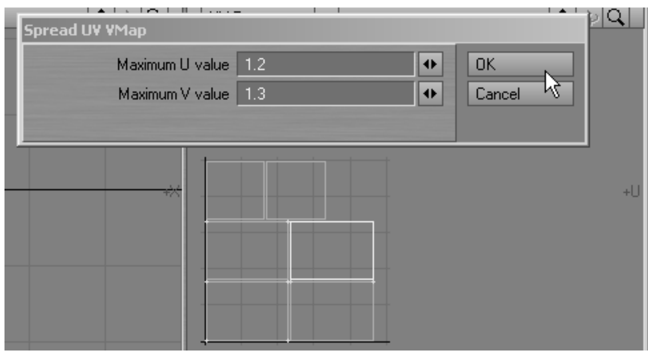
The Flip UVs command (**Map** > Texture: **Flip UVs**) reverses the UV map along the U, V, or both axes. Note that you may not get acceptable results if there are discontinuous UVs. You may want to unweld, flip, and then re-weld.



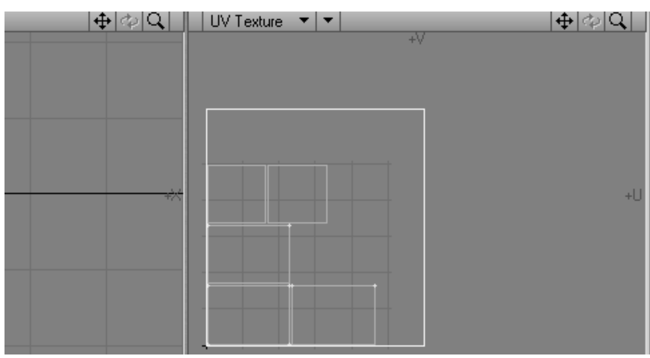
Left: Before flipping. *Right*: After flipping

Spread UVs

The Spread UVs command (**Map** > Texture: **Spread UVs**) operates on four-point polygons (a.k.a. *quads*). Basically, it sizes the polygons' UV map to the entered values.



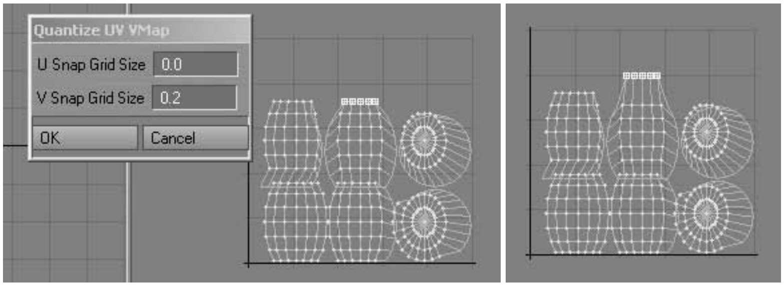
Single quad selected



Result of using Spread UVs

Quantize UVs

The Quantize UVs command (**Map** > Texture: **Quantize UVs**) snaps selected UV points to the specified U or V value (0 to 1).



Left: Before quantizing. *Right*: After quantizing

DISCONTINUOUS UVS

VMaps now support *discontinuous* values across polygon boundaries, which are useful for UV texture coordinates, gradient weights, and other VMap-controlled surfacing parameters. *Discontinuous UVs* are now automatically used when appropriate with Modeler's modeling tools. This means you should rarely encounter the dreaded *seam problem*, discussed below.

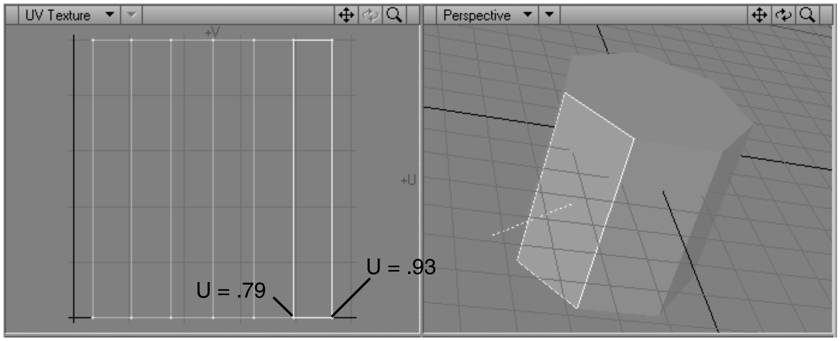
WARNING
Discontinuous UVs are now created in the normal course of (Modeler) business, thus a firm understanding of them is imperative.

That Dreaded Seam Problem

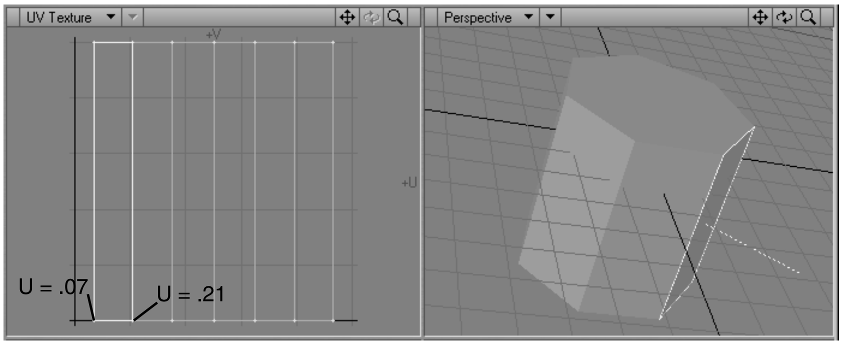
Prior to LightWave 6.5, when you used any of the automatic UV creation options on continuous geometry (like a sphere or cylinder), you would often encounter weird *seam polygons* where most of the UV texture would wrap backwards along a seam in the geometry. The texture on this polygon would be mapped from something like 0.93 to 0.07 and the entire image would get squished backwards onto this polygon—usually not the desired result.

NOTE
The following illustration was created using pre-6.5 Modeler. You will not normally be confronted with this seam problem. However, this will give you a better understanding of discontinuous UV Maps.

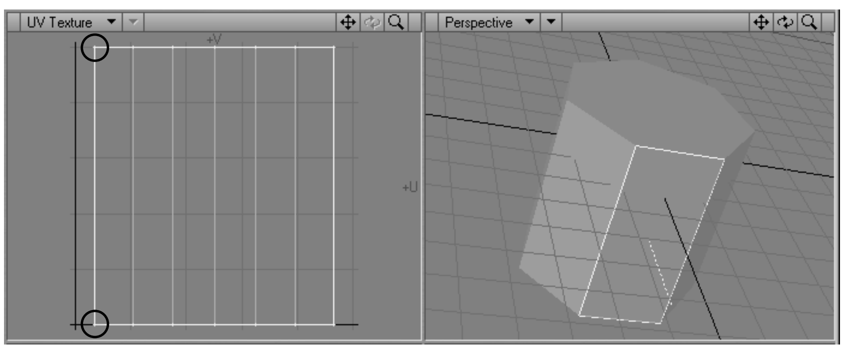
To illustrate, below is a seven-sided cylinder and UV map. Note where the selected polygon is on the cylinder and UV map.



Next, let's skip a polygon and note its position on the UV map, below.



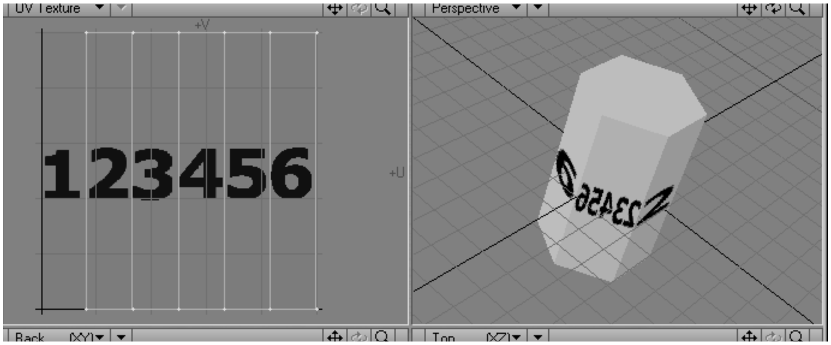
So far, we've seen that two non-contiguous polygons are on the right and left edges of the UV map. Where do you think the UV points are for the polygon in between? (See below for the answer.)



Did you guess correctly? Nearly the entire image—less a small portion on the left and right—is mapped backward on this polygon.

The fundamental reason for this is that the nice continuous projection is actually discontinuous at this *seam*. Ideally, the vertices for the circled UV points would be double-valued, using a 1.07 for this selected polygon, and a 0.07 for the one selected in the previous screenshot.

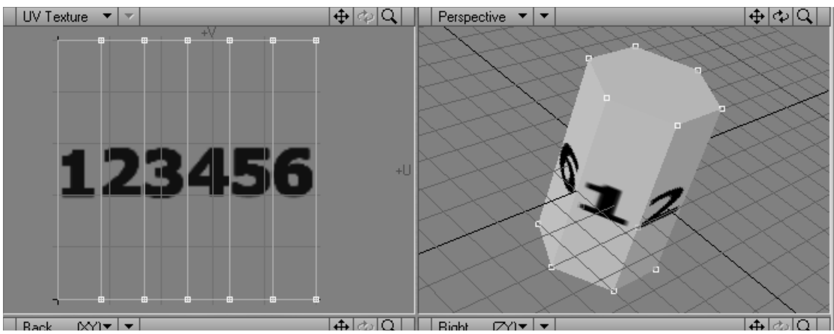
Below is an example (on a six-sided cylinder) where the effect of the seam polygon is readily visible.



Instead of the "1" being mapped onto the polygon, the polygon is getting the "23456" part of the image mapped backward!

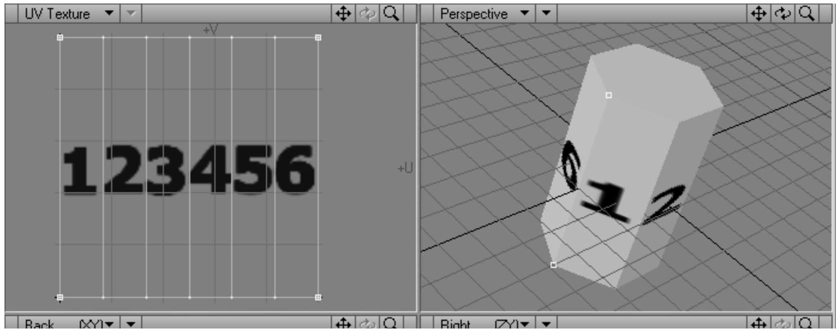
Enter Discontinuous UVs

Below is an example where the UV map was created with discontinuous UVs.

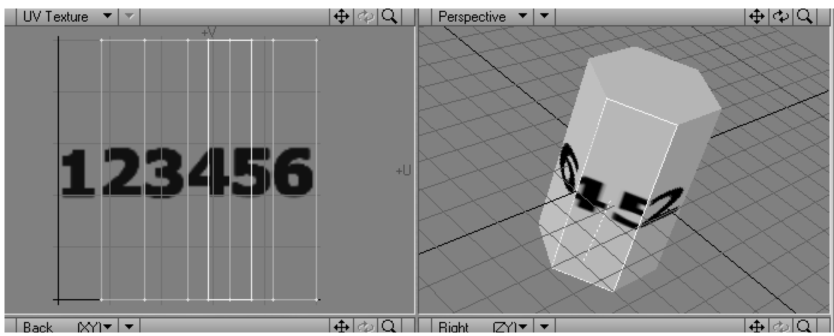


Notice on the UV map, above, how there are no visible points for the left edge of polygon 1. The reason is that the two points on the right edge actually hold two sets of UV positioning data: one set for the right side of polygon 6 and the other set for the left side of polygon 1.

If you use the Unweld command on the two right-edge points, this separates the edge between polygon 6 and polygon 1. Since polygon 1 now has its own left edge, the points become accessible. See below.



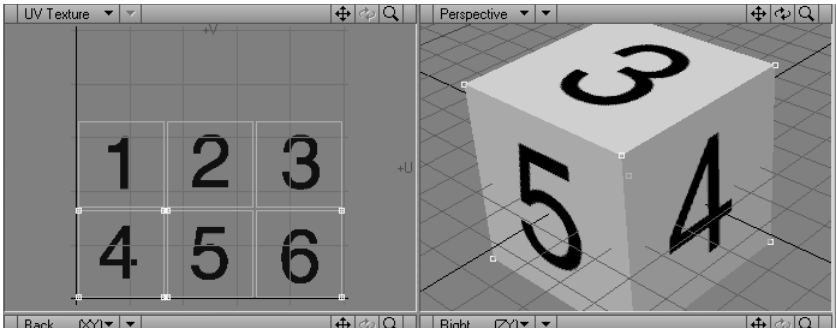
Below, we have unwelded all four points of polygon 1 and then dragged the floating polygon in the UV Texture viewport to straddle the 4 and 5. This demonstrates how a polygon with all points discontinuous can be dragged anywhere without disturbing the UV mapping of other polygons.



NOTE

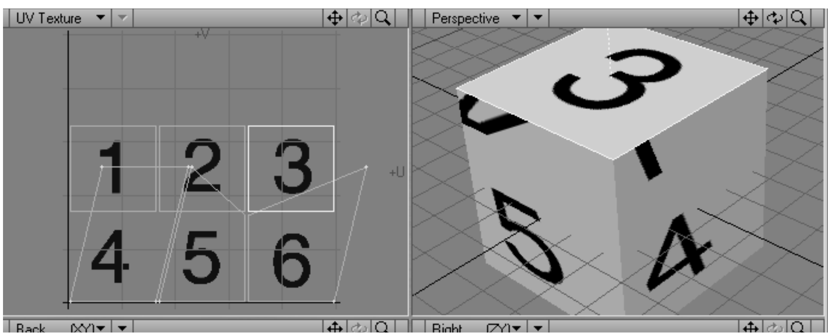
The discontinuous state of VMaps will survive a merge points operation.

Below, the Make UVs Atlas mode was used on a simple box. This illustrates an extreme discontinuous UV situation. Here, each polygon is independent with respect to UV mapping.



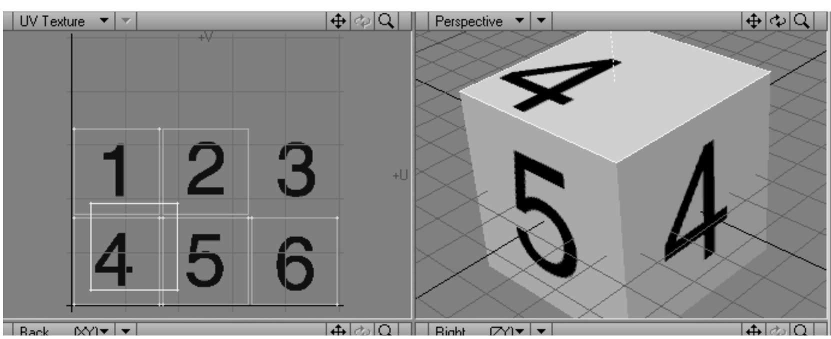
An extreme discontinuous UV situation

If you attempted to select and move polygon 3, the four points that include the discontinuous UV data for polygon 3 also move—messing up polygons 4, 5, and 6. Polygon 3 is unaffected by the move because you cannot change the discontinuous UV data unless you have access to the points (by unwelding).



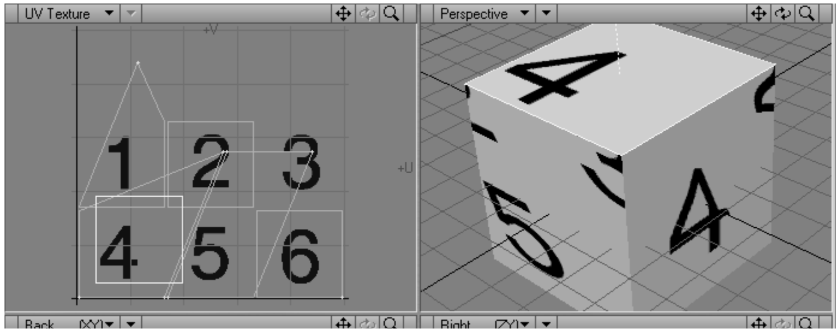
Moving polygon 3 affects other polygons

Now, if we select polygon 3 and unweld it first, we are free to move it anywhere without affecting the other polygons.



After unwelding, moving polygon 3 has no effect on other polygons

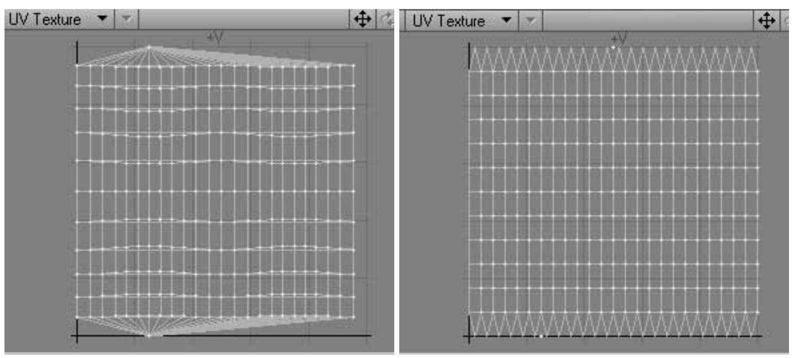
It is not necessary, but let's say we unselect all polygons and merge points. Then, when we reselect and move polygon 3, it again affects the other polygons.



After point merge, moving the polygon affects others

When Modeler creates UV maps, it creates *discontinuous UV coordinates*, as appropriate, to prevent seams in wrap-around maps. It also tries to do a better job at the poles, although polygons that straddle the axis of a rotational mapping will still get distorted.

For example, the UV coordinates should fit the sphere much better, removing seams at the wraparound edge and adding seams at the poles. Images can be mapped this way with much less distortion than before.



Left: Previous version. Right: New version

Unweld Command

The **Unweld** command (**Detail** > Points: **Unweld**) creates multiple copies of the selected points so that none are shared by two polygons. Each polygon is given its own copy of the selected vertices, and VMap values for the polygon are made continuous over the new vertices. This tool is the key to being able to edit discontinuous UVs.

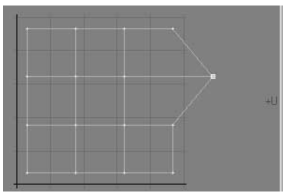
Merge and Weld Commands

When points are merged with the Merge Points (**Construct** > Reduce: **Merge Points**) or Weld Points (**Detail** > Points: **Weld**) command, any introduced discrepancies between VMap values are resolved with discontinuous UVs.

Merge points will no longer merge points that are at different locations in any of their morphs. This prevents distortions of the topology or discontinuities in the morph VMaps. It also provides a method to force seams in the mesh.

UVs Outside the Box

You may have noticed that UV coordinates can appear outside of the *UV box*. That is, have U or V values less than 0 or greater than 1. The texture is essentially *tiled*, so these points are still attached to some part of the texture. It is sort of like the old Asteroids video game where the ship moves off the edge, but appears on the opposite side.

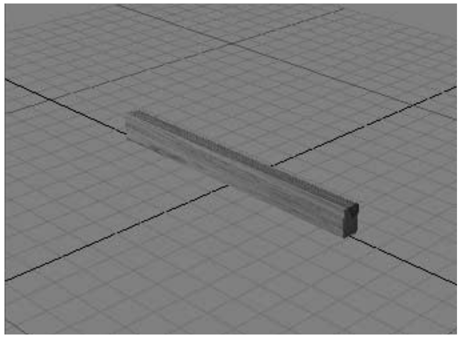


It may help you understand the result by thinking of two rotation keys in Layout of, say, 0 and 720 degrees. From the viewer's perspective, the orientations *at those keyframes* are the same; however, there is a (motion) *path* between the keys that LightWave interpolates. For UVs, there is a texture *path*. LightWave interpolates the texture between the UV coordinates.

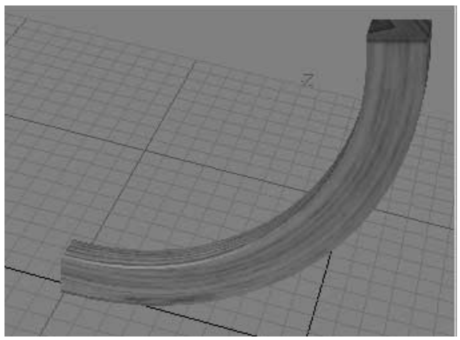
Let's say you had U coordinates at .9 and 1.3. Now although 1.3 and .3 would be at the exact same position on the texture, there would be a big difference in the interpolation path of .9 to 1.3 and .9 to .3. The latter would be backwards—essentially the dreaded seam problem, discussed previously.

UV Mapping in Everyday Modeling

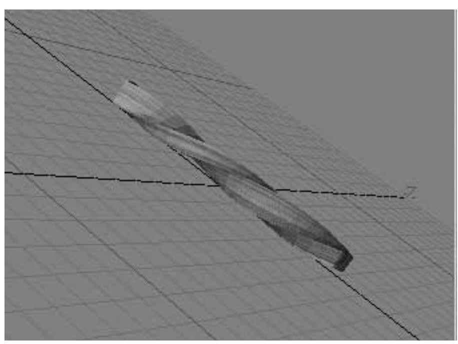
UV mapping is not just for organic irregular surfaces and should be used any time basic projection mapping might be difficult to apply to the finished surfaces. Here is a simple segmented box with a wood image UV map.



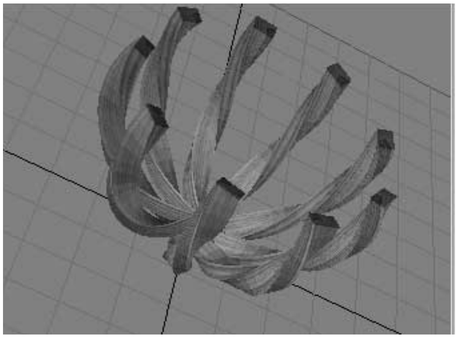
Big deal, eh? A simple cubic image map could do this. But what if you wanted to bend your piece of wood?



Wow! Notice how the grain follows the curvature of the bend. “A little more difficult,” you say. Perhaps it’s more difficult, but I guess you could somehow bend the image in a paint program to match the geometry.



What about adding a little twist?



We've finished *cloning* around with this illustration, but I think you can see just how powerful UV mapping is, even for non-organic objects. The real secret is to plan ahead and create your UV maps early.

PER-POLYGON UV MAPPING

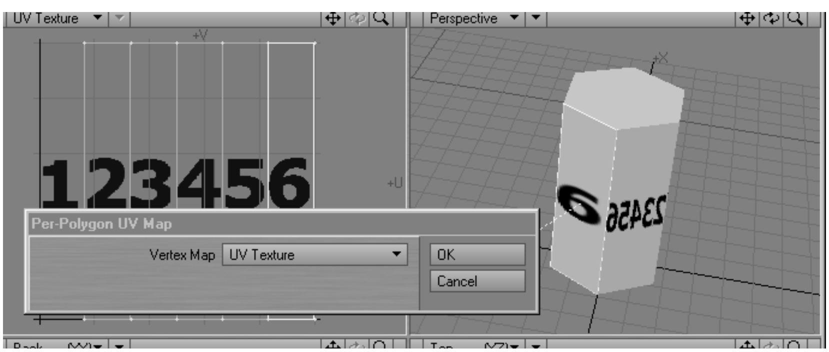
The predecessor to discontinuous UVs is *per-polygon UV mapping* (or *polymap* for short) introduced in LightWave 6.0b.

NOTE
For UV textures created using version 6.5 and after, generally, there is no need to use these per-polygon UV mapping tools.

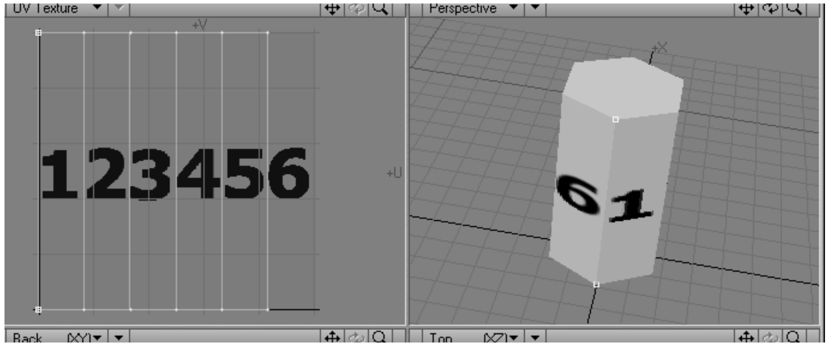
Per-Polygon UV Mapping solved the same problems that discontinuous UVs solve, but in a less elegant and more proprietary manner. It basically overrode the normal continuous UV map used by a texture with new UV values.

Below is an example of using Lock UVs to Polygon:

Here we will use Lock UVs to Poly (**Map** > Texture: **Lock to Poly**) on the 6 polygon because we plan to move the right UV points to fix the 1 polygon. If we didn't do this, the 6 polygon's UV map would get wrecked when we fixed the 1 polygon.



Next, we select the two right points on the UV map and move them to the far left.



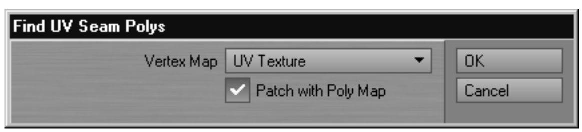
The 1 polygon is now fixed using normal UVs. The 6 polygon continues to look fine since we locked it in before moving the points.

To delete a per-polygon UV map:

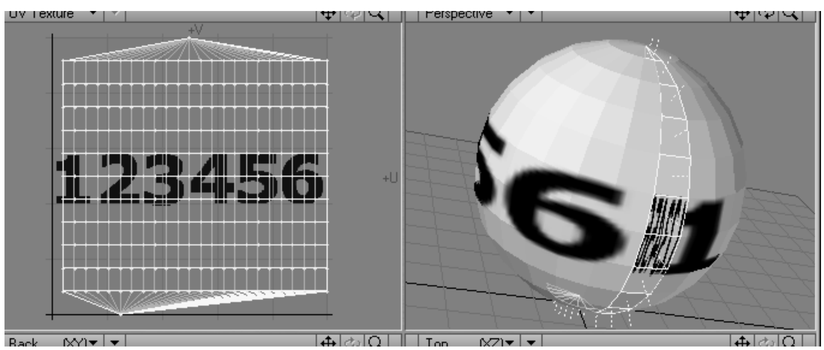
Select the polygon(s) and choose **Map** > Texture: **Clear Poly Map**.

Select UV Seam

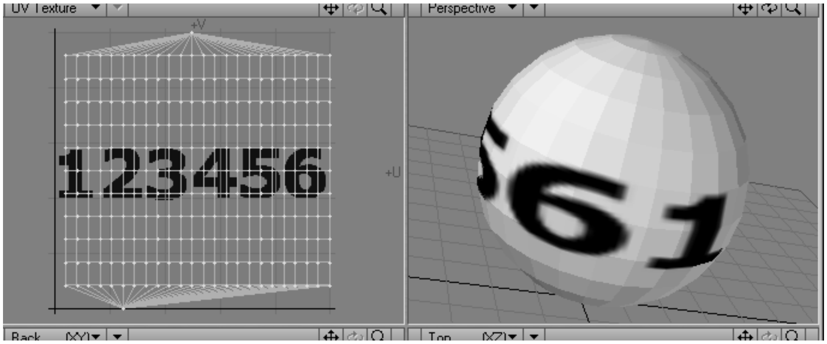
The Select UV Seam command (**Display** > Selection: **Maps** > **Select UV Seam**) automatically selects polygons where the UV values span more than half the image—thus assuming that they are *seam polygons*, discussed previously. If you activate **Patch with Poly Map**, it will automatically turn these into per-polygon UVs.



Below, only the seam polygons have been selected.



Next, you see the seam polygons after using the **Patch with Poly Map**.



Select UV Seam is limited, because it will not set UV values past 1.0 or less than 0. Thus, it is possible that some important portion of your image will get missed (e.g., you really needed a 1.2 U or V value).

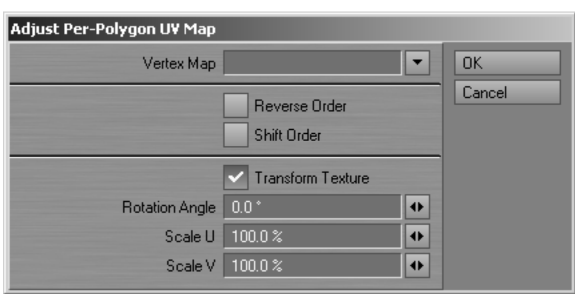
Editing Polymaps

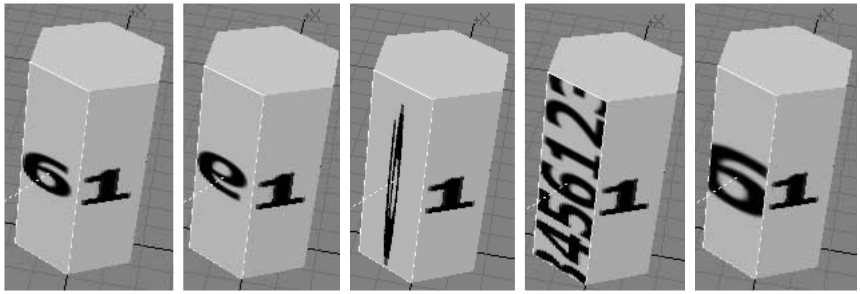
LightWave uses many of the modeling tools used for editing vertex positions to also modify UV values. This works smoothly because the VMaps, like the mesh, are continuous. Moving a point moves it in all the polygons that use it. With polymaps, however, any operation on a vertex also must specify which polygon's view of that vertex is edited. (Polymaps are *locked snapshots* of some UV mapping of a polygon.)

You can select polygons using per-poly UVs by choosing the Select by Poly Map command (**Display** > Selection: **Maps** > **Select by Polygon Map**).

The tools for editing polymaps in the Adjust Polygon Map function (**Map** > Texture: **Adjust Poly Map**) include 2D transformations of the UV coordinates, as well as adjusting the connection between texture points and the polygon's vertices. The power of these operations is limited in comparison to the tools for editing VMaps. Note that you need to select the polygon(s) first.

You can use this function on discontinuous UVs by entering the map's name in the **Vertex Map** field. For polymaps, leave the field blank.





Left to right: 6 is per-poly mapped. **Reverse Order**. **Shift Order**. **Rotation Angle** 90 degrees. **Scale V** to 60%

A quick method for editing a highly discontinuous texture is to use the **Unweld** command to duplicate all the points in your model, so that no polygon shares a point with any other. After this is done, you can move entire textured polygons anywhere over the texture image, without pulling the neighboring polygon's texture.

NOTE
Unweld removes per-polygon UV mapping. Then, you can use the normal Modeler tools, which support discontinuous UVs.

Leaving the model physically discontinuous like this is usually undesirable, but assuming the points haven't been moved, the **Merge** command will automatically fuse these points. Merging the points will, of course, destroy their independent VMap values, so Lock UVs to Polygon (**Map** > Texture: **Lock to Poly**) should be used on the polygons prior to the re-merge.

Unwelding all points is an extreme, but effective method. Unwelding selected points and locking the appropriate polygons individually is also quite feasible on low-polygon models, or where there are few discontinuities.

Selected per-polygon UV maps can also be converted to discontinuous UVs by choosing **Map** > Texture: **Poly Map to UVs**.

Using Separate *Editing* UV Maps

Another approach to editing is to copy the main UV map and edit the copy. You fix just the problem polygons by locking them based on the copy UV map. This approach assumes that most of the polygons, the non-fixed ones, will be fine with the original UV map. You can make subsequent changes to the per-polygon UV mapped polygons without affecting other polygons.

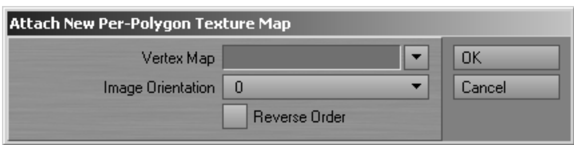
HINT
You may want to use a different surface name on your problem polygons—while editing—that points to the UV map copy. After the p-map has been locked, you can apply the regular UV mapped surface to

it. (Remember, since there is only one per-polygon map per polygon, it doesn't matter which UV map is used in the surface texture.)

Quad Polygon Maps

A convenient use of polymaps is to simply apply an entire image to a polygon. This makes sense in the case of polygons, which like images, are four-sided (i.e., quadrangles or *quads*). These are particularly useful in the case of a low-polygon model where the face of a model matches the textured parts, and every polygon edge is also a texture seam.

The New Quad Polygon Map command (**Map** > Texture: **New Quad Polygon Map**) generates polymaps like these easily. It will take selected quads and assign their corners to the four corners of UV space. If you enter a name in the **Vertex Map** field, discontinuous UVs are created. (If the name is already a UV map for the object, it will be altered. Otherwise, a UV map will be created.) If the field is blank, a poly map is created instead.

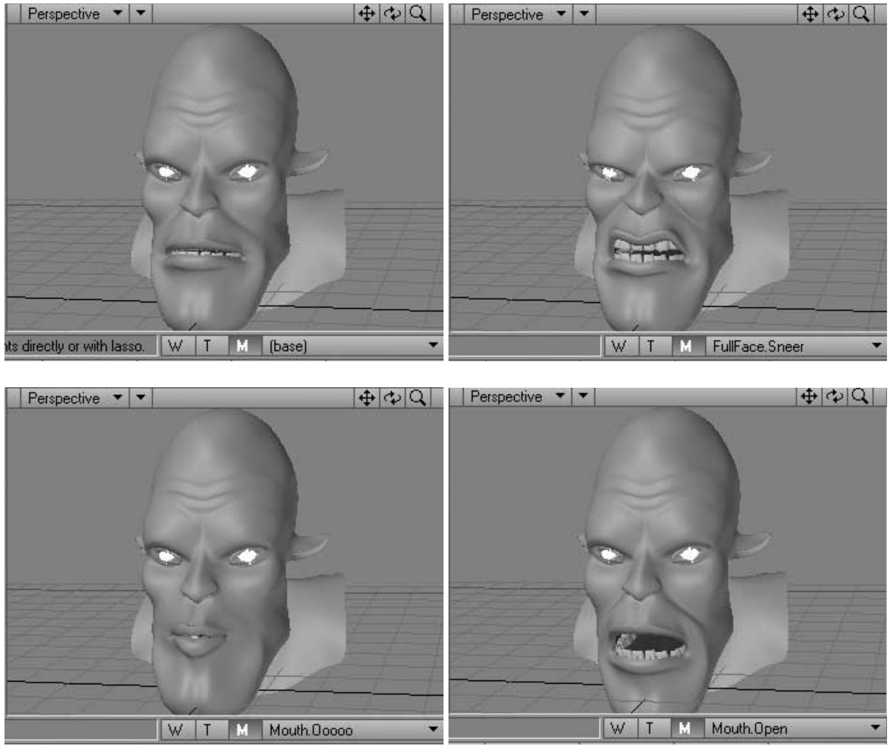


Since the choice of which UV corner goes to which quad corner is basically arbitrary, the order and orientation of the image can be selected. It will likely take a little trial and error to get the results you want.

The Polygon Normal UVs command (**Map** > Texture: **Poly Norm UVs**) projects the texture UVs along the polygon normals, mapping the image flat on the normal. It is something like a per-polygon atlas mode or the New Quad Polygon Map operation performed on non-quads. The results on a cube would be the same as cubic image mapping.

ENDOMORPHS

Sometimes, deforming an object using bones does not provide the accuracy you may need for a particular project. LightWave also lets you force an object to change its shape using another object. This is called *morphing*. Morphing provides absolute control over the final shape, something displacement maps and bones usually cannot offer. You may wish to use morphing to move the muscles in an arm or face, to get a flower to bloom, or to transform a car into a plane. The different states or poses are generally referred to as *morph targets*. You might think of them as different *poses* of the same object.



This object is in the LightWave6_5\[6] Projects\Character\EndomorphHead directory. It's called StuHead_WithEndomorphs.lwo

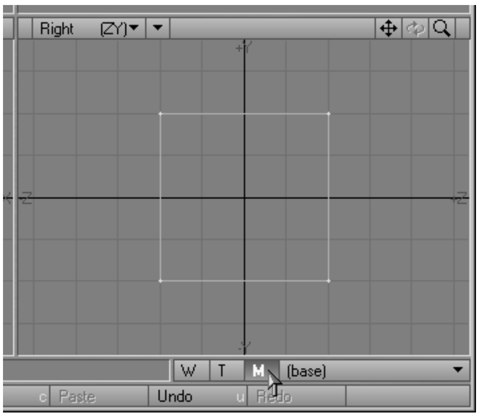
You handle morphing through *morph maps*, which are special VMaps (yes, just like point weighting). Each morph target defines different positions for each associated base-object point. All of the morphing information is contained in a single object file, called an *endomorph*.

LightWave has two types of morph targets. **Relative** targets are affected relatively for any changes to the base object. **Absolute** targets are not affected by <u>positional</u> changes to the base object, although changes to the number of points are always reflected.

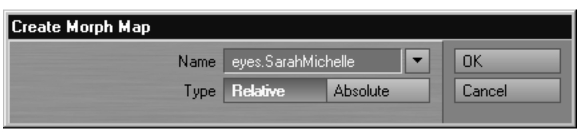
NOTE
See Chapter 9 for information on how to animate Endomorphs.

To set up a morph map:

1. Create your base object in Modeler.
2. Click the **M** button in the lower-right corner to activate the morph map mode. (base) should become selected in the selection pop-up menu, which appears to the right of the WTM buttons.



3. Select **(new)** from the selection pop-up menu to the right. The Create Morph Map dialog will appear. Enter a name using a *group.pose* format, something like HEAD.SMILE or EYE.CLOSE and set the **Type**.

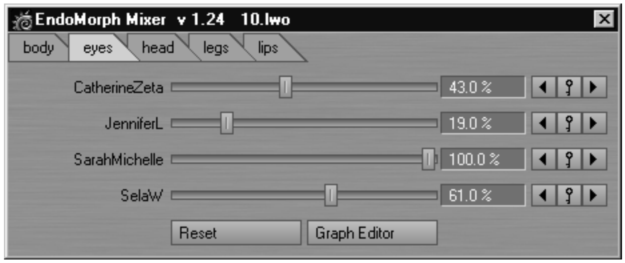


The pop-up menu on the Create Morph Map dialog lists existing targets.

NOTE
If you change an existing target's **Type**, it applies only to changes after that time.

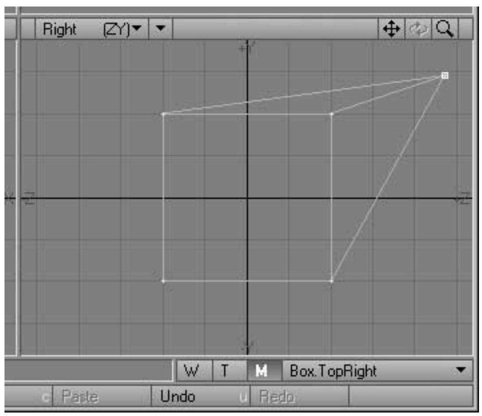
Poses with the same group appear on the same tab of the control interface called Morph Mixer.

NOTE
Below is an example Morph Mixer panel shot (see Chapter 9, “Animating Endomorphs”).

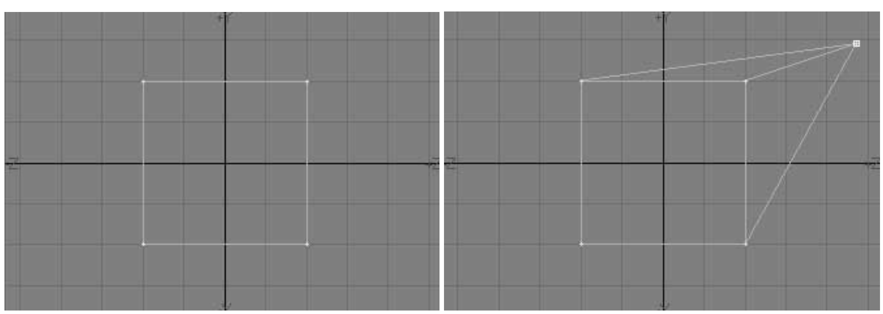


4 Click **OK** to create the map. The new map will appear in the selection pop-up menu.

5 Modify the new map in some way.

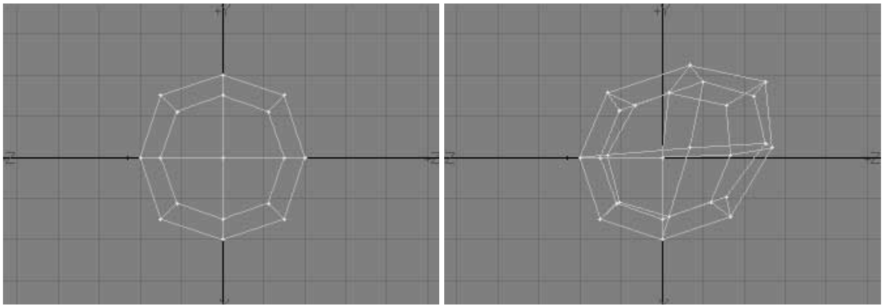


Now, if you use the selection pop-up menu and go between the (base) object and the morph map you created, you will see that the (base) is not affected by the change to the morph map.



Left: (base) object. *Right*: morph map

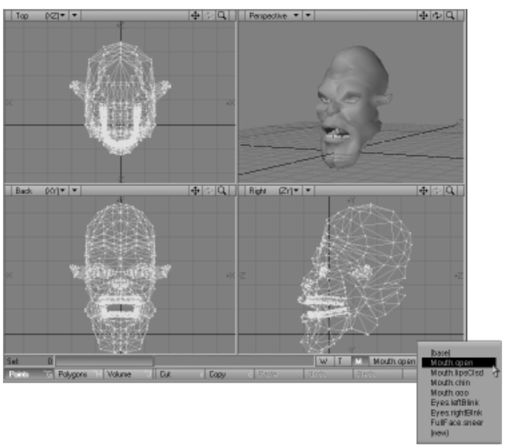
However, if you used the metaform command (which adds geometry), for example, that change is reflected in the (base) object.



Left: (base) object. *Right*: morph map

WARNING
We strongly suggest that you do modifications, like adding and subtracting geometry, on the base object and use the morph maps only for posing (i.e., moving points). Otherwise, you can end up with incorrect point orders.

Below is an example using a more complex object with several morph maps.



Inserting a Layer into a Morph Map

You can insert a background layer into a Morph Map using the **Bkg to Morph** command (**Map** > Morph: **Bkg to Morph**). Simply place the geometry you want to insert into the map in the background. Then execute the command. A dialog prompts you for a map name. Point count and order must still match, just as in any morph target.

HINT

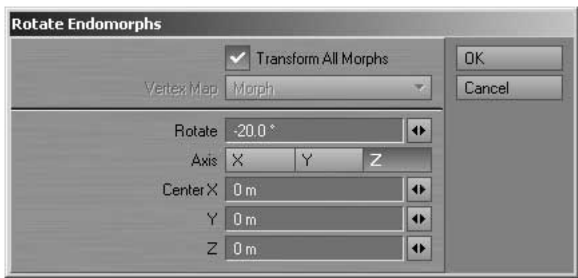
You can use the Bkg to Morph command to insert an object saved with Layout's Save Transformed Object command (**File** > **Save > Save Transformed Objects**) that has been deformed in Layout.

Merging Morph Maps

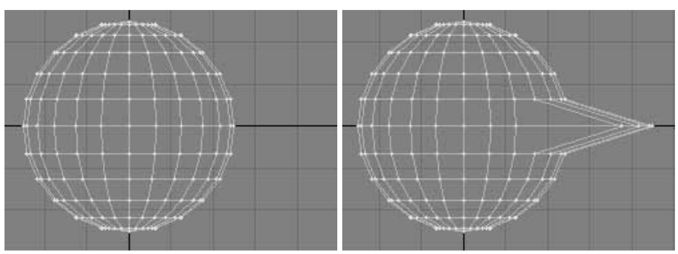
Use the Apply Morph command (**Map** > Morph: **Apply Morph**) to merge a specified percent of a (relative) morph map into the base. Note that all relative morph maps—including the one selected—are affected by changes to the base.

Rotate Morph Function

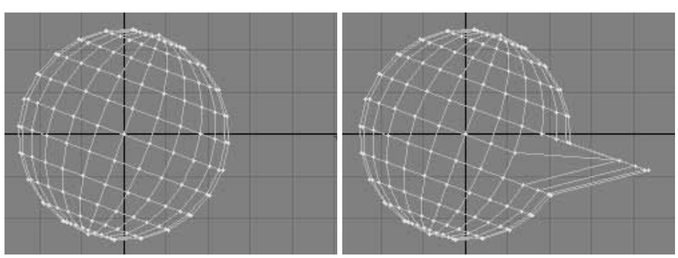
Rotating the base of an EndoMorph with the normal Rotate tool often does not add the proper amount of rotation to its morph targets. You may achieve better results with the Rotate Morph function (**Modify >** Rotate: **Rotate Morph**).



Here is an example:

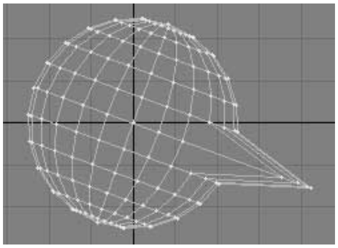


Left: Base object. *Right*: Morph target



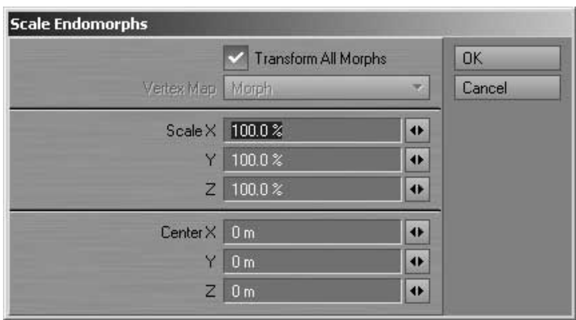
Left: Base rotated with the Rotate tool. *Right*: Morph target

Notice how the extended points rotate somewhat—basically, they have moved only the amount that those points moved in the base object with the rotation. Probably not what you expect nor want. Below is the morph using Rotate Morph instead, with the same settings used to rotate the base.

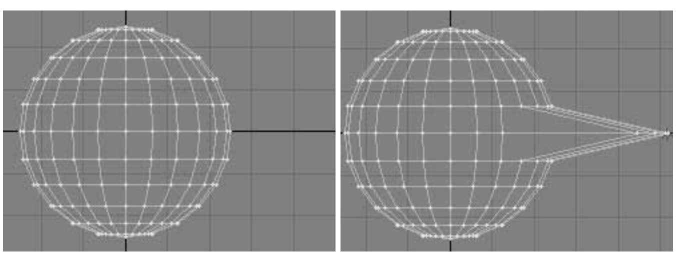


Scale Morph Function

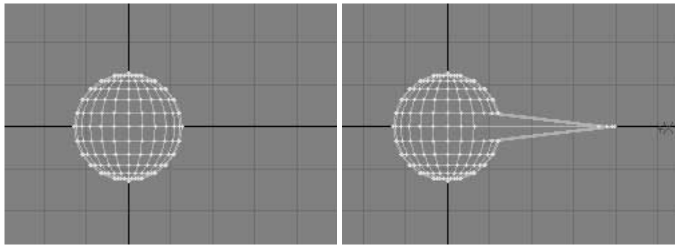
Scaling the base of an EndoMorph with a normal sizing tool may not add the proper amount of scaling to its morph targets. You may get better results with the Scale Morph function (**Modify >** Stretch: **Scale Morph**).



Here is an example:

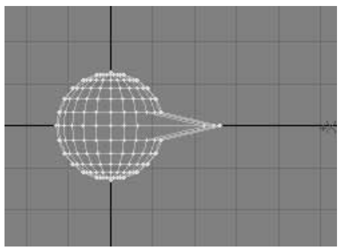


Left: Base object. *Right*: Morph target



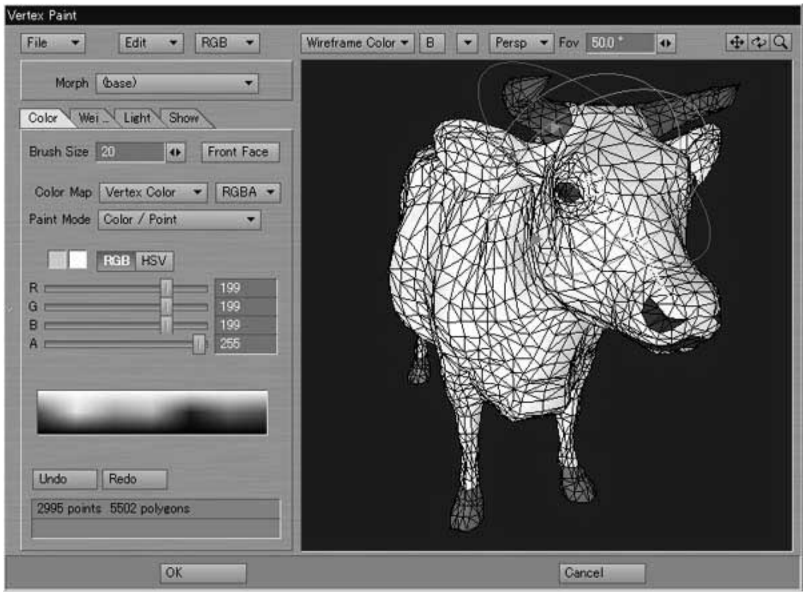
Left: Rotated base. *Right*: Morph target

Notice how the extended points scale somewhat—basically, they have moved only the amount that those points moved in the based object with the scaling. Not what you expect nor want. Below is the morph using Scale Morph instead, with the same settings used to scale the base.



VertexPaint

VertexPaint (**Map** > Weight & Color: **VertexPaint**) is a plug-in for painting vertex color maps and weight maps on bones. The display window gives you a 3-D view of the object loaded into Modeler. You paint your object in this view.



Main VertexPaint panel

NOTE

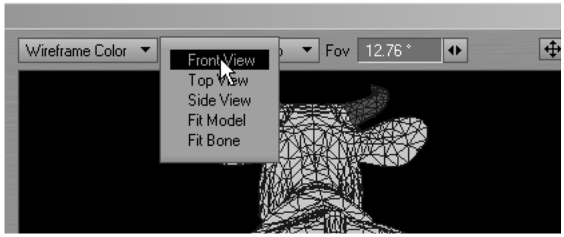
References to *bones* really means *skelegons*, since (technically) bones cannot be used in Modeler directly.

The view controls operate just like Modeler's, so dragging with Alt key rotates, Shift + Alt pans and Ctrl + Alt zooms. There are also corresponding drag buttons in the upper right corner.

Two types of projections are available for displaying the object in 3D space: **Persp** (perspective), the default and **Ortho** (orthogonal). Sometimes painting vertex colors is easier using the orthogonal mode.

When using the perspective mode, you can modify the zoom factor using **Fov** setting. Higher values simulate a telephoto lens zoom and lower values simulate a wide angle lens.

You can quickly switch to certain preset views by choosing one from the pop-up menu (down arrow) just to the left of the projection mode pop-up menu.



Preset views pop-up menu

Component and Blend Display

LightWave uses the point's alpha information as a mixed value ratio for blending point and surface colors. The formulas look like:

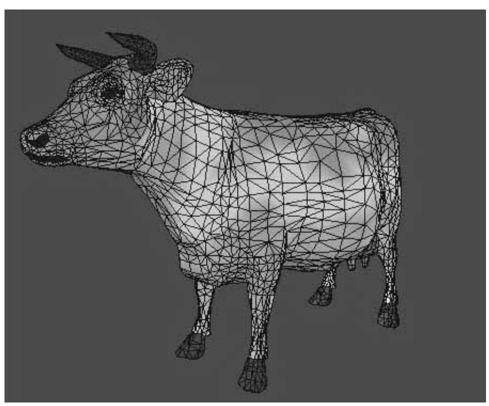
Color.Red = Vertex.Red * Vertex.Alpha + Surface.Red * (1.0 - Vertex.Alpha)

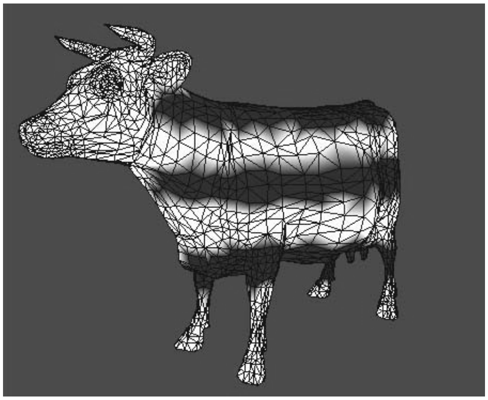
Color.Green = Vertex.Green * Vertex.Alpha + Surface.Green * (1.0 - Vertex.Alpha)

Color.Blue = Vertex.Blue * Vertex.Alpha + Surface.Blue * (1.0 - Vertex.Alpha)

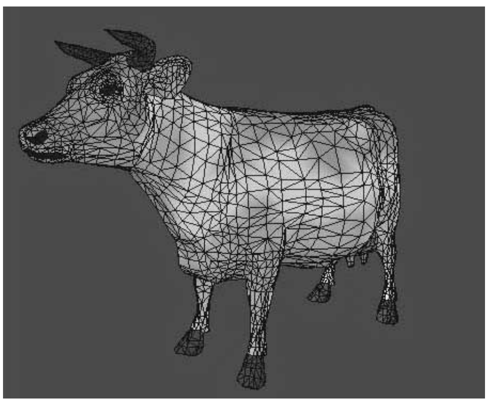
When you set the rendering mode to **Vertex Color** or **Wireframe Color**, pressing the **B** (blend) button will display both the point color and alpha channel.

When you paint, you apply both color and alpha values when VertexPaint is in blend mode only by pressing the **B** button. Otherwise you apply color or alpha values based on the mode set by the pop-up menu just to the right of the **Edit** pop-up menu button. This is the *component mode* (i.e., **B** button off).


When in component mode and state is set to RGB paint mode, only the point color is rendered



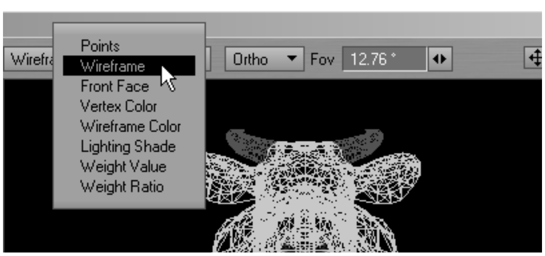
When in component mode and state is set to Alpha paint mode, brightness of alpha is rendered, with black as 100 percent transparent and white as opaque.



When in blend mode and state is set to either RGB paint or Alpha paint mode, the point color and the surface color are mixed when rendered.

Rendering Modes

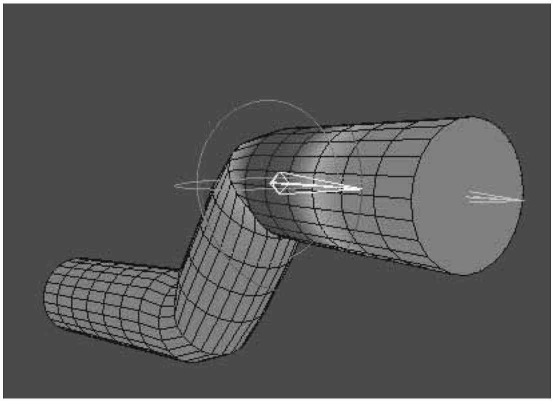
You can view the object in various rendering modes using the pop-up menu just above the window on the left.



Render modes pop-up menu

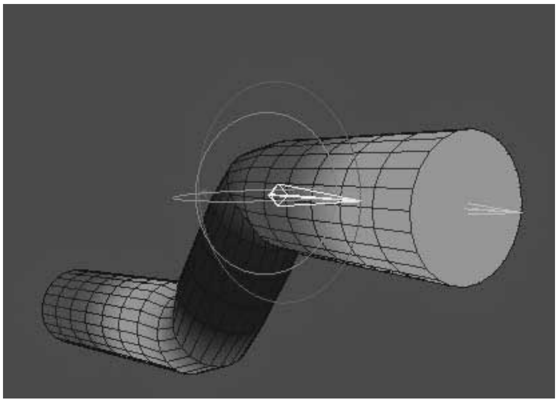
Points The points that form the object are rendered. The color of the point is set to the point color. Lights are not used.

Wireframe	The object is rendered as a wireframe. The point color is set by the points of each line. When the color at the start of the line differs from the color at the end, it is displayed as a gradation. Lights are not used.
Frontface	Only the polygons with surface normals that face the camera are rendered as wireframe. Lights are not used.
Vertex Color	The object's polygon is rendered using the color information from the current vertex color map. Lights are not used.
Wireframe Color	The object is rendered using the color information from the current vertex color map with an overlay of wireframe lines. Lights are not used.
Lighting Shade	The object is rendered using the lights set in the light mode. The color attributes for the polygon surfaces are used but the vertex color map is not rendered.
Weight Value	The weight value set for the current weight map is displayed in red. Unaffected areas are displayed in gray. The closer the gradation is to gray from the maximum red, the less influence the bone has on those points. This is a convenient mode to view your weight map when you have finished painting the map.



Weight Value render mode

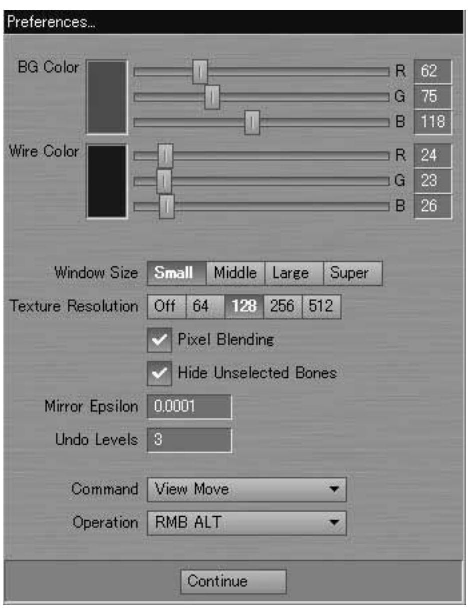
Weight Ratio The ratio of influence on each point by each bone is rendered using a color gradation. When no bone exerts influence over a point, it is rendered as gray. Before bone influence is calculated, the colors red, blue, and green are repeated in order. Each point color is compensated on the basis of the percentage of influence of the bone.



Weight Ratio render mode

Preferences

When you select **Preferences** from the **File** menu, you can customize many settings.



VertexPaint preferences

Changing the RGB values for the **BG Color** will modify the background color of the 3-D view. When the Preference panel is moved or closed, the view will be updated to reflect a change. This setting is saved between sessions.

The second item modifies the RGB values of the **Wire Color** in the 3-D view. This affects the following rendering styles: **Wireframe Color**, **Weight Value**, and **Weight Ratio**. When the Preference panel is moved or closed, the view will be updated to reflect a change. This setting is saved between sessions.

The **Window Size** option will change the size of the VertexPaint window. You must exit and restart VertexPaint for the changes to take effect. If you choose a size larger than your desktop, parts of the screen will not be visible.

Hide Unselected Bones specifies how unselected bones are displayed

Mirror Epsilon is the error quantity used when you paste bone weight values symmetrically on the X axis. (See “Copying and Pasting Weight.”)

Undo Levels is the maximum number of undos in Color and Weight mode.

Default Keyboard/Mouse Operations

The chart below summarizes keyboard/mouse operations.

Operation	**Command**
LMB + ALT	Rotate view
LMB + CTRL + ALT	Zoom in/out of view
LMB + ALT + SHIFT	Pan view
LMB	Rotate view (using Rotate button)
LMB	Zoom in/out of view (using Zoom button)
LMB	Pan view (using Move button)
LMB	Paint point color (Color mode)
RMB	Set the size of the color paint brush (Color mode)
LMB + CTRL + SHIFT	Select color from 3D view, like eyedropper (Color mode)
LMB	Select color from color palette (Color mode)
LMB	Paint bone weight (Weight mode)
LMB + CTRL	Rotate bone
LMB + SHIFT	Select bone
RMB	Set the size of the weight paint brush (Weight mode)
LMB	Move Lights (Light mode)

LMB + CTRL + SHIFT	Select color from 3-D view (Light mode)
LMB	Select color from color pallet (Light mode)
LMB	Select polygon (Show mode)
RMB	Deselect polygon (Show mode)
LMB	CTRL-+ SHIFT Rectangle selection of polygon (Show mode)
RMB + CTRL + SHIFT	Rectangular deselect of polygon (Show mode)
a	Center object
A	Center current bone
,	Zoom out
.	Zoom in
F	Front view
T	Top view
S	Side view
F1	Points Render mode
F2	Wireframe Render mode
F3	Front Face Render mode
F4	Vertex Color Render mode
F5	Wireframe Color Render mode
F6	Lighting Shade Render mode
F7	Weight Value Render mode
F8	Weight Ratio Render mode
u	Undo
up arrow	Select next bone
down arrow	Select previous bone
c	Copy current bone weight
v	Paste current bone weight
V	X axial symmetrical paste of current bone weight
x	Clear current bone weight
+	Scale up current bone weight
-	Scale down current bone weight
r	Reset current bone to rest position
R	Reset all bones to rest position

Customizing Keyboard/Mouse Operations

You can change keyboard/mouse operations, by selecting the command with the **Command** pop-up menu and then the corresponding keystrokes on the **Operation** pop-up menu.

If the keyboard/mouse operation is already being used by another command, the operations for the two commands will be swapped. Close the Preference panel to set the new Command+Operation combination.

Modifying Modes

VertexPaint offers four operating modes. You change modes by selecting the relevant tab on the left side of the interface. The contents of the **Edit** menu will change to the commands for the current mode.

Color tab	Vertex color map edit mode
Weight tab	Bone weight edit mode
Light tab	Light settings mode
Show tab	Polygon show/hide setting mode

Setting the Morph Map

If you are using EndoMorphs in Modeler, this menu lets you change the object deformation state to reflect the selected Morph map. If there are no Morph maps in the object, the drop-down menu will be inactive.

Color Mode Tab

Before you can begin painting points, you must first define at least one vertex color map. If a color map is already applied to the object, VertexPaint sets the first color map as the current map.

NOTE
If there is no information in the vertex color map, it will not be loaded into VertexPaint. If you want to define your vertex color maps in Modeler, you must first add some values (color) to the map.

To create a new color map in VertexPaint, you must be in Color mode. Choose **Edit** > **Create Vertex Color Map**. A dialog box appears where you name your vertex color map (this name must be unique). Then click the **OK** button. If the name already exists a number is appended to the name.

To modify the name of the current vertex color map, choose **Edit** > **Rename Vertex Color Map**.

To delete a vertex color map, set the map you wish to delete as the current map and choose **Edit** > **Delete Vertex Color Map**.

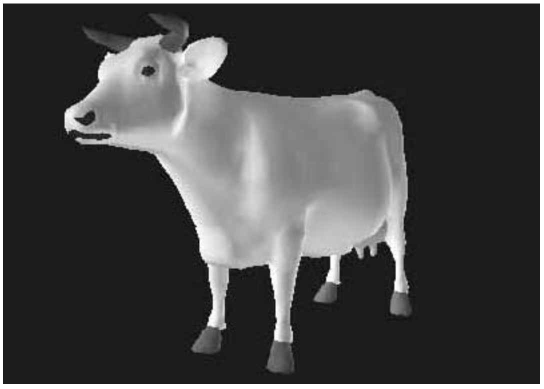
There are two types of vertex color maps: **RGBA** (color + alpha channel) and **RGB** (color only). The first includes alpha channel information, the other does not. VertexPaint defaults to RGBA, but you can change the type using the pop-up menu. Modeler will not be updated until you close VertexPaint.

You can reset the point color of the selected map to the surface color of the object by choosing **Edit** > **Reset To Surface Colors**.

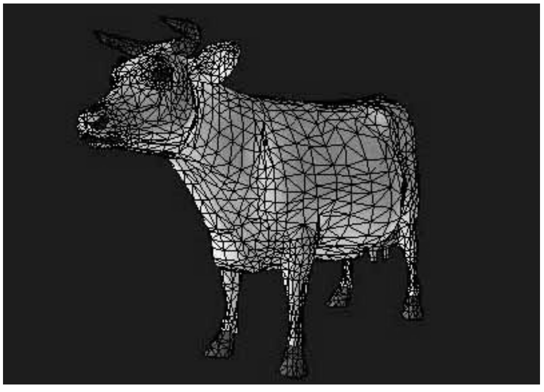
Baking the Color

Vertex paint can use the light information set in the Light mode tab to bake (OpenGL) color information into the color map. Before baking the surface to a color map, you can verify the results using the **Lighting Shade** rendering mode. To adjust the way the object is lit, use the options on the Light tab to change the strength and position of lights used to calculate the light mode.

Select the color map for which you want to apply the baked information, and choose **Edit** > **Burn Shading Colors**. Note that the whole object must be within the 3-D view for this to work properly. Any portion of a polygon outside the 3-D view will not receive point color value and possible errors may occur.



Light calculations. This is viewable in the Lighting Shade rendering mode.



After choosing Burn Shading Colors, the baked point color will be rendered, as seen here in Wireframe Color rendering mode:

Point Color

When you use the brush, there are three **Paint Modes** available for painting point color.

Color / Paint	Paint a single point
Color / Index	Paint a polygon point index unit
Color / Polygon	Paint a single polygon unit

To start painting, select a color map from the **Color Map** pop-up menu, then drag your LMB over the object in the 3-D view. You can use either the color sliders or the palette to select the color you want for painting.

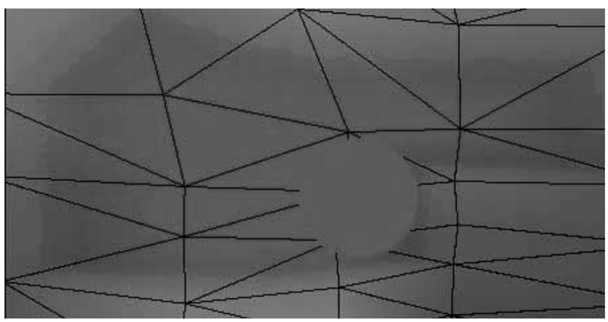
The slider operates in two modes **RGB** and **HSV**. Each slider affects the numeric entry field to the right of its location. Values for all sliders are displayed in a range from 1 to 256, although floating point values are used internally by VertexPaint.

The palette gives the user a visual way to select the active color for painting. Use the LMB within the palette to select a color. Use the slider to select alpha colors.

You can also select the active color directly from the object in the 3D view. This works similarly to an eye-dropper tool. To do this, hold the CTRL + SHIFT keys simultaneously and click on the color using the LMB. The color is displayed in the active color selection box.

Painting a Point

When you select the paint mode **Color / Point**, painting will affect points. The **Brush Size** and **Front Face** parameters affect the painting of a point.



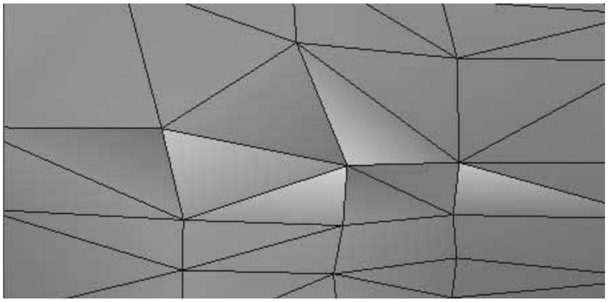
Color / Point mode

The brush size is represented by a disc, which is the color of the active paint color, and sits below the mouse pointer when you press the LMB in the 3-D view. To adjust the brush size, use the **Brush Size** slider and numeric entry field under the mode tab. You can also drag your RMB in the 3-D view to adjust interactively.

When you paint, all points receive paint whether they face you or not. The paint brush will pass through the object and paint points on the opposite side. If you activate the **Front Face** button (to the right of the **Brush Size** slider), only the front face will receive color.

Painting a Polygon Point Index

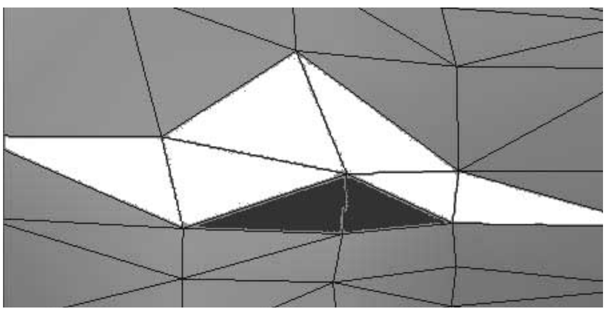
When you select **Color / Index**, you apply paint based on the polygon point index. When you paint in this mode, only the point index of the polygon surface receives paint. **Brush Size** and **Front Face** settings are ignored. Which polygon point index receives color is based on the polygon nearest to the cursor when paint is applied.



Color / Index mode

Painting a Polygon

When you select **Color / Polygon**, you apply paint to a polygon. When you paint in this mode, only the polygon surface receives paint from the brush. **Brush Size** and **Front Face** settings are ignored. When paint is applied, the polygon beneath the mouse pointer receives color.



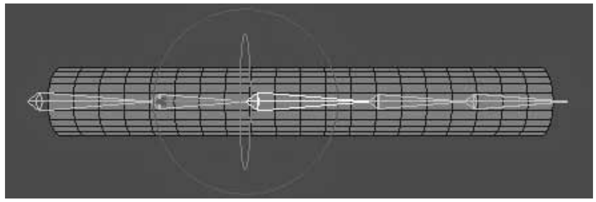
Color / Polygon mode

Weight Mode Tab

When you select a bone (i.e., skelegons), it is highlighted in yellow. By default, unselected bones are hidden. You can display all bones, active and inactive, by deactivating **Hide Unselected Bones** in the Preferences panel. Unselected bones are displayed as white.

VertexPaint offers three methods for selecting bones:

- In Weight mode, select the name of the bone from the **Bone** pop-up menu.
- Use your arrow keys to cycle through the next and previous bone.
- Hold the SHIFT key on the keyboard and click with the LMB to select a bone directly in the 3D view

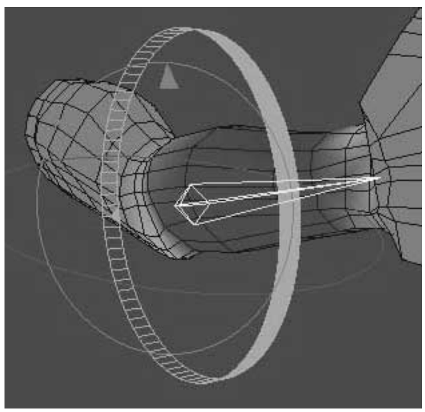


Bones as they are displayed when Hide Unselected Bones is deactivated

In addition, you can rotate a bone more easily when it is centered in the 3-D view. To center the currently selected bone, press the A key or select **Fit Bone** from the pull-down menu (represented by a downward triangle).

Rotating Bones

VertexPaint lets you test bone deformation on the object while you work on weight maps. The selected bone is displayed with HPB rotation controls. The display rings affect rotation as follows: red - Heading (Y axis), green - Pitch (X axis), blue - Bank (Z axis).

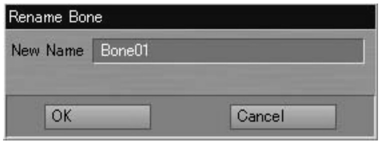


While holding the CTRL key, drag your LMB to rotate the bone. To rotate only one axis, drag right on top of an axis. The selected axis will be indicated by a thick ring.

To reset a bone to its default state (Rest Position), make sure the Weight tab is selected. Then choose **Reset the Bone** from the **Edit** menu. To reset all bones, select **Reset All Bones**.

Renaming Bones

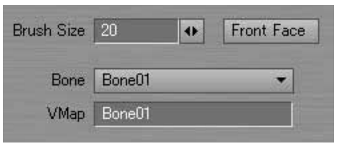
To rename a bone, choose **Edit** > **Rename Bone** in Weight mode. Enter the new name for the bone in the dialog and press the **OK** button. When you exit VertexPaint, a dialog box will ask you to update Modeler. Press **OK** and the changes will take effect in Modeler.



Bone renaming dialog

Renaming Weight Maps

To rename a weight map, edit the name of the weight map directly in the **VMap** field. When you exit VertexPaint, a dialog box will ask you to update Modeler. Press **OK** and the changes will take affect in Modeler.



Name field of weight map

Copying and Pasting Weight

To copy the weight values of the current bone to the VertexPaint clipboard buffer, press the C key or choose **Edit** > **Copy**. To paste the weight values to the weight map of the selected bone, press the V key or choose **Edit** > **Paste**.

You can also paste the weight values to another bone symmetrically on the X axis . Select the bone you wish to paste the values to and press SHIFT + V or choose **Edit** > **Paste-X**. When the weight map value is pasted, the point coordinate is searched for symmetrical positions along the X axis.

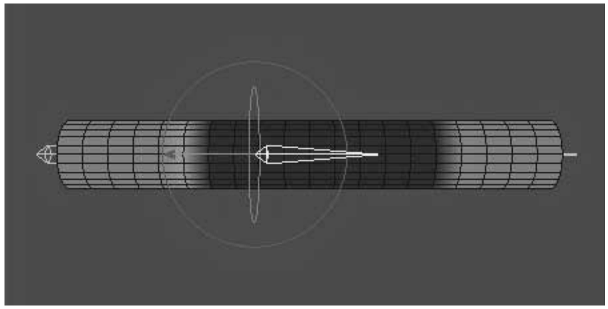
NOTE
You can set the computing error quantity when searching the axial symmetrical point by setting its value in the **Mirror Epsilon** setting found in the Preferences panel.

Clearing Weight

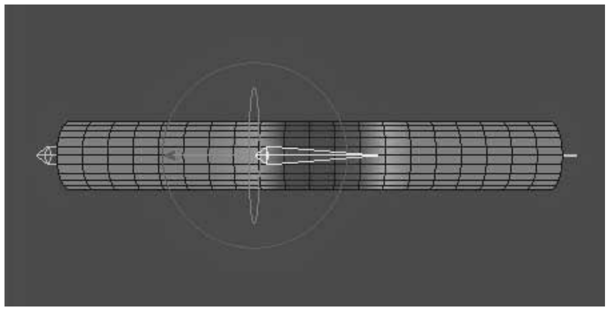
To clear all values from the weight map of the current bone, press the X key or choose **Edit** > **Clear**.

Normalizing Weight

When you use the paint feature in VertexPaint to calculate bone weight for Modeler, the values can sometimes exceed the range of 0.0 to 1.0. When you normalize the weight value (to a range of 0.0 to 1.0), the bone is easier to control. To normalize the weight values, choose **Edit** > **Normalize**.



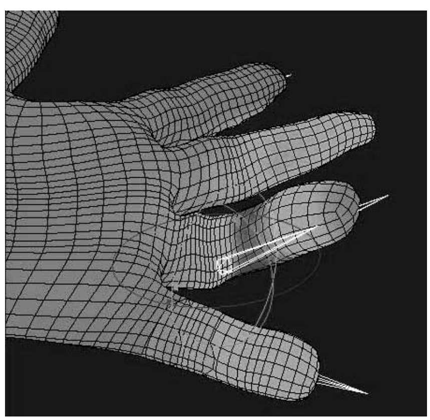
A weight map that has not been Normalized



A weight map that has been Normalized

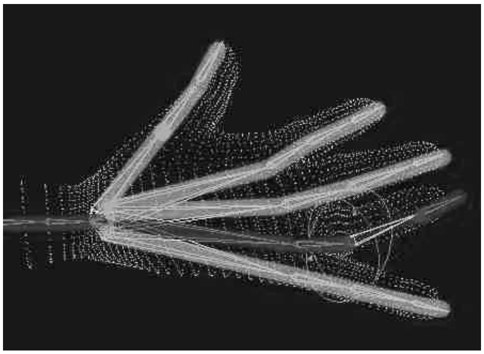
Trimming Weight

Sometimes independent bones can cause a divergence across the object by deforming areas controlled by other bones. This occurs when the weight maps for the independent bones were created using the calculated weight falloff function (see “Calculating Weight”), for example, when you arrange bones for the fingers of a hand. When falloff is calculated, only distance to the bone is measured to set the point weight value.



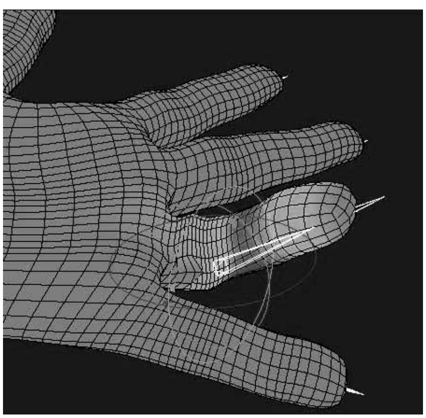
Before applying Trim Branch. Because there is a weight influence from the bone of the ring finger on the middle and little finger, when the ring finger is bent the middle and little fingers are also deformed.

When you choose **Edit** > **Trim Branch**, it trims the bone weight values along the route of the bone system for the selected weight map. It first seeks the primary bone of each point (the point to which the bone is closest). When a point is found that is not part of the selected bone system, it is selected and the weight value is cleared for that point in relation to the current bone. The route of the bone system and its divergence from the parent is shown below.



System of the selected bone. The darker bones represent the system of the selected bone, whereas the lighter bones represent the bones of other systems.

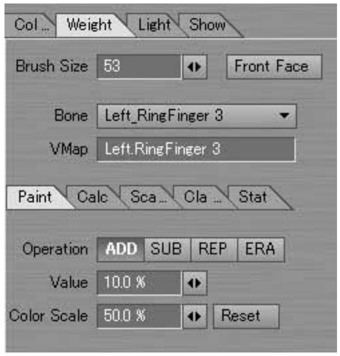
When you perform this command on a bone divergence such as the joint of a finger, sometimes the points in the joint will not deform smoothly. We recommend that you execute the command first along only certain bones of the divergence. In addition, trimming may not work well depending on the arrangement of the bones. If this is the case, use the ERA and SUB functions of the weight brush to manually adjust the value of the weights.



After applying Trim Branch. The weight values that affected the middle and little fingers are cleared of their value, so the ring finger can now be deformed independently without affecting the other fingers.

Painting Weight Values

To paint weight values in Weight mode, select the Paint subtab. Different input fields and operations are displayed for painting weight values. Choose an operation and enter your desired values, then drag your LMB in the 3-D view over the points you wish to paint.



The **Brush Size** and **Front Face** parameters function the same as when you paint point color.

Operation lets you set the type of operation that occurs when you paint weight values along the object's points.

The **ADD** operation adds the value set in the **Value** field to the weight map along the object. Drag your LMB over the points where you wish to modify the value. The change is applied once to each affected point. To modify the point value more than once, release the LMB and repeat the process.

The **SUB** operation subtracts the value set in the **Value** field to the weight map along the object. Drag your LMB over the points where you wish to modify the value. The change is applied once to each affected point. To modify the point value more than once, release the LMB and repeat the process.

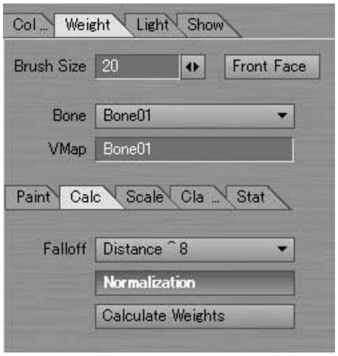
This **REP** mode replaces the current weight value with the value set in the **Value** field.

The **ERA** mode erases the value of the weight for the affected points and sets them to 0.0.

The **Color Scale** value adjusts the display strength of the red color gradation in the **Weight Value** rendering mode. Adjusting this value does not affect the influence exerted in the actual weight map.

Calculating Weight

To calculate the weight map using falloff in weight mode, select the Calc subtab. The parameter fields used to calculate weight are displayed. These parameters work in basically the same way as BoneWeight in Modeler.



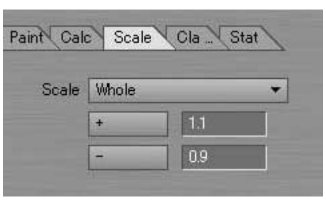
Falloff sets the method for calculating the value of the weight map. With **Linear**, the points closest to the bone receive a value of 1.0, all others are set to 0.0. With **Distance**, the point weight value is set to the inverse number of the distance from the bone, creating a gradation of influence. With **Distance ^ 2**, the point weight value is set to the inverse number of the distance to the second power from the bone, creating a gradation of influence, and so on.

When **Normalization** is active, the value of the weight map is calculated using a range from 0.0 to 1.0.

When **Calculate Weights** is clicked, the weight map is calculated using the above settings. Weight maps for all bones are created.

Scaling Weight

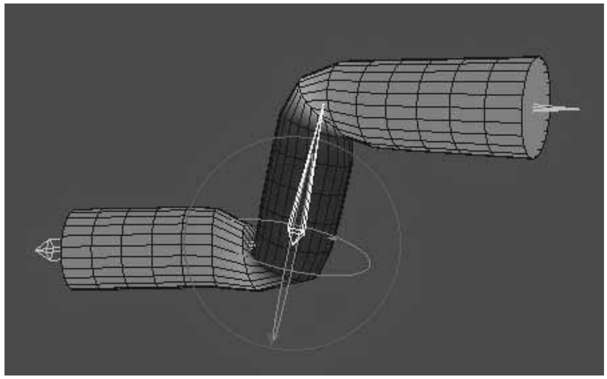
Select the Scale subtab in Weight mode to display the parameters for scaling weight map values.



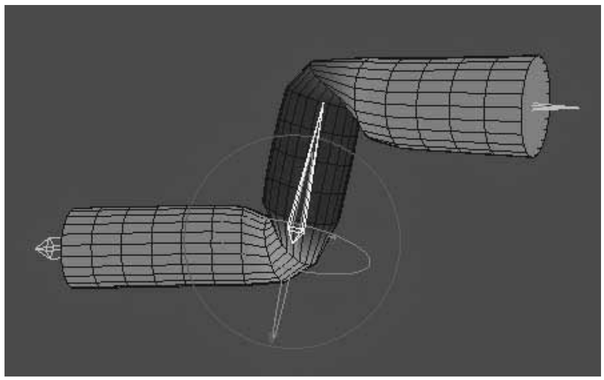
Scale sets the method for scaling weight values. With **Whole**, the scale is applied evenly along the weight map. With **At Start Joint**, the scale is applied from the base of the bone. With **At End Joint**, the scale is applied from the tip of the bone.

Clicking the **+** button will increase the weight value based on the value in the field to the right. Pressing the + key will also perform this function.

Clicking the **-** button will reduce the weight value based on the value in the field to the right. Pressing the - key will also perform this function.



Increasing the weight map value using the At Start Joint scale modifier.



Increasing the weight map value using the At End Joint scale modifier.

Clamping Weight

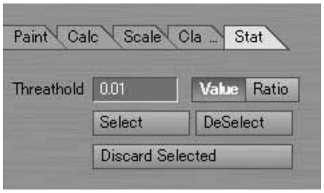
To activate the clamp weight, selected the Clamp subtab. When deformation for a single skin object is calculated, calculation speed is directly related to the number of matrixes that must be calculated for the bone. Calculation speed can be improved by truncating the weight value for points that receive little influence from the bone. When dealing with the deformation of single skin objects for real-time animation, this becomes an element of great importance for performance tuning.

N of Matrix sets the maximum number of matrixes, when calculating the deformation for each point. The weight value of each point influenced by the current bone is set to the number of **N of Matrix** in order of greatest influence. For example, in one mesh bone, 20 is clamped with the **N of Matrix** setting of 5. You keep the weight value of the points influenced by the bone that are greater than 5, and the value of points whose influence is below 5 are cut off and set to 0.0.

The **Clamp** button applies the clamp function based on settings.

Status of Weight

Select the Stat subtab to set values for displaying weight maps.



Threshold sets the value for the selection process. Any weight value less than the value set in **Threshold** will be selected. The entry field to the right allows you to set this value either by weight **Value** or as a **Ratio** (a percentage value based on the normalized weight value from 0.0 to 1.0).

When the **Select** button is clicked, all points whose weight value is less than the value set in **Threshold** are selected and displayed as yellow. To deselect the points, press the **Deselect** button.

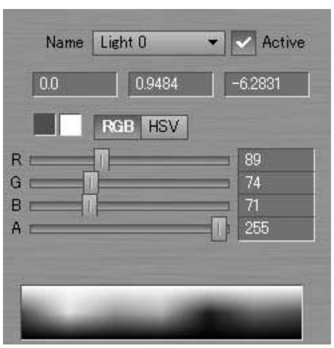
When **Discard Selected** is clicked, the selected points of the weight map, displayed in yellow, are set to a weight value of 0.0 and deselected.

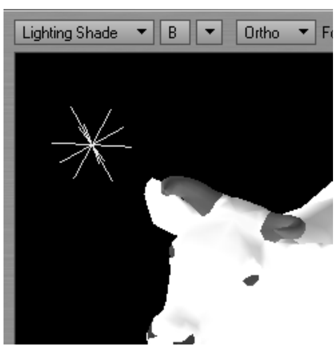
Lights Mode Tab

You can set the number of lights and their positions/values, which are used in the **Lighting Shade** rendering mode. In addition, the light settings here are used by the **Burn Lighting Colors** command (**Edit** pop-up in Color mode) for baking color information into the color map.

You can use eight lights with VertexPaint. This is the maximum number of active lights with OpenGl.

You can select lights from the **Name** pop-up menu or by clicking on them in the 3-D view. Lights look like yellow null objects when selected.





When you turn on the **Active** check box for the light selected on the **Name** pop-up menu, the light is set to its active state. When you uncheck this box, the light becomes inactive and does not cast light in the scene.

You can drag the selected light in the 3-D view to reposition or enter XYZ coordinates in the numeric entry fields under the **Name** pop-up menu.

The color of each light can be set independently. The color slider and palette are used the same way as in Color mode.

NOTE
VertexPaint lights operate independent of normal Layout lights.

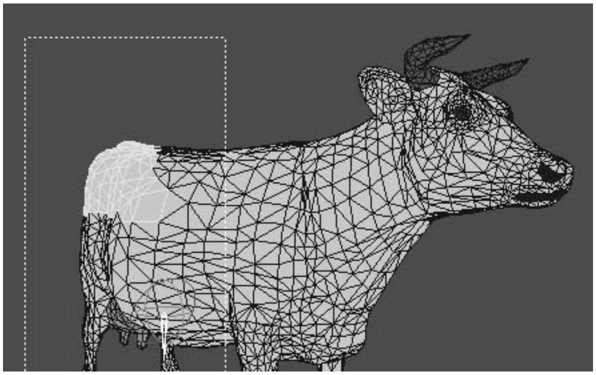
Show Mode Tab

The Show mode tab offers several ways to hide portions of the object while you work. The following operations are available for selecting polygons to hide.

Selecting Polygons

In the 3-D view, drag your LMB over the polygons you wish to select. If you use the RMB, the current polygons are deselected.

You can select polygons with a bounding box by holding the CTRL while you drag your LMB in the 3-D view. When you release the LMB, any polygons within the box are selected. If you drag with the RMB while holding CTRL, any polygons within the rectangular box are deselected.


Bounding box selection

When you choose **Select All** or **Hide All** from the **Edit** menu, the whole object is selected or deselected.

Show and Hide

There are two selection modes on the **Select** pop-up menu: **By Polygon** and **By Surface**. In **By Polygon** mode, the buttons work as follows:

Show	All polygons shown
Select	All polygons are selected
Hide	Selected polygons are hidden
Hide UnSel	Unselected polygons are hidden

If **By Surface** is used, select the surface you want to work on from the **Surface** pop-up menu. In this mode, the buttons work as follows:

Show	Polygons on the selected surface are displayed.
Select	Polygons on the selected surface are selected.
Hide	Polygons on the selected surface are hidden.
Hide UnSel	Unselected polygons and unselected surfaces are hidden.

Frequently Asked Questions

Q. Why is the mesh of the SubPatch not displayed?

VertexPaint does not support SubPatch objects. In Modeler, either press the TAB key to reset the SubPatch to polygon mode, or use the Freeze command to convert the SubPatch to a polygon mesh.

Q. The vertex color map was created, but the point color is not rendering in Modeler. Why?

Make sure you have the map selected on the Advanced tab of the Surface Editor under the **Vertex Color Map**.

chapter 29
Organic Modeling Tools

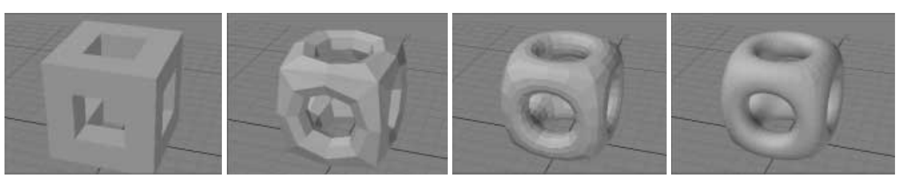
Chapter 29: Organic Modeling Tools

LightWave includes some powerful tools to help you create soft-edged organic objects. The *SubPatch* method is a parallel modeling mode that works with the same modeling tools and commands. *Metamesh* objects are sort of like interactive hypervoxels. *Spline patching* is a technique of creating a basic shape out of linked curves and then smoothly patching the areas between the curves with a mesh of polygons.

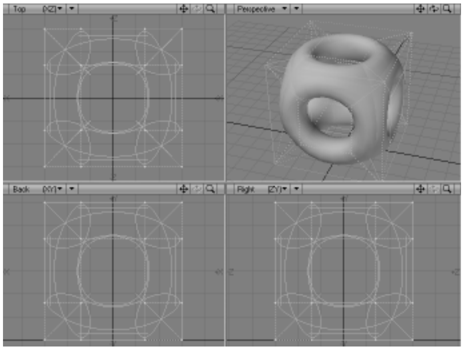
SUBPATCHES

Standard polygonal surfaces need many polygons to approximate a smooth surface. Even so, a smooth surface made up of polygons will eventually reveal its inherent sharp-edged nature, if it's examined closely enough. Although you can create such polygon-heavy objects, it is often difficult to manipulate and manage them from the perspective of memory-consumption and editing.

A "SubPatch," an abbreviation for subdivision patch, is a "bi-cubic patch." The idea behind SubPatching is to repeatedly refine the control mesh until you achieve a smooth surface, called the "limit surface." The SubPatch modeling mode is not unlike a real-time implementation of the Metaform (**Construct** > Subdivide: **Subdivide**) command.



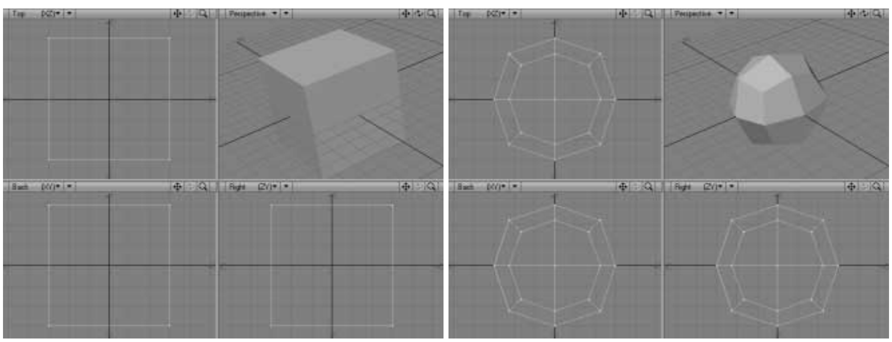
MetaForm performed three times in succession on the same object (no surface smoothing)



Same object with SubPatch activated.

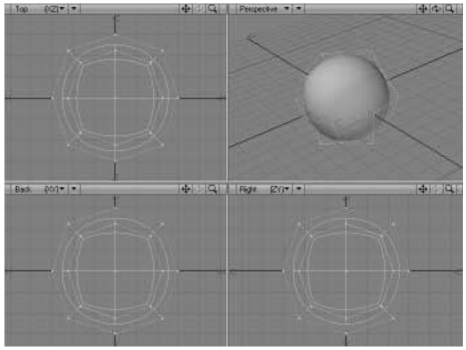
CREATING A SUBPATCH OBJECT

Your starting control mesh must begin with an object consisting of three- and/or four-point polygons. A good place to start is to make a simple box and then use Metaform (**Construct** > Subdivide: **Subdivide**) once (or maybe twice depending on your needs). This starts you off with a nice cage made only of quads (four-point polygons).



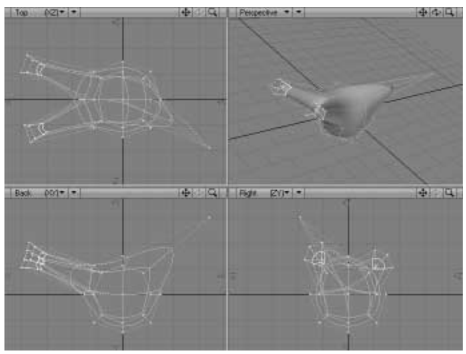
Left: Simple box. Right: Processed once with Metaform

You activate SubPatch mode by pressing the TAB key or choosing **Construct** > Convert: **SubPatch**. In most instances you want to affect the entire object, so no polygons are selected when you press the key. If polygons are selected, only those will have the SubPatch mode turned on. This can get confusing because the polygons become unselected. If you press the TAB key again, the SubPatch state of all polygons is reversed. If you do this by accident, use **Undo** and make sure all polygons are unselected before turning SubPatch on again. (You can also use the Polygon Statistics panel to select/re-select just the SubPatch surfaces.)



SubPatch mode enabled

To change the shape of the SubPatch object, you manipulate the shape of the polygonal cage (i.e., control mesh) using most of the standard modeling tools, including Bevel, Smooth Shift, Metaform, and so on, or by just dragging points and polygons around. Don't use tools that may create greater than four-point polygons, like Boolean.



Editing polygon cage affects SubPatch object

You can select the SubPatch vertices by clicking either on the true (polygon cage) vertex location or the location of the vertex when mapped on the SubPatch surface.

When you use SubPatch, you often manipulate the polygons into what would normally be non-planar, degenerate, or *cattiwompus* polygons. Since the polygonal cage is merely a reference for the SubPatch object, this is perfectly acceptable.

HINT

If you end up editing a very complex cage, the display can become confusing. In such a case, the cage is often close in shape to the SubPatch surface. Thus, you may be able to edit the normal polygons and just flick the SubPatch mode on to check the results.

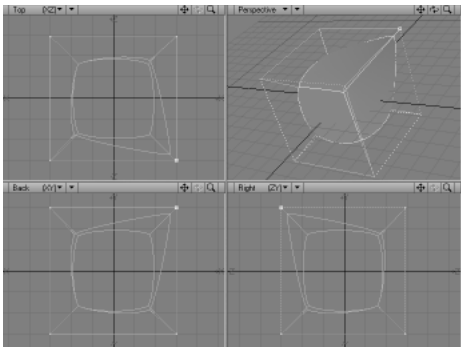
USING WEIGHT MAPS

SubPatch objects can utilize a special type of weight map that affects the bias of control points. You can thus create objects with much greater control and accuracy without adding complexity to the cage.

To adjust a SubPatch weight map:

1 Click the Weight map button in the lower-right portion of the interface, which is marked with the letter **W**, and choose **SubPatch Weight** in the pop-up menu to the right. (Weight maps are discussed in Chapter 28.)

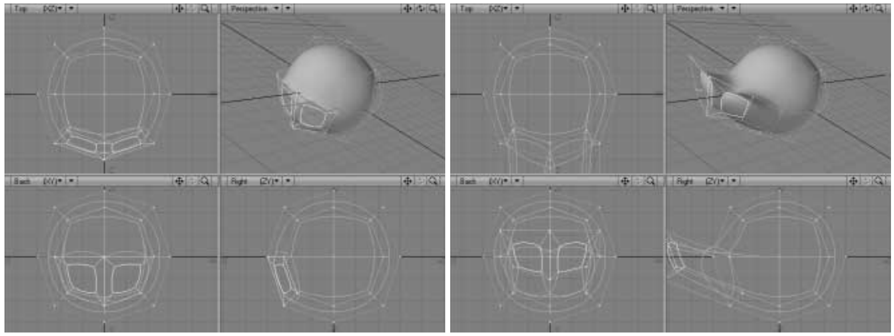
2 Use any of the weight-editing tools, like the Weights tool (**Map** > Weight & Color: **Weights**) to adjust point weights. You can increase or decrease the bias of individual control points on the SubPatch surface, which can sharpen (or unsharpen for negative values) the curvature of the object.



ADDING MORE DETAILS

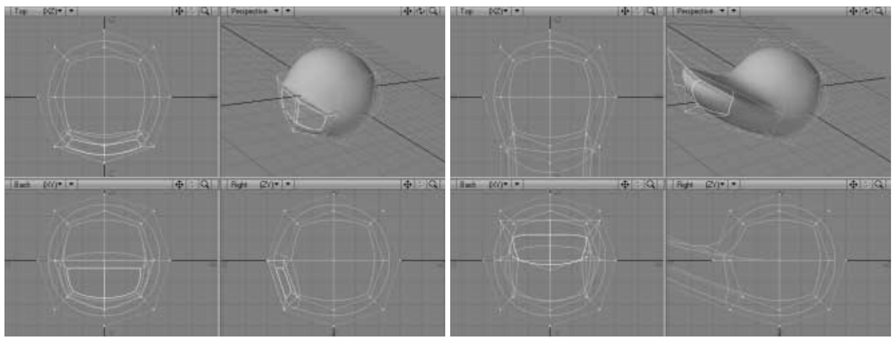
Adding more polygons to a specific area is somewhat problematic. Remember, your object must be composed only of quads (or triangles). Fortunately, there are a number of ways to easily add detail to a SubPatch cage.

If you are working with a single, selected polygon, the Bevel tool (**Multiply** > Extend: **Bevel**) is perfect because it will essentially add four new quads along the edges of the original polygon. Moreover, if you keep the **Inset** and **Shift** values at 0 (just click your RMB in a viewport), the bevel won't change the surface shape—although it will affect the underlying SubPatch form. (The beveled polygon sits in the same location so you can't see the new sides.) The polygon is still selected after the beveling operation, so you can then fine-tune the shape with the tools Stretch, Move, and so on.



Beveling out two polygons

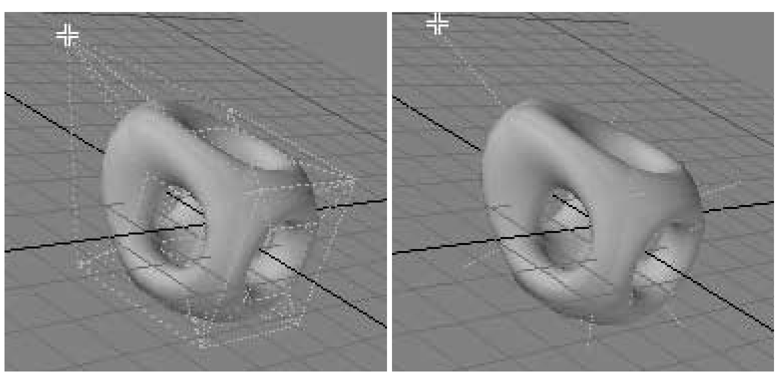
The Smooth Shift tool (**Multiply** > Extend: **Smooth Shift**) works similarly, but keeps a group of selected polygons together. Just click your RMB in a viewport to use an **Offset** of 0.



Smooth Shifting two polygons (*left*) and then moving them (*right*)

SPECIFIC DISPLAY OPTIONS

The display options for the Cage and Guides viewport specifically affect how SubPatch objects are displayed. Some folks find the cages clutter the screen too much and prefer to just grab the end of the guides to move points on the control mesh (See Chapter 27).



Left: Cage on. *Right*: Cage off

NOTE
See Chapter 9 for a discussion on using SubPatch object in Layout.

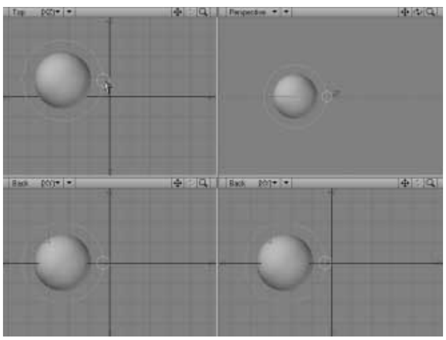
META-PRIMITIVE OBJECTS

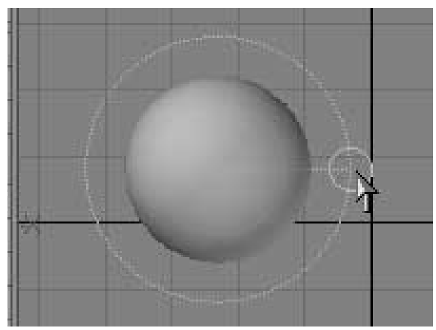
Meta-primitives are another cool modeling tool that comes in several flavors: plain Metaballs, Metaedges and Metafaces. Meta-primitives are similar to HyperVoxels except they are polygon-based, and their surface and interaction is more apparent and viewable in Modeler. In fact, Meta-primitives are a good place to start before you set up with HyperVoxels, because of their real-time feedback. All Meta-primitives will interact with each other, even though they are different types, but only if they are in the same layer.

To draw a Metaball object:

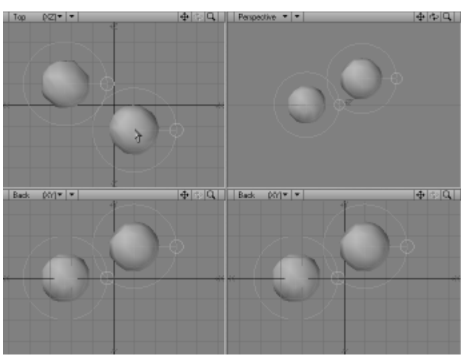
1 Activate the Draw Metaballs tool. (**Create** > Elements: **Metaballs**)

2 Click in a Modeler viewport. Use any Rendering Style other than Wireframe and you will see the metaball.

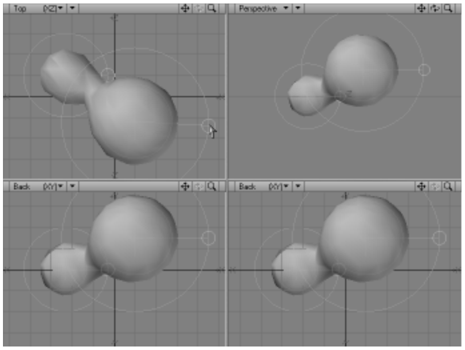




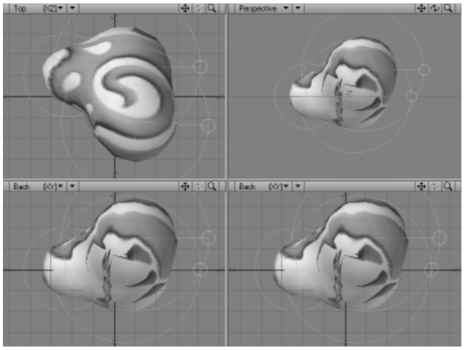
3 You can move the metaball by dragging its center handle. You can change its size by dragging the sizing handle that looks like an orbiting circle. If you click anywhere else, another metaball will be created.



4 If you move or increase a metaball so that its surface approaches the surface of another, they will adhere to each other, like the stuff in a lava lamp.



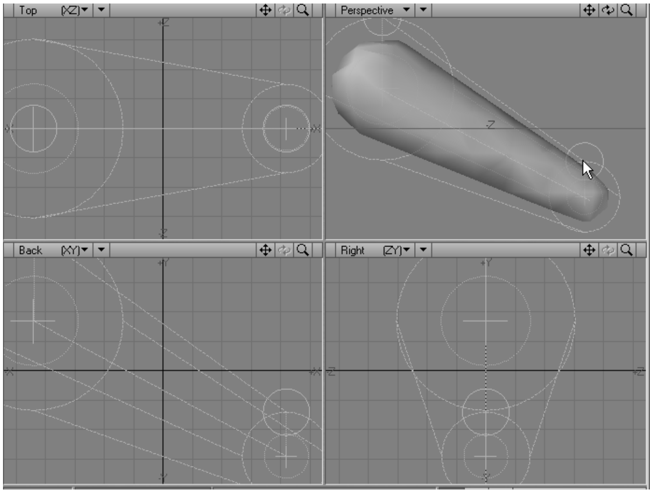
5 Drop the Draw Metaballs tool (i.e., reselect it or press the SPACEBAR).



Surface texture added

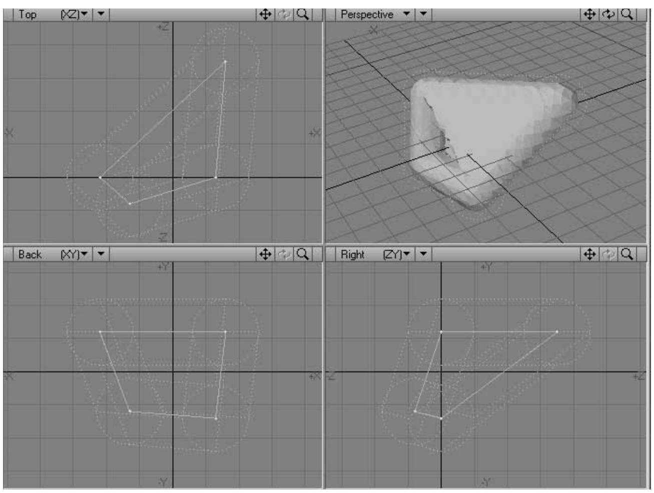
Metaedges and Metafaces

Metaedges and Metafaces are essentially multi-point versions of plain Metaballs. To draw Metaedges, you can choose **Create** > Elements: **Metaedges**. Your initial clicking point establishes the first point. Hold you LMB down and drag out the second point. Other than there being two points, this tool works just like the Draw Metaballs tool.

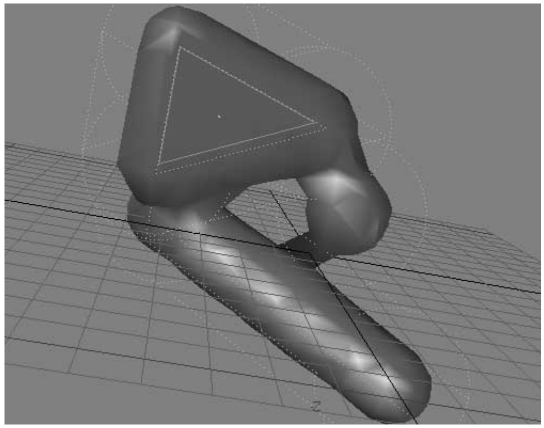


Metaedges

There is no draw Metafaces tool. They must be created from three- or four-point polygons or curves. Just select the polygon/curve and choose **Construct** > Convert: **Make Metafaces**.



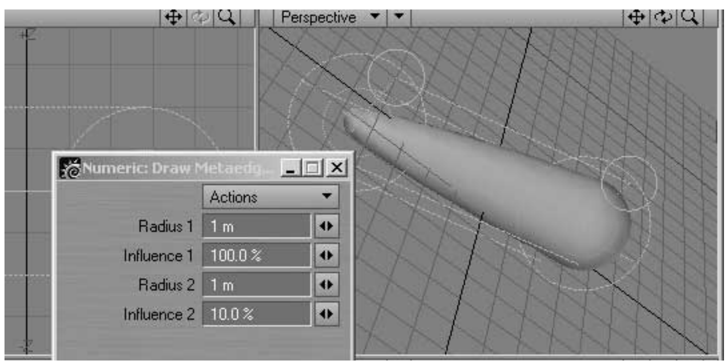
Metafaces



Metaedge, Metaface, and Metaball interacting together

Radius and Influence Settings

The numeric panel has **Radius** and **Influence** settings. **Radius** is simply the size of the Meta-primitive ball. Adjusting the **Influence** changes the interaction between balls. Setting a negative **Influence** will *invert* the primitive causing it to *eat away* neighboring Meta-primitives instead of blending. (See “Editing Meta-primitives” for example.)



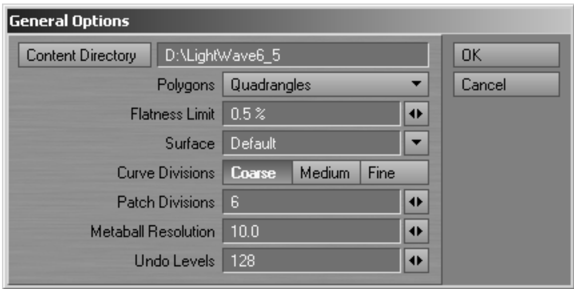
Metaedge with different Influence settings

Converting to Metaballs and Metaedges

You can also convert *points* into plain Metaballs using the Make Metaballs command (**Construct** > Convert: **Make Metaballs**). Similarly, Metaedges can be created from two-point polygons or curves. Simply select the polygon/curve and choose **Construct** > Convert: **Make Metaedges**.

Metaball Appearance

The amount of detail for *displaying* Meta-primitives is handled by the **Metaball Resolution** setting on Modeler's General Options panel (**Modeler** > **Options** > **General Options**). The value represents the number of subdivisions (pixels per metaball on the screen). To get a smoother surface, increase the value. There is no limit.



For a discussion on rendering and Layout display, see Chapter 9, "Metaball Display and Render Levels."

You can also toggle the Meta-primitive mesh on/off by choosing **Construct** > Convert: **Toggle Metamesh**. You may want to do this if you have a lot of Meta-primitives and your display refresh is too slow. This is not just a display change and is similar to toggling off SubPatches. If you save a Meta-primitive with the mesh off, you will not be able to see the surface in Layout.

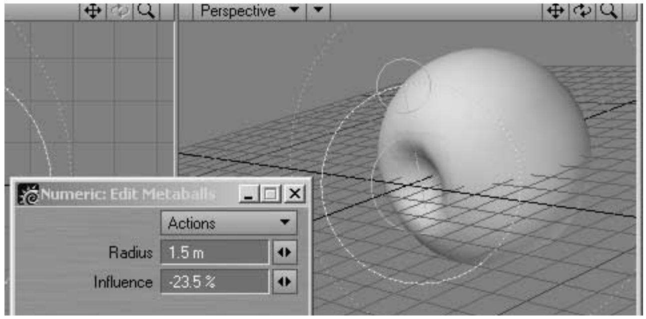
You can surface Meta-primitives as you would any object; however, you are allowed only one surface per layer.

Animating Meta-primitives

You can animate Meta-primitives the same way you would animate any object. Of course, you won't take advantage of their coolness unless you animate point positions. For this, you can use features like morph targets, bones, displacement maps, and so on. Note that only Meta-primitives in the same layer will interact with each other.

Editing Meta-primitives

Activate the Edit Metaballs tool (**Detail** > Other: **Edit Metaballs**) to edit existing Meta-primitives. On the numeric panel, you can adjust the **Radius** and **Influence**.



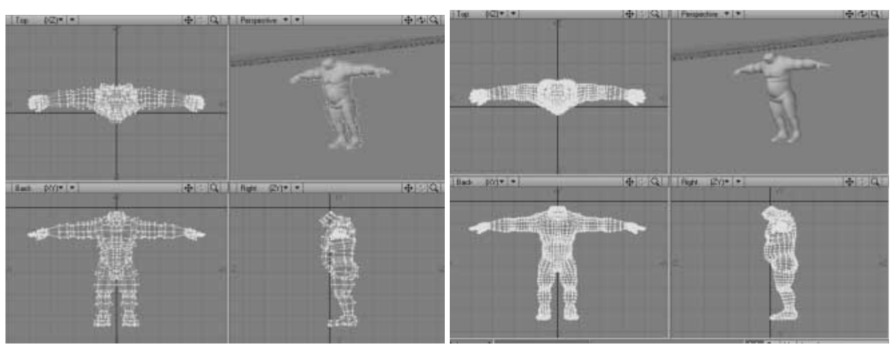
Negative Influence eating-away neighboring Meta-primitive

Deactivate the tool by dropping it, reselecting it, or hitting the SPACEBAR when you are done.

PATCH LEVEL AND CONVERSION TO POLYGONS

Although you can use a SubPatch model or Meta-primitives as is, occasionally you may want to convert it into a normal polygon mesh.

To convert to a polygons mesh, use the Freeze command (**Construct** > Convert: **Freeze**). The amount of polygonal detail in the object is also determined by the **Patch Divisions** setting on the General Options panel (**Modeler** > **Options** > **General Options**). (See Chapter 20 for more information.)



Once frozen, the control mesh no longer exists. As such, you should save the object before you perform a freeze just in case you later decide you want to modify its shape.

NOTE
The **Patch Divisions** setting on the General Options panel also affects the level of smoothing in the viewport displays.

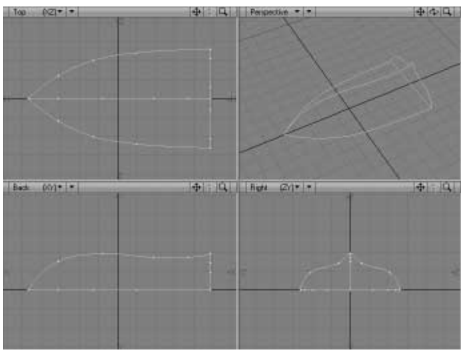
NOTE
The Freeze command supports discontinuous UVs. It will also generate all the morph target objects, as if they had been *frozen* individually. For geometry that changes topology when morphed—like metaballs—the currently selected morph generates the frozen mesh and all other morphs are discarded.

SPLINE PATCHING

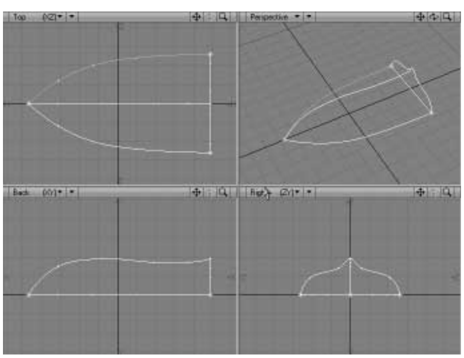
Modeler's feature for spline patching lets you model with curves and then smoothly spread a mesh of polygons over them.

To use spline patching:

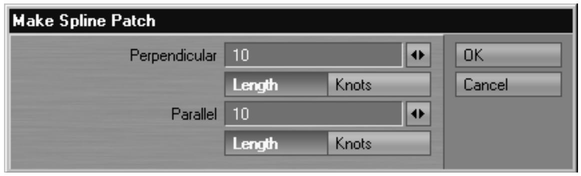
1 Make three or four curves that create one or more fully enclosed areas. Note that the curves share end points.



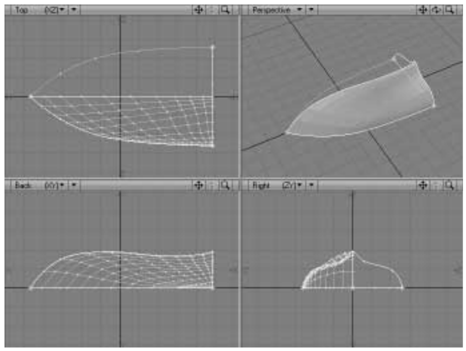
2 In the Polygon Selection mode, select the curves that surround the enclosed area. Do this in a clockwise or counterclockwise fashion. Remember which curve you selected first! It's okay if you also select some polygons in addition to the curves, they will be ignored in the patching operation.



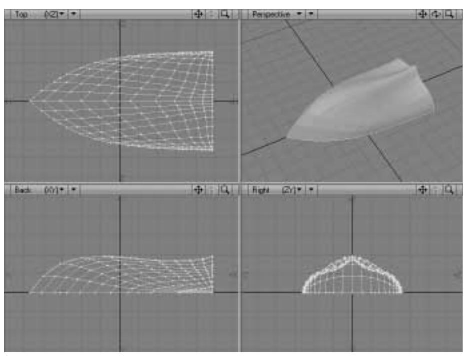
3 Choose **Multiply** > Combine: **Patches** > **Make Spline Patch**. The fields for **Perpendicular** and **Parallel** refer to the number of polygons to put along the side that is perpendicular or parallel to the first curve you selected. **Knots** and **Length** let you choose to space your polygons equally according to the overall length or relative to the knots or points on the splines.



4 Click **OK**. You should see a polygon mesh added to the area within the three curves you selected.



5 Repeat this for the other areas. If you patch an opposing side, select the curves in the same order and direction. (For this boat hull, the outer curve was selected first, then the other curves were selected in clockwise order.)



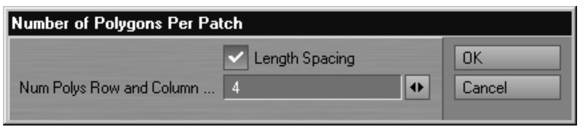
Sometimes a spline cage will be patched unexpectedly. There are some fixes you can try. All of the selected curves should flow in the same direction, based on their heads and tails. If they don't, you can flip them and try the patch again. (See Chapter 22 for more information on curves.) You can also try selecting the curves in the opposite direction.

Sometimes the resulting surface normals face the wrong way. Just select at least one polygon on that side. Choose **Display** > Selection: **Sel Connect** to select all of the connected polygons in the patch. Then, press the F key to flip the polygons.

If there are not enough curves or they do not share points properly, an error message appears. If this happens, check the joining points carefully to make certain they are not merely overlapping. Choose either **Construct** > Reduce: **Merge Points** or **Detail** > Points: **Weld** to correct.

AUTOPATCHERMK

The AutoPatcherMK command (**Multiply** > Combine: **Patches** > **AutoPatcherMK**) automatically patches a spline cage. You can set the number of rows and columns of the patch between 0 and 20. To space the polygons according to the length of the curve, select the **Length Spacing** checkbox. To insure success, never have more than four knots in a curve.



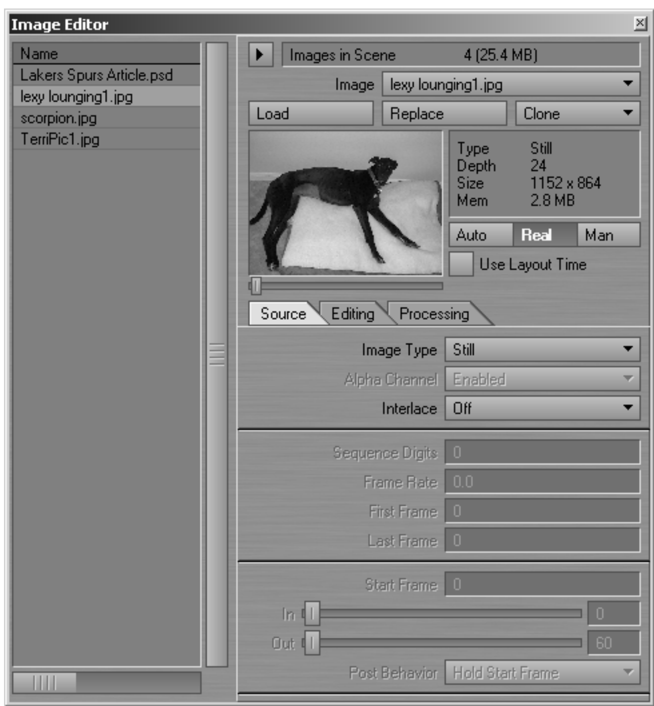
WHY SPLINE PATCH?

Often it is much faster and easier to create *patch-type* objects with SubPatch, but there may be times that only spline patching will get the desired results. For example, if you need to create a smooth area that will adhere to existing polygons, spline patches work better because you can align the edge using exact points. Since SubPatch uses the points as a reference only, it is difficult—if not impossible—to align the resulting SubPatch object with existing geometry.

chapter 30
Image Editor

Chapter 30: Image Editor

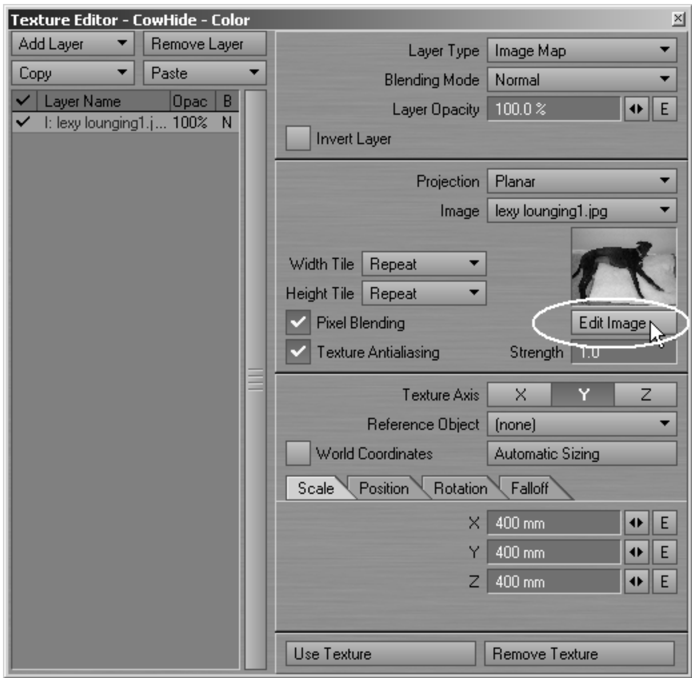
The Image Editor handles functions related to images and video clips. You can load images and video clips from this panel to make them available for various functions where images are needed, like for surface textures. You can also adjust certain image characteristics, like brightness and contrast, without actually affecting the original file. If you load an animation file (e.g., AVI) or image sequence, you can specify the beginning and ending frames to use and set what happens before and after the sequence begins and starts.



The Image Editor

QUICK ACCESS TO IMAGE EDITOR

Pop-up menus for image-loading throughout LightWave include time-saving functions that bypass the Image Editor. You can always edit loaded images by clicking the related **Edit Image** button that appears near image-loading pop-up menus.

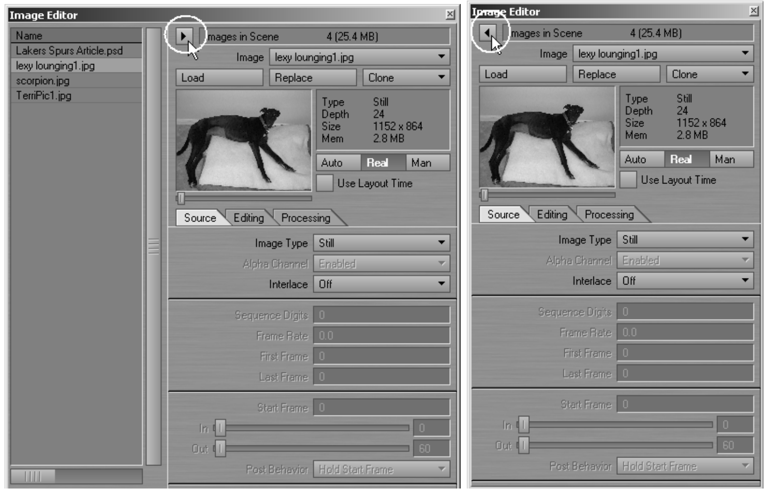


The Edit Image button on the Texture Editor

THE IMAGE EDITOR FILE LIST WINDOW

The controls on the various tabs to the right will affect the selected image file. The image will appear in the preview window below. If the item is a sequence/animation, you will display the first image.

You can collapse/uncollpase the list window by clicking the left-facing arrow button.



Left: List window open. *Right*: List window closed

FILE-HANDLING OPTIONS

You can select the current image by either clicking on it in the list window or by selecting it from the **Image** pop-up menu.

Click the **Load** button to load a new image file to add to the list. Use the **Replace** button to replace the selected image in the list window. To delete an image from the list, simply select it and press the DELETE key.

File-handling options

The **Clone** pop-up menu has two items. **Instance** creates a new entry for the *cloned* clip and lets you apply independent settings on the Editing and Processing tabs. The duration (e.g., In/Out) is linked to the original clip. Changes to the original image will affect any *instances* (made from that original). However, you can always make multiple instances and apply changes to only the instances, leaving the original untouched.

Duplicate creates a completely independent copy of the original that you can adjust in any way you want; however, it is available only for image sequences.

THE PREVIEW WINDOW

The preview window normally displays the selected image. If you select a sequence/animation, you can shuttle through it by dragging the slider.

The options to the right of the preview window affect how the window is updated. The **Auto**matic setting updates only when you have finished a change and released the mouse button. Changes include dragging the slider, making adjustments on the Edit tab, selecting images, and so on. **Real**time, on the other hand, updates the window interactively. **Man**ual requires you to click on the window to update. **Use Layout Time** uses the Layout time slider instead of the slider beneath the preview window (if you opened the Image Editor from Layout).

You may also change the above settings by right-clicking on the preview window. Clicking **Open with IView** or double-clicking on the preview window brings up the image in the Image Viewer.

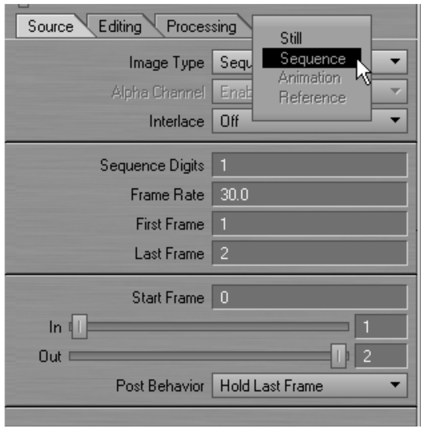
For speed and memory reasons, preview images use 24-bit data and filters are applied to the preview image, rather than being applied to the whole image and shrinking it down. In most cases, this is sufficient for previewing and much faster.

SOURCE TAB

The **Image Type** pop-up menu displays the type for the currently selected image. Generally, this setting is auto-detected when you load the file. However, you can specify a single image in a sequence of images by leaving its **Image Type** to **Still**. **Reference** is a clone of an image, as discussed above.

Loading a Sequence

If you wish to load a sequence of files, you can select any image in the sequence—it doesn't need to be the first. Initially, the sequence will be treated as a **Still**. However, you can just change the **Image Type** to **Sequence** and LightWave automatically analyzes the overall sequence of files and completes the related fields on the panel.



The Sequence tab

Each separate file must have the same name, followed by a number. The numeric sequence may be staggered, but the name must be consistent. For example, this is a valid image sequence: FLY001.TGA, FLY002.TGA, FLY005.TGA, FLY007.TGA, FLY010.TGA.

By default, each number corresponds to a frame. If images are missing in a sequence, then the previous image is used until the next image in the sequence comes up. Thus, with the sequence SIENNA001, SIENNA002, SIENNA005, SIENNA006, ... SIENNA002 is used at frames 2 through 4.

Alpha Channel

Be careful using files with an embedded alpha channel. This can cause unexpected results, like some or all of the image not appearing on surfaces when rendered, due to the alpha channel masking—the alpha channel is not taken into account in OpenGL. If appropriate, set **Alpha Channel** to **Disable** to ensure any alpha channel information is ignored.

Interlace

If the image is interlaced (e.g., digitized video), indicate whether the fields are interlaced with even or odd lines first using the **Interlace** pop-up menu. Non-interlaced images should set this option to **Off**.

Sequence Digits

For sequenced images, **Sequence Digits** sets the number of digits in the sequence filenames. When you load an image sequence, LightWave examines the way that the files have been named in an attempt to automatically detect the base filename and the numeric extension. You can change **Sequence Digits**, if LightWave has difficulty discerning the numbering scheme used.

For example, assume you have a sequence of 60 images named PARK001.TGA through PARK060.TGA. You select **Load**, and click on PARK045.TGA (since selecting any image from the sequence will work). When the **Clip Type** is changed to **Sequence**, LightWave determines that the base filename is PARK (its name in the list window will show PARK (SEQUENCE) and that there are three **Sequence Digits**.

Frame Rate

Frame Rate controls the incrementing of an image sequence with respect to the scene's **Frame Per Second** setting on Layout's General Options tab of the Preferences panel. So, if you set this to, say, 15 and your scene Frame Per Second is 30, the image sequence will increment every two frames.

Image Sequence Settings

The **First Frame** and **Last Frame** settings display which files (or frames for an animation file) are treated as an image sequence in LightWave. This is computed when you load an animation file or switch the Image Type pop-up menu to Sequence (after loading a single image from the sequence).

First Frame defines which image file in your sequence is the beginning. So, if your sequence is named IMGSEQ003...IMGSEQ100, your first frame is 3. For animation files, your first frame is set to 0.

NOTE
If an image is missing in the sequence, the first preceding image that exists is used.

Last Frame defines which image file ends your sequence. So, if your sequence is named IMGSEQ003...IMGSEQ100, your last frame is 100.

Because of the infinite possibilities for sequence naming, LightWave may not be able to correctly set the **First Frame** and **Last Frame** fields. If so, input the correct values.

NOTE
You should not use the First Frame and Last Frame settings to trim a sequence. It may lead to unpredictable results. Use the In and Out sliders for trimming.

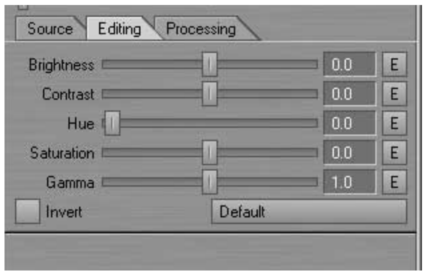
Use the **In** and **Out** sliders to trim the ends of the sequence or animation.

Start Frame specifies the Layout frame when the sequence will begin.

The **Post Behavior** settings determine what happens to frames after the defined sequence. You can hold the start or last frame, or loop the sequence.

EDIT TAB

You can independently adjust (and envelope) various image parameters for the selected image on the Editing tab, if you want the images altered before LightWave uses them.

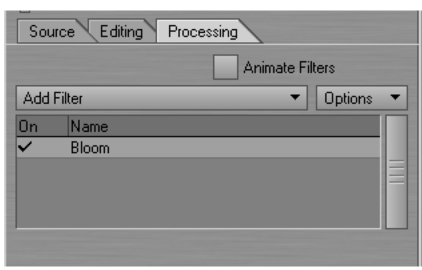


The Edit tab

NOTE
These image operations do not affect the actual file on your hard drive.

PROCESSING TAB

Access this tab to add image-processing filters. You may use any of the non-post-processing filters described in Chapter 14. Image filters added here will not update the Preview window interactively unless the **Animate Filters** option is activated.



The Processing tab

chapter 31
Surface Editor

Chapter 31: Surface Editor

With only a few exceptions, LightWave objects are composed of one or more polygons. What each polygon looks like is defined by its *surface attributes* or usually just *surface* for short. Groups of polygons can share the same surface and the polygons need not be contiguous. The polygons can even be in different objects.

SURFACING OBJECTS

Let's take a simple cube. It has six sides and, therefore, at least six polygons. Each side could have its own individual surface, all six sides could use the same surface, or any combination in between.

The polygons that make up a surface are given a name, usually in Modeler. The name can be changed, however, in the Surface Editor.

The surfacing information associated with the surface name is stored in the object file. When an object is later loaded, the surfaces used in it are also loaded, including any image files referenced in the surface attributes.

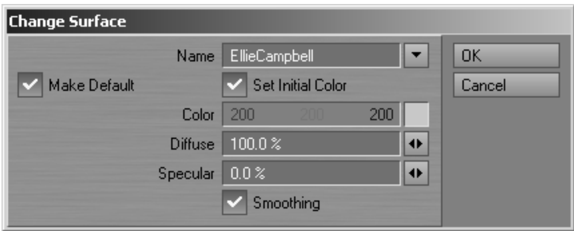
WARNING
Surface settings are saved as part of the associated object file(s), not in the Scene file. To save surface settings, save the object.

To assign surface names to polygons:

1 In Modeler, select the polygons.

HINT
When you rename all of the polygons that already have a surface name, use the Statistics panel (**Modeler** > **Windows** > **Statistics Open/Close**) to select them. Also, if you are naming all polygons, you don't need to select any.

2 Choose **Detail** > Polygons: **Surface** to open the Change Surface dialog.



Change Surface dialog

3 Enter the surface name for the selected polygons and click **OK**. You can also select an existing surface name from the pop-up menu.

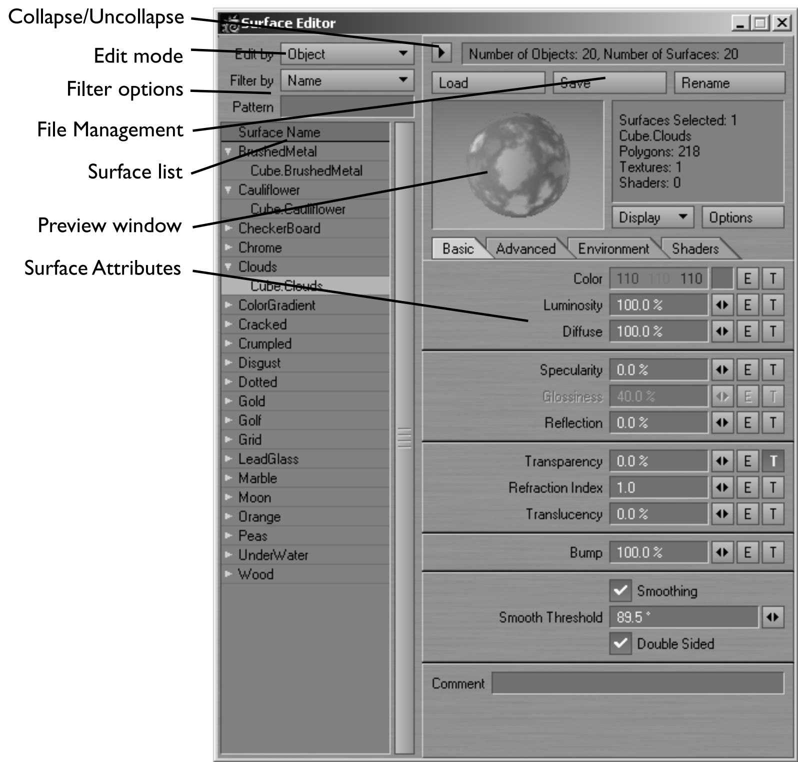
You should strive to give your surfaces meaningful names. For example, say you had a car object. One surface might be called "chrome," which you apply to the bumper, door handles, antenna, etc. "BlackRubber" might be another surface, which you apply to the tires, rubber gaskets, etc. You might use "RedMetal" for all of the exterior painted metal parts of the car.

4 To set some basic surface settings, activate **Set Initial Color** and adjust the available settings as desired. You can perform more advanced surface editing using the Surface Editor.

5 The **Make Default** option automatically updates the Surface option on Modeler's General Options panel (**Modeler** > **Options** > **General Options**). This controls the default surface name used when you create new geometry. (See Chapter 20 for more information.)

THE SURFACE EDITOR PANEL

You give your named surfaces their characteristics on the Surface Editor panel; you can access this panel from both the Layout and Modeler modules.



You can collapse/uncollapse the surface list by clicking the arrow button. When collapsed, a pop-up menu is added near the top for displaying and changing the current surface.

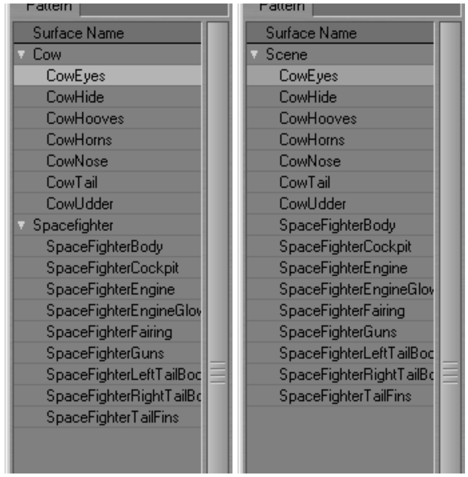
Surface list collapsed

Surface Edit Modes

LightWave offers two edit modes when you work in the Surface Editor: **Object** and **Scene**, which are *discrete* and *global* edit modes, respectively. These are selected using the **Edit by** pop-up menu. While surfaces are always saved within the object file itself, there are times when many objects share the same surface name. In Object mode, each object's surfaces are protected from changes made on other objects with the same name. When in Scene mode, all surfaces are *flattened*, so that any item in the scene that uses the current surface name receives the same changes to the surface attributes.

The Edit mode is saved from session to session and affects how LightWave handles objects loaded with identical surface names. If the Scene edit mode is active, the last loaded object's surfaces will control.

When working in Scene mode you will notice that the Surface List shows only a *raw* list of surface names. While working in Object edit mode, you will see a complete list of the loaded objects with their surfaces listed beneath the object names.



Left: Object mode. *Right*: Scene mode

Object Edit Mode

The default mode, Object, gives you many items in a scene with discrete surface settings. For example, you may have two goblins, HobGoblin and BobGoblin loaded into a scene and each may have a surface called Skin. In Object mode, LightWave internally holds an object composite library for each item. So, while they both have a surface called Skin, LightWave sees them as HobGoblin/Skin and BobGoblin/Skin so that you can make changes to either one without worry of interfering with the other.

When you change from Object to Scene mode, the last-loaded item determines the attributes for shared surface names.

Scene Edit Mode

The Scene mode is very handy if you want many objects to share a surface. With this mode enabled you can quickly make changes to a group of objects by changing their common surfaces. Internally, LightWave just drops the object composite library mentioned above, so that surfaces are referenced by their raw surface name, without object listing.

For example, you may have a troop of soldiers that all share a common uniform surface Jungle_Camo_Jacket. Internally, LightWave simply manages the surface name Jungle_Camo_Jacket rather than Soldier.01/Jungle_Camo_Jacket, Soldier.02/Jungle_Camo_Jacket, etc. If you want to change the base color of all soldier jackets, you can work in **Scene** mode and make a single change that propagates throughout all items with a surface named Jungle_Camo_Jacket.

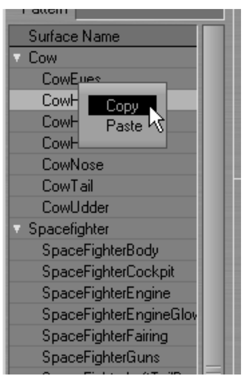
Surface List

The large text window on the left lists the surfaces available for edit. The **Filter by** pop-up menu lets you filter elements shown in the surface list window. **Objects** shows all objects and their related surfaces. **Name** shows all surface names. **Texture** shows all surfaces that use procedural textures. **Shader** shows all surfaces that use surface shaders. **Preview** lists all surfaces that are visible in the image currently in the render buffer (you need to render a frame first). This is determined on a pixel-by-pixel basis. (Note that the Preview setting is only available if the Surface Editor is accessed from Layout.)

Filter by works in conjunction with the **Pattern** input field, which is a simple *include* filter. Any item that includes any of the text you enter will appear in the list window. You don't need to specify wildcards, like asterisks; when you leave the input field blank, you include all surfaces in the **Filter by** category.

Select a surface to edit by clicking on its name in the list. If the object name is listed, you can expand or collapse the displayed surfaces by clicking on the arrow to the left of the object name.

When you right-click over the surface list window, a pop-up menu appears. You can **Copy** the selected surface to a memory buffer. Then, select a different surface and **Paste** the contents of the buffer over the settings of the selected surface.

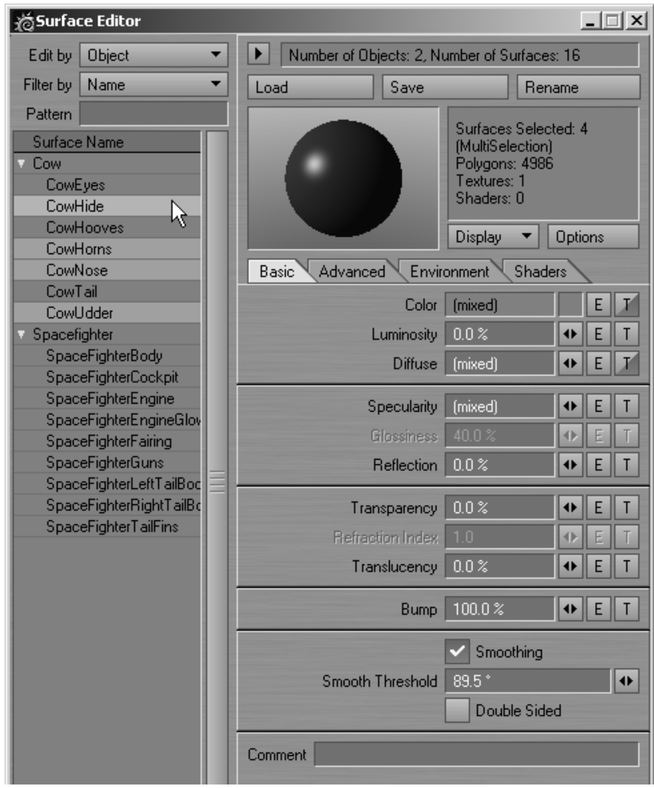


Surface Name list pop-up menu

Mass Surface Changes

You can select multiple surfaces in the Surface Name list and make mass surface changes. Hold the SHIFT key to select a range of surfaces or hold the CTRL key to select/unselect surfaces independently.

Parameters with input fields that have different settings for the selected surfaces will show (mixed) in the field. If textures are different, the **T** button will appear in an intermediate state. Changing a surface attribute changes it for all selected surfaces, including *mixed* states. *Shift-clicking* on the **T** or **E** button removes the texture or envelope for all selected surfaces.



Multiple surfaces selected

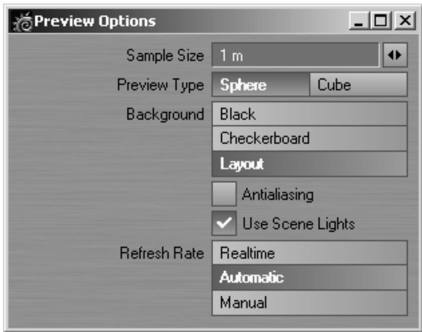
NOTE
Currently, you cannot make mass changes to surface shaders.

Preset Shelf and VIPER

If you would like to use VIPER (Layout only) or the Preset Shelf, click their respective buttons on the main toolbar. See Chapter 3 for more information.

Preview Window

The Preview window shows you surface samples for the active surface. You can access several preview options by clicking the **Options** button.



Preview window options dialog

Sample Size is the diameter of the sampled area on the surface. You get the most accurate preview if you set this option to the approximate surface size to which you are applying the current attributes. **Preview Type** sets the shape of your sample surface.

You have several options for the preview **Background**. **Checkerboard** is nice for surfaces with some level of transparency. If you select **Layout**, it uses Layout's backdrop settings. You can reduce *jaggies* in the preview by using the **Antialiasing** option.

If **Use Scene Lights** is active, the lights from your scene will be used instead of the default preview light. This option is applied as if the preview object was at the Origin and also affects VIPER. Obviously, this is only available if the Scene Editor is open from Layout.

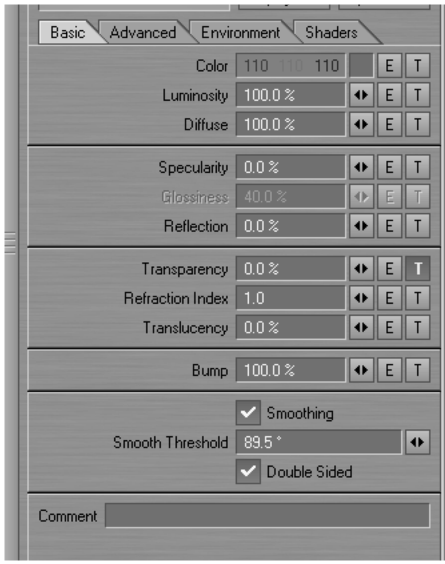
The **Refresh Rate** setting determines how the preview is refreshed when you make changes to surface settings. If you set the rate to **Realtime**, the preview updates as you change interactive controls; for example, if you adjusted a mini-slider. When it is set to **Automatic**, the preview updates when you release the control. **Manual** requires you to click on the preview window to update.

You can further tailor the preview to reflect only certain internal channels (buffers) instead of the normal **Rendered Output** with the **Display** mode pop-up menu. This is very useful when you want to determine the effects you get on a specific channel. For example, if you apply a diffuse texture, you may want to see the effects of your settings on the actual diffuse channel.

NOTE
You can quickly set options by using the pop-up menu that appears when you hold the RMB over the Preview window.

BASIC SURFACE PARAMETERS

The surface attributes on the Basic tab are fundamental for virtually all objects. Each represents a certain characteristic of a surface's appearance. You may never have thought about it before, but if you look at the surface of objects around you, they differ by much more than their color. Are they shiny? Dull? Do they reflect other items in the room? Can you see through the material? These characteristics help you determine what materials objects are made of.



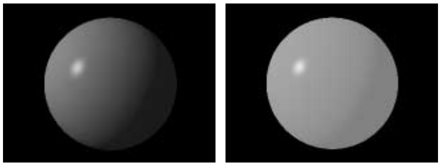
LightWave attempts to divide the different surface characteristics into controllable parameters.

Color

Color is probably the most obvious surface parameter. It doesn't take much experience to know that if we want something to look like a banana, we need to make it yellow, right? However, since you are dealing with a 24-bit color palette and, thus, over 16 million colors, there are probably thousands of shades of yellow. Moreover, other settings, such as Diffuse, can have a dramatic effect on the final rendered color.

Luminosity

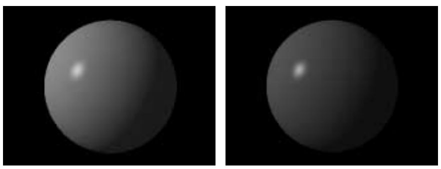
Luminosity refers to how much a surface appears to glow of its own light. However, unless you use Radiosity (**Lights** > Global: **Global Illum**), luminosity does not have any actual light emitting properties—you need to add an actual light for that. A value of 0% is most common for this setting, unless a special need arises such as the surface of a modeled light bulb. This surface property is not the same as Glow on the Advanced tab.



Left: Luminosity = 0%. Right: Luminosity = 100%

Diffuse

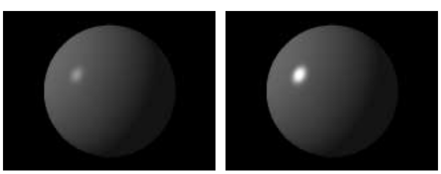
Diffuse (sometimes called *diffusion*) is the amount of light scattered by a surface. A high level scatters a lot of light, and therefore, the surface appears bright. A low level absorbs most of the light, and therefore, the surface appears dark and dull. Metal and dirt surfaces are good candidates for a low **Diffuse** level. Common values are 40% to 80%. Surfaces must have some diffusion for shadows that cast on them to be visible.



Left: Diffuse = 100%. Right: Diffuse = 60%

Specularity

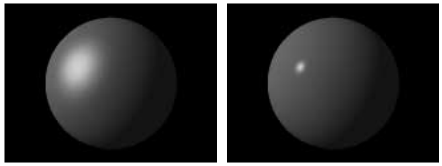
Specularity is a kind of reflection or highlight that occurs on the surface of smooth or shiny objects. This highlight is really the reflection of the light source. High Specular levels are commonly used on glass spheres, chrome bumpers, and so on. How the surface reflects this highlight tells the observer if the surface is dull, smooth, shiny, hard, or even metallic. Generally, the highlight assumes the color of the light source that causes it, but you may change this with the **Color Highlights** settings (Advanced tab).



Left: Specularity = 25%. Right: Specularity = 100%

Glossiness

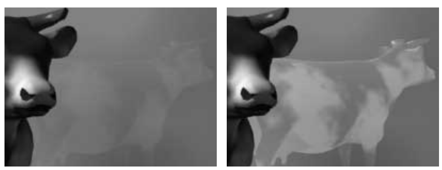
When some level of **Specularity** exists, **Glossiness** determines how a highlight spreads out. A low setting creates a large highlight, whereas a higher setting creates a smaller highlight.



Left: Glossiness = 5%. Right: Glossiness = 50%

Reflection

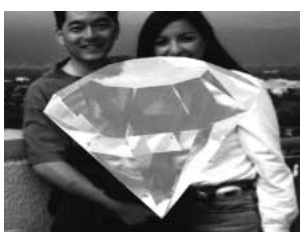
Reflection determines how much a surface shows a reflection of its surroundings. A mirror is an example of a highly reflective surface. LightWave will ray trace reflections or you may use options that fake them, which can substantially reduce rendering time. (See the discussion on the reflection options on the Environment tab, later in the chapter.)



Left: (Ray-traced) Reflection = 25%. Right: (Ray-traced) Reflection = 100%

Transparency

Transparency lets you make a surface *see-through*; it is the opposite of opacity. Now, whenever light passes through a transparent object, the light will bend. An example of this can be seen when you look through a glass of water or down through a clear pool of water. The amount of this light bending is controlled by the **Refraction Index**.



Transparent diamond

Refraction Chart Index

Material	Index
Vacuum	1.00000
Air	1.0003
Cinnamon Cat	1.136
Carbon Dioxide, Liquid	1.200
Ice	1.309
Water	1.333
Acetone	1.360
Ethyl Alcohol	1.360
Sugar Solution (30%)	1.380
Alcohol	1.329
Flourite	1.434
Quartz, Fused	1.460
Calspar2	1.486
Sugar Solution (80%)	1.490
Glass	1.500
Glass, Zinc Crown	1.517
Glass, Crown	1.520
Sodium Chloride	1.530
Sodium Chloride (Salt) 1	1.544
Polystyrene	1.550
Quartz 2	1.553
Emerald	1.570
Glass, Light Flint	1.575
Lapis Lazuli	1.610
Topaz	1.610
Carbon Bisulfide	1.630
Quartz 1	1.644
Sodium Chloride (Salt) 2	1.644
Glass, Heavy Flint	1.650
Calspar1	1.660
Glass, Dense Flint	1.660
Methylene Iodide	1.740
Ruby	1.770
Sapphire	1.770
Glass, Heaviest Flint	1.890
Crystal	2.000
Diamond	2.417
Chromium Oxide	2.705
Copper Oxide	2.705
Amorphous Selenium	2.920
Dougbrainium	3.039
Iodine Crystal	3.340

NOTE
Also see the discussion on the refraction options on the Advanced tab, later in the chapter.

NOTE
To control the amount of edge transparency, you must add a shader, discussed later.

Translucency

Translucency allows back lighting. The effect can be seen when someone (or something) stands behind a curtain and you see a silhouette through the material. Other examples include seeing bones in your hand with a flashlight in your palm, or bugs crawling on the underside of leaves.

Translucent curtain

Translucency is similar to **Transparency** in that all lighting properties, like color and luminosity, will show through. The obvious difference is that translucency doesn't add a *see-through* effect.

If you want to make a silhouette visible, something must cast a shadow on the back side. Note that you do not need rear-facing polygons for the back side, nor must you use a double-sided surface to *catch* the shadow.

Bump Map

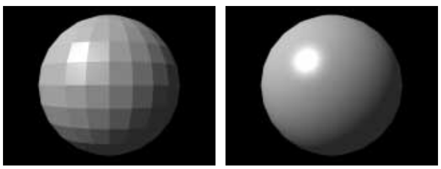
Nearly all real world surfaces have some amount of texture or bumpiness. Such bumps are apparent due to the way light falls across the surface. A **Bump** map is applied to a surface to give the appearance of a texture. However, no real geometry is added and if you looked across the surface from the side, the illusion could be ruined. Shadows and edges can also reveal this lack of geometry in certain circumstances. The default setting is 100%.

If you want your bumps to actually displace geometry, try the Bump Displacement option (Object Properties) discussed in Chapter 9.

Left: Bump mapped ball. *Right*: Close-up reveals straight-edge polygons and no bumps

Surface Smoothing

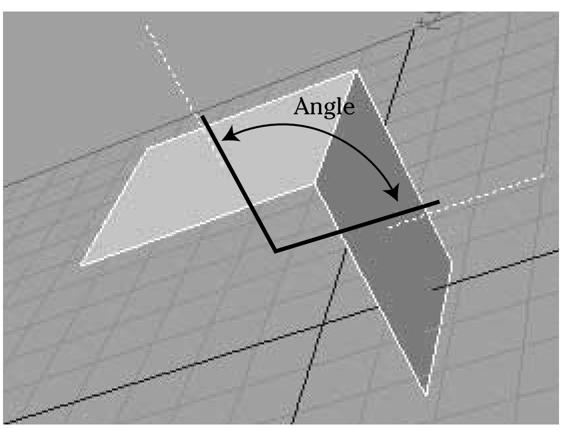
Smoothing causes objects to appear to have smoothly rounded surfaces even though the object is composed of flat-faced polygons. To do this, LightWave uses a technique known as *phong shading*. If the edges of two smooth-shaded polygons share vertices (points), they appear as one continuous, curved surface. The shared edge between them is no longer visible.



Left: No Smoothing. *Right*: Same surface with Smoothing

Smooth Threshold

By default, LightWave will not smooth across two polygons if the angle between them is 90 degrees or sharper, unless you adjust the **Smooth Threshold**. This value adjusts the range of the smoothing function; it defines the angle beyond which LightWave will not smooth over the seam between adjoining polygons. The angle is measured using the surface normals of the adjoining polygons. If this angle is less than the **Smooth Threshold**, the surfaces are rendered smoothly. The default of 89.5° assures that any surfaces at right angles (90°) or greater to each other are not smoothed.



How angle is measured between two polygons

Sometimes, due to the way an object is modeled, you may get unexplained rendering errors on smoothed surfaces. Try increasing the **Smoothing Threshold** to correct for such errors. We recommend using smaller increases first, although for extreme displacement-mapped objects, you may try fairly high values if rendering produces a mixture of jagged and smoothed results.

NOTE
As with **Bump**, **Smoothing** does not actually change the surface's geometry. As such, the edges of a ball can still expose the straight polygon nature of the object. If this becomes a problem, make the actual geometry smoother. You can use modeling tools like **Metaform**, for example.

Sharing Points

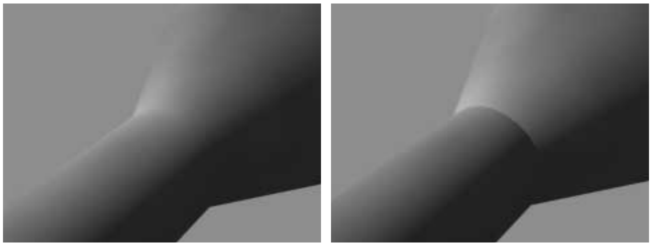
The concept of sharing points is important in surface smoothing. Unlike the real world, objects, and their points, can occupy the exact same space. A simple cube has eight points and six sides. Any one point is shared by three sides. However, you can create a seemingly identical cube that doesn't share any points or sides. Each side would be a totally independent polygon, thus there would be 24 points (six sides times four points).

The obvious reason for using shared points and sides is that it creates a more compact object. Remember, the simpler the object, the faster it renders and the lower the amount of resources used (i.e., RAM). However, sometimes you will find that you want the polygons separated.

Using Separation

Surface smoothing can occur only where two polygons share *all* of the points along an edge. However, sometimes this smoothing is not what you want. For example, if you were modeling something where you wanted a visible physical break, like a seam, you'd want to cut the polygons away and then paste them right back in. You may also find that when you have a flat surface that rounds over a corner, separating the flat polygons from the initial rounded polygons gives a more realistic look.

If you examine objects around you, you should notice that they all have seams. Very few objects in the real world are perfectly smooth with edges that align. Separating polygons can add subtle detail to objects that give them a real world look.



Left: surface smoothing. *Right*: separated polygons

NOTE
Separating polygons will double the points along the previously shared edge since each polygon now has its own set of those points.

HINT
Using the shortcut keys, x and v, to cut and paste, is a quick way to separate polygons.

This technique lets you control smoothing by using a single surface name. Although you could create independent surfaces with identical settings—except one has smoothing on and the other does not—you may prefer to have one with smoothing on and use this separation technique to create the seam.

Double Sided

Sometimes you want to see both sides of a polygon, such as when you want to go inside a model (and wall thickness is not a concern). Also, if you import objects from other 3D modeling programs, either directly or through a conversion utility, some polygons may end up facing the wrong direction, which causes the object to render improperly. Click **Double Sided** to treat all polygons on the selected surface as two-sided.

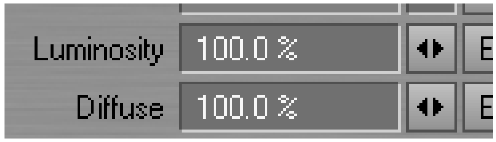
As a consequence of using Double Sided, rendering time will increase somewhat because LightWave must calculate more polygons. Note also that Double Sided is only a surface attribute. The object, when loaded into Modeler, will show the true single-sided polygons. To create true double-sided polygons, see Chapter 26.

Comment

You can add a comment to the surface in the **Comment** field.

NUMERICAL SETTINGS

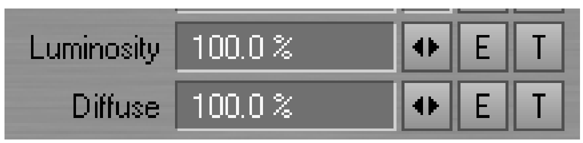
All of the numerical settings have mini-sliders. You can enter a numerical value into the input field or use the slider. The range of the slider is based on realistic settings; however, the input field will accept values beyond the maximum and minimum possible with the slider, including positive and negative values. Keep in mind, except for maybe Luminosity, values outside the normal range of 0 to 100 percent will be unrealistic.



Numerical input fields

ENVELOPES AND TEXTURES

Very few things in the real world have constant surface properties. Color patterns, discoloration, nicks, scratches, and such, can all subtly contribute to an object's appearance. The real world is anything but consistent, but that is what makes things look *real*. The **E** and **T** buttons let you use envelopes and textures, respectively, instead of a static numerical value. The proper use of these features often results in a much more realistic surface. The Texture Editor is discussed later in the chapter.



Envelope and Texture buttons

Envelopes let you vary a value over time. For example, instead of Luminosity having the same value throughout your animation, it can differ on each frame. However, in any particular frame, it will have that value over the entire surface. To vary the value over the surface area, you must use a Texture, discussed next.

Envelopes use the Graph Editor, which is discussed in depth in Chapter 8.

NOTE
Envelopes are not limited to surface values, so you will find envelope **E** buttons throughout LightWave.

TEXTURE EDITOR

While you can vary a parameter over time with an envelope, the parameter is not dynamic over the surface: the value for a particular frame is the value for the entire surface. Textures, on the other hand, essentially let you vary the value dynamically over the surface (as well as in time). A Color texture is probably the easiest illustration. Instead of using the same color on a surface, you can map a picture (a color texture *image map*) onto the surface that lets the color differ over the surface area.

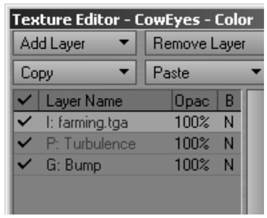
The Texture Editor

You can use image maps for more than just color. On other surface parameters, the color element is ignored and just the brightness of each pixel varies the value over the surface. For example, you can use an image map with Specularity to suggest scratches on a surface. Textures let you divide a surface into thousands of smaller areas and apply a different level of the surface attribute to each individual area.

In addition to **Image Map** textures, you can also use **Procedural Textures** and **Gradients**. Procedurals are more or less mathematically computed images. Gradients act as a type of envelope, except that the value can change over a condition other than time. Select these options by using the **Layer Type** pop-up menu.

Texture Layers

LightWave can layer surfaces infinitely; once you set the original texture, you can add another layer by using the **Add Layer** pop-up menu. You can choose to add any of the layer types—the type can be changed later if necessary.



The layer list window shows you all of the layers. The first column indicates whether the layer is active (checked) or not (blank). You can toggle the status by clicking in the column. The next column shows the layer name, which indicates the type of layer. Next is the opacity percentage, followed by the Blending mode (e.g., + for Additive).

You can choose which layer to work on by simply clicking on it in the window. To remove a layer, select it and then click **Remove Layer**.

Copy and Paste

You can copy the currently selected layer or all of the layers in the Surface list to a memory buffer by using the **Copy** pop-up menu. For the Paste operations, **Replace Current Layer** will replace the selected layer with the buffer contents, even if the buffer contains more than one layer. **Replace all Layers** will clear all existing layers and then add the buffer contents. **Add to Layers** simply appends the buffer contents to the end of the list.

NOTE
The buffer contents will remain until you quit LightWave.

Layer Order

Texture layers are always listed in order from top to bottom. New layers are added on top of existing layers. Keep this in mind when you set up multiple layers. For example, if you want an image map with fractal noise across the entire surface including the image, you must map the image first and add the fractal noise afterwards. Of course, you can reorder layers by dragging them with your mouse.

NOTE
The base surface settings—those set on the Surface Editor's Basic tab—always sit *beneath* all layers.

Blending Layers

To set the opacity of a texture, use the **Layer Opacity** field. Reducing the value from 100% makes the overall texture more and more transparent. Setting it above 100% can force a texture layer to unnatural values.

The **Blending Mode** determines how the layer is blended with other layers. With **Normal**, underlying layers will be totally covered (i.e., *replaced*) by the texture, assuming **Layer Opacity** is 100%. The texture acts like an alpha matte; thinner areas of the texture allow the underlying layers to show through. If **Layer Opacity** is 50%, you get 50% of the layer and 50% of the underlying layers. **Additive**, *adds* the texture (times the **Layer Opacity**) to underlying layers.

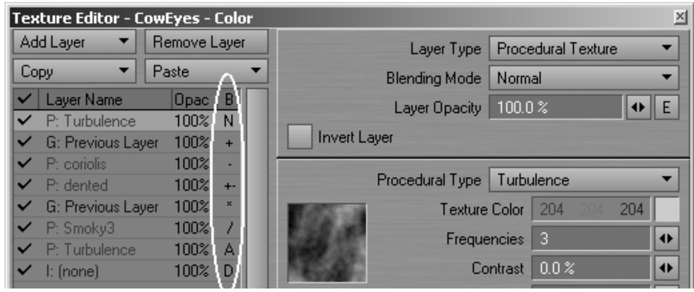
NOTE
In versions prior to 7, "Normal" was called "Additive." The current Additive mode is new for version 7.

To achieve an even blend between multiple texture layers, use **Normal**. Then divide the number of the layers into 100% and use the resulting value as your texture opacity value. For example, the first (bottom-most) layer would be set to 100% (100/1), the second layer will be set to 50% (100/2), the third layer will be set to 33% (100/3) and the fourth layer will be set to 25% (100/4).

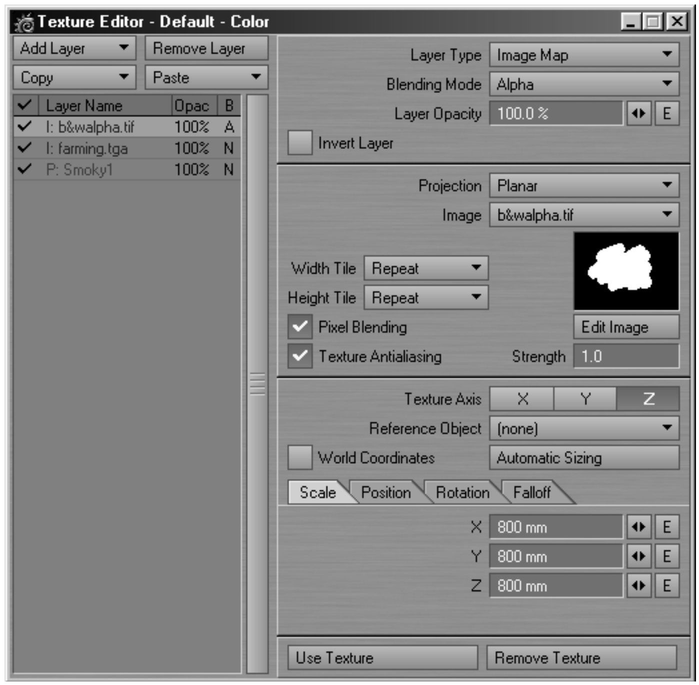
Subtractive subtracts the layer from the underlying layers. **Difference** is similar to **Subtractive** but takes the absolute value of the difference. **Multiply** factors the layer by the underlying layers. Multiplying by darker colors will darken the image, while brighter colors will brighten. **Divide** multiplies the underlying layers by inverse of the layer. This generally has the opposite effect of **Multiply**.

The **Alpha** blending mode makes the layer an alpha channel on the preceding layer. In other words, it *cuts out* parts of the preceding layer and makes those areas transparent. White in the alpha image creates opaque areas and black creates transparent areas. Shades in between will do a little of both. If the image, procedural, or gradient has color, the alpha image is based on the brightness of the areas.

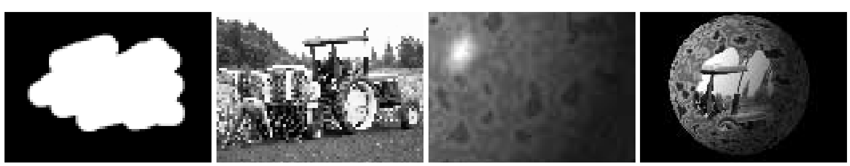
Each layer's blending mode is indicated in the right-most column of the layer list.



Layer blend mode indicator



Top layer: alpha image. Middle: image being cut. Bottom: Smoky1 procedural



Left to *right*: The three layers from top-most to bottom-most, the Surface Preview window

Texture Displacement displaces (distorts) layers above it, similar in effect to a bump map.

Texture Displacement

LAYER TYPE: IMAGE MAPPING

Except with the Color surface attribute, when you use image maps the brighter the data, the higher the setting, and the darker the data, the lower the setting. A totally white image is the same as a 100 attribute value and a totally black picture is the same as a 0 attribute value. As such, if you use a standard RGB picture, you cannot get values higher than 100 or less than 0, but it will usually contain a wide array of brightness values. However, if you use a high dynamic range image, discussed in Chapter 16, you can exceed this limit.

NOTE
To see image maps in the Layout view, make sure the viewport is set to Textured Shaded Solid. You can change this on a viewport's titlebar.

Image Map Projection

Since images are usually rectangular and surfaces may or may not be, you must tell LightWave how you want the image map *projected* onto the surface. The common projection types settings are **Planar**, **Cylindrical**, **Spherical**, **Cubic** and **Front**.

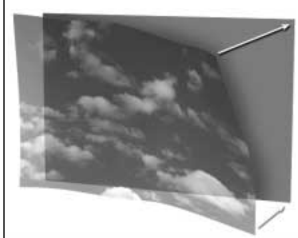
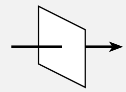
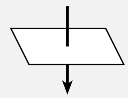
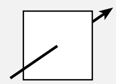
NOTE
See Chapter 28 for information on **UV** mapping.

Generally, you should pick the shape that best describes the surface shape. (Note that this is not necessarily the object's overall shape since that may be made up of many surfaces.) For example, if you were mapping a label image on the sides of a soda can, you'd use **Cylindrical**. For a planet, you'd use **Spherical**. For a wall, **Planar** would do the trick. For a brick, **Cubic** might be best.

What about a die? **Cubic**? This may be a trick question. Since a die has a different number of dots on each side, you'd use a different **Planar** (or UV) map on each one.

Planar Projection

Planar projection will project an image onto a surface as if you were projecting the image through a slide projector onto a wall. Planar image maps are best used on flat, or nearly flat surfaces like the sides of buildings and the screens of video monitors.

Planar Image Mapping	Axis	Diagrams	Examples
	X		Mapping insignia onto the tailfin of an aircraft
	Y		Mapping aerial photo onto terrain
	Z		Mapping a picture onto a TV screen

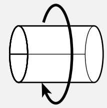
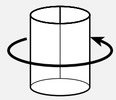
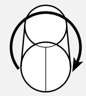
For the X and Y axes, Planar images are *projected* from the positive-axis side of a surface towards the negative-axis side. This means that the image appears correct when viewed from the positive side and it appears reversed if you view it from the negative side. For the Z axis, Planar images are projected from the negative side.

HINT

If you encounter this reversing effect and it isn't what you want, you can reverse an image back by making the **Scale** value negative for the axis that needs to be reversed.

Cylindrical Projection

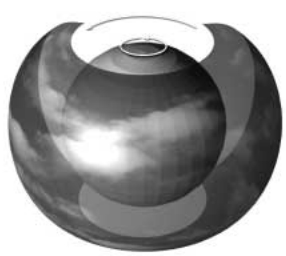
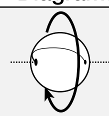
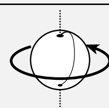
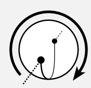
Cylindrical projection wraps an image around the selected axis like a paper towel wrapped about a cardboard tube. By default, an image is wrapped so it covers the surface once, allowing the side edges of the image to meet on the *back* of the surface. A soda can or tree trunk are both good examples of surfaces that would use Cylindrical projection.

Cylindrical Image Mapping	Axis	Diagrams	Examples
	X		Mapping detail onto a pipe segment
	Y		Mapping label onto wine bottle or soda can
	Z		Mapping detail onto an engine exhaust

Cylindrical projection is always wrapped about a surface so the top of the image appears towards the positive-axis side of the **Texture Axis**.

Spherical Projection

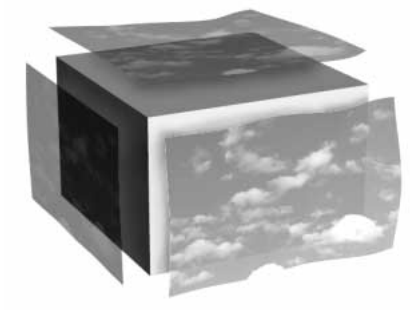
Spherical projection wraps an image around a surface as if you were stretching a flat piece of rubber around a ball, but without having to worry about the edges all meeting. Planets, basketballs, and marbles could all use a **Spherical** projection.

Spherical Image Mapping	Axis	Diagrams	Examples
	X		*Rarely Used*
	Y		*Mapping world map onto a globe*
	Z		*Rarely Used*

Spherical projection does not use **Scale** parameters. Images are wrapped completely around the surface (using the *wrap* values, discussed later). **Spherical** projection is always oriented so the top of the image appears toward the positive-side of the **Texture Axis**.

Cubic Projection

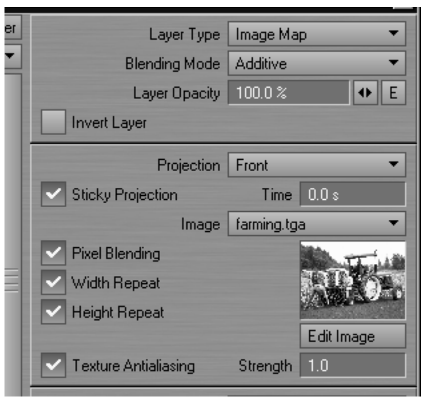
Cubic projection is essentially the same as Planar, except that you cannot select a **Texture Axis**. **Cubic** projects the image from all three axes at the same time. The image is projected like **Planar**, except simultaneously along all three axes. Use **Cubic** where you wish to apply the same image to all sides of a rectangular shape, such as an image of tiling bricks wrapped about the sides of a building, or wallpaper on the walls of a room.



Cubic Projection

Front Projection

The concept of **Front** projection is very simple and quite similar to a *chroma-key* effect. However, instead of applying an image where a color screen is, it replaces the selected surface(s) with a selected image.

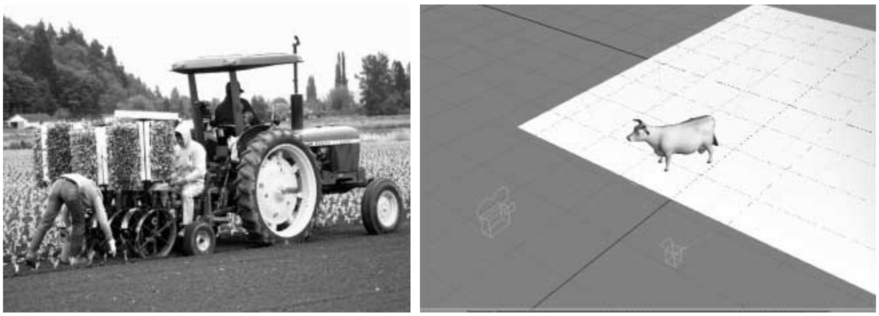


In most cases, the image you select for a front projection map is the same image you use for the **Background Image** on Layout's Compositing tab of the Effects panel (**Scene** > Effects: **Compositing**).

Scale, **Position**, and so on, are not relevant with Front projection. It is always the size (and frame/pixel aspect) it would be if loaded as a **Background Image**. As such, changing the **Resolution** or **Pixel Aspect Ratio** on the Camera Properties panel will also affect the Front projection.

Front projection is used primarily for *comp* (compositing) work where you combine LightWave objects with a live-action background image or sequence. A common example occurs when you want a LightWave object to cast a shadow (believably) onto the image or *behind* a portion of the background image.

The image used in the surface and the background will *pin-register* the surface to the background, letting you go in front or behind. Your object then appears to interact with the environment. You can cast shadows or cause reflections from a regular 3D object onto the surface that is front projection mapped.

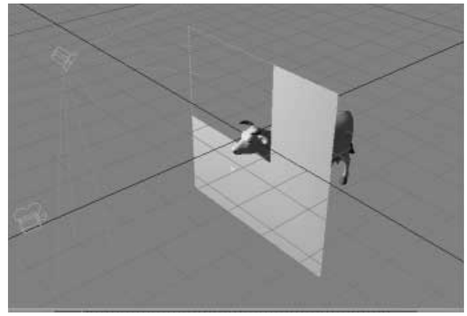


Left: Original image. *Right*: Layout screenshot

The ground object is just a flat box with **Front** projection image mapping that uses the same image as the background. Its job is merely to catch the cow's shadow. The cow is placed with her feet slightly below the level of the ground object, so it looks like she is standing in dirt.

Front-projection mapped ground surface catches cow's shadow

Here is an example of *poking* an object out from behind something.



Left: Original image. *Right*: Layout screenshot

The result. Notice the shadow on the window frame.

Another example is to use an image of trees as your background image and fly a UFO between them so the UFO appears to go in front of some trees and behind others. All you need to do is model some rough shapes that match the trees you wish to fly behind (they could even be flat planes).

Another good example for Front projection is to create a flat plane and align it to an image of an ocean or a lake. Front projecting the water surface onto it lets you place an object beneath the water and push it through the surface. Submarines and sea creatures will appear to *break the surface* this way.

The hardest part of front projection is aligning the objects, matching lighting, and getting the right camera angle. Using **Background Image** as the **Camera View Background** on Layout's Display Options tab of the Preferences panel (**Display** > **Display Options**) helps a lot. You also must search for the right balance of Luminosity and Diffuse for the Front projection surface so that the object's true shape is not revealed by shading.

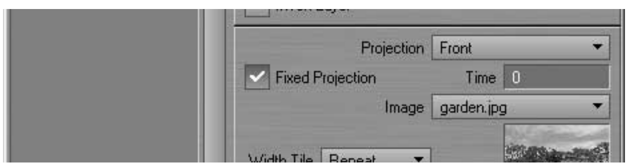
NOTE
Don't use the **Soft Filter** (Camera Properties panel) with Front projection. It will soften only the image used on the object surfaces, not the background.

Fixed Projection

Front projection surfaces will always look the same no matter where you move the object or which way you rotate it. The image does not normally stick to the surface. However, if you activate the **Fixed Projection** option (previously *Sticky Projection*), it *fixes* (i.e., locks) the projection from the camera's perspective at the specified **Time**.

The default unit of measure for **Time** depends on the **Frame Slider Label** setting on the General Options tab of the Preferences panel in Layout or the **Time Format** setting on the Display options panel, Units tab in Modeler. You may specify the unit of measure by appending *f* for frames or *s* for seconds to the entered number (e.g., 22f for frame 22, 31s for 31 seconds). You may also enter SMPTE timecode, like 00:00:01:16 for one second, frame 16. The entry is converted to the default unit of measure.

Use **Fixed Projection** to create parallax with two-dimensional images by using a technique called *camera mapping*. (Use the **Reference Camera** setting to select the camera, if you have multiple cameras in your scene.)

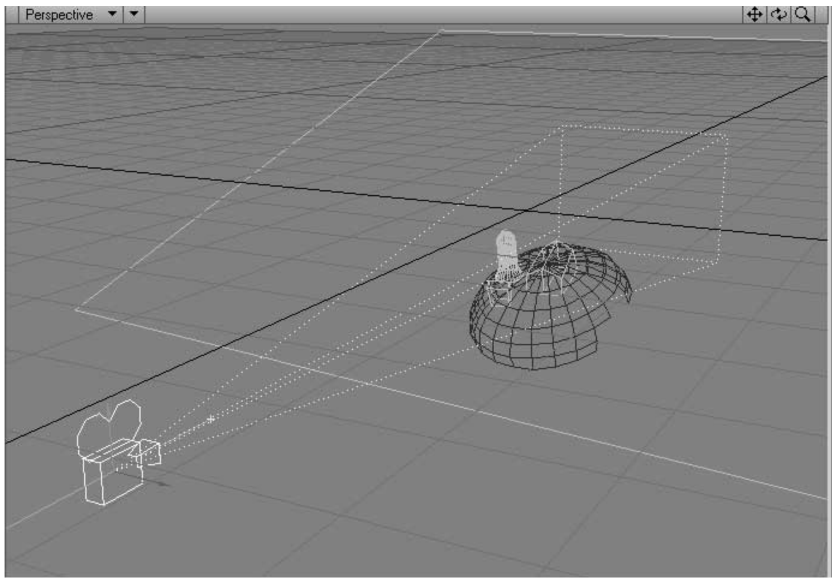


Essentially, you set the frame where the texture will be pin-registered to the background (like normal front projection mapping). On all other frames, the texture is stretched to compensate for more or less of the texture being visible, which is caused by factors like the camera being moved.

For example, if you had a picture of some buildings, you could place the LightWave camera in the same place as the original camera in relation to some simple 3D buildings. Then, project the picture onto the buildings and lock it at frame 0. You'll also need a *doctored* background image/object—also using Fixed Projection—that reveals stuff behind the buildings. If you move the camera forward, it will appear to fly into or around the buildings.

Left: Front projection map for buildings and ground. *Right*: Front projection map for background.

HINT
Use your paint program's rubber stamp function to erase areas where background Fixed Projection surfaces will be revealed.



Background image FP mapped to large plane. Foreground image FP mapped to other simple geometry

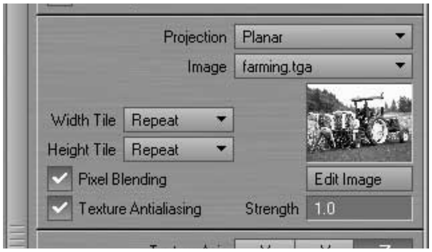
Left: Initial frame. *Right*: Last frame. Camera flies into image

NOTE
Obviously, there are great limitations in getting three-dimensional data out of a two-dimensional picture. However, short slow-moving sequences can be pretty realistic.

Image Properties

After you select one of the **Projection** settings, use the **Image** pop-up menu to select or load an image/sequence. Clicking the **Edit Image** button launches the Image Editor panel. Here you can adjust certain aspects of the image(s).

The available image settings on the Texture Editor panel vary depending on the **Projection** setting. Two settings always appear, however. **Pixel Blending** smoothes out the pixelization that can occur when a camera gets close to the mapped surface. **Texture Antialiasing** reduces artifacts in finely-detailed image maps and yields a softer look. Select it to help reduce or remove *scintillation*—those annoying lines of flashing pixels and moire patterns that are especially noticeable when fine lines in a texture image converge in perspective (particularly on television displays). Generally, choose **Texture Antialiasing** when a pattern is somewhat distant in the scene, such as a receding checkerboard pattern, although it can help with other busy texture patterns and images as well.

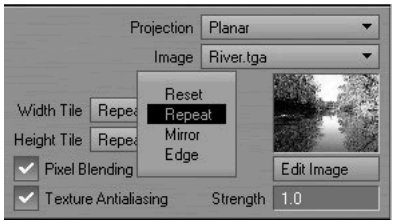


NOTE
Do not confuse **Texture Antialiasing** with **Antialiasing** on the Camera Properties panel. The latter one works primarily on the edges of objects.

The antialiasing **Strength** value determines the amount of antialiasing. The default setting of 1 should be used in most cases; however, the level may need to be raised or lowered slightly to better match the particular image. This value can be set to higher levels to add blurring effects to the image.

NOTE
Texture Antialiasing can cause the mapped image to appear blurry.

Planar, **Cubic**, and **Cylindrical** image map projection have tiling (i.e., repeating) options, which you can set independently for the horizontal and vertical repeating directions.



Reset Reset is the *no-repeat* mode. The underlying surface will be visible if the image is smaller than the surface area.

Repeat Repeat tiles the image until it fills the surface.

Mirror Mirror tiles the image, but flips the image horizontally or vertically.

Edge Edge extends the edges of the image to cover the surface. This setting works best when the outer edges are a solid color.

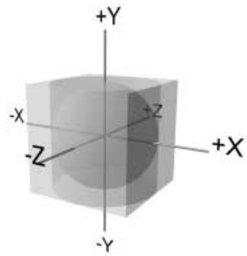
NOTE
The repeating options are relevant only if the **Scale**, **Position**, and **Rotation** settings (at the bottom) are set such that the image will not fill the surface area.

Cylindrical and **Spherical** have wrapping options that set how many times the image appears horizontally **Width Wrap Amount**, and vertically, **Height Wrap Amount** (not applicable to **Cylindrical**). This sets how many times you want the image wrapped for the given Scale values. Generally, you'll want this set at 1.

NOTE
Wrap amounts can be negative, which will reverse the image as it is wrapped around in the opposite direction.

Texture Placement

Once you set the image properties, you must now scale and position the image. Textures are initially positioned referenced to the **X**, **Y**, or **Z Texture Axis**. For **Planar** projection, think of the axis as a nail used to hold a photograph to a wall. For **Cylindrical**, think of the axis as the cardboard center of a roll of paper towels. **Spherical** can be thought of in much the same way. You cannot select an axis for a **Cubic** map, however.



For example, a soda can would use a **Y Texture Axis** because it sits vertically. The fuselage of an airplane, on the other hand, would probably use the **Z Texture Axis**.

NOTE
You should generally model objects so that they face the positive Z axis.

If the **Texture Axis** is perpendicular to a surface (such as the sides of a box using the same surface name as the *projected* side, the image will run *through* the surface.

Surface Size, Position, and Rotation

Scale defines the size of the surface to which the texture is applied. **Position** defines the coordinates of the center of the texture on the surface. **Rotation** defines how much the texture rotates around the center **Position**.

NOTE
Different combinations of **Texture Axis** and **Rotation** can achieve the same results.

Generally, you want to match these settings with the actual surface. **Automatic Sizing** causes LightWave to calculate the **Scale** and **Position** of a selected surface and input those values. It computes an imaginary bounding box around all areas using the current surface. This will fit an image map *perfectly* onto the surface. (Of course, it's our definition of perfect.)

HINT
Use **Automatic Sizing** for starting values when using procedural textures.

If the **Scale** is smaller than the surface size, an image map will tile across the surface (assuming you have one of the repeat options active). You might also see the tiling effect if the **Position** is not set to the actual surface center.

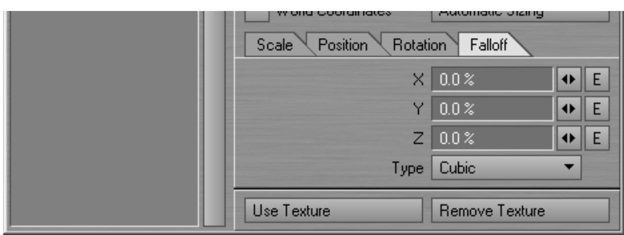
Because LightWave's textures are three-dimensional, there is no rule stating that the center of a texture must be located somewhere within the bounds of the surface. You can place the texture center outside of the surface and still get a texture to appear across the surface. Textures *extend* in all directions throughout the virtual space in Layout, but appear only on the surfaces that are set to display them.

There are definitely times where you don't want the texture sized exactly to match the surface. Such is the case with procedural textures, which usually look better when the **Scale** is smaller than the surface size because it adds more detail. Another example is when you use only part of an image that is texture mapped.

Falloff

The **Falloff** values specify the apparent falloff for every *unit of distance* moving away from the texture center, defined by the **Position** settings. (The unit of distance is defined by the **Default Unit** setting on the General Options tab of the Preferences panel.) When you animate the Position with an envelope, the position at frame 0 is used as the center for falloff purposes.

The **Type** pop-up menu determines the shape of the falloff. (In previous versions, the falloff was always cubic.) *Linear* Falloff types (i.e., LinearX, LinearY, and LinearZ) operate only in one direction. If you use 100% LinearX, for example, the texture will fall off only in the positive X direction from the texture center. To fall off towards negative X, use -100%.



With Cubic, the falloff occurs on both the positive and negative sides. You can achieve a linear-type falloff, but in both directions, by using Cubic and setting the two axes you do not wish to fall off to 0%.

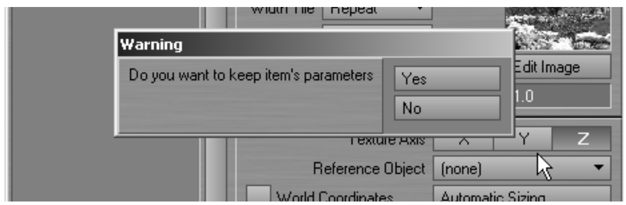
Using the Reference Object

Although you can explicitly set the position, size, and rotation of a texture, you can also assign a **Reference Object**—normally a Null object—and get similar results. The reference object makes all of these settings operate relative to itself. You can better control the texture animation by animating the reference object. Moreover, you can use different reference objects for different surface textures (e.g., surface color, diffusion, specular, and so on).

NOTE
An object's pivot point does not affect surfacing.

Freezing Reference Object/Camera

If you set a **Reference Object** (or **Reference Camera** used with Front Projection mapping) and then select **(none)** from the related pop-up menu, a dialog appears that asks, "Do you want to keep item's parameters?".



If you click **Yes**, you will store the reference item's state at the current frame in the object. (Don't forget to save it.)

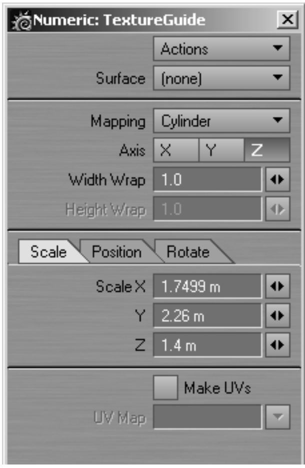
World Coordinates

Normally, textures are *locked* to a surface and travel with it, as the object is moved, stretched, rotated, or deformed. Selecting **World Coordinates** will lock the texture to LightWave's Origin instead of those of the surface. Moving the object now will result in the object traveling *through* the texture. All texture settings continue to be valid, but now are relevant to the world coordinates.

World Coordinates can create the look of light beams moving through a fog. You create this effect by using transparent light beams with a fractal noise texture for **Color** and/or **Luminosity** activated. As the light beam is moved about, it will move through the fog created by the fractal noise. Using **World Coordinates** on a **Transparency** map with falloff can make a spaceship *cloak* (turn invisible).

TEXTURE GUIDE TOOL

Modeler's Texture Guide tool (**Map** > Texture: **Texture Guide**) can interactively set position, rotation, and scale, as well as create a UV Map, for a surface texture. You can select the affected surface using the **Surface** pop-up menu.

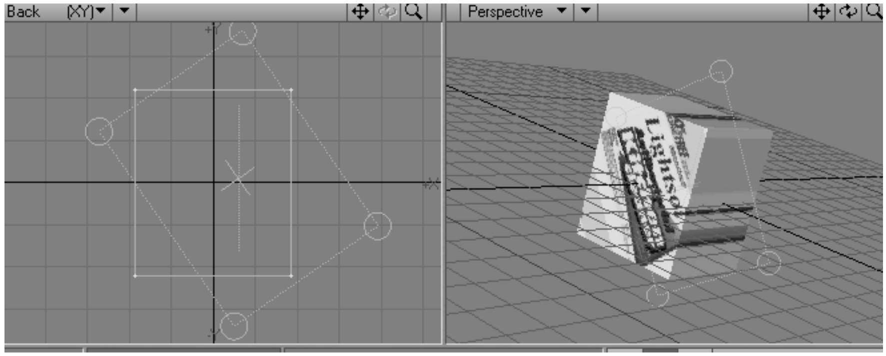


You can move the texture by using the center drag handle. For **Planar** and **Cubic** mapping, to resize the texture, drag any of the circle handles in the corners. To rotate **Planar**, spin the guide by dragging one of the rotation handles—these are lines coming from the center. (Rotation is sometimes difficult, so you may want to set this numerically.)

For **Cylinder** mapping, you can interactively adjust the mapping axis size. There are no interactive adjustments for **Spherical** mapping.

You can automatically create a UV map by selecting that option and entering a new UV map name or choosing an existing UV map from the pop-up menu.

The **Camera** mapping mode lets you create a UV map from the perspective of any point in 3D space, similar to pointing a camera in Layout. You can rotate the perpendicular axis by dragging the tip of the pyramid. The X handle moves the entire outline.



Use Texture Guide interactively while viewing in a Texture mode viewport.

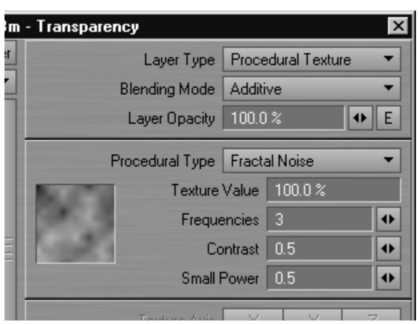
Note that you can affect only the first layer, if you have multiple layers in the texture. As such, you may need to drag a layer to the bottom of the list (in the Texture Editor), if you want to use Texture Guide on it.

NOTE
OpenGL will show only the first layer if a texture has multiple layers.

NOTE
Remember this is a tool, so deactivate it by selecting it in the menu again (or select another tool).

LAYER TYPE: PROCEDURAL TEXTURE

An alternative to using an image as a Texture is to use built-in mathematical computations called *Procedural Textures*. These serve the same purpose as image maps; however, since they are mathematically computed they are applied seamlessly on a surface. Thus, you do not need to worry about a projection method.



NOTE
All Procedural Textures are available for use on each surface attribute.

To add a procedural texture layer:

Click on the **Add Layer** button and select **Procedural** from the pop-up menu. You can also change an existing layer's type by changing its **Layer Type** to **Procedural Textures**.

NOTE
Generally, you cannot see procedural textures in a viewport. However, see Chapter 14 for a procedure to use the TextureFilter filter to make this possible.

Texture Color and Texture Value

Unless you are adding a texture to surface **Color**, you must specify a **Texture Value**, which is the value for the texture pattern at its most intense points. This can even be a negative value.

If you are adding a texture to surface **Color**, you do not define a value but a **Texture Color**. The color chosen here will be the color of the texture at its most intense points.

Unless the texture pattern fills all of the surface area, there will be transparent and semi-transparent areas, revealing underlying layers. In areas where the texture is below 100%, the texture will blend with underlying layers.

You can change the background color used in the thumbnail window by right-clicking on it—a color selection dialog will appear. You can also drag the texture around with your LMB.

Texture Scale

Generally, you want your scale to be a fraction of the surface size to properly see the texture on the surface. Unless you know the exact dimension of your surface, it may be handy to click **Automatic Sizing** first and then modify the resulting **Scale** values.

HINT
Use LightWave's math feature and append, say, "/4" to the automatic-sizing values. This divides the value by four. For example, adding "/4" to a value of "24 m" would look like "24 m/4" (don't enter quotes). When you move to the next field the result is calculated automatically!

Procedural Texture Settings

The following is a list of the Procedural Textures. Remember that although they are described in essentially surface color terms, they can also be used for Diffuse, Specularity, Transparency, and so on.

Brick

Brick produces an array of symmetrically spaced bricks. The **Texture Color** is the color of the mortar. **Mortar Thickness** determines the width of the mortar. At 1.0, no brick is visible. **Fuzzy Edge Width** makes the outer dot edges soft and faded. Since procedural textures are three-dimensional, you may need to tweak size and position values if a surface cuts through mortar and that is all you see on an edge.

Bump Array

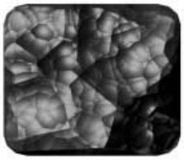
Bump Array produces an array of symmetrically spaced circular dots that appear to have depth. You can use this texture to create a golf ball surface.

Radius sets the radius of each bump used in the bump array pattern. **Spacing** sets the spacing between each bump.

Checkerboard

Checkerboard generates a tiled appearance on an object's surface, like that of a checkerboard or chessboard. Your XYZ Scale values set the size of each square in the pattern. For example, to see ten checkerboard squares on the surface of an object, divide its length by ten. A 50-meter cube would yield a **Scale** of 5 (for X, Y, and Z) to achieve ten squares.

Crumple



Crumple is a very detailed texture that gives the appearance of a surface that was *crumpled* under pressure. You can use it to simulate crumpled paper, hammered metal, ice cubes, stucco, and even cauliflower.

Frequencies refers to the number of different scales of detail added to the pattern. A value of 1 makes a surface appear partially crumpled so that it has only a few large dents. A value of 5 makes a surface appear very crumpled so that it has many smaller dents.

Small Power affects the appearance of the large and small features added to a surface. The default values create the appearance of a surface with large and small dents. A higher **Small Power** (1.0 or above) causes the smaller dents to be shaded with the same degree of intensity as the larger dents so that the surface becomes busier and the larger dents lose some of their distinction. A lower **Small Power** (under .50) makes for less distinction between the large and small features.

Crust

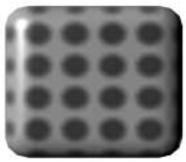
Crust is one of the more complex textures. It produces raised circular splotches on a surface. This is good for barnacles, warts, or moon craters.

Coverage determines how much of the crusty pattern covers the surface. A low value (0.0 to 0.5) makes many small islands of crust across the surface, whereas if you use a high value (0.6 or higher) the crust pattern begins to cover the surface color underneath.

The patterns that display **Ledge Level** are shaded so that the surface appears to have raised edges. **Ledge Level** shifts the position of the *ledge* that defines the edges of the pattern further outward from the pattern itself, into the surface color. The values that make visible changes in these patterns are subtle, so change them in small increments until you understand their effects well (on the order of 0.1 at a time).

Ledge Width alters the apparent depth of the *valleys* formed by the ridges in the pattern. The values that make visible changes in these patterns are subtle, so change them in small increments until you understand their effects well (on the order of 0.1 at a time). **Ledge Width** will affect the sharpness of the slope falling off of the discoloration pattern. The default values create a ridge just along the outside of the pattern discoloration. Higher values move the ridge further outward from the pattern, lower values move it inward. Most useful values will fall between 0.3 and 1.0.

Dots



Dots produces an array of evenly spaced dots. The dots can optionally have a soft edge. **Dot Diameter** determines the size of the dots in the dot pattern. At 1.0 the edges of dots will touch. **Fuzzy Edge Width** makes the outer dot edges soft and faded

FBM

FBM is another fractal pattern that uses the same settings as **Fractal Noise**.

FUN FACTS
Fractional Brownian motion (FBM) is named after botanist Robert Brown, who observed the random movement of pollen grains. Einstein later explained that the movement was due to surrounding molecules moving in random directions.

Fractal Noise

Fractal Noise produces a random fractal pattern. It is undoubtedly the most commonly used texture, since it can quickly mask somewhat the computerized look of 3D imagery. Use this as a bump map for terrains and brushed metal (use an exaggerated value along the *grain* direction). As a transparency map, **Fractal Noise** can generate realistic-looking clouds. As a surface color (or diffuse) map, it can give a weathered look to surfaces, such as grass or dirt ground. You'll find numerous possibilities for this texture. (Also see the **Turbulence** texture.)

Frequencies affects the level of detail produced in the noise pattern. Increasing this level will increase the variation in the pattern. Values above 6 are not useful (the level of detail is so small it may not be noticeable, and it will increase rendering time unnecessarily). **Contrast** adjusts the blending of the texture. The higher the level (above 1.0), the greater the contrast, and the more pronounced the pattern. Values lower than 1.0 (between 0 and 1.0) produce a less stark, more softly blended pattern. **Small Power** refers to the amount of intensity applied to both large and small details. It changes the intensity of the smaller details. A higher **Small Power** (1.0 or above) causes the smaller dents to be shaded with the same degree of intensity as the larger dents so that the surface becomes busier and the larger dents lose some of their distinction. A lower **Small Power** (under .50) makes for less distinction between the large and small features.

Use very small XYZ **Scale** values in a bump map to add random pits to surfaces.

Fractal Noise with small XYZ Scale values

NOTE
The Fractal Bumps option in previous versions of LightWave—only available for bump maps—was really **Fractal Noise** with **Contrast** and **Small Power** set to .5.

Grid

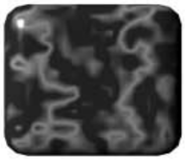
Grid generates a grid across an object's surface. The grid is a three-dimensional grid. Therefore, lines project into all three dimensions. Often, you want a 2D grid superimposed on an object's surface, like graph paper. In such cases where you use the grid texture and see unwanted portions of the grid showing up, try expanding the size of the grid texture along that axis (this expands it off the surface of the object). For example, if you map a grid onto a one-meter ball, the texture scale of the Z axis can cause the appearance of ripples that break up the nice graph paper look of the X and Y texture scale. Using a texture scale of 0.25m, 0.25m, 1m will get the proper look.

The **Line Thickness** value determines the thickness of the lines in the grid pattern.

HoneyComb

HoneyComb produces a *hive* of activity in the form of a honeycomb (*won't cha be my baby...*). The **Texture Color** is the color of the lines. **Line Thickness** determines the width of the lines. **Fuzzy Edge Width** makes the outer dot edges soft and faded.

Marble



Marble produces fractal-like patterns that imitate marble veins. The pattern is calculated in veins that wrap around a selected axis, much like rings wrap around the center of a tree.

Frequencies sets the level of detail produced in the pattern. The higher the value, the better-looking the pattern becomes, but the longer the rendering time as well. (A value above 6 is not useful here, because the variance is microscopic in size and virtually invisible.) The **Turbulence** value should be a fraction (often one-half) of the **Vein Spacing** value. It determines how close a vein may come to a neighboring vein. **Vein Spacing** sets the distance between veins in the marble pattern. **Vein Sharpness** sets the level of blending/contrast of the pattern. The lower the value, the more blending that will occur. Higher values will produce very sharp, distinct veins.

HINT
Select a **Vein Spacing** and **Texture Axis** that you like first, then set **Frequencies**, **Turbulence**, and **Scale** afterward.

Ripples

Ripples and **Ripples2** produce the appearance of fluid waves or ripples to a surface. Use small ripples to simulate water, or large ripples to put a wave in a flag.

Wave Sources determines the number of ripple sources. The higher the value, the greater the number of areas rippling with waves. A value of 1 will create a single ripple pattern, like that of a solitary water droplet falling into a pond. Values higher than 16 are not recommended; they require longer rendering time and may not add to the appearance of the texture.

Wavelength controls the distance between the ripples. The lower the value, the closer the waves will appear to each other. **Wave Speed** sets the speed of the ripples.

Looping Wave Ripples

In order to loop the movement of ripples throughout the course of the animation, use this formula to determine the proper **Wave Speed**: **Wavelength** divided by the number of frames over which the pattern should loop equals the **Wave Speed** (i.e., **Wavelength**/# of frames to loop = **Wave Speed**).

Smoky1, 2, and 3

The **Smoky** textures use the same settings as **Fractal Noise** with the addition of a **Turbulence** control, which lets you adjust the pattern disturbance.

Turbulence

Turbulence combines fractal noise layers, each at a different scale (or frequency). It produces an effect very similar to **Fractal Noise**, but will normally yield more eye-pleasing results. It also gives better control over the range of frequencies found in the noise, from smooth to detailed, grainy textures, due to the small scale noise that is added in. The settings are the same as those for **Fractal Noise**.

Underwater

Underwater produces the rippling pattern effect of refracted light, much like you see on the bottom of a swimming pool. You can also use this texture to simulate nighttime sky effects such as the Aurora Borealis, changes in cloud patterns, or even electrical shocks.

NOTE
You can generate actual real world effects by using Layout's Caustics feature (**Lights** > Global: **Global Illum**), but rendering times will be significantly greater.

Wave Sources determines the number of ripple sources. The higher the value, the greater the number of areas rippling with waves. A value of 1 would create one ripple. Values higher than 16 are not useful. **Wavelength** controls the distance between the ripples. The lower the value, the closer the waves will appear to each other. **Wave Speed** sets the speed of the ripples. **Band Sharpness** sets the level of blending/contrast of the pattern. The lower the value, the more blending that will occur. Higher values will produce very sharp, distinct bands.

Looping Wave Ripples

In order to loop the movement of ripples throughout the course of the animation, use this formula to determine the proper **Wave Speed**: **Wavelength** divided by the number of frames over which the pattern should loop equals the **Wave Speed** (i.e., **Wavelength**/# of frames to loop = **Wave Speed**).

Value

The **Value** procedural lets you create a layer of uniform value or color. You can use this to composite layers with alpha channels.

Veins

Veins produces a series of raised areas separated by canals or veins. This is great for cracked mud, leaded glass, stone walls, leaves, and so on.

Coverage determines how much of the vein pattern covers the surface. A low value (0.0 to 0.4) applies the vein color only to veins themselves, whereas a high value (0.5 or higher) causes the vein color to fill in between the veins, overtaking the surface color underneath.

The patterns that display **Ledge Level** are shaded so that the surface appears to have raised edges. Ledge Level shifts the position of the *ledge* that defines the edges of the pattern further outward from the pattern itself, into the surface color. The values that make visible changes in these patterns are subtle, so change them in small increments until you understand their effects (on the order of 0.1 at a time).

Ledge Width alters the apparent depth of the *valleys* formed by the ridges in the pattern. The values that make visible changes in these patterns are subtle, so change them in small increments until you understand their effects (on the order of 0.1 at a time). **Ledge Width** affects the depth of the veined ridges. Values just above the default setting have the most visible effect. The default values create a ridge just along the outside of the pattern. Higher values shift the ridge further outward from the pattern, lower values move it inward. Most useful values fall between 0.2 and 1.0.

Wood

Wood is similar to **Marble**, but produces a pattern imitating the rings in a piece of wood.

Frequencies sets the level of detail produced in the pattern. The higher the value, the better-looking the pattern becomes, but the longer the rendering time as well. (A value above 6 is not useful here, as the variance is microscopic in size and virtually invisible.) The **Turbulence** value should be a fraction (often one-half) of the **Ring Spacing** value. It determines how close a wood ring may come to a neighboring ring. **Ring Spacing** sets the distance between rings in the pattern. **Ring Sharpness** sets the level of blending/contrast of the pattern. The lower the value, the more blending that will occur. Higher values will produce very sharp, distinct rings.

HINT

Start by selecting a **Ring Spacing** and **Texture Axis** that you like first, then set **Frequencies**, **Turbulence**, and **Scale** afterward.

Additional Procedural Textures

A number of additional procedural textures are available; these textures are based on the noise and fractal routines presented in the textbook *Texturing and Modeling: A Procedural Approach* by David Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steve Worley (Morgan Kaufmann Publishers, 2nd Ed., July 1998, ISBN 0122287304).

Fractals are the most commonly used tool in computer graphics for adding visual complexity to 3D surfaces. Fractals create this complexity by repeating a common underlying noise pattern at various scale sizes, and accumulating those scaled patterns into a final surface texture.

The following parameters are common to most of the following procedural textures:

Increment controls the fractal dimension of the texture. When set to zero, the fractal function looks like white noise. As the value gets larger, the fractal will become smoother.

Lacunarity is the amount of change in the frequency of noise that occurs between each successive iteration of the fractal calculation. As this parameter increases, the average size of the gaps between the scaled patterns will also increase.

Octaves are the number of times the texture is scaled down to a smaller pattern, and added back again to the larger pattern. Larger values add more small details to the final texture, but also increase rendering times.

Offset exists in all the multi-fractal procedurals (where the fractal dimension varies throughout the texture). When set to zero, the fractal dimension will change greatly throughout the texture. This causes the roughness of the texture to vary greatly across the surface of the texture. Larger values cause the roughness (or smoothness) to be more consistent across the entire surface of the texture.

Threshold specifies a threshold value that is used to determine whether the texture should be displayed or not. If the procedural texture value is higher than threshold, that value modifies the surface. If the procedural's texture value lower than threshold, that value does not alter the existing surface attribute at all.

Noise Type uses Perlin noise as the most common and fastest noise function available in these procedurals. The other options (Value, Gradient, Value-Gradient, Lattice Convolution and Sparse Convolution) are different implementations of noise written by Darwyn Peachey and described in chapter 2 of *Texturing and Modeling: A Procedural Approach*, referenced above. While these implementations may provide better quality noise, they are definitely slower than Perlin Noise.

Coriolis

Coriolis is a texture used to simulate the shearing effect of the atmospheric flow on earth caused by the earth spinning faster at the equator, and slower towards the north and south poles.

Coriolis Scale scales the value calculated by the coriolis texture. Its effect is to vary the contrast between the *clouds* and the underlying surface. Smaller values create less contrast, and larger values display a high contrast between the clouds and surface.

Coriolis Twist is the amount of twist or rotation of the clouds from the poles to the equator.

Coriolis Offset is added to the value calculated by the coriolis texture. Larger values create more dense clouds and smaller values result in fewer, thinner clouds.

Cyclone

Cyclone is a turbulence texture with a single vortex, used to simulate cyclones and hurricanes.

Cyclone Radius is the maximum radius of the cyclone. Outside of this radius, the clouds will not appear twisted.

Cyclone Twist is the amount of twist or rotation of the *clouds* within the cyclone radius.

Cyclone Offset is added to the value calculated by the cyclone texture. Larger values create more dense clouds and smaller values result in fewer, thinner clouds.

Dented

Dented is a turbulent noise function that creates crumpled dent patterns. **Scale** adjusts the magnitude of the texture output.

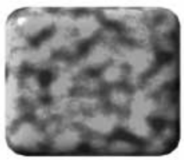
Power is the fractal dimension of the dent texture. A value of 1.0 looks like crumpled paper. 3.0 makes smaller, isolated dents on the surface.

Frequency is the frequency of the dents, affecting the detail of the pattern.

FBM Noise

FBM Noise is a typical homogeneous fractal noise function since the fractal dimension does not vary.

Hetero Terrain



Hetero Terrain is a multi-fractal texture that is smoother at lower elevations and progressively gets rougher as the altitude increases.

Hybrid Multi-fractal

Hybrid Multi-fractal is another multi-fractal texture that smoothes the *valleys* of the texture at all altitudes, not just at lower elevations.

Multi-fractal

A *multi-fractal* is a function whose fractal dimension varies depending on the location of the point being shaded or displaced.This is similar to FBM, except that it uses multiplication in its inner loop computation rather than addition.

Puffy Clouds

Puffy Clouds is a *thresholded* FBM noise function that creates soft, puffy cloud patterns.

Ridged Multi-fractal

Ridged Multi-fractal is a hybrid multi-fractal texture that uses a threshold value to create ridges in the terrain.

Turbulent Noise

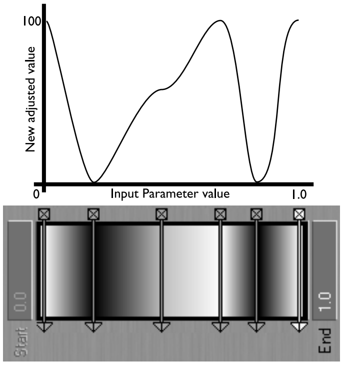
Turbulent Noise (formerly TurbNoise) is a modified FBM texture that uses an absolute value function, adding a turbulent effect to the texture.

LAYER TYPE: GRADIENT

While envelopes let you vary a setting based on time, a gradient lets you vary a setting (color, luminosity, etc.) based on features like height of surface bumps, or distance to another object. So, for example, with an envelope, you can make the Diffuse level 20% at frame 23. With a gradient, you can make the Diffuse level 20% if a surface is 3.23 meters from the Cow object or the slope angle of the surface is 26.2 degrees.

The **Input Parameter** determines what feature the setting varies over. Think of gradients as *filters* for the selected **Input Parameter**. For example, if the surface is bumpy, colors can be different based on the height of the surface being colored if the **Bump** option is used. You can even use the preceding texture layer.

Gradients use *gradient ramps* (the colored bar) to depict the value change. You define different values along the bar. LightWave automatically determines the in-between values. Essentially, the different colors along the gradient correspond to texture color/values for the related **Input Parameter**. The gradient is just a colorful (and clever) way to create a graph.



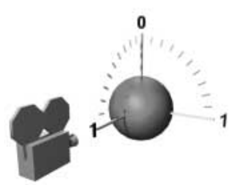
Gradient and equivalent XY graph format

The **Input Parameter** defines what item will dynamically control the parameter.

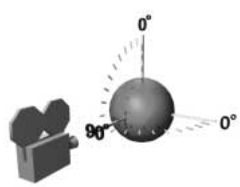
Previous Layer Previous Layer uses the brightness data from the immediately underlying layer.

Bump Bump uses the height of surface using the Layout's **Default Unit** on the General Options tab of the Preferences panel.

Slope Slope changes the parameter based on the angle (in degrees) of the surface relative to the ground (flat on Y axis).



Incidence Angle Incidence Angle uses the angle (in degrees) of the surface relative to the Camera.



Light Incidence Light Incidence is similar to **Incidence Angle**, but the angle is relative to the selected light. Use the **Light** pop-up menu that appears.

Local Density (HyperVoxels only) Local Density uses the density of the HyperVoxel.

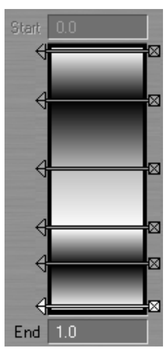
Distance to... The Distance to... settings change the parameter independently based on the distance from the surface to the selected camera or object. The X, Y, and Z distance settings measure the distance only along the respective axis. If you select the distance to an object, an **Object** selection pop-up menu will appear.

Weight Map Weight Map uses the weight map selected on the **Weight Map** pop-up menu.

NOTE
Other **Input Parameter** options may be available depending on what feature the gradient texture controls.

The Gradient Bar

OK, once you determine your **Input Parameter**, you need to set the value of the new texture color/value. The Gradient Bar is where you graphically set your key values. (A *key* is characterized by three numbers: its value, parameter, and alpha.) As with envelopes, you can create, edit, and delete keys. The keys change the color of the gradient and ultimately the parameter values.



If the parameter is a color, the bar is in color and color transitions appear from one key to the next. If the parameter relates to numeric values, the bar appears in grayscale. The lowest set key value is black and the highest is white.

The **Start** and **End** fields, at the top and bottom of the Gradient Bar, respectively, set the beginning and end of the visible area of the gradient. Whether or not these values are editable, depends on the **Input Parameter** you selected; however, their default values should work for most situations. Also, you can set keys outside this range.

To add a key:

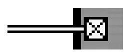
1 Click on the Gradient bar and hold down your LMB. A key marker that looks like an arrow will appear.

If you continue to hold the mouse button down, you can drag to refine the position of your key. The **Parameter** field will change as you drag your mouse. Note that this value is not limited to whole numbers.

NOTE
To prevent accidentally adding a new key, you can drag the key using the arrowhead.

2 Release the mouse button to create the key.

To delete an existing key:

Click on the end of the arrow. 

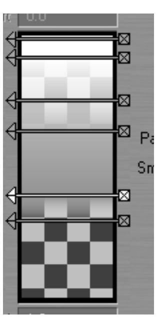
The Current Key

The **Value/Color**, **Alpha**, and **Parameter** fields show the corresponding settings for the *current key*. This is the key that is highlighted in the Gradient Bar. Click on a key's arrowhead to select it as the current key. When the panel is active, you can also use your UP and Down Arrowkeys to select the preceding or next key as the current key. (Make sure that you do not also have an input field active.)

NOTE
When you edit a surface Color, the *value* is a color selector. All other types of textures have a numeric input field for value.

The **Parameter** setting determines the current key's numerical position on the bar. Note that this setting can have decimals. The **Value**/**Color** is the texture value or color for the current key.

The **Alpha** setting is the amount of opacity. Higher values are more opaque: 100% is fully opaque and 0% is fully transparent. You can see a background checkerboard when you use **Alpha** settings below 100%.

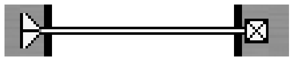


Changing Key Values

You can change the current key's **Color** or **Value**, as the case may be, by adjusting that setting. You see the gradient bar change as you adjust the setting.

To prevent a key from being changed:

Right-click on the arrowhead. It will invert, indicating the key is fixed and cannot be altered.



Changing Key Positions

You can move a key—but not past a neighboring key—by dragging the key's arrowhead with your mouse.

NOTE
Although you can also drag the middle of the arrow, it's not recommended because you can accidentally add a new key.

You can also numerically move a key by entering a value (including a decimal if needed) into the **Parameter** field. Note that you still cannot move the key past a neighboring key.

Smoothing Between Keys

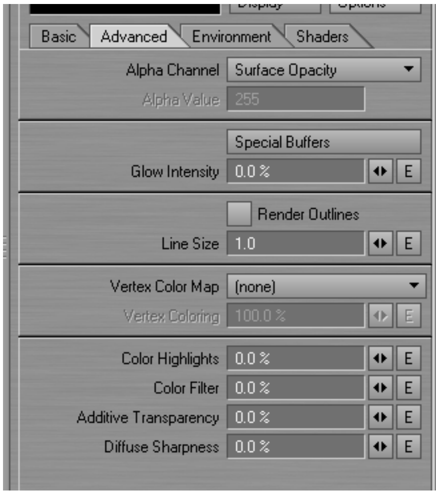
By default, the values between a key and the preceding key (next one above it on bar) are transitions in a **Linear** fashion. Setting the Smoothing pop-up menu to **Spline** will apply an *ease-in/ease-out* transition to the preceding key. The **Step** setting will hold the preceding key value until the next key.

Scale and Shift

You can scale and shift the key <u>positions</u> by dragging up and down on the **Scale Keys** and **Shift Keys** buttons. Similarly, you can scale and shift the key <u>values</u> by dragging on the **Scale Values** and **Shift Values** buttons. **Invert Keys** reverses the values of the keys. This will not affect the position of keys on the bar, however.

SURFACE EDITOR: ADVANCED TAB

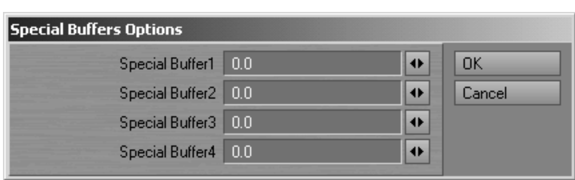
The **Alpha Channel** option affects the contents of LightWave's alpha buffer. Thus the alpha image is saved when saving alpha images is active on the Render Options panel (**Rendering** > **Render Options**). This option is applied on a surface-by-surface basis; each surface can have a different setting.



Surface Opacity, the default, uses the opacity of a surface to create the corresponding alpha area. Thus, if a surface has some level of transparency, the alpha channel information will be somewhere between 0 and 255. If you select **Unaffected by Surface** the surface has no effect on the alpha image. When you select **Constant Value**, the **Alpha Value** field can be set to a specific number from 0 to 255. **Shadow Density** uses the density of shadows in creating the alpha image.

The **Special Buffers** function works in conjunction with certain image or pixel filters and lets you make surface-by-surface adjustments. Essentially, the settings are arbitrary values associated with a surface and can be used any way the filter wants. For example, the Corona image filter can use special buffers for its Threshold Mask setting.

When you click the **Special Buffers** button, a dialog will appear with four input fields. The values you enter depend on what the filter seeks.



NOTE
Use values from 0 to 1 for older filters requiring 0 to 255 integer values.

The **Special Buffer 1** field relates to the first filter in the image or pixel filter list on the Processing tab of the Effects panel. **Special Buffer 2** relates to the second image filter, and so on. The **Special Buffer 4** field works on the fourth filter and all that follow it.

Adding a **Glow Intensity** amount will render a glow around the selected surface. The overall amount and intensity of the glow are determined by the settings entered for (glow) Intensity and Glow Radius on the Processing tab of the Effects panel (**Scene** > Effects: **Image Process**). The glow will appear in the color of the surface, even if it is texture mapped.

NOTE
Enable Glow must be active on the Effects panel in order for LightWave to calculate glows around surfaces.

Render Outlines renders the outline of a polygon rather than its face. All of the polygons belonging to the current surface will render with their edges appearing in the surface color. If you apply an image or texture map to the surface, the outlines will show portions of this mapping. Use this option to surface open frameworks or the gaps around hatches. **Line Size** is outline line size in pixels. Lines are drawn with the current surface color.

The **Vertex Color Map** pop-up sets the vertex color map you wish to use. This feature colors the object vertices. (See Chapter 28 for details.)

Color Highlights causes a surface's own color to *show through* a specular highlight. Normally, a specular highlight shows in the color of the light striking the object. For example, a white light illuminating a red ball leaves a white specular highlight on the surface of the ball. But with **Color Highlights**, the highlight appears red. A color highlight blends the object's surface color with the color of the light source, which causes a more metallic appearance. When a reflection map is used, reflection maps are also tinted by the surface color.

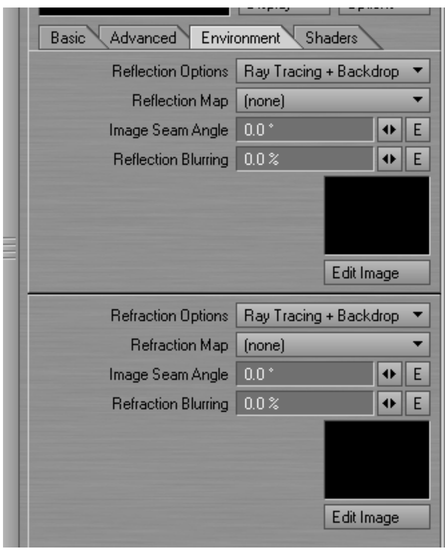
Color Filter is available when a surface has some degree of transparency. It allows the surface color of an object to tint all of the objects seen through it. A good example is a green wine bottle. Light shining through the bottle tints all objects seen behind it with green.

With **Additive Transparency**, the color of an object is added to the colors of the objects visible behind it. (Obviously, the surface must be somewhat transparent.) The surface therefore appears brighter in color and intensity, as if it was overexposed. The brighter color will tend ultimately toward white (255, 255, 255), because this is the brightest it can reach. Use this control to enhance the glow of rocket flames, jet exhausts, and the like.

Diffuse Sharpness causes a more severe *shadow cutoff* zone on objects rather than the gradual falloff of light. It enhances the appearance of contrast between the lit and unlit sides of the object. **Diffuse Sharpness** is usually used for very large-scale objects, like planets.

SURFACE EDITOR: ENVIRONMENT TAB

Reflection Options determines what you want reflected onto the surface. You can choose between different combinations of the backdrop (Background Image), actual ray tracing, and an image map. **Spherical Reflection Map** causes the reflective surface to reflect the image selected as the **Reflection Map**. This reflection is a spherical image that is mapped onto the inside of an all-encompassing sphere that surrounds the scene at an infinite distance.



Environment tab

Image Seam Angle determines where LightWave places the *seam* of the reflected image. The 3D universe is considered a mathematical sphere, with all rendering occurring inside this sphere. LightWave uses the **Reflection Map** image to wallpaper the inside surface. In this manner, the image can be reflected off the different surfaces on objects within the scene, no matter which angle they face.

Unless the image you choose is seamless (i.e., it has *matched* edges), a visible seam appears where its edges meet. You may or may not see this seam in the reflection, depending on where you point the camera. If the seam is visible, you can adjust where it falls by changing the **Image**

Seam Angle. This setting corresponds with heading in Layout—as if you were standing at the center of the Layout grid (0, 0, 0), and looking straight ahead into the positive Z axis.

The **Reflection Blurring** value allows you to add soft ray-traced reflections. The value controls the spread of the semi-random reflection rays. This is a sensitive settings and small values go a long way.

With (*left*) and without (*right*) Reflection Blurring

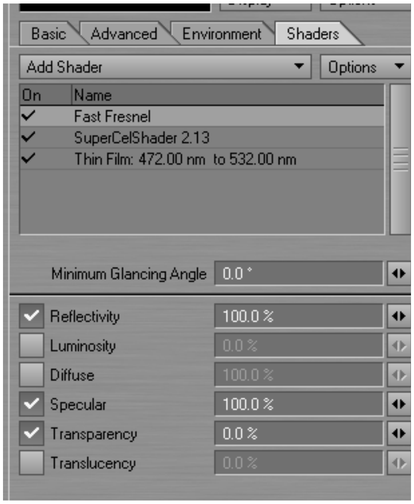
Refraction Options have the same available settings but affect what is refracted through the surface. Note that if you select **Spherical Refraction Map**, you will not see any objects through transparent surfaces. If you select **Ray Tracing + Spherical Map**, you should see other objects being refracted, with the spherical map behind them. The rays will bend and go off until they either hit an object or the map.

NOTE
If you use Ray Tracing options, you must activate the **Ray Trace Reflection** and/or **Ray Trace Refraction** options on the Render panel (**Rendering** > **Render Options**). These will also take much longer to render.

NOTE
Obviously, your success in using either of these mapping options depends on how closely the viewer can scrutinize the affected surface.

SURFACE EDITOR: SHADERS TAB

LightWave lets you use surface shaders to change surface attributes. Here you can add, remove, and reorder shaders, as well as change their settings. To add the shader, select it from the **Add Shader** pop-up menu. To access its settings, double-click on the shader in the list. Depending on the shader, the options, if any, will either appear in their own dialog or below the shader list.



Shaders tab with options showing for selected shader

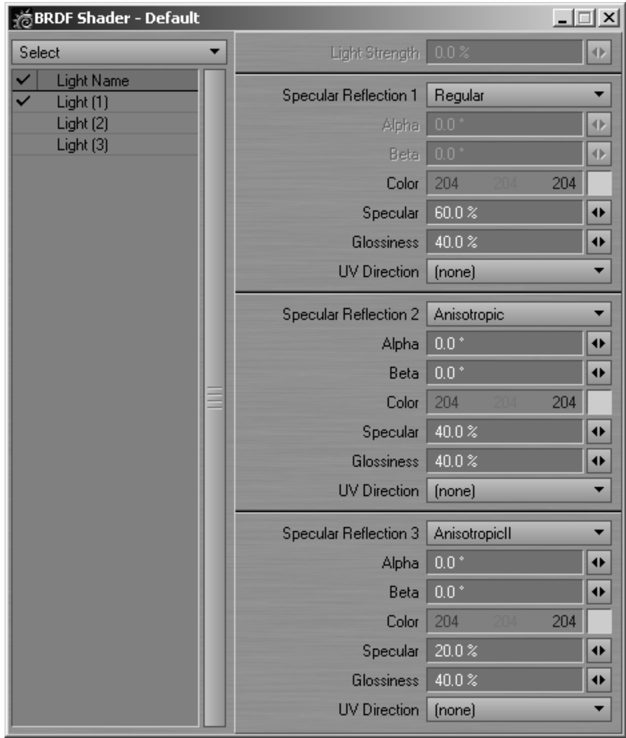
NOTE
If a shader does not appear in the pop-up menu and you think it should, you may need to add the plug-in. See Chapter 4 for information on how to do this.

BRDF

BRDF (Bidirectional Reflectance Distribution Function) is really a multi-function shader. First, you can totally exclude specified lights from affecting the surface. You do this by unchecking the light in the list.

BRDF's second function is to let you stack up to three specular highlights with different settings. Multiple specular layers are important for surfaces like the car paint on a shiny red 1996 Mazda Miata named *Seiko*, where the first layer of paint is the color red and has a less glossy specular highlight and the top layer of paint is actually a sealer that is clear and high gloss (at least when it's been washed recently). So you can now have a low-gloss colored specular under a high-gloss white specular.

Real world surfaces like machined metals reflect light unevenly, yielding what is often called a brushed-metal look. This is called *anisotropic* distortion, and it's the shader's third function. Compared to a smooth surface, these surfaces will have a softer and broader specular highlight.

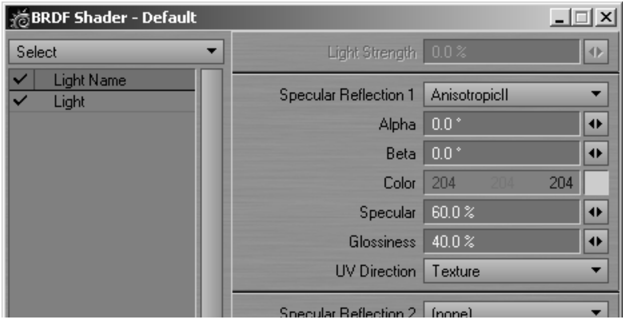


BRDF panel

On the left, you select which lights to affect. On the rest of the panel you set up multiple specular highlights, and make the shape of the highlight different by applying anisotropic distortion. The BRDF specular settings work with the normal Specular surface attribute, so you must have some Specular on the surface to see any change.

For *anisotropic shading*, you must know two directions lying in the surface. The specular reflection will be stronger along one direction. An example is a surface that has long highlights along fine grooves and short ones across them. With the **Anisotropy** and **AnisotropicII** reflection types, you can set **Alpha** and **Beta** settings, which essentially determine two different angles of disturbance. These effectively add tiny grooves to the surface that make the specular highlight imperfect—sort of like a *micro bump*.

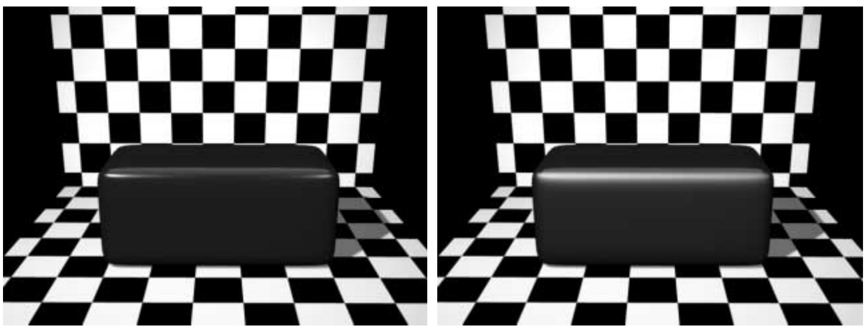
With **AnisotropicII**, you can specify a UV map to define these directions. The **Alpha** parameter is essentially an angle for groove sharpness and blends the plain reflection with the anisotropic reflection—90 degrees is 0% anisotropic. **Beta** is the angle between the the U direction and the maximum reflection, and so it rotates the grooves in relation to the UVs.

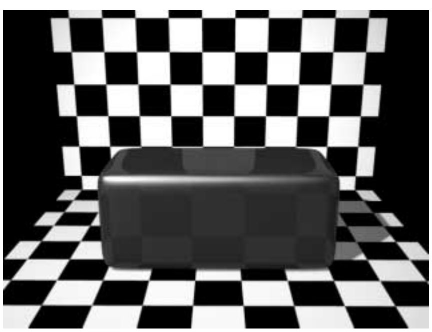


BRDF using AnisotropicII

NOTE
The **UV Direction** map works only with **AnisotropicII**.

There are many models for anisotropy and two are featured in this shader. Basically, these differ in the pitch and bank variance on a surface of the grooves.


Left: Normal high-gloss specular. *Right*: Normal low-gloss specular


Low- and High-gloss Regular specular using BRDF

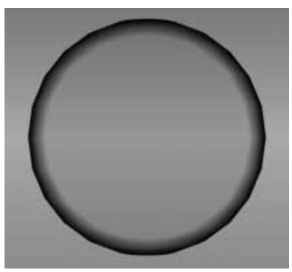
Edge Transparency

The Edge Transparency shader affects the degree of clarity and definition for the polygon edges belonging to a transparent object.

Opaque creates an adjustable edge. **Normal** creates a solid edge. **Transparent** causes the edges of the object to blend into its surroundings.

Left to right: Opaque, Normal and Transparent edges

Edge Threshold is available for **Opaque** or **Transparent** transparency settings only. It determines the amount of blending between the surface color and the transparent edge of the object surface. This transition zone may be wide and softly blended over the face of the surface, or it may be sharp and seen as a thinner line. **Edge Threshold** is normally set at 1.0. Use lower values for a sharper transition. Use higher values for a softer transition.



Edge Threshold = .5

NOTE
When you fine-tune for glass-like objects, we recommend you use **Limited Region** (Camera Properties) to render a small portion of the image at full resolution. Test rendering at lower resolutions may cause jaggy edges that won’t be visible at your higher final resolution.

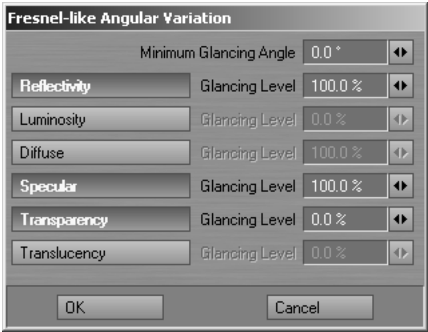
Fast Fresnel

In the real world, the angle between the viewer and a surface affects the amount of light that is reflected and refracted. For example, a lake is often nearly clear close to the viewer and gets gradually more reflective farther way.

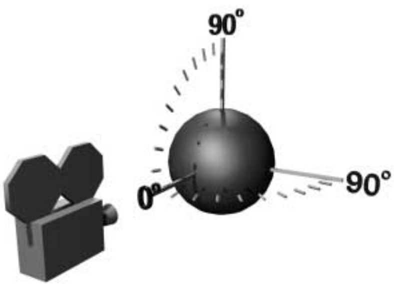
The FastFresnel shader works in combination with the basic surface parameters that you set and then modifies those settings based on viewing angles and the FastFresnel parameters.

FUN FACTS

The fresnel effect was discovered by French physicist Augustin Jean Fresnel (pronounced "*fra nel*")

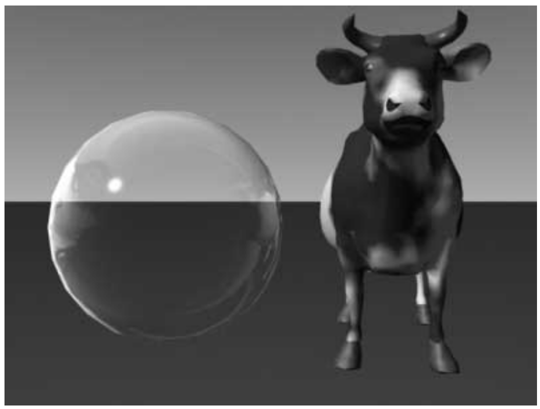


The *glancing angle* is measured from the surface normal, thus 0 degrees is any surface normal that points directly at the camera and 90 degrees refers to any normal that is perpendicular to the camera.



The regular surface value always occurs looking directly at the surface. However, you can adjust the point when the value begins to migrate towards the FastFresnel settings using the **Minimum Glancing Angle**. At a 90-degree angle, the attribute becomes equal to the value set on the FastFresnel panel. Essentially, the regular surface settings change to the FastFresnel settings as the angle of incidence goes from the **Minimum Glancing Angle** to 90 degrees. (Although, the **Minimum Glancing Angle** can be 0 to 89 degrees, you'll usually want this at or near 0 to create a wide range for the effect.)

Semi-transparent reflective ball without FastFresnel

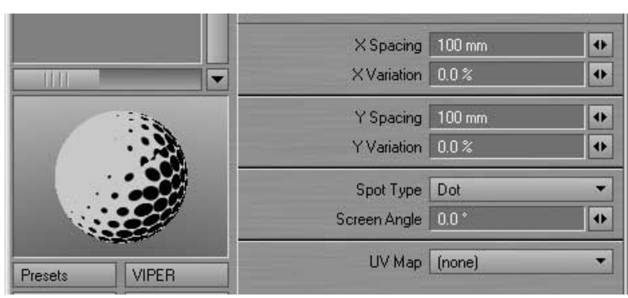


Default FastFresnel settings (regular Reflection=0 and Transparency=100)

Parameters that default to 100, like **Reflectivity**, naturally get stronger as you reach the higher glancing angles. Parameters that default to 0, like **Transparency**, naturally get less intense as you reach the higher glancing angles.

Halftone

In print, halftone screens are made up of dots that control how much ink is deposited at a specific location. Varying their size and proximities creates the illusion of variations of continuous color.

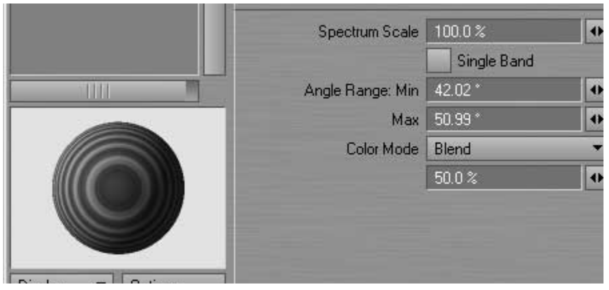


The **Spacing** and **Variation** settings set the spacing of the pattern and a percentage that you can use to vary it (for randomness), respectively. You can choose different pattern shapes with the **Spot Type** pop-up menu. The **Screen Angle** value lets you change the angle of the pattern from the true horizontal.

If you specify a **UV Map**, the shader determines a direction in the surface to align the lines or the crosshatching. HalftoneShader is calculated like a cubic texture, if no **UV Map** is specified.

Interference

The Interference shader adds the sort of distortion seen on an oil slick. This *interference* pattern is caused by the light reflecting between two layers of various thickness. In the case of an oil slick, light reflects between the water and the oil. This often appears rainbow-like, where you can see all spectral colors swirling about. This shader adds all spectral colors in a banding fashion and uses incidence angles to vary which colors are visible.



Spectrum Scale determines how far through the color spectrum the shader will travel across the slope of the surface. This is dependent on the **Min** and **Max Angle Range** settings. For example, the default settings of **Spectrum Scale**=100%, **Min**=42, and **Max**=50 tell the shader to travel through the entire spectrum (100%) as the angle of incidence changes from 42 degrees to 50 degrees, or a delta of 8 degrees. The spectral range colors are red, orange, yellow, green, blue, indigo, and violet. If you change only the **Spectrum Scale** to 50%, the surface travels only through red, orange, yellow, and green, across the same angle.

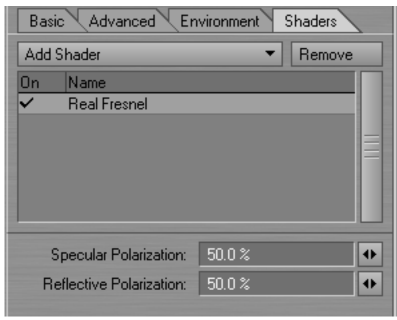
Activating the **Single Band** option restricts the spectral change to a single ring between the **Min** and **Max** angles. In effect, this keeps the texture from repeating across the entire surface.

The **Add Color Mode** adds the color values of the interference pattern to the original surface colors. If a pixel was originally shaded at 100, 100, 100, and the interference color for that pixel was to be 0, 0, 100, the resulting pixel value will be 100, 100, 200. The **Multiply** option multiplies the original pixel values by a number between 0 and 1. If the original pixel value was 255, it is multiplied by one. If it was 0, the new pixel value is multiplied by zero. Intermediate values are altered on a sliding scale.

The **Blend Color Mode** blends the pattern with the original surface attributes using the percentage field. At 50%, the default, the interference pattern is seen on top of the original surface with a strength of 50% of its own color values.

RealFresnel

The RealFresnel shader is similar to FastFresnel, but based on real physics and thus features few user-definable controls. It is essentially set up to create a transparent item by calculating falloff for the transparency value with the *Fresnel* equation.



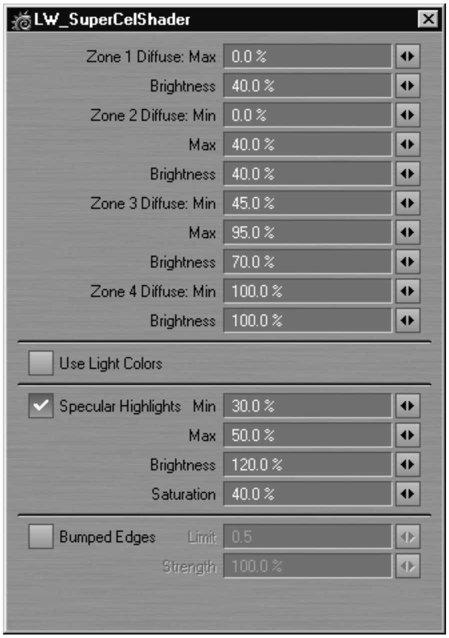
Specular Polarization and **Reflective Polarization** are the values for those surface attributes that will be used when the camera is perpendicular to the surface.

RealFresnel shader

RealFresnel is more accurate than FastFresnel, but will probably require more time and effort to find settings that will work for your scene.

SuperCelShader

SuperCelShader alters the shading algorithm to make large bands of solid color across a surface—like a cartoon—rather than a smooth gradient between light and dark areas. This shader should be used with **Silhouette Edges**, **Unshared Edges**, **Sharp Creases**, and **Surface Borders** (Object Properties panel, Edges tab). (See Chapter 9, "Polygon Edges.")



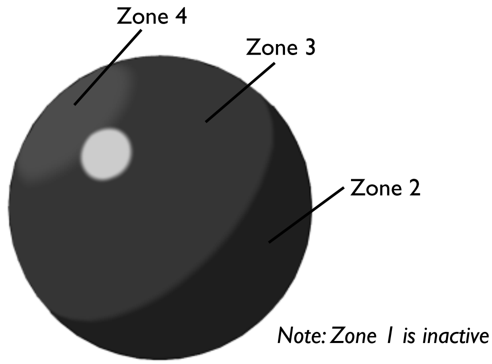
Defining Color Zones

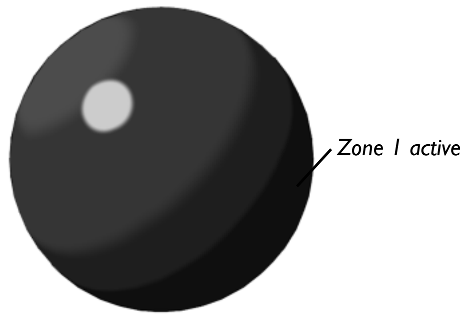
SuperCelShader has four possible color zones (1 through 4). An area's amount of *diffuseness* (i.e., brightness) determines how many zones the surface is broken down to. The zones go from darker to lighter from 1 to 4, respectively. Essentially the Min and Max settings define the color zones.

Min and Max values should range from 0 to 100% and must get progressively higher. For example, the **Zone 3 Diffuse: Min** must be higher than or equal to the **Zone 2 Diffuse: Max**. If the Min value is the same as the lower zone's Max, a hard edge is created between the zones. Separating the values will smooth the color transition.

The default settings for the panel disable Zone 1 by making the **Zone 1 Diffuse: Max** value equal to zero. Note that the Zone 1's diffuse minimum is always zero and Zone 4's diffuse maximum is always 100%—there are no input fields for these.

Notice the 5% difference between the **Zone 2 Diffuse: Max** and the **Zone 3 Diffuse: Min** values, creating a slight smoothing between the color transition.





Brightness

The **Brightness** settings control how bright the corresponding zone will be. These settings are totally independent; however, in most cases, you want them to get progressively higher. Values should range from 0 (black) to 100% (brightest), although higher settings are possible.

Colored Lights

Activating the **Use Light Colors** option will tint the surface with the light color. Normally, the normal Surface Color attribute determines the surface's color.

Specular Highlights

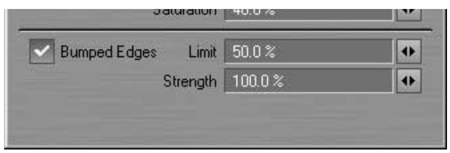
If **Specular Highlights** is not active, the specular highlights of the surface appear normal (i.e., as without SuperCelShader). Activating this option results in *cartoony*-looking specular highlights.

The **Min** and **Max** specular settings define which part of the specular highlight to show. A value of 0% equates to its darkest part and 100% equates to its brightest part—this range can often be hard to see in a normal specular highlight because they are naturally bright. Making the

Min and **Max** values the same creates a hard edge to the highlight, while separating the values creates a smoother transition. The higher the **Min** value, the smaller the specular highlight will be.

Left: Low Min setting equal to Max setting. Right: High Min setting equal to Max setting

Brightness controls how bright you want the overall highlight to be. **Saturation** determines how much of the surface color you want to saturate the highlight.

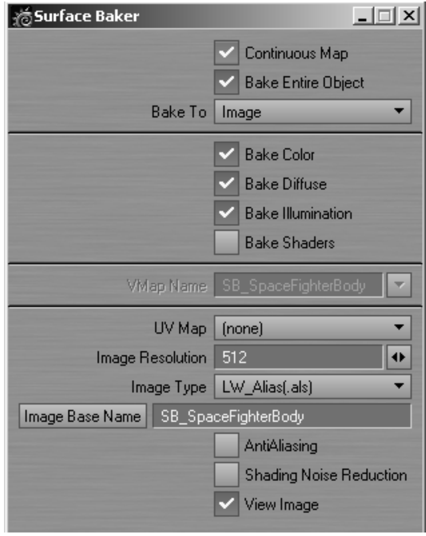


In some cases, you want to *soften* the edges of cel-shaded surfaces to avoid the harshness that can sometimes occur. Activating the **Bumped Edges** option makes the edges *soft and fuzzy*. This has no direct effect on the shading of the zones, other than near the edges where the surface normal is angled toward 90 degrees from the camera—like the edges of the ball in the preview window.

Limit basically defines the minimum angle—of the surface normal relative to the camera—to be affected. 100% is a surface normal pointing at the camera and 0% is perpendicular to the camera. As **Limit** is set towards 100%, more edges are affected. **Strength** determines how much to affect the edges. Values greater than 100% begin to *eat away* the edges.

Surface Baker

The Surface Baker surface shader allows the *baking* of lights, shadows, textures, and so on. You can save this information to an image to UV map back onto the object, or you can apply the information directly to the object's vertices. After you use Baker, you can see complex textures in real-time in your viewports!



If the surface has sharp creases or is not smooth, you may not want to interpolate across polygon boundaries. Uncheck **Continuous Map**: this essentially turns off polygon smoothing and the map will change sharply at each vertex. You might, for example, keep this option off if you were *baking* a room and did not want to smooth between the walls.

Bake Entire Object is a time-saving option that lets you add Baker to a single surface and create its results for all of the object's surfaces.

You can independently choose to bake in color, diffuse shading (e.g., bumps and diffuse textures), illumination (all the lighting from the scene, including shadows, projection images, *even radiosity and caustics*), as well as other shaders in the surface baking computation.

Baking the illumination takes the lighting environment into consideration for the *diffuse reflection*. If **Bake Illumination** is off, it's like the surface is calculated in an environment without lights and 100% white ambient color. If it is the only option on, then you bake only the light intensities that reach the surface and discard the surface settings.

NOTE
When baking surface shaders, Baker can be anywhere in the shader list. It does not need to be at the top or bottom.

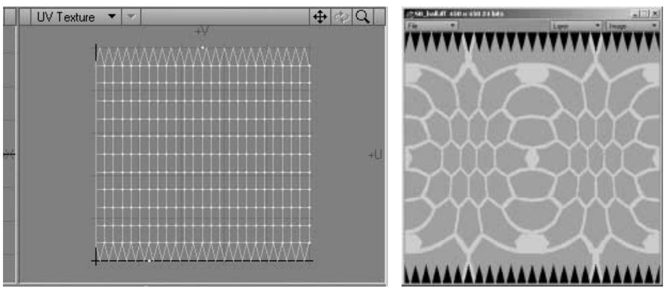
Baking Tips

Due to the complexity of the computations, only polygon faces with three or four vertices can be baked. Other faces are ignored. (Tripling your polygons will ensure you meet this requirement.) Moreover, some portion of the surface you are baking must be visible to the camera at the time of rendering. It is not necessary for the entire surface to be visible, just some portion of it. (If the surface is not visible, the plug-in does not get executed during the render.)

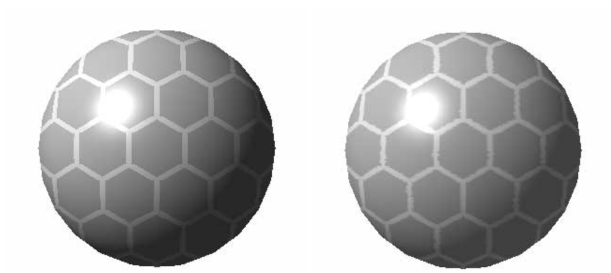
You cannot bake elements that depend on the relative position of the camera and the lights, like incidence angle gradients, transparency, refractions, and reflections. These will not bake correctly. However, you may use the incidence angle from a light as the **Input Parameter** for gradients (but not the incidence angle from the camera).

Image Mode

With the **Image** mode (**Bake To** pop-up menu), each polygon is *painted* onto the final image based on its UV coordinates. Once Baker creates the full image, you simply UV map it onto the surface, as you would normally. With the proper UV map, the surface is seamless.



Left: UV map created for a sphere. Right: Image created with Baker



Left: Normal procedural texture. Right: UV mapped image

The UV-mapped version, above right, is slightly grainier and not as sharp, but you can imagine how much rendering time this could save if the original surface had multiple layers of procedurals.

The key to pulling off this effect is starting with a good UV map. If the map cannot acceptably line up the UVs with the appropriate parts of the image, you will get unacceptable results.

The **Image Resolution** (Image mode) sets the pixel width and height of the image to be created.

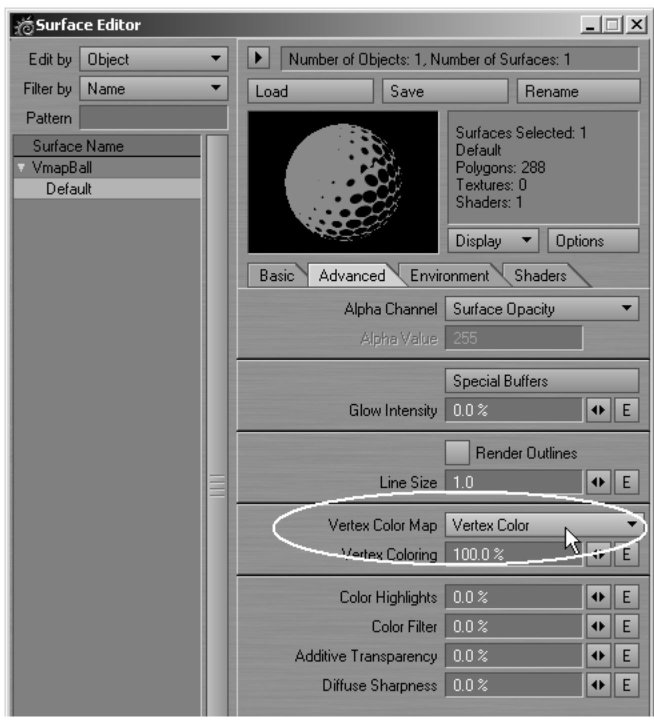
Use the **Image Base Name** button to navigate to a directory to save your image files. Enter a base filename. A frame counter and format extension will be added automatically. Because the frame counter will be incremented, you can save an image sequence of your surface.

If you choose an **Image Type** that supports alpha channel, the wireframe view of the mesh is saved to the alpha channel.

There are options at the bottom of the panel to toggle **Antialiasing** and **Shading Noise Reduction** algorithms on or off. If **View Image** is checked, baked images will be displayed after rendering.

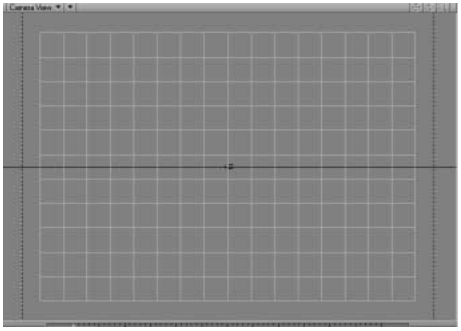
Object Mode

With the **Object** mode (**Bake To** pop-up menu), the surface is sampled at each vertex and assigned to a vertex-shading VMap called a *vertex color map*. Essentially, color information is stored with the points and thus becomes part of the object. To use this map on the surface, go to the Advanced tab and select it from the **Vertex Color Map** pop-up menu. Set the blending percentage in the **Vertex Coloring** field.

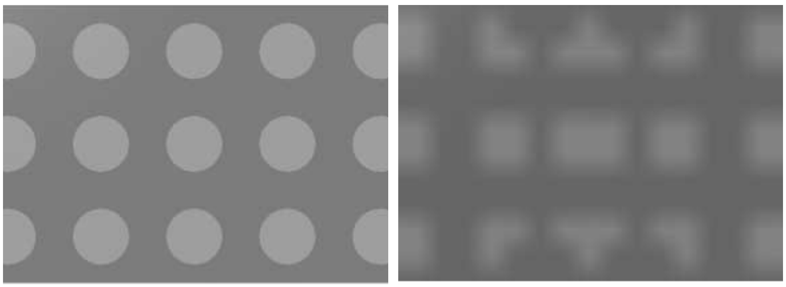


Vertex Color Map settings

The surface area between each vertex is interpolated, which obviously means the result can be only an approximation of the original surface. A higher density of points (i.e., vertices) in the surface will increase accuracy.



Segmented flat box



Left: Normal rendered surface. Right: vertex color map version

NOTE
Since the Object mode samples only at each vertex, it computes much faster than the Image mode.

NOTE
Baking to a vertex color map saves RGBA data, using floating point values for RGB, not limited to 1. The alpha value is 1.0. Baking to an image can also save RGBA data, using floating point values for RGB, not limited to 1. However, the alpha channel is saved only if the file format supports it and the wireframe is saved in the alpha channel.

To use Baker:

1 If you plan to use the Image mode, you must first create a UV map in Modeler.
2 In Layout, add the Baker surface shader to the target surface.
3 Select the surface attributes you want to *bake in*.
4 Select the **Bake To** mode.
5 Set the vertex color map name (**VMap Name**) or **Image File Name**, depending on mode used.
6 Close the panel.
7 Render a frame (F9).

8 Remove the Baker shader from the surface or deactivate it.

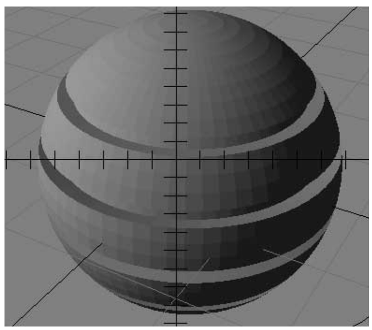
NOTE
Vertex color maps are discussed in Chapter 24.

Instant Radiosity

You can use Baker and radiosity to compute an *accessibility map*. An accessibility map shows how accessible a point is on the surface, which lets you create *dirt*, *weathering*, or radiosity lighting effects. Surfaces are normally darker in grooves and creases, not only because less light reaches these areas, but also because dirt has a tendency to accumulate there.

To simulate this effect:

1 Here is the groovy object in Layout before turning down the lights.



2 Set lights and ambient intensity to 0%. Enable radiosity. Set the backdrop color to white. The object will turn black in the viewport.

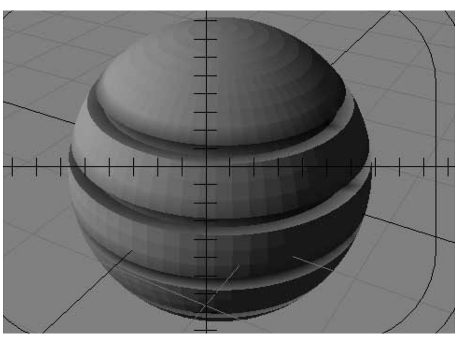
3 Set the surface color to white with 100% Diffuse.

4 Add Baker to the surface and **Bake to Object**.

5 Render a frame. You'll see something like the image that follows. Only the grooves get some shading due to radiosity.

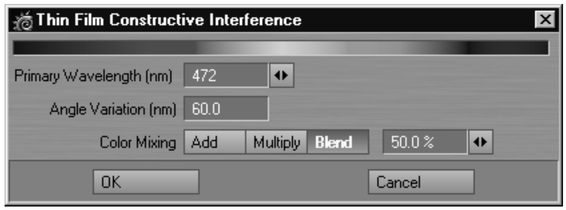
6 Remove Baker and add the **Vertex Color Map** you just created. Set your basic surface parameters as you would normally (e.g., change the color, reduce Diffuse, and so on).

7 Notice how the object now has some nice shading in its grooves, compared to the image in step one. You've basically preprocessed radiosity, which can save a lot of time, particularly on static objects. Note that since you apply coloring only to the points, the mesh needs to be sufficiently detailed and have a good uniform distribution of vertices.



ThinFilm

Similar in effect to Interference, the ThinFilm shader also changes the color spectrum based on the surface's angle to the camera. It can be used for effects like an oil film on water. **Primary Wavelength** is the color in the spectrum that the shader will use as its base color. You may either enter the wavelength value or simply click on a color in the spectrum. **Angle Variation** is the angle at which the colors will start to shift.

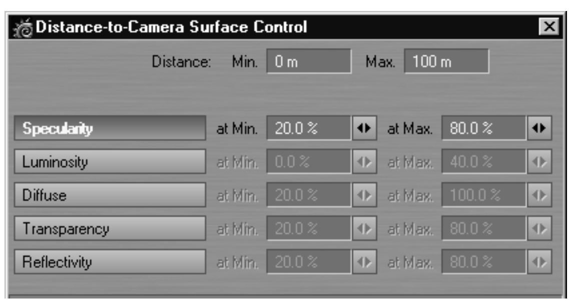


When **Color Mixing** is set to **Add**, it adds the color values of the interference pattern to the original surface colors. If a pixel was originally shaded at 100, 100, 100, and the interference color for that pixel was to be 0, 0, 100, the resulting pixel value will be 100, 100, 200. The **Multiply** option multiplies the original pixel values by a number between 0 and 1. If the original pixel value was 255, it is multiplied by one. If it was 0, the new pixel value is multiplied by zero. Intermediate values are altered on a sliding scale.

When **Color Mixing** is set to **Blend**, it blends the pattern with the original surface attributes using the percentage field. At 50%, the default, the interference pattern is seen on top of the original surface with a strength of 50% of its own color values.

ZShader

ZShader lets you vary the values of certain surface attributes over a specified distance from the camera.



Define the distance from the camera range using the **Min** and **Max** fields. Activate the attribute you want to vary by selecting its button. Enter the values for the minimum and maximum distances in the fields provided. In-between values will be interpolated.

Surfacing Ideas

By no means should you restrict yourself to using, say, a procedural texture for what its name implies. The unlimited combination of settings, layers, and different surface attributes yields an infinite number of visual possibilities.

You are encouraged to load and render the various objects that come with LightWave. Study their surfacing techniques. Change them if you want. However, do not fall into the trap of thinking the surfaces are perfect. Many of them are far from it. Surfacing is an art form and 3D artists each have their own opinions on the best approaches. You need to develop your own.

Surface Management

You can save surface attributes in a special surface file that you can re-use later. (Your Surfaces subdirectory should contain a multitude of pre-defined surface files.) Use the **Load** and **Save** buttons to do this. Use the **Rename** button to rename the surface.

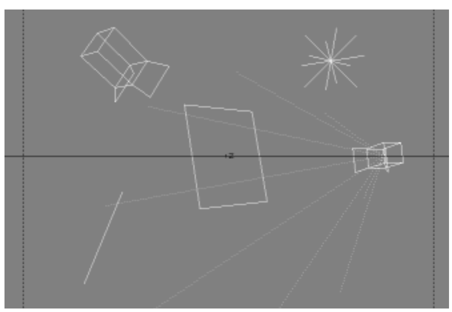
NOTE
When you load a surface file, remember that the surface texture **Scale**, **Position**, and so on are also loaded. Thus, if the object you apply the surface to is a different scale, you may need to tweak texture positioning data.

chapter 32
Shadow and Light

Chapter 32: Shadow and Light

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 it's **Light Intensity** to zero or deactivate the light on the Scene Editor (see Chapter 12).

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. LightWave also features radiosity and caustic light effects, which can add tremendous realism to your scenes.



NOT THE REAL WORLD

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. This is actually a handy feature because unlike on a movie set, you can place lights anywhere, including in front of the camera!

You can also place lights *inside* of objects. For example, you may have a solid ball inside of a box and you may want the inside walls of the box to be lit. Placing a point light inside of 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 behavior 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.

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 (**Display** > **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** > **Add** > **Lights** and select the desired light type from the submenu.

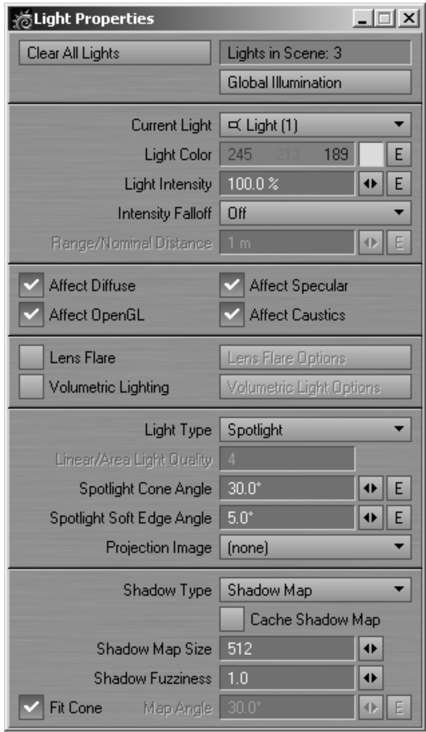
NOTE
A dialog prompts you for a name when you add lights. Click **OK** to accept the default.

To remove a light:

Select the light and choose **Items** > **Clear** > **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.



EXCLUSIONS

You can exclude lights, radiosity and caustics from affecting certain objects. (See Chapter 9 for more information.)

SAVING LIGHTS

Make sure you turn the lights off after you exit a room... No wait. That's saving electricity. 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 like the light from the sun. Surfaces receive an infinite amount of parallel light rays traveling in the direction that the Distant light points. Distant lights are handy when you want equal illumination on objects in a scene.

Distance light

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 below the object or one meter above. 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.

Point light

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 level of reduction.

The **Range/Nominal Distance** values set the distance from the light where the light's intensity is zero.

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.

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

The Spotlight

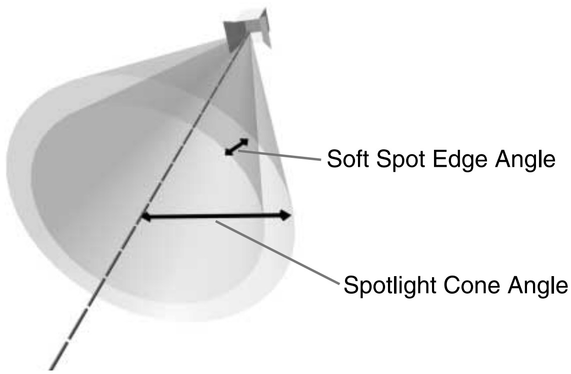
Spotlights behave like their real world counterparts, sending light out in a specified direction and size. Spotlights are probably the most useful type of light available to the LightWave animator. Flashlights, car headlights, and searchlights are all good examples of a Spotlight.

Spotlight

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.



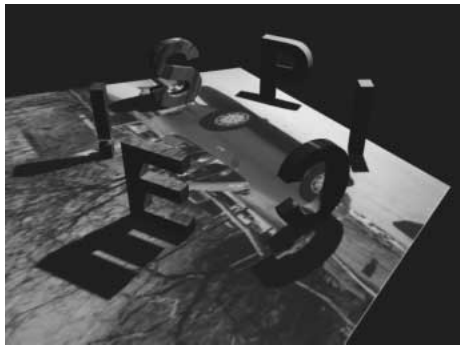
A **Spot Soft Edge Angle** of 0 degrees creates a Spotlight with a hard edge, while a setting less than or equal to the **Spotlight Cone Angle** creates a soft-edged light.

Hard-edge spotlight

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.



Projected image

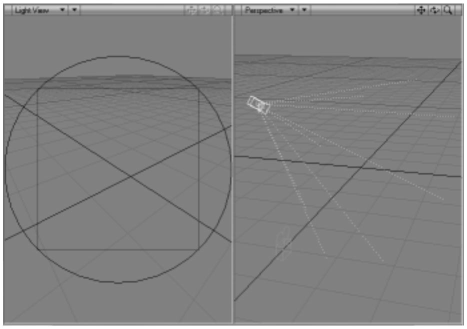
NOTE
The projected image will not conform to the circular shape of the cone angle. To do this, you could place a *cookie-cutter* object (e.g., a flat box with a round hole) in front of the spotlight or use a paint package to edit the image and fill the unwanted area with black. You may also want to use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

Spotlight Viewport Display

You can manipulate how Spotlights display in Layout. First, when you selected the **Spot Cone Angle**, it is represented by lines emanating from the light. If you interactively adjust the cone angle (**Lights** > **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**.



Light view for spotlight

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 lamps.

Linear light

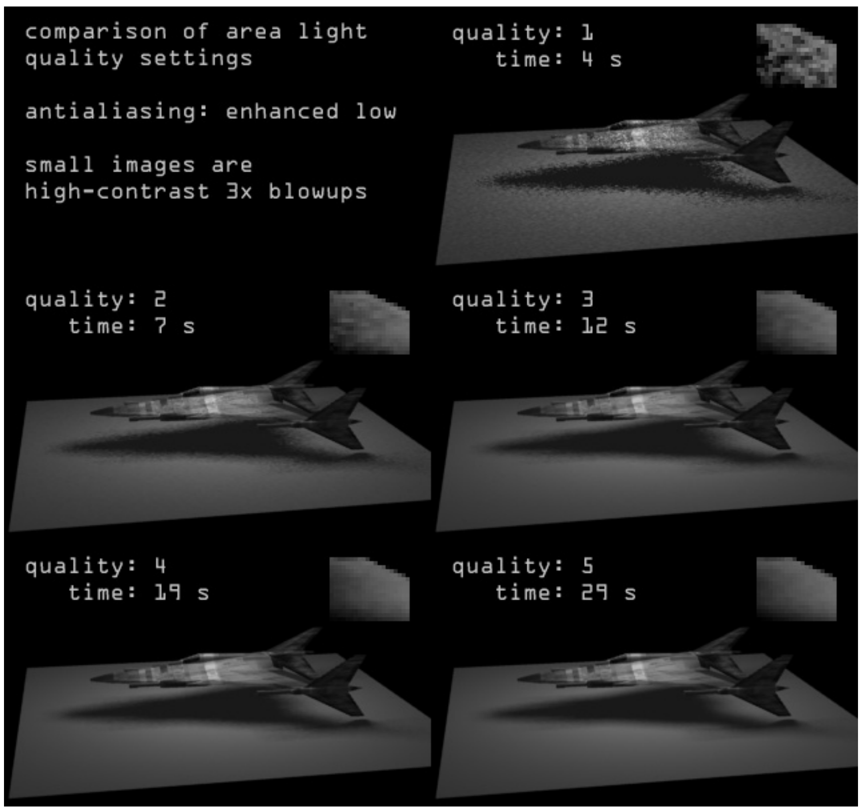
Now, if you could turn a four-point double-sided polygon into a light, you'd have an Area light. Light is sent out equally in all directions, except along the edges. This type of light might be used for flat light panels.

Area light

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 fuzzy geometrically-accurate edges.

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 antialiasing levels.

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



A comparison of quality settings and examples of their effect on image quality and rendering time

The Cow Light

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 (Global Illumination panel).

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

The intensity of lights 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 where the values of light sources exceed a certain brightness.

NOTE
A surface hit with multiple lights that exceed a total of 100% is not necessarily a bad thing, 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 (**Lights** > **Lgt Intensity**) to adjust the intensity of selected light(s) by dragging your mouse.

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.

OTHER 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 spot on a surface, but you don't want it to affect the rest of the lighting.

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. Many times a scene 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 specularity for these lights you can overcome the problem. For similar reasons, there is a **Affect Caustics** option.

The **Affect OpenGL** option is for display purposes only. It will not affect 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 use a limited number of lights to affect your OpenGL display (see the Display Options tab of the Preferences panel).

ME AND MY SHADOW

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.

NOTE
If you couldn't tell, the preceding light type illustrations all had shadows active.

With shadows inactive

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.

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. It's funny to think about it, but a house is dark inside during the day only because of the shadows from the wall.

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 soft edges.

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.

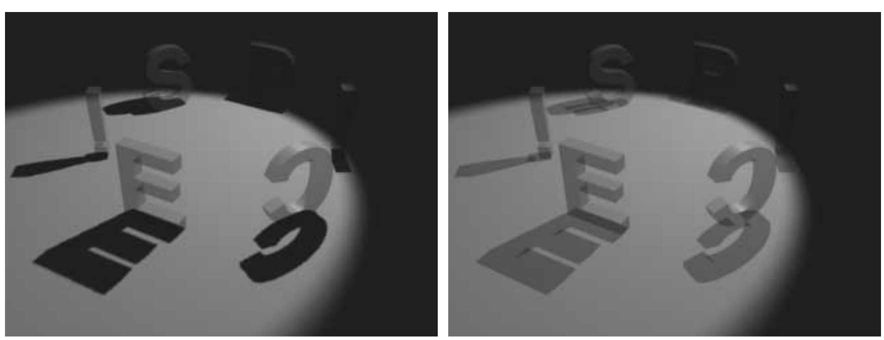
Shadow Mapping

An alternative to ray-traced shadows is the **Shadow Map** option, available only for Spotlights (for a limited time, if you call now...) 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.



Left: Shadow Map. Right: Ray Trace shadows

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 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 pixelated 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 x 512 x 4 = 1,048,576 bytes = 1 megabyte). 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.

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 in Chapter 9, "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.

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 panel (**Lights** > Global: **Global Illum**) 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.

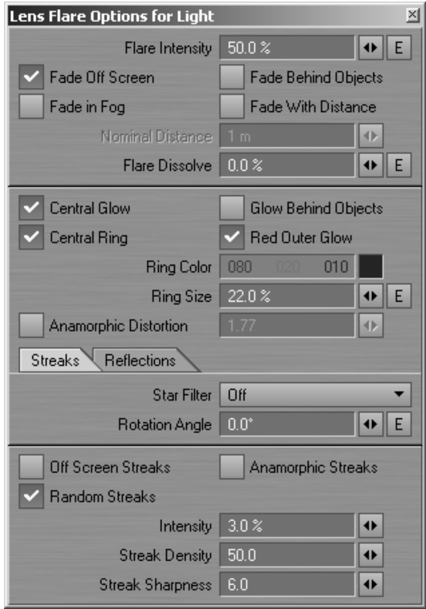
Lens Flare Options

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 phenomenon, like fire, glowing, and explosions.

To set up a lens flare:

1. Open the Global Illumination panel (**Lights** > Global: **Global Illum**).
2. Make sure there is some percentage of **Global Lens Flare Intensity**. 100% should work in most instances.
3. Activate **Enable Lens Flares**.
4. Select the light.
5. Click **Lights** > **Lens Flare** to turn the option on for the selected light.
6. Click **Lights** > **Flare Options** and set the options as desired.

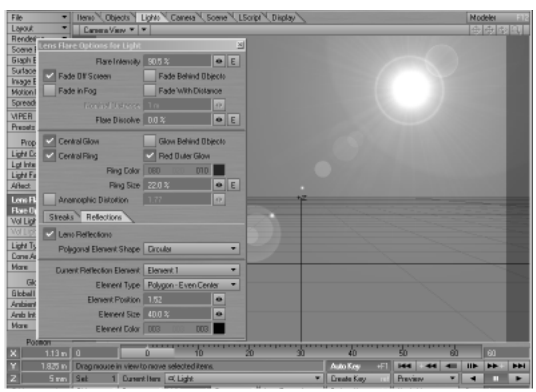
NOTE
You can also access the **Lens Flare** state and Lens Flare Options panel from the Lights Properties panel.



Lens Flare Options 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 (**Display** > **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.


Lens Flare visible in viewport

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 (**Scene** > Effects: **Volumetrics**). 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. *Right:* With **Fade Behind Objects**

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%.

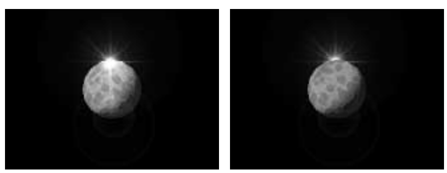
Enter a **Flare Dissolve** value to adjust the transparency of the lens flare effect. This option is handy when you wish to see large streaks of light coming from the lens flare but do not want a bright hot spot at the center of the flare. The higher the **Flare Intensity**, the larger the streaks (if 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. *Right:* With **Glow Behind Objects**

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.

4 + 4 Point

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.

Random streaks

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.

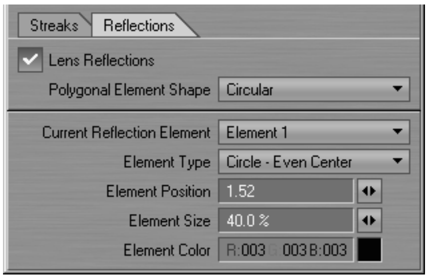
NOTE
A very low **Streak Sharpness** value (like .0001) will blur the streaks into each other to create a glow-like effect.

Lens Reflections

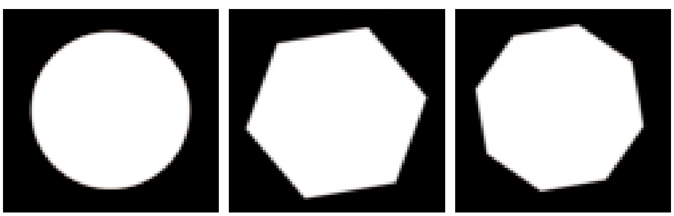
The **Lens Reflections** option adds reflections of light in the LightWave camera, as if it were shooting the scene through a standard camera lens assembly. This effect emulates the multiple lens elements that make up a typical lens. Such flares commonly occur whenever you aim a camera at an intense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

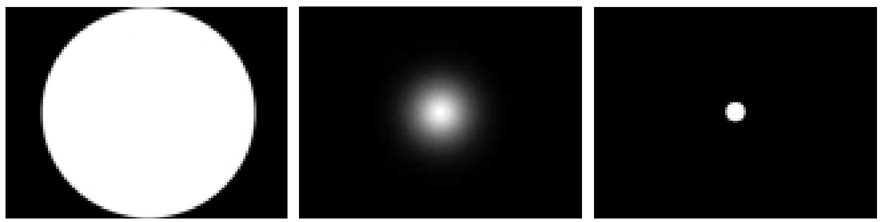
You can modify the default **Element Type**, **Element Position**, **Element Size**, and **Element Color** for each of the sixteen reflection elements. To edit an element, first select it using the **Current Reflection Element** pop-up menu. The elements are aligned in a straight line going through the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

Left: Position = -1. *Middle*: Position = 0. *Right*: Position = 1

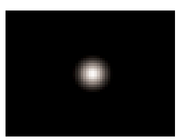
An **Element Size** of 100% vertically fits the element to the camera resolution. It can be clipped or there can be space to the left and right, depending on the relative camera resolution Height setting.



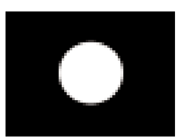
Left: Size = 100% (Even Center **Type**). *Middle*: Size = 100% (Bright Center **Type**). *Right*: Size = 10% (Even Center **Type**)

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

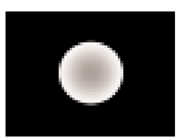
The **Element Type** setting determines the shape and density characteristics of the reflection.



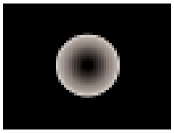
Circle - Bright Center

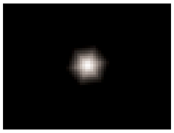


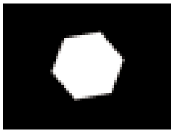
Circle - Even Center

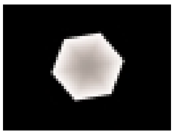


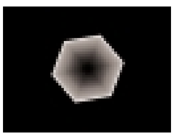
Circle - Dim Center

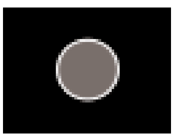

Circular Ring


Polygon - Bright Center


Polygon - Even Center


Polygon - Dim Center


Polygonal Ring

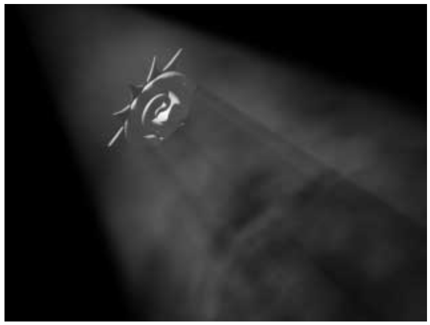

Rainbow Ring

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

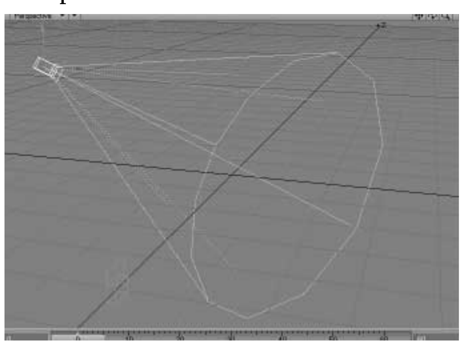
Lights > **Vol Lighting** turns a selected light into a *volumetric light*. This gives lights—or more accurately their beams—physical volume. These effects are common in everyday life and can play a key role in creating dramatic and realistic environments.



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

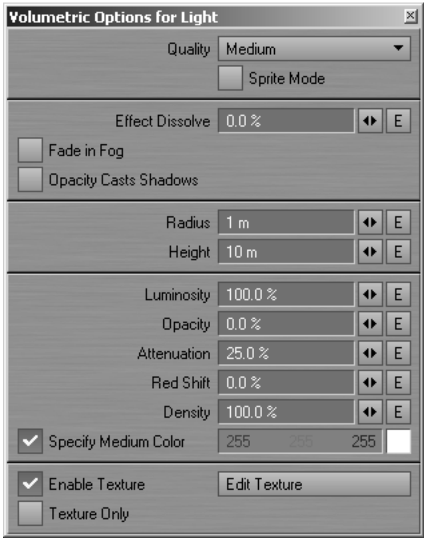
NOTE
Also see the **Volumetric Antialiasing** option on the Volumetrics tab of the Effects panel, discussed in Chapter 15.

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



NOTE
Linear and Area lights cannot be volumetric.

To display a selected light's volumetric options, choose **Lights** > **Vol Light Opts**.



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.

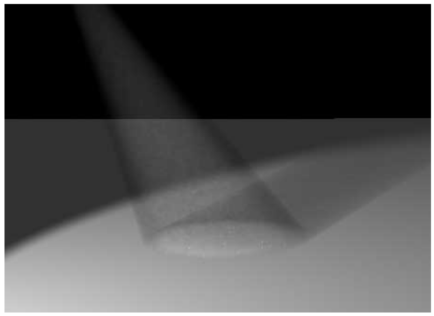
Quality Use the lowest **Quality** setting that achieves acceptable results. Obviously, higher settings will take longer to render.

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.)



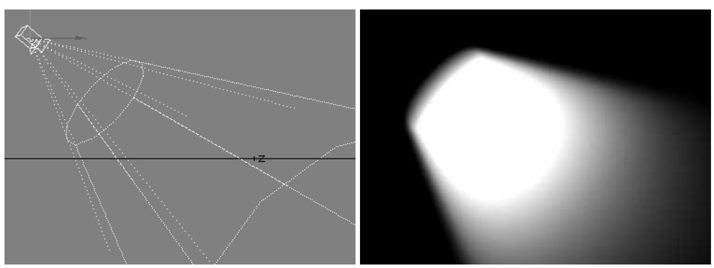
Volumetric light casting ray-traced shadow

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.



Left: Cone Base adjusted higher than 0. *Right*: Rendered result.

Height

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

Luminosity

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

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 that can cause dense areas to get very bright. You may need to increase **Luminosity** to compensate for increased **Opacity** settings.

Attenuation — **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 effect attenuation always exists.

Red Shift — The **Red Shift** setting works in conjunction with **Attenuation**, controlling the interior behavior 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.

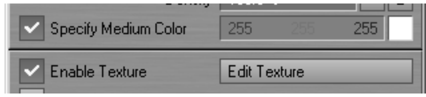
Density — **Density** controls the global density of the medium. This essentially works like a multiplier for **Luminosity** and **Opacity**.

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

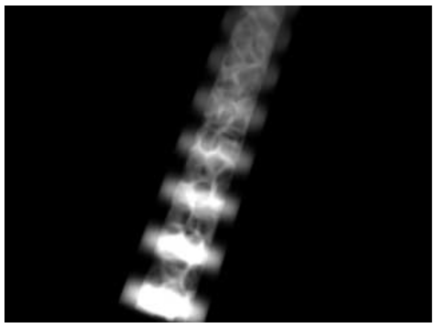
To add a texture to the volumetric effect using the Texture Editor, click **Edit Texture**. The **Enable Texture** option must be active to apply the texture settings. Deactivating it will not affect any existing settings.

The **Texture Only** option means that the texture is not blending with the base color, but replacing it totally. (See Chapter 31 for more information on how to use the Texture Editor.)

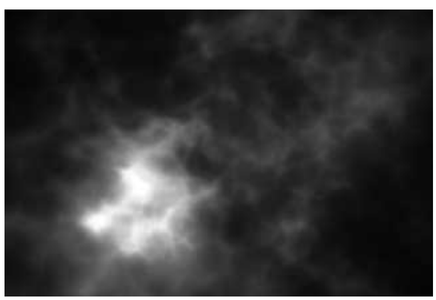


NOTE
The axis of the light beam for unidirectional lights is the Z axis.

Here is an example of just two volumetric distant lights. The inner light uses a Vein procedural texture and the outer light uses Ripples. The secret to this effect is to use one wave source and place the Ripples Z position far away (i.e., negative Z) so that the ripple is relatively flat.



Learn to think *outside the box* with volumetric lights. Below, a single volumetric point light fills the screen with a cool cloud effect.



NOTE
The point light, above, was placed at the Origin with no rotation. The Crumple texture was animated along the Z axis. A gradient using the **Previous Layer** (the texture) as the **Input Parameter** was used to add color highlights and *drop out* the lower parts of the texture using low **Alpha** values. The Volumetric light **Luminosity** was cranked up to around 400%. **Texture Only** (Volumetric Options panel) was also activated.

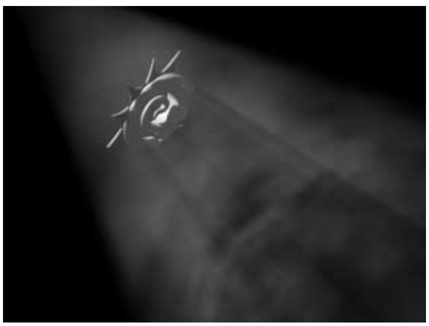
The VIPER Window and Presets

To use the Preset Shelf or VIPER window, activate those options on the main Layout interface.

NOTE
You can preview animated textures with VIPER. See Chapter 3 for more information.

Volumetric Shadows within Lights

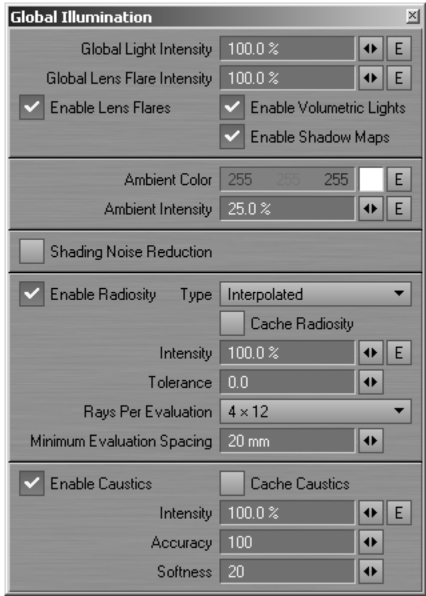
To achieve volumetric shadows within lights, make sure you enable shadow maps (**Lights** > Global: **Enable** > **Enabled Shadow Maps On/Off**), if you are using shadow map shadows. If you are using ray-traced shadows, activate the **Ray Trace Shadows** option on the Render Options panel (**Rendering** > **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 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.

GLOBAL ILLUMINATION PANEL

You access the Global Illumination panel by choosing **Lights** > Global: **Global Illum**. Here you can adjust some of the global lighting controls, including radiosity and caustics.



The **Global Light Intensity** value is an overriding *volume* control for all lights. 100% is the normal setting. However, you can ramp all of your lights up or down, or even create an envelope. The **Global Lens Flare Intensity** is a similar control for lens flares.

You must activate **Enable Lens Flares**, **Enable Volumetric Lights**, and **Enable Shadow Maps** when you are using any of these features. (You may find it quick to toggle these settings from the **Lights** > Global: **Enable** menu.) These *switches* are quick ways to turn these options off globally for test purposes, without losing any of your settings. However, note that they do not turn off the light sources themselves.

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 Global Illumination panel (**Lights** > Global: **Global Illum**). 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 of surfaces. Although LightWave has a radiosity option (discussed later), these *faking* techniques offer greater control and faster rendering times.

HINT
Most professionals considered the default **Ambient Intensity** of 25 percent too high. Settings below 10 and sometimes as low as 0 are common. However, the default setting lets beginners see something in their rendered images, even if they aim lights incorrectly.

Shading Noise Reduction

The **Shading Noise Reduction** option is designed to reduce the graininess in the shading of diffuse surfaces from linear/area lights or using radiosity with low **Tolerance** values. This option will add some time to rendering, but will result in smoother shading. For radiosity, the alternative of using a higher **Rays per Evaluation** setting would add much more to rendering time.

Radiosity

Without using LightWave's radiosity option, all surfaces are lit *directly*, with lights or ambient light. *Radiosity*, the scattering (reflection) of light off of diffuse surfaces, causes surfaces to become (indirect) light sources—like in the real world—generally resulting in much more realistic images.

Image by DigiMania

Copyright ©1999 Stuart Aitken

Image by Bob Quinn

In LightWave, the scattered light includes surfaces that generate their own light through use of the Luminosity surface attribute. The images that result from a radiosity renderer are characterized by soft gradual shadows. Radiosity is typically used to render images of the interior of rooms, due to the high amount of bounced light, and can achieve extremely photo-realistic results for scenes that comprise diffuse reflecting surfaces.

There are no lights in this scene. Just luminous panels. Image by Mike Ash

LightWave can calculate secondary rays bouncing from surfaces or coming from the atmosphere. This adds a very subtle, but photo-realistic effect. When combined with the *high dynamic range* calculations, the renderings become astoundingly realistic.

Looking at the images below, we can see a room that is illuminated from a distant light source (the sun). This image shows us a standard ray-traced image with shadows. Notice how many details of the scene disappear as they are obscured by the shadows in the room. This looks unnatural to the human eye, as a room in midday would never be this dark.

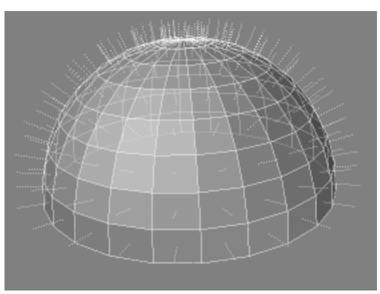
The next image shows the scene with radiosity turned on. Immediately we can see an added level of realism as LightWave calculates the secondary rays and the room is illuminated from *bounced light*. In this example, however, the light source is limited to 100%, which is a natural choice because it is actually the limit in many applications. After all, what good is a bright light when the pixels can't really show beyond 100%? (Brilliant question!)

LightWave has some special uses for pixels that are exceptionally bright. Take a look at the next image. This one uses a light set to 200%. Notice now that the floor is illuminated a little more brightly. Also notice that the entire room gets a little brighter. Though the displayed pixels on the floor can never go beyond 100% luminous, the bounced light from that pixel will carry the extra data, the *high dynamic range* data. This allows the bounced light to illuminate the room more strongly.

The final image uses a light set to 1000%. This time you can see the floor light is still the same brightness but the secondary rays are definitely much stronger as they dramatically light up the room—just as it would look at mid day.

How Radiosity Is Calculated

LightWave approximates radiosity using “projection hemispheres.” These basically “sit” on surfaces, each projecting out multiple radiosity rays at various angles using the theoretical normals of each polygon in the hemisphere. This is called an “irradiance evaluation.”



An illustration of a projection hemisphere

If a radiosity ray strikes a surface that scatters light, some amount of that light illuminates the surface where the ray originated—colors are determined in the usual (non-radiosity) way, which can include the effects of luminosity, mirror reflections, caustics, and so on. The light-scattering surfaces are essentially extra little light sources, used instead of ambient lighting, that affect the diffuse shading of the current surface. Evaluated and non-evaluated areas are blended to compute the final effect.

NOTE

If you activate the **Unseen by Rays** option on the Rendering tab of the Object Properties panel, that object's luminous surface is not considered a source of light for radiosity purposes.

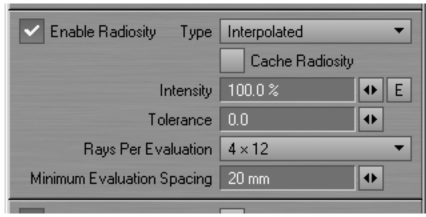
Performing a full irradiance evaluation (tracing hundreds of rays) every time a point must be shaded is too time-consuming. Fortunately, unlike direct lighting, with its concentrated sources that can cause sudden changes across a surface (e.g., shadow boundaries), the indirect lighting that radiosity is meant to handle tends to change gradually over a surface. LightWave can use the results of each previous evaluation and smoothly interpolate between them. Only when no previous evaluations are close enough to the point being shaded, is it necessary to fire a new set of rays.

What is *close enough*? Each time LightWave performs a full evaluation, it estimates the size of the area that can produce valid results. This depends on factors like how far the rays traveled. If all rays went a great distance before hitting anything, then the indirect light must be fairly constant and the calculated irradiance should be good for a large area. But if several rays hit nearby objects, then the indirect light might be varying more rapidly across the surface and the irradiance evaluations should be more closely spaced.

The **Tolerance** setting is a scale factor on these valid area estimates. With a large **Tolerance,** the renderer reuses previous irradiance values more often rather than computing new ones, and rendering time is reduced at the expense of accuracy (i.e., in some areas, small local changes in indirect lighting might be missed). With lower **Tolerance** values, full hemispherical evaluations are performed more frequently—a zero value won't interpolate at all (always do a complete new evaluation). As such, **Tolerance** acts as a limit on the amount of error allowed in the radiosity calculations, although there will always be some errors due to using a finite number of rays.

Evaluation points tend to crowd closer together in corners and other confined areas, so to prevent the renderer from spending too much time in those areas, a minimum spacing distance (**Minimum Evaluation Spacing**) is enforced (unless **Tolerance** is zero). The maximum limit on the valid ranges of the evaluation points is 100 times the minimum spacing.

The Radiosity Settings



Activate **Enable Radiosity** for LightWave to render this phenomenon.

In order of increasing complexity, the radiosity **Type** choices are: **Backdrop Only**, **Monte Carlo**, and **Interpolated**. **Monte Carlo** uses a zero Tolerance value. **Interpolated** allows you to set the Tolerance value. **Backdrop Only** evaluates only the rays that hit the backdrop. It is faster than **Monte Carlo** for environmental illumination, especially in scenes with “expensive” textures or lighting. However, it does not account for diffuse inter-reflection, luminous surfaces, and so on.

Cache Radiosity saves radiosity data for subsequent render passes and frames, which can significantly reduce rendering time. The results can be inaccurate if objects or lights are animated, but this option works particularly well with scenes like walk-throughs in which only the camera moves.

You can increase or decrease the overall amount of radiosity by adjusting the **Intensity** setting from its default value of 100%.

Tolerance can be set from 0 to 1, but the default is .3. When **Tolerance** is greater than zero, LightWave tries to save rendering time by interpolating between stored values and doing a full-radiosity evaluation only when needed. This works if the indirect lighting on a surface varies smoothly and gradually, which is normally the case.

The *n* by *n* **Rays Per Evaluation** selections (e.g., 8 x 24) represent the number of sides and segments of the projection hemisphere (yeah, like when you model a ball), which determine the number of radiosity rays sent out for evaluation. As you might expect, the higher the density, the more accurate, but the longer rendering will take.

NOTE
See also the **Shading Noise Reduction** option on the Global Illumination panel, discussed earlier.

LightWave attempts to optimize the number of hemispheres in a scene. The **Minimum Evaluation Spacing** sets how close at a minimum the hemispheres can be. The maximum is 100 times this number. The lower the setting, the greater the radiosity accuracy, but the longer the render time. The default setting is 20mm.

Lighting Considerations

Radiosity must consider *global illumination*, which means accounting for all lighting, whether direct or indirect. This includes indirect light provided by the backdrop, sometimes referred to as “skylighting.”

When the radiosity rays are fired, the color and brightness of the closest element hit by each ray (other than direct light sources, which are accounted for separately) are added into the global illumination for that point. It doesn't matter if a ray hits another diffuse shaded polygon, a luminous polygon, or the backdrop—whatever is hit will be taken into account.

Of course, if the backdrop is blocked by objects, it won't affect shading. For example, in a scene with an *infinite* ground plane object, the lower half of the backdrop gradient won't matter since no rays will ever reach it. Similarly, when shading the floor of an enclosed room, only the areas of the backdrop seen through open doors, windows or skylights will contribute.

Moreover, with gradient backdrops, orientation matters. If the sky is dark blue at the zenith, but bright at the horizon, then the sides of an outdoor object may get more skylighting than the top. This is because radiosity rays are fired more densely near the direction of the surface normal than around the base of the sampling hemisphere for each shaded point. Thus, light coming in perpendicularly is more important than light coming in at a glancing angle.

VolumetricRadiosity command

The VolumetricRadiosity command can be used to control whether volumetrics are taken into account by radiosity rays. This is enabled by default, but can be disabled to avoid potentially long rendering times. You will need to add this command to a menu or shortcut to access.

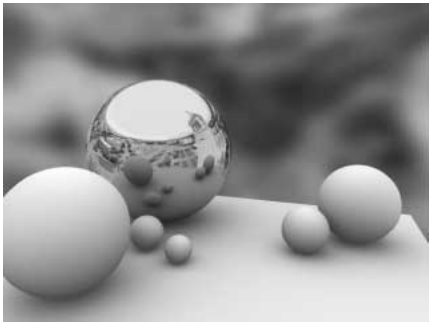
Radiosity and High Dynamic Range Images

High dynamic range images (HDRI) contain color and brightness information beyond what is possible in standard 24-bit image formats. One of the most obvious uses for them is with radiosity.

NOTE
See the *Appendix* for more information on HDRI and the file formats that support it.

A neat trick is to enclose your scene in luminous polygons mapped with HDRI. In this manner, you can actually light a scene without any conventional lights! You can simulate environments by using photographs with HDRI data and illuminate 3D worlds.


Left: No lights. Lit by HDRI and radiosity. *Right*: Lit by single Area light and radiosity. Both images by Terrence Walker.



Lit by HDRI. Image by Jason Bickerstaff.

NOTE
See the *Appendix* for more information on HDRI.

Ambient Light

Ambient light, controlled by **Ambient Intensity** on the Global Illumination panel (**Lights** > Global: **Global Illum**), is not directly added to surface points that are shaded with radiosity—where radiosity rays originate. However, it is added to points that are shaded without radiosity—where radiosity rays hit a surface. If not for this, a new evaluation point would be spawned at the *hit points* and render time would explode.

Ambient light will still brighten every surface, but only indirectly, after bouncing off other surfaces. Thus it can simulate light that would have come from further radiosity bounces.

As a result, the same exact polygon can be lit in two different ways during the rendering of the same image! This is because shading is a temporary thing that LightWave computes on the fly. Just because a polygon is shaded with ambient light when hit by a radiosity ray doesn't preclude it from being shaded without ambient light in the part of the frame where it's seen directly by the camera—polygons are forgetful (forgiving?) and don't *remember* being hit by those rays.

For example, let's say we have a white floor polygon and a red ceiling polygon, 25% ambient light and no direct lights or other objects. If you render this without radiosity, ambient light will show the floor and ceiling as dark gray and dark red polygons. However, with radiosity active, the floor will appear red too.

The floor is red because it no longer gets direct ambient light. Instead, it is lit by radiosity rays that reach out and find the red ceiling—which is shaded with ambient light when hit by these rays—and that red indirect light illuminates the white surface, resulting in red. When the ceiling is rendered, it too is now lit only by radiosity rays that are hitting the white floor (which now gets ambient light as far as those rays are concerned) and that white indirect light illuminates its red surface, so it ends up red too.

Using Radiosity

"Blotchiness" in areas lit by radiosity occurs when different evaluations come up with different results. If the lighting environment is complex and there are not enough rays to properly sample it, then two or more nearby evaluations can be different enough to cause blotchiness. This can occur, for example, when **Tolerance** is set too high.

Consider a scene lit only by a single luminous polygon. The smaller that polygon is (as seen from the surface being shaded), the larger the number of rays necessary to ensure that at least one of them hits it. If only one of two adjacent radiosity evaluations contains a ray that hits that polygon, then the other evaluation will result in a dark patch on the surface. You can minimize this effect by increasing the *evenness* of the environment—that is, making the luminous polygon bigger or dimmer or using more rays to increase the density of the sampling pattern.

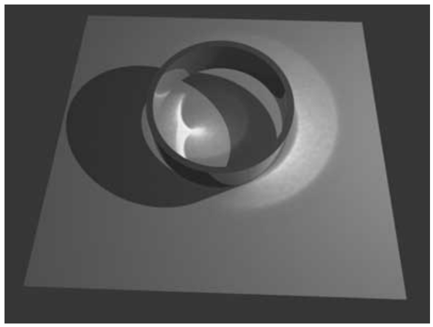
The same principle applies to indirect lighting. A brightly lit area is very much like a luminous polygon as far as radiosity is concerned. For example, when shading a ceiling, more rays are required if the adjacent wall has a few tight spotlights shining on it than if the wall is lit by a broad floodlight. Specular highlights or caustics can also act as tiny indirect light sources that can require a higher sampling density.

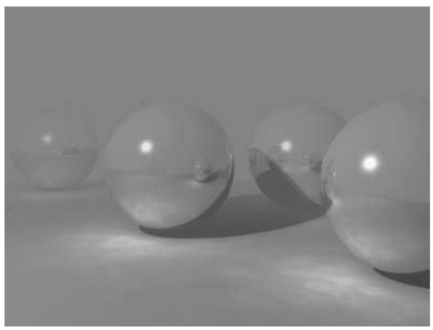
Without using radiosity, lighting is relatively simple. Basically, it amounts to a small range of light intensities and a linear mapping of the resulting surface brightnesses into image pixel values. Radiosity, however, is essentially a *physically-based* simulation, with all of the complications that go with it. In the real world, light intensities vary over a huge range, and so it is with radiosity.

Radiosity by its very nature tends to be dim. Ambient light will still be a factor and can help brighten the scene. You may want to process rendered images with a gamma correction filter for radiosity renderings viewed on a computer monitor. This boosts pixel values in shadowy areas, and makes radiosity effects more noticeable.

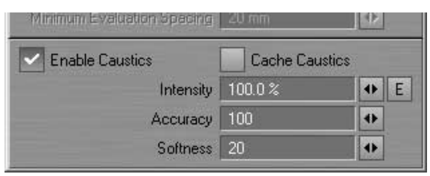
Caustics

Generally, a caustic occurs in the real world when light reflects off a curved surface or refracts through a transparent surface so that it is focused on a small area. Light through a wine glass is a good example. With a more complex surface, the caustic can create a random pattern like those seen on the floor and sides of a swimming pool.





Activate **Enable Caustics** to render this effect in your scene. **Cache Caustics** saves caustics data for subsequent render passes and frames, which can significantly reduce rendering time. The results can be inaccurate if objects or lights are animated, but this option works particularly well with scenes like walk-throughs in which only the camera moves.



Intensity is a scaling factor on the brightness of the caustics. If a light shines on a disco ball, casting dots of light around a room, and you want to halve the intensity of the light source without affecting the brightness of the dots, you can just double the caustic **Intensity** to compensate. This parameter does not affect rendering time nor interact with the other two caustics settings.

Accuracy (1-10000) determines how many caustic rays are *fired* to compute the caustics. The time needed for the initial rendering pass is directly proportional to the **Accuracy** setting, so you can reduce the setting to speed up rendering. However, if you need to accurately render sharp-edged or intricate caustic patterns, you may need to increase the value. You may also need to increase it if the reflecting or refracting objects in the scene are small compared to their distances from the light source—they are a harder target for the rays to hit.

Softness (1 to 100) determines how many nearby caustic rays to take into account when rendering. It affects the speed of the *polygon* rendering pass (as opposed to the caustics preprocessing). The more rays are averaged together, the more blurry the effect. If the caustics are too *noisy*, this setting can be increased. Reducing it will produce sharper caustics, which may require a higher **Accuracy** setting.

NOTE
The default values for **Accuracy** and **Softness** should be fine in most cases.

If you do not want specific lights to contribute to the caustics effect, deactivate the **Affect Caustics** option on the Light Properties panel.

Radiosity and Caustic Caching Considerations

You should not use the cache options for radiosity and caustics when using distributed rendering on multiple machines. Each machine may have a different cache, which can cause flickering.

Unfortunately, not using caching will probably result in flickering too—even when you render on just one machine. Generally, the most stable method is to render on a single machine with caching active. This assumes, however, that nothing changes during the scene that would affect the lighting (otherwise caching should not be used). Flickering may be less severe if you increase **Rays Per Evaluation** for radiosity or **Accuracy** for caustics.

Appendix

Appendix

MACINTOSH INFORMATION

Right Mouse Button Action

To apply a RMB action (i.e., drag or click), hold the COMMAND (APPLE) key while you press the mouse button.

Memory Allocation

To change the amount of memory allocated to LightWave 3D, select the icon and choose **File** > **Get Info**.

Command Line Files

On the Macintosh platform, you can use a special text file to specify executable options. (On the Windows platform, these are entered as command line parameters in a DOS window or shortcut icon). The file must be named the same as the application followed by a space and then the word CMDLINE.

For example, if the file LIGHTWAVE CMDLINE contained only the following line, the Hub would be disabled and the config path would be redirected to SOMEPLACEELSE.

-0 -CHD:SOMEPLACEELSE

The same holds for Modeler, the Hub and lwsn. An example of the contents of a LWSN CMDLINE file might be:

-2 HD:apps:purple:programs:job1 HD:apps:purple:programs:ack1

YOUR DISPLAY

Depending on your computer platform and display cards, you can run LightWave in a number of different screen resolutions and color depths. However, the larger your resolution and the more colors you are using, the slower your LightWave interface will be. LightWave does require at least a 16-bit display to function.

NOTE
Even though you may be using a lower number of display colors, you can still save 24-bit images when you render your final output.

FILENAMES AND EXTENSIONS

Avoid using spaces in hard drive, directory, and file names for scene, image, and object files: this can possibly cause problems later if you use LightWave's distributed rendering feature.

LightWave uses several filename extensions for the different types of files it uses. The following list gives you some examples of these extensions:

.env	Envelope data
.lwo	LightWave object
.lws	LightWave scene file
.mot	Motion data
.p	Plug-in
.srf	Surface attribute file

Generally, LightWave will automatically add the appropriate extension to filenames when saving files, if one is not provided. However, this is not the case with most plug-ins.

Additionally, there are numerous standard filename extensions (.bmp, .iff, .tga, .wav, etc.) that LightWave uses, but are not specific to this application.

FOREIGN OBJECT SUPPORT

You may *load* foreign object formats supported by LightWave directly into Layout or Modeler. However, Modeler must be used to *save* an object using a foreign object format. To do this in Modeler, choose **File** > **Export** and select one of the **Export_** options.

Imported Formats

OBJ	M, SA, UV
DXF	M, SA
3DS	M, SA, SSP, T, UV
FACT	M, SA, SSP, T

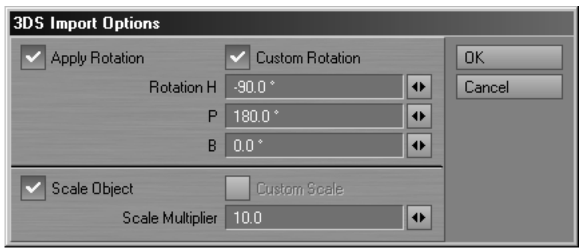
Exported Formats

LW5	(5.x LWO format) M, SSP, T
OBJ	M, SA, UV
DXF	M, SA, T, UV
3DS	M, SSP, T, UV

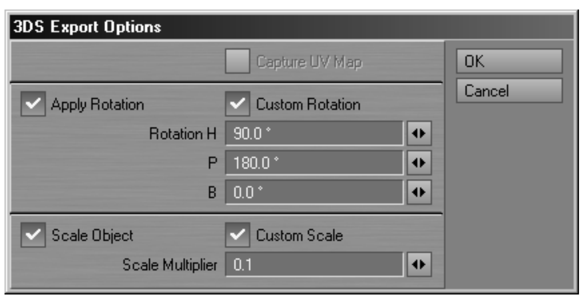
M = Geometry mesh
SA = Surface assignments (LW default properties)
SSP = Standard surface properties (color, glossy, etc.)
T = Texture maps (non-procedural)
UV = UV texture information

NOTE
Before exporting to a foreign object format, save the object as a LightWave .LWO file first.

When appropriate, an options dialog may appear when you load or save a foreign object format file. Some trial and error may be needed to get acceptable results.



3DS import option dialog



3DS import option dialog

NOTE
Generally, you should merge points after loading a 3DS object.

IMAGE AND ANIMATION TYPES

You can load and save various image and animation formats. What is available to you depends on the platform you are using, as well as the plug-ins you have active. Please check the NewTek Web site (www.newtek.com) for additional plug-ins as they become available.

The **Flexible Format**, **TIFF LogLuv**, and **Radiance** image formats contain high dynamic range data (discussed below). In addition, they store the data with floating-point accuracy instead of using integers. This approach results in much more accurate image data.

NOTE
When a 32-bit format is selected for saved RGB images, the 8-bit Alpha image is stored along with the 24-bit RGB data. However, when you save alpha images directly, you save only the alpha image, even if the format is 32-bit.

NOTE
The filename extension for Radiance images is .HDR. If you have any of these images using the .PIC extension, you should rename them.

Images and Memory

Mipmapping is a process where a series of images with progressively lower resolutions are created from a base (full-resolution) image. Each pre-processed image is a power of two smaller than the previous level. As textures move farther away from the camera, they obviously must be scaled down. However, because the mipmaps are pre-processed and loaded into memory, they can be used immediately for textures instead of continually shrinking down the base image; this saves rendering time, but requires more memory.

Eight-bit grayscale and eight-bit index-color images remain in eight-bit form internally. For color images, their mipmaps, however, are 24-bit since they must include colors in between color palette entries. Grayscale images remain at eight-bit for mipmaps. Floating-point images use 96 bits per pixel and have 96-bit mipmaps.

Here are some examples dealing with a 512 x 512 image in various formats, and how much memory is used for the base image and the first couple mipmaps:

512 x 512 eight-bit grayscale image:

Base image	262,144 bytes (512 x 512 x 1)
First mipmap	65,536 bytes (256 x 256 x 1)
Second mipmap	16,384 bytes (128 x 128 x 1)
Total	344,064 bytes

512 x 512 eight-bit color-mapped image:

Base image	262,144 bytes (512 x 512 x 1)
First mipmap	196,608 bytes (256 x 256 x 3)
Second mipmap	49,152 bytes (128 x 128 x 3)
Total	507,904 bytes

512 x 512 24-bit image:

Base image	786,432 bytes (512 x 512 x 3)
First mipmap	196,608 bytes (256 x 256 x 3)
Second mipmap	49,152 bytes (128 x 128 x 3)
Total	1,032,192 bytes

512 x 512 floating-point image

Base image	3,145,728 bytes (512 x 512 x 12)
First mipmap	786,432 bytes (256 x 256 x 12)
Second mipmap	196,608 bytes (128 x 128 x 12)
Total	4,128,768 bytes

High Dynamic Range Images (HDRI)

In computer graphics, color is displayed as a *triplet* value: red, green and blue. These values typically range from 0 to 255. Those 256 steps of color represent eight bits and together all three channels make up a 24-bit image. This means the maximum amount of color or luminance variation an image is allowed is merely 256 steps.

NOTE
See the discussion on radiosity in Chapter 32 for additional information on high dynamic range images.

In the real world, the human eye can perceive a much higher range of brightness and color values. Film can also react to a much wider range. Video cameras, however, are limited to a fixed range that fits closely to the same 256-step limit.

When exposed to *high dynamic range visuals*, such as a sunset or a desert landscape, the lens of a camera will produce some level of artifacting. Some of those artifacts can be seen as blooming areas of brightness, color bleed, luminance spill, lens streaking, and many other *visual cues* that tell the viewer there is a very bright light source in the scene.

These very bright surfaces can also contribute to the overall lighting of a scene. For example, sunlight streaming into a room will bounce off the floor and add a subtle illumination to the walls and ceiling that would otherwise be left dark (i.e., radiosity). All of these effects can be seen in images captured by devices that do not support high dynamic ranges.

Because computer graphics applications were designed to output to devices that would not understand pixel values above RGB 255, 255, 255, most applications do not provide for any value to exceed these limits. LightWave, however, calculates all internal data without limits and with IEEE floating-point accuracy. This means that when LightWave points a light at a surface, while the final rendered pixel may reach only RGB 255, 255, 255 for pure white, internally that pixel may have reached ten times that amount. This may not seem significant at first glance—white is white after all—but, if we look at how LightWave utilizes that data, it becomes very exciting.

Import/Export

LightWave can utilize high dynamic range detail, as it is generated internally (e.g., in the case of a very bright light) or from data in image files. This can be imagery generated from a series of photographs taken at various exposures and composited (see *Recovering High Dynamic Range Radiance Maps from Photographs* by Paul E. Debevec and Jitendra Malik at HTTP://WWW.CS.BERKELEY.EDU/~DEBEVEC/RESEARCH/HDR/) or data rendered in LightWave saved in one of the high dynamic range formats.

Once these images are imported into the system, they can be used just like any other image in LightWave 3D (e.g., as a texture, background, etc.). During the rendering process LightWave will respect the extra data in the image to assist in secondary lighting and other calculations.

Imagine using a high dynamic range image as an environment wrap (e.g., using the Image World environment plug-in. See Chapter 14.), which also illuminates the scene. With the appropriate imagery you can illuminate a scene without any lights and the results will match the look and feel of the original photograph.

Image by Terrence Walker.

Once LightWave finishes rendering, you can export images with the same high dynamic range data. This lets you bring that data back into LightWave or into compositing applications that support such data.

Using this extra data in the compositing process is very important as it can more accurately represent imagery as it would look if it were recorded directly to film. For example, compositing applications could use the extra dynamic range data to calculate the amount of diffuse bloom or color bleed from one pixel to the next.

Internal Compositing

Another area where high dynamic range imagery is supported is in LightWave's own internal compositing through pixel and image filters. Any filter can be designed to take advantage of the high dynamic range data with floating-point accuracy. This way, high dynamic range data can be leveraged in the post-process phase with included filters and by third-party additions.

Preview Compression Codec

The NTCodec.dll adds LightWave's preview compression scheme into the Windows AVI codec. You just need to install it if you want to play/load compressed preview anims into other Windows applications/players. See the ReadMe.txt file located in the NTCodec directory for installation instructions.

Configuration Files

The various LightWave configuration files are stored in your Profiles directory under Windows NT or the Documents and Settings directory under Windows 2000. Generally, you will not need to edit these files directly. On the Macintosh, they are in the System:Preferences folder (with long names).

Custom Configuration Files

Layout (Lightwav.exe), Modeler (Modeler.exe) and the Hub (Hub.exe) all support a -c command line argument that allows you to save your configuration files somewhere other than the system default (e.g., C:\Winnt\Profiles). To use, add a -c<path> parameter when you launch the appropriate executable.

Example: Modeler.exe -cD:\Monkey

NOTE
For Windows systems, just create a shortcut.

The -c argument sets up entries that the Hub tracks. So, if you click the Modeler button on the Layout interface, for example, it will run with the proper config file.

You must run Layout and Modeler with this argument at least once to store the entry in the Hub. Moreover, you must continue to use this argument when you run Layout or Modeler from an icon. If you don't, the config path entry in the Hub will change to the default. This can happen, for example, if you run Layout or Modeler directly from their .EXE icons.

The path must be valid or the default will be used. Do not add a trailing backslash unless you are using the root directory. If your path includes spaces, enclose the entire argument in quotes, like "-CD:\WHY DID I USE SPACES\CFGS."

A -p<path> parameter can also be added to select a different plug-in database file (LWEXT3.CFG). The path can be either a directory or a complete filename.

Layout Startup Command

If the LW3.CFG configuration file contains a line beginning with "StartupCommand," Layout will try to execute the rest of the line as a command after the main interface is first opened. The command can have arguments and it can refer to a generic plug-in or script.

Customizing Layout Viewport Navigation

The mouse directions used for viewport navigation can be customized by editing the WorldNavigation entry in the LW3.CFG configuration file. It has four boolean values that correspond to panning via icon and ALT key, and rotation via icon and ALT key. Zero values mean that the viewpoint moves in the same direction as the mouse, and *ones* mean that the world appears to move with the mouse. For example, the default 6.0 is 0 1 0 1. The old 5.6 setting would be 1 1 0 0.

Customizing Camera Presets

The camera resolution presets are stored in the LW3.CFG configuration file using the keyword "ResolutionPreset." Each preset includes a width and height, pixel aspect ratio, mask settings (left, top, width, and height) and a name that will appear in the pop-up menu. They can be modified, deleted, or added to. If no presets are found in the file, the original list is restored. See the configuration file for examples.

LIGHTWAVE LIMITS

- unlimited points and polygons per object
- 1,023 points per polygon
- 4,096 bones per object
- 32,000 cameras per scene (see note)
- 32,000 objects per scene (see note)
- 32,000 lights per scene (see note)
- 65,536 object layers
- unlimited images
- unlimited surfaces

NOTE
Because of memory considerations, the available number of cameras, objects and lights per scene is set at 100, 1000, and 1000, respectively, in the LW3.CFG file. The values may be increased—within the limits above, but not decreased below 100; however, this will require more memory, so keep the new values as small as possible. Do not edit the CFG file while LightWave is running.

MEASUREMENT UNITS

LightWave supports a large number of units of measurement. When entering numeric values into input fields, you may specify the unit of measurement using the following abbreviations:

Abbreviation	**Description**
um	Microns (one-millionth meter)
mm	Millimeters (one-thousandth meter)
cm	Centimeters (one-hundredth meter)
m	Meters
km	Kilometers (one thousand meters)
Mm	Megameters (one million meters)
mil	Mils (one-thousandth inch)
in or “	Inches (2.540 centimeters)
ft or ‘	Feet (.3048 meters)
kft	Kilofeet (One-thousand feet)
mi	Miles (5,280 feet)
nmi	Nautical miles (1.151 miles or 1852 meters)
f	frame
s	seconds

HINT
Since the **meter** is the default unit of measure for the **SI** or **Metric** unit systems, entering the meter equivalent is often easier than typing in the abbreviations. For example, enter .01 for centimeters, .001 for millimeters, 1000 for kilometers, and so on.

DO THE MATH

Basic math functions are supported in numeric input fields. You can input a string such as “12ft+14m” and get an answer of “17.6576m.”

Math Functions

Functions

abs(x)	absolute value
acos(x)	arccosine, return value in radians
acosh(x)	inverse hyperbolic cosine
asin(x)	arcsine, return value in radians
asinh(x)	inverse hyperbolic sine
atan(x)	arctangent, return value in radians
atan2(y,x)	arctangent of y/x, return value in radians
atanh(x)	inverse hyperbolic tangent
bessi(m,x)	Bessel function Im(x)
bessj(m,x)	Bessel function Jm(x)
bessk(m,x)	Bessel function Km(x)
ceil(x)	round up
cos(x)	cosine, x in radians
cosh(x)	hyperbolic cosine
dbessi(m,x)	derivative of Bessel function: Im'(x)
dbessj(m,x)	derivative of Bessel function: Jm'(x)
dbessk(m,x)	derivative of Bessel function: Km'(x)
djroot(m,n)	nth non-zero root of Jm'(x)
exp(x)	e (2.718..) raised to the power of x
fact(n)	factorial (n!)
floor(x)	round down
jroot(m,n)	nth non-zero root of Jm(x)
ln(x)	natural logarithm (base e)
log(x)	logarithm to the base 10
noise(x,y,z)	Perlin noise
rand(n)	Random value, seed n
sgn(x)	-1 if x<0, 0 if x=0, +1 if x>0
sin(x)	sine, x in radians
sinh(x)	hyperbolic sine
sqrt(x)	squareroot
tan(x)	tangent, x in radians
tanh(x)	hyperbolic tangent

Constants

_acres_per_sq_km	247.1
_air_density	1.293
_air_mol_mass	.02897
_atm_per_psi	.06804
_avagadro	6.0220e23
_boltzmann	1.3807e-23
_c	2.997925e8
_cm_per_in	2.54
_coulomb_const	8.98755e9
_deg_per_rad	57.2958

_earth_esc_spd	1.12e4
_earth_grav	9.80665
_earth_mass	5.98e24
_earth_radius	6.37e6
_earth_to_moon	3.844e8
_earth_to_sun	1.496e11
_eps0	8.85419e-12
_erg_per_joule	1e7
_eulers_const	.57721566490153286061
_ft_per_m	3.280839895
_g	6.672e-11
_gas_const	8.314
_gauss_per_tesla	1e4
_gm_per_oz	28.34952313
_golden_ratio	1.61803398874989484820
_h	6.6262e-34
_hbar	1.05459e-34
_joule_per_btu	1054.35
_joule_per_cal	4.184
_joule_per_ftlb	1.356
_joule_per_kwh	3.6e6
_kg_per_slug	14.59
_km_per_mi	1.609344
_knots_per_mph	.86897624
_lbs_per_kg	2.204622622
_lit_per_gal	3.785411784
_me	9.1095e-31
_mn	1.67495e-27
_moon_grav	1.62
_moon_mass	7.35e22
_moon_period	2360448
_moon_radius	1.738e6
_mp	1.67265e-27
_mu0	1.256637e-6
_oz_per_gal	128
_pasc_per_atm	101325
_pasc_per_psi	6895
_pasc_per_torr	133.32
_pi	3.14159265358979323846
_qe	1.60219e-19
_solar_const	1350
_speed_sound	331
_sun_mass	1.99e30
_sun_radius	6.96e8
_watts_per_hp	745.712
_zero_deg_cels	273.15

Other

ifeq(a,b,t,f)	if-equal, returns t if a equals b otherwise returns f
iflt(a,b,t,f)	if-less-than, returns t if a is less than b otherwise returns f
ifgt(a,b,t,f)	if-greater-than, returns t if a is greater than b otherwise returns f
iflte(a,b,t,f)	if-less-or-equal, returns t if a is less than or equal to b otherwise returns f
ifgte(a,b,t,f)	if-greater-or-equal, returns t if a is greater than or equal to b otherwise returns f

COLOR CHART

Blacks	R	G	B
black	0	0	0
ivory black	41	36	33

Greys	R	G	B
cold grey	128	138	135
grey	192	192	192
slate grey	112	128	144
warm grey	128	128	105

Whites	R	G	B
antique white	250	235	215
azure	240	255	255
bisque	255	228	196
blanch almond	255	235	205
cornsilk	255	248	220
eggshell	252	230	201
floral white	255	250	240
gainsboro	220	220	220
ghost white	248	248	255
honeydew	240	255	240
ivory	255	255	240
linen	250	240	230
navajo white	255	222	173
old lace	253	245	230
seashell	255	245	238
snow	255	250250	
wheat	245	222	179
white	255	255	255
white smoke	245	245	245

Reds	R	G	B
brick	156	102	31
cadmium red	227	23	13
coral	255	127	80
firebrick	178	34	34
indian red	176	23	31
maroon	176	48	96
pink	255	192	203
raspberry	135	38	87
red	255	0	0
salmon	250	128	114
tomato	255	99	71

Oranges	R	G	B
cadmium orange	255	97	3
carrot	237	145	33
orange	255	128	0
orange red	255	69	0

Yellows	R	G	B
banana	227	207	87
cadmium yellow	255	153	18
dougello	235	142	85
forum gold	255	227	132
gold	255	215	0
goldenrod	218	165	32
melon	227	168	105
yellow	255	255	0

Browns	R	G	B
beige	163	148	128
brown	128	42	42
burnt sienna	138	54	15
burnt umber	138	51	36
chocolate	210	105	30
flesh	255	125	64
khaki	240	230	140
rosy brown	188	143	143
raw sienna	199	97	20
raw umber	115	74	18
sepia	94	38	18
sienna	160	82	45
saddle brown	139	69	19
sandy brown	244	164	96
tan	210	180	140

Blues	R	G	B
blue	0	0	255
cobalt	61	89	171
dodger blue	30	144	255
indigo	8	46	84
jackie blue	11	23	70
manganese blue	3	168	158
midnight blue	25	25	112
navy	0	0	128
peacock	51	161	201
powder blue	176	224	230
royal blue	65	105	225
slate blue	106	90	205
sky blue	135	206	235
steel blue	70	130	180
turquoise blue	0	199	140
ultramarine	18	10	143

Cyans	R	G	B
aquamarine	127	255	212
cyan	0	255	255
turquoise	64	224	208

Greens	R	G	B
chartreuse	127	255	0
cobalt green	61	145	64
emerald green	0	201	87
forest green	34	139	34
green	0	255	0
lawn green	124	252	0
lime green	50	205	50
mint	189	252	201
olive drab	107	142	35
sap green	48	128	20
sea green	46	139	87
spring green	0	255	127
terre verte	56	94	15

Magentas	R	G	B
blue violet	138	43	226
jasoa	160	102	211
laker purple	153	51	250
magenta	255	0	255
orchid	218	112	214
plum	221	160	221
purple	160	32	240
violet	143	94	153

“It is a poor craftsman that blames his tools.”

Index

Index

0-9

A

B

D

E

F

G

H

J

K

L

M

N

O

P

Q

R

S

T

U

V

W