



Worley Laboratories
405 El Camino Real #121
Menlo Park, CA 94025 USA

G2 Manual

Chapter 1 QuickStart Guide

G2 is a huge system that integrates into nearly every part of LightWave Layout. Luckily you can start using the core G2 features and learn the other powers either as you need them or just as you work through the G2 manual. This quickstart guide is just a fast overview, but it's enough to let you start using G2 with your own projects.

Before you can use G2, you'll need to install and license it.

Visit [our downloads section](#) and download the G2 plugin. Uncompress it and copy the **g2.p** file into your LightWave plugin directory. Use Layout's "Add Plugins" (Layout->Utilities->Add Plugins) to install G2. Quit Layout to save the changes. Restart Layout.

You'll find new G2 plugins in the Surface Editor's Shader panel and the Master Plug-ins panel. If you try to open the G2 Master or Shader plugin interfaces, you'll be asked for your G2 license code. Follow the instructions (sending your "Machine ID" to Worley Labs, and getting the license code from us in response), and enter the code to activate the plugin. After this, you'll be able to start exploring.

G2's key feature is its previewing system. Even if you never use any of the advanced G2 lighting and shading features at all, the G2 previewing system will change the way you use LightWave. The previews give you interactive, render quality feedback of lighting changes you make in Layout. G2 does this by capturing surface textures and raytraced transparency and reflection rays during one "F9" render. After the render data is captured, lighting changes (including adding or deleting lights, changing light positions, intensities, types, falloffs, colors..) are shown interactively in the G2 preview. The preview also shows changes to the many, many lighting and shading options of G2 itself.

One way to start exploring G2 is with the **G2_simple_head.lws** demonstration scene. This is not a great piece of art, just a convenient scene that already has G2 applied to start. The first step you'll almost always do in any project is to add a G2 preview window. The easiest way to do this is with the G2 Generic plugin called "G2:Add Preview." (You can use LightWave's menu editor to make this a button on your main Layout menu!)

You'll see a new window appear: the G2 Preview. It will show just empty green at first because it hasn't captured your scene's appearance yet. The window is nonmodal, movable, and resizable so you can adjust its position on your screen easily. You can open as many preview windows at once as you like.

Render the scene with LightWave's "F9" control. When the render is finished, you'll see the preview window reappear showing the rendered picture of a basic head model. There are two rows of buttons at the top of the preview, representing the different contributors to the image. You can toggle these on and off to see the effects of the channels, for example by turning off specular. You can shift-click a channel to see just that single channel. Shift-click the channel again to see all the channels.

The preview updates when you modify LightWave lights. Pick a light in Layout and open its Properties panel. Adjust the light's intensity or color. You'll see the G2 preview immediately update. You can change *any* light attribute, including the type, falloff, shadow color, exclusions, or cone angle. You can *move* the light in Layout and rotate and scale it and still see the effects update. You can also add, delete or clone lights.

You can pan the preview's view by pressing the left mouse button in its view and dragging. You can *zoom* into or out of the preview by using the right and middle mouse buttons. The quality of the zoomed preview depends on the resolution of the rendered F9 image, but even lower resolutions work well. There's no need to use LightWave's antialiasing.

You can have as many simultaneous preview windows as you like, even showing views from different cameras or frames at the same time. The easiest way to make a new preview view is to visit the desired frame and camera you wish, use "G2:Add Preview" to create a new preview window, and hit F9 to render.

The buttons along the top of the preview act as tabs to reach other pages of preview controls. Click on the "Lights" button. The small purple button with a spiral arrow on it will let you cycle through the lights in your scene one at a time. This lets you see the effect of the individual lights on your scene. You can pick any set of one or more lights with the larger light picker button. Press the "Show All" button to show all the lights at once again.

You can learn many detailed facts about any pixel of your preview image by opening an "Info" window with the button at the bottom right of the preview. After you open this floating Info window, you can move the mouse over any pixel in the preview and see all of the numerical contributions to that pixel, both by different channels and by different lights.

After you're used to the preview, you can start experimenting with G2's lighting and shading options. If you single-click on a surface in the preview, a G2 window will appear showing the G2 *shader* that's applied to that surface. (Your model needs to have G2 applied to its surfaces for this to work.) You can also open the G2 shader panel from LightWave's Surface Editor.

G2 surface windows are nonmodal. You can have as many G2 panels open as you like, but by default new panels will *replace* old ones in order to avoid cluttering your screen. If you want the current panel to stay open (so other surfaces open as a new independent window), toggle the "Pin" button at the bottom of the interface.

The G2 shader panel opens showing the first page of surface options, "Boosts." These controls scale the surface's properties by a multiplier to make the effects stronger or weaker. This lets you interactively adjust, say, the Diffuse channel to be at the strength you like. (And, of course, every change is immediately previewed!) There are several G2 specific boost abilities like the ability to change the strength of received shadows and bump mapping effects. Note that if your basic surface has a 0% strength for a channel (like a 0% transparency) the boost can't multiply that 0 into something visible, so you'll want to set the base LightWave surface value first, and use F9 to capture it.

You can explore the buttons along the left side of the shader which show different categories of G2 features. All of the controls are definable for each surface, but there are also global G2 settings called "Master" settings. The "Open Master" button on the Shader panel will open the Master control panel.

The Master panel *also* has a "Surface Boosts" page. This lets you make additional *global* boost changes to your scene, for example, to make specularly stronger on every surface at once. The "Image Process" page shows controls for standard brightness and gamma processing of the output image. All of G2's renders and previews use full floating point color so there are no 24-bit quantization artifacts.

Other pages of the Master panel allow you to adjust Gamut limits (which allow you to define what happens to over-bright pixels), and to set up light groups for editing and previewing those sets at once.

There's a lot to explore in G2! Understanding the preview system is the first step to allowing you to get the effects you imagine. The rest of this manual goes through a detailed description of G2's abilities, but it's easy to experiment. The Skin Tools are a good place to start. Try using the preview to look at just the Diffuse effects in your scene, and show just one light at a time. Then apply the Skin Tools "Multilayer Subsurface Scattering" to see how the shading model affects the surface.

One last warning: G2 is addictive! But that's what makes it such a useful tool. Enjoy it!

Chapter 2 Introduction

G2 is a powerful system for lighting and shading. {You already know this or you wouldn't have purchased it.} It may seem daunting to start using this giant system, but really it's quite easy to start since you can learn one piece at a time. By the time you read this, it's likely that you've already played with G2, especially if you used the Quickstart guide, but hopefully the longer instructions can fill the details of the basic system.

2.1 Installation

G2 works with LightWave Layout version 8.3 or later. All of the G2 tools are contained in the single **g2.p** file. The software uses about one megabyte of hard disk storage. Installation is the same for PC and Mac versions of LightWave.

You can always get the latest version of G2 from our web site. You should visit [our downloads section](#) online to make sure you have the "latest and greatest" version.

To install the plugin, copy the *g2.p* file into your LightWave plugin directory (you can pick any subdirectory that you like). Once copied into this directory, LightWave still needs to be told about the installation. First close all running copies of LightWave. Start a fresh copy of LightWave and use the "Layout/Plugins/Add-Plugins" chooser and use the file requester to select the *g2.p* file.

Licensing

Piracy is unfortunately a major problem with plugin tools. They are small, easy to copy, fun to play with, and are given less respect than a standalone application, which makes plugins attractive to "share." This is extremely illegal, but unfortunately not uncommon.

The impact of piracy is lower sales and higher prices. This isn't what we want; we want to give you underpriced and overpowered tools!

The most practical way we can do this while still making some profit for future software releases is to protect the plugins from piracy. G2 uses the dongle you already have for LightWave as its method of protection. G2 is *not* tied to a specific PC, just a specific *dongle*. Some people install LightWave both at home and work, and just carry their dongle with them. There's nothing wrong with this, and G2 will have no problems.

The licensing won't affect your work after it is set up. Licenses for multiple dongles, usually at a studio, are also easily handled, even if the plugin file is being shared over a network.

After you install G2, the plugin can load, save and render even if it has not been not licensed yet. However, the *interface for each G2 panel will not be accessible until the plugin is licensed*. These interfaces can be opened in several ways but the easiest panel to get to for initial licensing is via the "Pixel" panel. To reach this panel, use the LightWave "Scene" tab and find the "Image Process" button on the left. Use "Add Pixel Plug-in" in this LightWave panel, and find G2 in the plugin list.

When you try to open the G2 interface (by double-clicking the "G2" name) the plugin licensing panel will open. It tells you the version and date of G2, and your "Machine ID," which is your internal dongle number. {There may be one or more numbers printed on your dongle itself, but this is not used by LightWave or plugins. I think these numbers can be used to predict lottery winners, though.}

You need a license code from Worley Laboratories to unlock the plugin interface. The license code will work only with your specific dongle, so we need to know the Machine ID number the plugin reports to create it. We also need your plugin serial number, which is printed on the back cover of this manual.

You can email, call, or FAX us to get your license code. The license screen has a button which should automatically start your mail program and compose email to us. {But it may not work on all systems, especially if you don't have a default mail program configured. In that case, you'll have write the email yourself. We just need your Machine ID and plugin serial number.} You need to add the serial number from the back of your manual and send it to us. We'll reply with the license code as quickly as possible; it should never be more than 24 hours.

If you have multiple plugin licenses, just repeat this procedure for each machine with a dongle. The plugin knows how to properly store multiple license codes even over a network. {If you have a *lot* of licenses (more

than 5 or so) you can email us and we'll describe how to license all of your dongles at once by editing the stored license file directly. This will avoid having to manually visit each machine, which is difficult and annoying for larger studios!}

Once you get your license code, just type it into the license panel. After licensing, the protection is transparent. You won't see the requester again unless you install on a new PC.

ScreamerNet machines do not need to be licensed. You can use the plugin on any number of ScreamerNet nodes as you wish. These nodes do not need licensing at all.

ScreamerNet and Multiple Machines

If several computers share the same LightWave directory over a network, make sure all of them have exited LightWave before installing any plugins. If you don't, the other copies of LightWave will overwrite the **LW.cfg** file. (LightWave uses this file to store plugin configuration information.)

If multiple machines share the same plugin file (because you have purchased multiple licenses, or you are using ScreamerNet on a render farm), be careful that each machine has a properly configured **LW.cfg** file. This is often done by manually installing the software on each computer and ensuring that each machine has its own copy of LightWave, **LW.cfg**, and plugin files.

A better alternative is for a single master server to hold these files and configure each machine to access it through the network. In this case, make sure the pathnames of the plugin files are not machine-specific. This is often done by setting up a network drive letter (like **Z:**) to store the LightWave and plugin files. Every machine, including the host server, mounts this network drive as **Z:** and always refers to it with that leading drive prefix. An absolute path like **C:\newtek\plugins\G2.p**, causes trouble because it's only valid for the server machine; the other machines don't necessarily have that file on their *own* **C:** drive and you'll get errors when they try looking there.

Network setup problems are especially hard to diagnose because ScreamerNet failures are difficult to analyze. ScreamerNet *silently ignores errors* such as plugins that it cannot find. If your ScreamerNet renders seem to be totally missing G2 effects, probably your plugin paths are incorrect and LightWave just can't find G2. Network installation of LightWave itself is a difficult topic that NewTek's technical support department can help you with if you're having a problem configuring your systems.

2.2 Legal Stuff

In simple terms, the software may be used on a single computer by a single user at one time. The single exception is the use as part of a rendering network via ScreamerNet. Purchase of a single copy allows *unlimited* use as a ScreamerNet client.

Worley Laboratories makes no warranties, expressed or implied, with respect to the quality, performance, or fitness for any particular purpose of the enclosed software. In no event will Worley Laboratories be liable for direct or indirect damages due to use of this program. This software may not be rented, lent, or leased. You may not distribute any part of the software or its documentation to any party. It's common sense.

Each license of the software is linked to one specific LightWave dongle you specify by obtaining a license code from Worley Labs. Your license allows you to use the software only on the machine equipped with that dongle. Your license allows the use of only a single copy of the software at one time, unless you have purchased multiple licenses.

The use or installation of the software implies agreement with these terms. Enforcement of these restrictions is via the copyright laws of the United States and the State of California. {This legal stuff is really tiresome but necessary. You don't have to read it. Hmm, but you probably already did, since this footnote is at the end. Sorry!}

2.3 Updates and Bugs

Plugins are often affected by bugs or problems with LightWave itself. NewTek periodically releases LightWave updates to fix software bugs on its [website](#). Check NewTek's website to ensure you have the latest LightWave version.

We release patches to our plugin software when we fix bugs or add new features. Our [support](#) section online contains the details.

We are admittedly perfectionists about bugs, and G2 has very few known problems. However, we have lists of LightWave problems which, in some cases, still affect G2. We try to use workarounds for these problems in the program, and document other problems here.

If you see strange behavior, first check the FAQ section. If you do find a new G2 bug, *we want to hear about it immediately!*

Reporting Bugs

When you find a bug, first make sure it *is* a bug and not a common problem. Read the FAQ section of this manual, and also read our website's [Support section](#) which has lists of known bugs and workarounds. Also make sure you have the latest versions of both G2 and LightWave to insure you're running the most modern versions.

Bugs that remain are easy to understand once they are isolated. If you do find a problem, start to *simplify* your scene to isolate it. This process can often teach you what the problem is, whether it's a software problem or not, and often whether there's an easy way around them.

Simplifying the problem is done step by step, just eliminating each extra layer of extraneous complication. Remove any objects that don't affect the problem. Remove extra lights if they don't matter. Turn off shadows if they don't matter. Take off surface textures. Change lights to simple point lights. Lower the rendering resolution to something fast like 320 by 200 as long as the problem is still visible. Turn off multithreading. Turn off reflection and transparency if possible. Remove unneeded plugins from the scene. Replace objects with a single polygon or at least remove extra geometry that's not needed. Try removing G2 and re-adding it to reset the values to G2 defaults. Try removing G2 altogether! (Sometimes the bug is inherent in LightWave, not G2). The basic idea is to simply remove as much extra distraction as possible and really isolate the problem in its simplest case.

It's extremely likely that somewhere during this process you'll find a step that gets around the bug. This doesn't mean there's no problem, but it does give a valuable clue, such as if the problem only shows up when using both multithreading and area lights. These clues can often let you find a workaround (maybe as easy as turning off multithreading, for example). They also give *us* clues for where the problem lies and how to solve it.

If you've simplified the scene but the problem still remains, contact Worley Labs. If you have found a new bug, your simplified scene is probably very small and easy for us to test. This will help us fix any G2 problem quickly. If you've found a LightWave problem, it will help the LightWave programmers fix their error as well.

2.4 Thanks!

Our thanks go out to each person and studio who helped make G2 so polished, useful, and powerful. These include our terrific beta testers: Stuart Aitken, Takahiro Arikawa, Stephen Bailey, Nicholas Boughen, Bill Boyce, Connon Carey, Jay Chernick, Gary Coulter, Don Culwell, Bob Eaton, Caroline Garrett, Zareh Gorjian, Steve Hofer, Shojiro Hori, Seiji Iseda, Kiyonobu Kitada, Joe Laffey, Akira Orikasa, Adrian van der Park, Rowsby, Syunichi Shirai, and Yuji Umoto.

Kei Nakamura was invaluable in helping with Japanese translation of this manual.

Nobuhiko Nabeshima consistently provided careful, well-thought-out suggestions and clarifications.

Kevin Stubbs not only helped with testing, but also provided enormous help in the totally underappreciated art of copy editing.

Rich Helvey also distinguished himself with his truly exceptional testing, suggestions, and examples.

Thanks, guys. We really appreciate the help in making this software!

2.5 Navigation around G2

G2 is a complex system and can be a little overwhelming at first! There are three different kinds of panels, and each one has pages of controls. Over 250 new buttons and names to learn!

But really, don't panic! The biggest and best feature of G2 is its totally interactive previews. This helps enormously since you see any changes take effect immediately, even as you're sliding a parameter value. Also, you can learn G2's features incrementally, ignoring entire classes of controls until you need to use them.

You'll interact with G2 using several different panels and methods.

First, there are individual G2 *shaders*, which are applied to your object surfaces in LightWave's Surface Editor. The G2 Shader panel controls settings that differ for each surface.

Next, there is a single G2 "Master" panel, which is a LightWave Master Class plugin. This is found in LightWave's "Master Plug-ins" panel. This G2 panel controls global scene settings for G2 such as image processing.

G2 also applies a "Pixel" filter, but this is added automatically and you never need to access it. If you try to open the Pixel filter's interface, the Master panel will open instead.

G2 also has eight "Generic" tools for performing certain convenient actions. These tools appear in LightWave's "Generics" plugin menu. Two of the G2 generics are used especially frequently. The first, "G2:Open Master", opens the Master panel with one click without having to find the LightWave Master Plug-ins panel.

The second common Generic is "G2:Add Preview", which makes a new G2 preview window, something that you'll be using almost continuously. The preview is nonmodal and resizable. You can add as many simultaneous previews as you wish.

LightWave also allows you to assign these Generics to a buttons in your main LightWave panel by using LightWave's "Edit Menu Layout" tool. This editor is opened via LightWave's "Layout -- Interface -- Edit Menu Layout" button. In this panel you can reconfigure your Layout interface to move and replace LightWave buttons and menus. Most G2 users have found it convenient to include the two common G2 Generics (Open Master and Add Preview) on their Default control tab. To create these buttons, open the "Plug-ins" entry in the left window and drag the "G2:Open Master" and "G2:Add Preview" items into the right window. You can place them into any LightWave menu you wish, or into the Top Group so they are always available. You can also assign these Generic plugins to hotkeys using LightWave's "Edit Keyboard Shortcuts" panel. Configuring LightWave's interface to include these Generics as buttons is optional but you'll find it useful as you use G2. You can always change the configuration later as you become more used to G2 and discover the setup that's most comfortable for you.

2.6 G2's Control Panels

G2's panels are all nonmodal, meaning that you can move and open and close them independently of LightWave's Layout window. You'll almost always leave one window open continuously since it doesn't interfere with your control of Layout.

There are different ways to navigate between G2 windows. There is always a battle for "screen real estate" because opening too many previews and controls panels will simply fill your screen and make it *harder* to find and change information, not easier. (But you can still do this if you like!)

By default, G2 tries to have only one control panel open at one time. If you have a G2 Shader panel open and then open a *new* G2 Shader panel, by default the new panel will *replace* the previous panel on your screen, so you still have only one panel open. This is also true for the Master panel, which will replace any open Shader panel. This behavior works very well at minimizing the clutter of multiple panels.

Sometimes you may *want* to have more than one G2 control panel open if you're often adjusting values in both. To allow this, G2's Shader and Master panels have a small button at the bottom labeled "Pin" that acts like a thumbtack. When this button is on, that panel is never replaced, and instead any new panel will open a new independent window. This override feature allows any number of panels to open at once. Sometimes

windows may overlap or even open directly on top of each other since they remember their previous locations. You can always move and rearrange the windows to adjust them to best suit your layout.

The Shader and Master panels each have subpages of controls, arranged by topic. This is really a tabbed interface control, though in G2 the categories are controlled by the bank of buttons along the left side of the panels.

The Shader panel's pages each have a small "C" and "P" button at the bottom left. These allow you to copy and paste the settings for that page into the corresponding page of another surface. Simply press "C" to copy the settings to memory. The "P" button will highlight to show that a paste is now available. You can press "P" to paste the stored settings into a new page. This is very convenient for making two different surfaces match behavior. It can also be used as a simple kind of scratchpad "undo" to allow you to test some settings and later restore them back to when you pressed "C." The paste buffer can be cleared by shift-clicking on a highlighted "P" button.

All of the G2 panels are interlinked for convenience, since it's often annoying to try to open LightWave's plugin panels to click on the G2 plugin directly. This is especially true for G2 Shaders, since you'll often be changing them to show different surfaces. You can always use LightWave's Surface Editor to manually select a specific G2 surface and open its G2 interface. But if you have a G2 surface panel open already, you can use the "Next Surface" button at the bottom to switch to the next G2 surface in the list. If you hold "shift" while clicking, you'll go to the previous surface. The G2 Master panel has an "Open Surface" button to switch to the last surface you've modified. And finally, the G2 preview window will open the G2 Shader panel for any surface you click on. Clicking on the preview is the easiest and most common way to open the G2 Shader panel you want.

The Master panel is most often opened with the Generic plugin as described previously. You can also open it from any Shader plugin by using the "Open Master" button at the bottom. Yet another way is to use LightWave's own Master Plug-ins panel to open the G2 Master directly. Finally, clicking on G2's Pixel plugin will also open the Master, for convenience.

Opening new Preview windows is almost always done with the Generic "G2:Add Preview." However, you can also use the Master panel to open a preview using the "Add New Preview" button in the Master's "Previews" page. You can also have a current preview window open a clone of itself by pressing the "Clone Preview" button in the Preview's Tools page.

It may seem that all of these panels are complex with many links between them. In practice, it gets comfortable quickly and all of the many links make it *easier* to navigate since you can get from one panel to another without having to use LightWave's plugin controls (unless you want to.)

When one or more G2 previews are open, sometimes you want to refresh all of them with a new F9 render. This can be annoying to do manually when there are several windows showing different frames or cameras. The generic "G2:Render Previews" will automatically render the proper frame(s) to update each open preview.

Since the G2 panels take up space on screen, it can be useful to hide and restore them. You can do this by using the Generic plugin "G2:(Un)Hide" to hide or unhide all G2 panels at once. You can hide or unhide just the preview windows with the "G2:(Un)Hide Preview" Generic.

Three final G2 Generics are useful for adding the G2 Shader plugin to different surfaces. If you pick an object (or set of objects) and use the "G2:Add To Objects" Generic, the G2 Shader will be added to each surface of each of the objects. "G2:Add To All" adds the G2 Shader to *all* of the objects in your scene. Conversely, you can pick one or more objects and use "G2:Remove From Objects" to strip off the G2 Shaders from those objects. (You rarely remove G2 unless you're trying to save new versions of your objects for another animator who doesn't have G2.)

Chapter 3 Preview System

The core of G2 is its preview system. You'll use it constantly. Its realtime feedback really changes everything about lighting and shading, since the immediate updates let you quickly find the exact best settings to make the appearances you want.

The G2 previews are easy to use, but there are also many, many options inside to give extra power and abilities. Initially, you can use the G2 preview as just a kind of interactive render display. As you learn more about it, you can start looking at more detailed display options to view your scene in new ways.

By the time you read this, you've probably already played with the preview system (using the Quickstart guide at the front of this manual).

3.1 What Previews Show

When you make a normal F9 render, the G2 preview system grabs a "snapshot" of your scene as you rendered it. This includes every surface property (including textures), reflection rays, transparency rays, shadow strengths, everything that's needed to recreate the image pixel for pixel.

This can take a lot of memory to save, but G2 has a smart compression engine to keep it manageable enough that you don't need to worry about the RAM use too much. You can always see how much RAM the previews are using with a readout on G2's Master panel. Most scenes use 30 to 50 megabytes, compressed. Rendering at higher resolution stores a more detailed preview. LightWave's Adaptive Sampling option will cause only half of the resolution to be stored in the preview, so we recommend you leave it off.

The previews *don't* store any lighting information! This may seem surprising, but it's actually the key to how G2 can exactly recreate the rendered image. The previews don't store the lighting directly, instead they store enough information to *recompute* the shading. This means that any *changes* you make to G2 settings (or information that G2 can get from LightWave, like light positions, intensities, types, ambient light, and so on) can be used to re-render the image with the new settings very, very, very quickly, in literally fractions of a second. The net effect is that you can interactively slide a parameter's value and the preview is fast enough to recompute the lighting over and over and over again and refresh the display in realtime.

If you pause and the shading parameters aren't changing any longer, G2 uses the spare time to add extra antialiasing to the preview, refining it to be more and more smooth and accurate.

This strategy is also what makes it possible to zoom and pan into the image. G2 doesn't interpolate the rendered colors of your image. That would just make giant blurry, blocky, pixels. Instead, G2 interpolates every single surface parameter, and then shades that new surface. The net effect is a preview that looks quite good even if the source render was very low resolution. (In fact, you might want to use very low resolutions for your F9 renders just because the G2 preview's new shading will give you a result that's superior to the F9!)

3.2 What Previews *Don't* Show

What doesn't update? There are three classes of effects that require a new F9 to "recapture" updated details.

First, LightWave does not allow G2 to sample new *rays* in interactive Layout, so any change that requires new rays won't be shown. G2 remembers the rays from the last F9, though, so those can be previewed and manipulated. So for example, reflection strength and tint is interactively previewed, but a change in what the rays show when you move an object or camera isn't. Shadows are also "fixed" because G2 can show updated shading, but any *updated* shadow positions from new light positions would require new rays (or a new shadowmap, which is equivalent.) Again, you can change shadow colors and opacities, but the *location* of the previewed shadows is fixed because changes from a moved light need to be sampled with new rays. A very few G2-specific settings don't update. Specifically, changes to the appearance of reflected ray gloss, since they need more rays. (Though gloss changes of environment maps *do* update since they don't need rays.)

The second limitation is that LightWave does not allow plugins to sample object *surfaces* so changes to surface texturing are also not updated and must be captured again with a new F9. So G2 isn't useful for previewing LightWave surface texture changes.

Finally, G2 can't query LightWave's *postprocess* effects such as volumetric lighting, lens flares, and pixel and image plugins. G2 has its own postprocess effects (like gamma and brightness controls) which *do* preview properly since G2 creates them and therefore knows about them inherently.

My hope is that future LightWaves will provide these three types of information to plugins. Computers are more than fast enough to provide them all without having to "freeze" and go into a special render mode, and LightWave itself previews surface textures and post process effects but just doesn't share the results with plugins (yet).

When you first read about these limitations you may think that the G2 preview is crippled, but in practice you really don't notice it very often. Admittedly, you do get so used to everything updating in realtime (literally over 150 different controls give perfect feedback) that when you hit just one that doesn't (like reflection gloss) you're kind of surprised.

Another common habit at first is to keep doing occasional F9s to "make sure the preview is up to date." After a while, you'll realize that you rarely have to.

3.3 Using Previews

The first step to using the G2 preview system is to open a preview window, as described [here](#). There are several ways you can make a new preview appear. By far the easiest is by using the "Generic" plugin named "G2:Add Preview." You'll use this so often that it's very useful to map the generic to a LightWave button so that it is easy to access at any time. Creating this menu tool is discussed on the same [page](#).

The preview window opens showing a blank green screen. This is a signal that there's nothing to show yet; you need to render a frame to capture the surface details for G2 to preview. This F9 render can be low resolution. Higher resolutions do create higher quality previews. There is no need to use LightWave Antialiasing when capturing previews, since the preview will only capture the first "pass" of the render and ignore other passes.

After LightWave has rendered the image, the preview will reappear and show your scene. However, this will be a fully *live* preview that will update in realtime with (most!) of your G2 setting changes.

The preview window is *nonmodal* so you can move and resize it to suit you best. It's common to leave the G2 preview window on top of LightWave's Layout window, but of course you'll rearrange things as you change projects or tasks.

Since the preview can be resized, it may be hard to see some details if the window is small. You can zoom into the preview to magnify your view by using the *right* mouse button to zoom in and the *middle* mouse button to zoom out. You can also use the small "+" and "-" buttons to zoom. (As a secret hidden feature, shift-clicking the "+" button will resize the window to display the preview at actual size, and shift-clicking "-" will make the window as small as possible.)

Holding the left mouse button and dragging in the preview window allows you to pan the view in the window to focus on any detail you wish.

The "Reset" button at the bottom will recenter your preview and set the zoom back to full size.

You can open more than one preview at a time! Each window has its own size, zoom, and pan. This is useful to show a full-frame view of your scene in one window, and a zoomed-in detailed shot of a character's head in another. All open previews update simultaneously as you change settings.

Output Components

By default, the preview shows all the effects in your scene, essentially the same final render you'd get with a LightWave F9. But the preview system is more versatile, since it can selectively show just *parts* of your render. This can help you isolate the causes of strange shading, or remove distractions like specularities while you work on background lighting.

The very top of the preview has a horizontal line of buttons that show different "pages" of preview controls. The first page, "Output," lists the different components of the final render underneath as a set of 11 small buttons. These are labeled with the basic types of effects that add up to make the final rendering. The first buttons, "Diffuse" and "Specular" are usually the most important parts of your image.

You can toggle each component on and off by clicking its button. For example, you might want to see your image without the effects of specularity, allowing you to focus on the remaining diffuse shading details. This toggle works for every channel, including ambient lighting, reflections, transparency effects, G2 scattering effects, even LightWave's Global Illumination (radiosity and caustics.)

For convenience, if you *shift*-click a component, then all *other* components will turn off. If you shift-click that isolated component again, all of them will be active again.

Three of the components are a little special. Turning off "Shadows" shows your scene with no shadows. Similarly, "Img Pro" (Image Processing) shows your image without any special G2 [image processing effects](#). Finally, "LW Effects" hides any areas of the preview that G2 does not control. This includes the surfaces of objects that do not have G2 on them. It also includes areas that show fog, Hypervoxels, or other postprocess effects.

Raw Channels

The preview can also show the base values of the objects. These are the raw (unshaded) surface colors, or other base surface properties like Diffuse or Specular.

Usually these channels will show only a solid shade unless your objects are textured. This view is very handy for seeing those texture effects, for example to see if your Specular surface property has the kind of noisy pattern you expect on it.

These "raw" channels are shown by using the "Channels" button at the top of the preview to visit the second options page.

Like the Output components, the raw Channels are selected by clicking on the appropriate channel, though you can select only one Channel at a time.

The color channels showing the RGB surface, the Global Illumination, and the Reflection and Transparency ray colors all show the color directly. The rest of the channels use a black and white value to show the intensity of the effect. If a property (like Diffuse) is stronger than 100%, the preview will tint pink then red to show a stronger and stronger "overdriven" value. Similarly, negative values are shown as deeper shades of blue.

Areas of the image that don't show a G2 surface will show a neutral deep green color.

Lights

The raw Channels are useful, but most often you'll be viewing the Output channels instead, because they show how your scene actually appears when rendered.

That appearance depends on the lights in your scene. Usually you'll want to see the effect of all those lights together, since that's the way the final image is rendered. But when editing, sometimes it's useful to focus on certain light effects separately, since the combination of all lights at once can be distracting and it can be difficult to understand what highlight or shadow is caused by which light.

So, like the Output components, G2 allows you to show the effects of single lights at a time, or any combination of lights. You can even cycle through the lights one at a time, which is very useful to check that each light is doing the "right thing." (It's surprising how many scenes have lights that are effectively useless because of their placement or type. You don't notice this because the other lights are doing all the work and you don't realize that the extra light is too dim or has bad placement or settings. Viewing the lights one at a time makes it easy to check for this situation.)

By default, the preview shows the effect of all of the lights in the scene. To change this, visit the "Lights" page of the preview window. The bottom row of controls is most important. The first button labeled "Show All" is lit by default. As it says, it makes the preview show the effect of all lights at once. If you turn *off* this button, you can *pick* the light or lights you want to see by using the light picker button next to the "Show All" button.

It's especially common to view only one light at a time, and cycle through the lights one after the other. This is so useful and common that there's a shortcut button to the right of the light picker. This small shortcut button has a corkscrew arrow on it. Pressing this button will turn off the "Show All" button and select the

first light in the scene. Pressing the shortcut again will advance to the next single light. It's extremely common to simply cycle through the lights one at a time by repetitively clicking on this shortcut. You can cycle *backwards* by shift-clicking the button.

Light Groups

Especially large scenes may have huge numbers of lights, enough that it's hard to cycle through hundreds of them. Often the lights fall into categories, like the headlights and tail lights of a car versus the street lights along the road versus the lights inside the rooms of a house on that road.

G2 has many tools for handling groups of lights at once, as described on page~\pageref{light_groups}. That section of the manual talks about boosting and restricting the effects of groups of lights at once. But groups can also be used for previews as well. If you define light groups (as described in the Light Groups section), the preview will activate the top row of buttons in the Preview's Light page. These buttons are labelled 1--8 corresponding to the same light group numbers. The group's button is lit by default, meaning "include these lights!"

By clicking off the group's button, that set of lights will be excluded from viewing even with the "Show All" button set. The shortcut "cycle" button to advance through the lights one at a time will also skip those excluded lights.

Even lights that are not in a group can be excluded, since they have their own "Ungrouped" button.

Viewing Frames and Cameras

Not only does G2 allow you to have multiple previews open at once, but different previews can show the effects of different captured frames, or views from other cameras. The Preview's "View" page contains three simple but useful ways to define the frame and cameras.

The left hand button changes the method the preview uses to define the displayed view.

This button defaults to "Last F9." Whenever you make an F9 render, that render will show up in the preview immediately. If you change cameras or view frames and F9 render again, the preview will again update and show the frame you just rendered. The preview acts very much like a viewer for the last F9 render you make.

If you want to *lock* the preview so that it keeps what it currently shows even if you make a new F9 render of a new frame or camera, you can cycle the mode button to "Manual," and set the camera and frame with the controls to the right.

The final mode, "Use Scrub," makes the preview interactively follow the frames you visit using LightWave's frame scrubber. When you first use this mode, it will be disappointing because most frames don't have a captured preview; you'll just see blank green previews in every frame except the last one you F9 rendered.

But, *if you have enough RAM*, you can make a *really* fun and easy way to edit the lighting of an animated scene.

G2's Master panel has a "Previews" page. One option of this page is "Automatically Free Previews at F9," which is on by default. This prevents G2 from wasting RAM by keeping old preview data around after you're done with it. But you can *deactivate* this feature and the previews will accumulate.

Now set your camera resolution to be *very, very low* (which saves preview RAM). Leave your preview's view mode as "Last F9" since you want it capture each individual render as it's created. Finally, render a sequence of perhaps 10 frames using F10. {This is all RAM limited. You can use higher resolutions and longer sequences if you have enough RAM, but start small at first and use the Master panel's RAM display on the Previews page to make sure you're not overdoing it. Running out of RAM makes most computers slow and unstable, it's best to avoid it. You can probably capture and preview 150 frames of an animation at a rendered resolution of 160 by 120, but only if you have at least a gigabyte of RAM. The G2 preview can be any size you like, just keep the *rendered* resolution low. }

After the sequence is completed, the preview will show the last frame of the sequence. Change the mode to "Use Scrub."

After following these instructions, you'll find the preview will update as normal, with all of its normal options. But, (and boy is it cool!) the preview will track any frame you scrub to. This shows every effect including the way a specularly flows across a car's hood as it drives by a light, or the way shadows fall onto a character as it enters a dark cave.

This "scrub following" mode is not the usual way of using previews, since it can take a *lot* of RAM (It depends on your scene, the camera resolution, and the number of frames, but it's easy to use hundreds of megabytes!). But as computers have more and more RAM (512 MB is quite standard now in 2002, and most professionals have 1 GB) this will become a smaller penalty.

You *can* use LightWave's automatic "Play" controls to make LightWave scrub through the scene automatically. G2's previews support this mode as well, though usually LightWave plays so quickly that the G2 preview's update will be very low resolution or even blank. It's probably not useful or practical to edit or preview animations with G2 using this method. (Then again, if your machine is *really* beefy, it may just be able to handle it, if not now, maybe later as CPUs get even more powerful.)

LUTs

LUT stands for "Look Up Table" which isn't very descriptive of the problem it solves.

Video behaves differently than film, with images that have different color and brightness behavior. And flat LCDs behave differently than computer monitors. And different computer monitors even look different. And even the same monitor can display an image differently based on its controls and video card.

So what you get is a mess! This results in the knotty problem of trying to ensure that the picture you see on your screen will still look the same after you've recorded it to video.

The most common solution is to apply some compensation correction to the displayed image to account for the difference between the way your monitor displays images and the way your final output device does.

It's sort of like a local deli shop that always makes the same mistake and gives you only 90 percent of the weight you order. This is annoying because your recipes or sandwiches will never have as much stuff as you wanted. But you can compensate for the annoying deli by simply automatically ordering an extra 10 percent each time. You change your order and ask for more than you need, because the deli will take that 10 percent off. You finally end up with what you wanted in the first place.

The same strategy works for displayed images. Your monitor may behave differently than what you want. But if you know how it acts, and you know how you want it to act, software can adjust the images to make your monitor behave like your final output device does.

With a monitor, the simplest way to measure its behavior is called "gamma." It's a single number that describes the way brighter pixels appear even stronger than you'd expect them too. The higher the gamma, the more powerful these bright pixels become, even though dimmer ones stay the same or even decrease. So it's something like a contrast control.

If you know your display's effective gamma, and you also know the gamma of your final output display (the one that your clients will use!) G2 can use these values to compensate and preview how the image *will* look on the new display. This compensation is applied *only* in G2's preview since it's a way of predicting how the rendered image will appear later on the new machine.

Gamma is not the only way to calibrate or measure this kind of response; it's just the most common. More precise calibration curves have been standardized to try to allow exact representations of how images are displayed. The "sRGB" standard is used for Internet file exchange. The "rec 709" standard is used for video. Both of these are carefully designed to be consistent and reproducible.

So where does the term "LUT" come in? It means "Look Up Table" and it's based on the way that these curves are often represented, as tables of numbers. This is especially common when using "old fashioned" 24 bit colors.

G2 has straightforward support for these compensation curves. Luckily G2 uses full precision colors, not just 24 bit, so it can apply exact gamma, sRGB, and rec709 curves without needing a table.

To apply these compensations, you first need to know how your *monitor* displays images. Is it calibrated to a certain gamma? Or to the sRGB or rec709 standards? If your monitor is truly calibrated, you'll know it because you have special software and even calibration hardware to directly measure the monitor's behavior.

If, like most people, you don't know what your monitor really does, it's probably best described by a default gamma, probably of a value around 2.2. You can measure approximate monitor gamma by eye with reference images (try an Internet search for "gamma chart").

You also need to know the calibration of your destination, the final display. If it's video, rec709 is probably appropriate. If it's another monitor, you'll want to find its gamma.

In the G2 Preview panel, the "LUT" page allows you to enter these two settings, one for what your monitor is calibrated to, and one for what your final destination uses. After you set these, the preview will convert its displayed colors to portray what the net effect will be on your destination.

This kind of compensation is used almost exclusively by the highest end professional community. If you're in the majority, and you don't have calibration equipment and never worried about display compensation, then you *don't need to worry about this*. The default settings leave G2's preview showing the same as the rendered output, and it's OK to simply ignore it! This is how you've worked with LightWave and Photoshop all along, and there's no need to change. But when you do need compensation support, you can come back to G2's LUT settings and the tools will be waiting for you.

Tools

The final page of Preview controls, labeled "Tools", holds four simple functions.

The first, "Show Pixels," turns off G2's surface interpolation. This shows you rendered pixels directly. This is always inferior to the interpolated view *except* that it better matches a zoomed version of the actual render. In particular, the interpolated view may show highlights or shading caused by the interpolated surface that don't exist in your actual render. By viewing the actual pixels, you can more easily determine if a strange artifact is your shading or just the preview's interpolation. (To be honest, you will rarely need to use this tool.)

The next button, "Clone Preview," opens a new G2 preview window. The new window will have the same size, options, and settings as the first one. This is especially useful when you want a side by side comparison of a difference (like showing Diffuse only in one window and Specular in another, or two different lights in the two windows.)

"Refresh" does just that. It forces the preview to refresh itself in order to clean up any artifacts or problems that may need an update. G2 does an excellent job of automatically refreshing when needed, but this manual override may be useful if there's some situation G2 misses. (In practice, you won't need to use this control either!)

Finally, a "Update Style" setting allows you to set what pixel size the preview uses when it takes its first pass updating the window. Normally the preview will pick an optimized level that trades off resolution for interactivity. Some testers find it useful to change to a *finer* update level, which will *slow down* the preview's updating so you can see more clearly the "old" effect compared to the "new" effect. If you have a super-super-fast machine, you can also use this finer level to increase the quality of the preview's initial update. Coarser settings are useful to keep interactivity on extremely slow machines. (In practice, you won't need Coarse modes. G2 updates very quickly even on an old 450 MHz CPU machine like I use. Yes, in 2002, I am using a 450 MHz CPU to develop G2.)

3.4 Info Window

As you move the mouse over the Preview window, you'll notice a convenient display of the RGB values of the color that's under the mouse pointer. This common kind of readout is in Photoshop and LightWave's popular QV image display too. But G2's preview takes this idea a lot further.

Click the small "Info" button at the bottom right of the preview, and a new floating Info Window appears. This window is packed with numbers, detailing just about every statistic of every pixel of your image. Just move your mouse across the preview window and you'll see the information update instantly.

The top of the window shows you the exact pixel you're viewing. It even shows fractions of a pixel, since the G2 preview lets you zoom *into* the LightWave image. It prints the words "G2 Surface" if the pixel shows a surface with the G2 shader applied, or "Not a G2 Surface" if it's a pixel that doesn't have G2 or the G2 object has been obscured.

Underneath, three lines display and detail the exact color of the pixel. The first line is "Shaded:" color, computed after lighting and shading. Notice that it's a high precision number so you can view the intensity of even ultra-bright pixels. The second line, "Processed:," shows the same color but after Image Processing effects (G2's gamma and brightness controls) have been applied. Finally, the "Clipped:" color is the net effect after Gamut Control, if you've activated it. (See page~\pageref{gamut}) This line also shows the final RGB pixel values for the color after it's been lowered down to 24 bit color for display. (High precision colors are great, but seeing integer values from 0--255 is a long, comfortable habit!)

The comparison button underneath the color readouts is described in the next section.

Moving down the window, the surface and object names are displayed. Next comes the contribution of each attribute like Diffuse, Reflection, or Ambient. There is also a percentage listed for each to make it easy to see which components dominate. Often you'll have a strange artifact or weird pixel, and this percentage helps you quickly realize the bright spot is from a reflection (or whatever.)

At the bottom, the contributions of individual *lights* are printed in a similar fashion. Again, it's easy to see which light is dominating the shaded effects.

3.5 Comparison Tool

Instead of displaying the value of the pixel you move the mouse over, you can *lock* the display by *shift* clicking in the preview window. This will open a new Info window that *sticks* with the pixel you clicked on. This means that even as you change G2 parameters or edit LightWave lights, the Info window will update to continuously show the numeric details of just that pixel. This can be useful when you're trying to see exactly how the color of a specific location is affected as you change a setting.

If you have a specific goal, a color you *want* to create, you can let G2 help you match that shade with the "Compare to:" tool. You may be trying to make your LightWave scene's shadows match the color of a background plate you're compositing into, or make highlights the same tint as a photographic reference. In any case, you can turn on the "Compare to:" tool and use the color picker to set your desired color. G2 then displays the difference between the current and desired color.

In fact, you can open *multiple* Info windows, and use the Compare tool in each one simultaneously. In this case, a single summary line is printed on the bottom of the G2 Master panel, showing the *total* error of all matches combined. This is useful when you want to match multiple shadows or color tints simultaneously. You can adjust lights or any G2 property and see whether your changes improve or worsen the overall match.

Chapter 4 G2 Shader Controls

The G2 Shader panel defines properties that affect each surface independently. As surface shaders, the per-surface G2 settings are stored in your object's file (not the scene file) so the properties you set will appear in the scenes that use that object.

There are 13 separate pages of controls in the G2 shader. But as the introduction mentioned, you don't need to worry too much about being overwhelmed. It's safe to ignore any option you don't need or understand, and just use the features you want and learn them as you need.

4.1 Boosts

The most fundamental changes you can apply to your object surfaces are *boosts*. G2 is not designed to texture objects, but you can use it to interactively adjust the basic properties of any surface. While you could use LightWave's own Surface Editor to do something such as increase the Diffuse value, this does not give you the real-time feedback that G2 does. G2's interactive boosts make it very, very easy to find the absolute best settings for these properties.

"Boosts" is the first page of the G2 Shader panel. At the top, it lists the basic LightWave surface properties including Luminous, Diffuse, and Specular. Each of these properties can be "boosted" by a multiplier to make the effect of that channel stronger or weaker. The default boost of 100 tells G2 to use 100 of the original surface property; in other words, leave the value unchanged. A boost of 50 would use a half-strength version of the original surface, and similarly 200 would use a double-strength version. These multiplying boosts affect even textured parameters.

Five of the boosts allow control over properties that you can't normally access with LightWave. The "Global Illum" channel allows you to brighten or dim the effects of LightWave's Global Illumination (radiosity and caustics) on the surface. "Ambient" controls how much the world's ambient light affects the surface.

"Respect Light Exclusion" makes the surface react only *partially* to LightWave's built in light exclusion abilities. Normally this LightWave feature makes a light either include or totally exclude an object, but G2 extends this to allow you to "exclude" a light only *partially*, to dim its effect instead of fully eliminating it. (There are several other alternatives to adjusting light intensities per surface or per object. See the [Light Groups description](#).)

"Shadow Opacity" allows a surface to receive only *part* of the shadows that fall on it. This is very useful when a shadow is simply too dark and you want to lighten the shadow slightly.

Finally, "Bump Mapping" lets you adjust the strength of all bump maps applied to the surface. This is especially useful when the bumps are caused by a complex gradient or image, since you don't need to change the map or texture settings and re-render to test your change.

Each of these Boost settings on the Shader panel is duplicated on the Master panel's "Surface Boosts" page. However, these Master settings affect every G2 surface in your scene. These global adjustments are especially useful for Ambient, Global Illumination, Reflection, and Shadow Opacity.

You can choose to have a specific Shader ignore these global settings. This lets you make an exception for just specific surfaces. For example, you may want to brighten ambient lighting for all surfaces except one. To do this, you could use the Master panel's Ambient boost to increase ambient light on all surfaces, but click off the "Include Master Surface Boosts" button on the G2 surface you don't want to brighten. This is a lot easier than manually boosting every surface in the scene in each G2 Shader panel.

4.2 Edge Colors and Effects

G2 can also apply boosts (and other adjustments) to the edges (the outside rim) of your object. This is a simple but surprisingly versatile way to add many kinds of realistic and special effects.

The most important technique is to use these edge effects to increase the reflection of the surface at the object's edges. This is a real-life behavior, called the "Fresnel" (pronounced FREH-NELL) effect. This

increase of reflection is easy to see in real objects around you if you study them. Even a normal piece of white paper becomes quite reflective when you hold it up level with your eyes and stare across it; you can easily see lights reflected when it's at this very glancing angle. (Try it sitting in front of your computer; the screen is bright enough to see.) Another good example is asphalt, which you don't normally think of as being reflective. But think about driving at night, when you can see the taillights of the car in front of you reflected in the pavement. (Some people believe this is a heat mirage effect, but it happens no matter what the temperatures of the pavement and air are.) Specular highlights are also stronger when they're at the edge since they are just a kind of reflection. The Fresnel effect also changes transparency, but it *decreases* the transparency at the edge.

While real-life Fresnel effects only apply to reflection and transparency, G2 allows you to affect the edges of other channels as well. While these shouldn't be called "Fresnel," and aren't necessarily physically justified, in practice they can make very interesting effects that would be difficult to otherwise create. {Originally G2 had a specific Fresnel tool which used the proper math formulas to recreate the physical behavior of light. We later replaced this with a more general set of tools that allowed you more customizable control even if it wasn't as mathematically justified. }

Edge effects are applied in G2's "Edge Effects" page in the Shader panel. Each of the basic surface properties like Diffuse, Specularity, Reflection, and so on is represented. By default, each channel shows "No Edge" so no special effects are applied.

If you click the "No Edge" button repeatedly, you'll see there are several different options to determine how the edges should be modified. The first is "Multiply" which *scales* the current channel value by the value you specify. So for example, imagine your LightWave Specularity value was 25. If you use G2's "Multiply" mode with a setting of 200, the Specularity value will be boosted up to 50 (25 times 200, i.e. doubled) at the edge of the object. If the G2 setting were 20, the final value at the edge would become 5 (25 times 20). Too much math? Don't worry too much, because the preview lets you see the final effect immediately.

Instead of *multiplying* the base value, you can pick the "Add" mode and simply add to the base. So a base Specularity of 25, with a G2 "Add" value of 50, would make a final value of 75 (25+50) at the edge.

The final mode is "Replace" which simply sets the edge value to the value you specify. So a G2 value of 75 would make the edge become 75 regardless of the original base value.

In each case, you can change the sharpness of the edge (how close to the actual rim) the effect occurs by using the "Edge" control. Yet again, the preview gives you instant feedback so it's not too overwhelming to figure out what value gives the effect you're hoping for. It can sometimes be useful to look at the [raw surface channels](#) in the preview, since you can see the changes in the raw channels more easily than the final shaded render.

An additional edge channel, "Edge Bump Boost" allows you to make textured bump maps stronger (or weaker) at edges, with a similar "Edge" sharpness control.

A final control, "Use Bump for Edges" makes G2 include surface bump mapping when deciding where to apply edge effects.

Edge Color

The edge controls continue on a second page of options that change surface *color* on edges.

The first settings adjust color Hue, Saturation, and Value. It may be surprising at first but these adjustments can create very interesting effects such as color-changing paints, pearlescent materials, and iridescent inks. These are all useful on many objects including tropical fish, "new style" American money, and fancy metallic car paints. Like the other channels, the "Edge" parameter changes the sharpness of the edge itself.

Surface colors can also be tinted by one or two extra shades. This applies different colors that you directly specify to the edges. Since there are two applied color slots, you can create a broad range of edge styles that again are useful for metallic paints and special effects.

4.3 Processing

The Processing page of controls is similar to the Boost settings described [here](#). However, these modify the surface color instead of the numerical channels like Diffuse or Specularity.

The controls allow you to change or boost the Hue, Saturation, and Brightness of the surface color. Like Boosts, you could edit these in LightWave's main panel, but you'd lose the interactive update of G2's previews. Also, changing a textured color in LightWave may involve editing an image map or a gradient, which G2 can modify much more quickly for simple changes.

In practice the most useful control is Hue shifting, which is great for small changes to surface hue, especially skin. If your texturing (and lighting) result in a surface that's just a little bit too reddish, you can use a small Hue adjustment to move it back to the shade you like. It's especially useful for color matching in a composite image since you may be trying to make a photograph and a rendered object share similar colors.

A second column of options allows *random* changes to the hue, saturation and brightness. This ability seems odd until you realize that it's invaluable when used on object clones. In LightWave, cloned objects always share the same surfaces and colors. You can move and scale the clones independently, but their colors are always identical to each other. But imagine that you have 100 alien spaceships flying through the sky and you want them to be different colors. Normally you'd have to make 100 copies of the object and manually change the colors for each one!

With the *Random* controls, you can make each clone adjust its base colors to a random variation independently. This might be subtle like making small skin tone variations in a group of characters, or overt like entirely different colors of cars. It can help to randomize rocks and boulders, making a pile of unique stones from one original object. Think of applications like cloned buildings to make a city, copied trees to make a forest, airplanes lined up at an airport, even jellybeans in a jar.

A final control, "Follow Master Desaturation" makes G2 ignore the Master panel's Distance Desaturation effects, discussed [here](#)

4.4 LSD: Luminous Shadow Darkening

In high end rendering, one of the most challenging tasks is seamless compositing. You have a "plate" (photograph or sequence from a video or film) that may show, for example, an empty stretch of desert. Your task is to convincingly add a giant sand worm to the desert. This kind of task is challenging but very common. (Well, not adding sand worms to the desert, but compositing in general.)

G2 doesn't help with modelling and animating a sand worm, or matching the camera view. What G2 can help with is the simple but extremely important task of adding the shadows cast by your geometry into the plate. This is challenging because your object's shadow needs to interact with the geometry of the world shown in the plate. You usually also have the requirement that the plate be unchanged except where there is a shadow; you don't want the entire plate to dim or brighten or the scene will be different than the original plate.

The solution to this problem is not an obvious one. It's also a problem that hobbyists don't often encounter, so they may not know the standard Hollywood studio techniques. The first step in the standard method is to build very simple "stand-in" geometry that crudely approximates the shape, position, and scale of the objects in the plate that will receive shadows. The appearance of these objects is irrelevant, and the geometry is often only roughly correct. It's not surprising to see a desk represented by a single polygon. A hillside is sometimes just a stretched and rotated sphere.

This geometry is crude because it's just there to determine the shape and position of shadows that fall on it. The actual geometry is made invisible by using the "Sticky Front Projection" plugin of the Taft collection from Worley Labs. The background plate is used as a background for the rendering, but the same image is also front projected onto the stand-in objects. If the object surfaces are set to be 100 luminous, with no diffuse or specularity, the rendered stand-ins have no shading and all you actually see is the seamless background plate.

For a flat desert, it's not hard to scale up an object that's just a single square and define that as the "ground." You position the camera so the horizon in the plate roughly matches the horizon you see in Layout's preview, so the polygon is located roughly where the desert ground is in the real plate. You Front Project the desert

image onto the luminous plane, and its rendered appearance will match your desert plate.

Adding Shadows with LSD

The Luminous Shadow Darkening (LSD) Tool on the Lighting panel in G2 is designed to solve the problems with this kind of compositing situation. Basically you want the front projected color value to be visible, unless a shadow falls in that area, in which case you want to darken the color.

This is what the LSD tool does when activated. It computes the shading for the point as usual, but then determines the strength of the shadow(s) that falls at that point. If the point is fully lit, the surface color is left unchanged *and no shading is done*. The surface is *also set to be fully luminous*. This means that the surface color, usually from the front projection map is now the output color in your rendered scene.

If, however, the point is shadowed, the LSD tool determines the strength of that shadow. It then sets the surface to be luminous, with no shading, as before, but it *darkens the surface color* based on the strength of the shadow. This is exactly the effect you want to composite in the shadow effect.

In fact, you have extra controls over this shadow. You can define the exact color of the shadow; shadows are rarely simply black! This ability is invaluable when trying to match the effects of lighting visible in the plate. If a real rock on the desert floor image has a shadow of a certain color, you can simply make your sand worms have the same color shadow. This helps enormously when doing photorealistic composition. You can also select the *opacity* of the shadow to try to match the lighting. If there's a lot of ambient light in the real plate, a shadow may be pretty dim and barely noticeable. You can change the opacity of the composited shadow to match this subtle behavior by only partially darkening the surface. A 100 opaque shadow is just a solid blob of color and doesn't feel realistic at all.

The controls of the LSD tool are extremely simple. The "Use" button activates the tool. You can also select the color of the shadow. The opacity of the shadows can be set with the normal G2 "Shadow Opacity" control on the Boost page.

LSD Strategies

Once you understand what the LSD tool is designed for, it's extremely simple to use. The basic steps are to build stand-in geometry, front projection map your plate onto it, apply G2, activate the LSD tool, dial in your shadow quality, and render. All of G2's other features work as they usually do; you can select lights to ignore or boost, you can change shading parameters, and you can use area light shadows. Area light shadows are particularly effective in composite shots since they mimic the real behavior of lights, and really help make a convincing illusion.

Even if you're not compositing geometry into photoreal scenes, you may find the LSD tool to be extremely useful. One application that turns out to be extremely similar to the fancy Hollywood compositing is in the production of web page graphics. Imagine that you have a background color for your page, and you want to create the illusion of 3D gadgetry sticking out of the page; maybe a toggle switch or active "click here" button. You can make this imagery work effectively by lighting the object consistently and rendering with the camera aimed vertically down onto the object. A finishing touch is to add a drop shadow to the item, which gives the impression of true 3D depth.

This drop shadow is almost always done in a 2D paint program by taking the rendered outline, turning it into a mask, blurring it, then stamping it down at an offset from the original geometry. How annoying! This sequence is nearly the same as the fancy plate composition shot though.

The approach with G2 is to make a flat plane and set it to the exact color of your web page background. Place your gadget geometry on top of this plane and light it. Apply G2 to the flat plane, and turn on the LSD tool. Your output render will have your 3D gadget geometry sitting on top of your background pattern, with the shadows from the geometry falling onto the flat plane. The color of the plane will be *unchanged* from the original color, except for the shadows. This is all done in one pass, so it's great for quick prototyping.

A final bit of history: The LSD technique was invented for a project for Computer Cafe in 1995. It was a simple and effective solution for quick composites. It was included in our 1996 Gaffer software, and also in our Sasquatch tool. It returns again in G2, though now it is partnered with G2's interactive previews, making it more powerful than ever.

4.5 PhotoMapping

Realistic surfaces usually have a lot of detail. Wood has very intricate grain patterns, skin has pores, wrinkles, and freckles, metal has pits and scratches, concrete has stains and cracks. Good surface texturing uses image maps, procedural textures, and other tools to try to recreate at least some of this detail on 3D objects.

It seems obvious that the most photoreal surface is the one shown in a real photo! As long as the picture shows fine surface details, isn't it an obvious source of detail for your object? There are two obvious problems with this, of course.

The first is the mapping itself.. how do you make it line up with your model properly? There are many tools for helping with this, including LightWave's UV map system and plugins like Sticky Front Projection (part of the Taft plugin collection from Worley Labs). Often you'll pose the photographs to make them as easy to map as possible, for example by taking a flat view of a brick wall, or a face viewed from directly in front instead of from an angle. G2 doesn't help you with this step of mapping.

The second problem with photographic maps is the lighting, shading, shadows, and specular highlights they include. What you usually want is an unlit "base" surface color for your objects, since LightWave (and G2!) will be adding shading on top.

This often creates obvious problems. If your wood photo has a shadow falling across it, the LightWave wood surface will be dark where that shadow appears. And it's not easy to "fix" this by just adding a light. That shadow is "baked" into the surface and adding a LightWave light won't properly brighten the dim surfaces. It's hopeless in a way similar to trying to use a flashlight to brighten a dim corner of a painting.

Even if there are no obvious shadows, there are often shading and highlights that are inappropriate. If you try to use this as a surface map, it's like the surface has shading applied to it *twice*, which is very strange and makes bright areas brighter and dim areas dimmer.

Photos are still often used to add detail because you often take the photo in a way to minimize the shading. And sometimes even if there is some shading in the map, it doesn't look *too* bad and the mis-lit artifact is an acceptable compromise in order to get the nice details the photo adds.

G2's "PhotoMapping" mode is designed to help with this second problem when using photos as texture sources. The goal of the G2 PhotoMap tools is to allow you to "de-bake" your mapped photos, attempting to derive the original "base" surface colors without the baked-in shading, shadows, and specularities. Then you can add new LightWave lighting and avoid (or at least reduce) the problems of strange double-shaded appearances.

This mode isn't magic, since in general there's not enough information (or fancy artificial intelligence) to analyze a photo to understand its lighting and how to remove its effects. However several surface lighting models use image processing to compensate for shading and specularities effects. You could imagine doing this kind of compensation manually in Photoshop, perhaps by playing with the image intensity curves or color ranges. This can work, but is slow because Photoshop doesn't have tools that are specifically designed for shading removal, and it's also annoying to keep re-testing each new trial map in LightWave.

G2's PhotoMap mode uses a image processing model to help you remove baked-in shading effects, and preview the final effects in real time with the G2 preview.

To use PhotoMap mode, you first need to apply your map to your object! G2 can't help you with this part. Different projects might use normal texture mapping, UV mapping, Front Projection maps, or plugins like Sticky Front Projection.

The next step is to use G2 (and its preview!) to find a good "debaked" appearance. But what is an unbaked surface? Basically we want to remove shading and specularities effects from the surface as much as possible. This new surface color is much "blander" than the original shaded photo, but *this is what we want*. A common misconception when setting up the surface texture is that it's supposed to look good by itself, but remember it's just a surface color and we'll add our own shading and shadowing later. I mention this now because if you don't think about the goal of surface mapping, you will be disappointed that no "magic cool astonishing" effect happens.

For example, imagine that you're trying to make a texture of a human face, and you've carefully projected a photo of a face perfectly onto your model. Now you're trying to make a base surface color from this projected photo. In theory, what should that base surface look like? We know there should be no highlights, so any specularity should go away. Even more importantly, there should be no shading, so all of the brightness effects from the surface shape will disappear. This will make a flat, boring surface color. The raw surface color will form a see kind of a flat plane of skin color with dirt and specks on it. It won't even look like a face.. the nose's shading is missing, and the cheeks flatten out. But this is *exactly* what we want. The shading details come from lighting and shadowing, and we'll be adding them back later. Since skin material looks pretty similar on the nose and cheek, we *want* the nose outline and profile to disappear.

Now that we have an expectation of what goal we're striving for, we can start to use PhotoMap mode in G2. You'll probably want to set up your model with different surfaces for each part of the model; that is, a skin surface, a lip surface, an eye surface, and so on. This is so you can use different PhotoMap compensation values for different regions.

We want to *see* the unshaded surface color as we manipulate G2's settings. Use the "RGB Surf" view of the G2 preview to see this raw unshaded color. When you activate this view, your surface probably looks great, just like the photo! But remember this is actually BAD, since that photoreal appearance is because of the shading and lighting that's already baked in. We want to *remove* that to get just the base colors and surface effects.

Photomap Controls

G2's PhotoMap tools are located in the shader's Processing tab. Turn on the "PhotoMap Surf" mode to 100. This activates the full image processing to remove the shading intensity differences and specular spots.

With the default settings, you'll probably see something that looks like a bland, washed out version of your original map. Remember that's not a bad thing! You can use four G2 controls to balance the processing to retain the most surface detail while still removing the shading effects.

The G2 "White Point" control is used to control how much color is allowed in the surface. This is usually used to "deflate" a white specular highlight in order to restore the original surface color it's obscuring. The White "Range" describes how much variation is allowable in the surface whiteness. Your goal is to examine the white specular areas of your map and reduce the Range setting until the highlight gets reduced or eliminated. The larger the Range value, the more detail will be preserved in your surface, but the more likely it will be for the highlights to remain. Luckily you can find the best compromise with the real time preview. After you set the White Range, you can also set the White Point itself to give the correct overall whiteness of the surface.

The Bright and Bright Range controls are similar. These remove shading and shadow intensity differences, equalizing the shading across the surface. Again, you adjust the Range value, making it small enough to remove obvious shading. You want a good compromise where the shading is mostly removed but as much surface detail is preserved as possible. When you have a good range, you can set the final overall brightness. Remember that it's OK when something like a nose seems to "disappear." This is a sign of success, since you've found a skin texture that matches the cheeks, making the boundary becomes less visible.

Sometimes you'll find that a very dark shadow or very bright highlight will become an "odd" color after processing. This is because the image itself lost the information about the surface. (If a photo pixel's RGB value is 255 255 255, what's the underlying surface color? You can't tell, since the RGB has been maxed out at full white.) G2 lets you add a small bias color to push the pixels of these uncertain regions to a more reasonable compromise. Increase "Tint Bias" to preferentially move towards a color shade that works with your own surface.

After you've found the best debaked surface color don't be disappointed that some of the liveliness of the photo is missing. We now have the huge benefit of being able to relight this surface *without* having the original shading added! You can also add new surface details (like extra dirt) by texturing the Diffuse and Specular channels. You probably don't want to apply texture to the color channel since that defines your photo image. {It's too bad LightWave doesn't have multiple color surface channels!}

Adding New Lights

This technique works extremely well for projects where you are adding *new* lights to an existing photo or sequence. We can restore the original photo's appearance, and have G2 simply *add* shading to this surface. This is a common and powerful trick in many LightWave compositing shots, but G2 gives you new power to make realistic additions and changes. Previously, with LightWave alone, you were forced to use the same photo as your surface color both for lighting and luminous compositing. As we know, that's wrong since the lighting is already baked into the image and we don't want to duplicate it.

G2 has a special PhotoMap-only luminosity mode that lets you use your original photo as a luminous surface to "match" the original photo. The "Base Lum" control in the PhotoMap panel adds a luminous version of your *original* surface map to LightWave's output. You don't need to set LightWave's luminous channel at all.

This allows a very very powerful ability that can be used over and over in many different projects. This surface-and-luminous tool of PhotoMap mode allows you to *add new lighting effects to a photo*. If you set "Base Lum" to 100, the rendered LightWave image will match the photo, pixel for pixel. But it will also add new LightWave surface lighting that *you* design. This surface lighting will not act on the photo color, but on the surface color that PhotoMap derived. This produces much greater realism when you add your own effects or lighting. This means that your actress's face will still look photoreal even after you animate your glowing logo flying in front of her, with the glow properly illuminating her skin as it goes past.

You can also experiment with even more modifications. Remember that LightWave and G2 support *negative* lights. If your photo has a highlight or shading from a direction you don't want to include, you can partially remove that effect by appropriately placing a LightWave light with negative intensity. (You may want to exclude this light from other surfaces by using Light Exclusion or Light Group Boosting) It's unlikely you'll be able to match the position and effect of the original photo exactly, but you can still make major modifications to the original photo lighting. You can virtually *move* a light in the real photo by placing a negative light in the original location used in the photo and adding a new LightWave light in a new replacement position. Adjusting the photo lighting is never exact or complete, but you can create convincing changes of soft lighting without strong shadows. The real time preview of G2 is invaluable for adjusting this effect.

4.6 Skin Tools

Human skin is very difficult to recreate faithfully since our brains are very used to seeing and studying even the most subtle variations in human appearance, especially human faces. This makes realistic skin very important, yet very difficult.

Realistic skin requires a good texture to form the pattern of freckles, pores, wrinkles, and even fine hairs on its surface. It's not easy to recreate all of these details. They can be added with procedural textures, painted with an image map (3D paint programs are especially well suited for this) or photographs can help make good starting points. G2's PhotoMapping mode can help with this last photographic method.

The other important realistic skin behavior is its *shading*. Skin is a complex material, mostly because it is very translucent. A large amount of light penetrates about a centimeter into your skin, (You don't believe me? Go into a dark room and shine a flashlight onto the back side of your fingers. You can see the light come through even the thickest part of your finger.) and reacts to all of the layers of tissue underneath. These layers make the light bounce and scatter in complex ways, resulting in the characteristic "soft glow" of skin. There are also very visible color effects due to the subsurface blood, no matter what your surface skin color is. G2's skin model uses a scary amount of mathematics to simulate and compute the way light reacts to these surface layers of tissue. Luckily this physics engine is pre-tuned to react to the kinds of scattering that happens in humans, so you can create realistic skin shading without having to study medical optics.

G2 models skin as layers of tissue, but you only need to concern yourself with the *visual* requirements of the model. In particular, the most important "decision" that G2 needs to make is just how thick the model's surface skin is at every point. Skin varies in thickness enormously. Where the skin is very thin, the blood-filled layers underneath are not very obscured, so the skin appears redder. Thick outer skin (called the epidermis) will obscure this blood layer and the skin will become paler. Lips are red because of their very, very thin epidermis. The heels of feet often have a yellowish tint because of their thick epidermis. Even when your skin becomes red (from sunburn or drinking) or pale (from extreme cold), it's because your body is

advancing or withdrawing blood from the upper layers of your skin.

G2 absolutely needs to know this skin thickness in order to calculate the way light interacts with it. But defining a "skin thickness map" is *not* intuitive nor easy. Luckily G2 uses a very artist-friendly way to make it very easy (and even somewhat automatic!) When you look at skin, it's usually a pink/peach tint (for white races). This comes from a mix of the red blood and the yellowish epidermis color. G2 uses this observation by calculating that if you have painted (or textured) a reddish surface tint, it's a sign that the epidermis is thinner. Less red (in proportion) implies a thicker epidermis. Luckily this technique works even for brown and black skin. Therefore G2 can learn from and react to skin thickness layers based on the intuitive skin surface layers that you paint or texture. These tints are exactly the ones that you intuitively add. They also match with the colors that are photographed!

This design allows G2 skin simulation to be quite accurate but not need a huge number of user settings to adjust mysterious internal parameters. The controls that you do have are much easier to use and understand, and combined with the realtime preview, you can reproduce accurate and realistic skin shading quite quickly.

G2's Skin Controls

To use the skin shading simulation mode, visit the Skin Tools page of the G2 shader interface. The simulation is activated with the main "Multilayer Subsurface Diffusion" control. Notice that higher values make your surface appear "softer." The rim of side and backlit objects also becomes brighter, as some of the light bounces through the surface towards the front. The effect can be subtle, but viewing only the Diffuse channel for only one light at a time makes the changed shading easier to perceive. A natural value for skin is around 60 but of course you should use the preview as your guide.

This shading is still inaccurate because it uses a generic epidermis layer that has equal thickness all over your surface. You can activate the variable thickness by increasing the "Epidermis Visibility" control. You'll notice that the shading will change especially on the edges of your object. This is caused by the fact that your view cuts through more epidermis at an angle than it does in a straight-on view. (It's sort of like why it's shorter to cross a road directly instead of walking across at an angle.) Since the light goes through more epidermis at these angles, both the shading and color change more. G2's layer design leaves the base skin hue UNCHANGED when viewed straight on, so your intuitive color mapping is completely natural.

The amount of thickness variation is controlled by the Epidermis Visibility, but you can tune the shading effect with the "Epidermis Scattering" control, which describes how much light is affected by the outer epidermis as opposed to the inner layers. Higher scattering tends to make the skin appear more evenly lit and brighter, and the color of the epidermis more pronounced.

The final control is "Blood Color" which is the shade G2 uses to derive this very interesting skin thickness. This also indirectly determines the epidermis color. (G2 can compute the effect of the blood layer alone, use that with your textured surface color to derive what color and and thickness the epidermis must be to be consistent.) If you change blood color, your shaded skin will often appear more colored *oppositely* since it affects this computation. That is, if you make the blood color lighter, the outer skin will appear darker, and redder blood will add more blue-green to the shaded skin.

The Skin Tools page has two other controls which are useful for making realistic skin. The first is "Diffuse Bump Adjust" which is particularly important. Since skin is so translucent, most of the colored diffuse reflection happens from deep beneath the surface. This means that the surface bumpiness and pattern actually does not change the shading of skin very much! However *specularity* is very affected by this surface texture. It is extremely useful to use a very low bump mapping strength for diffuse shading and a much, much higher strength for specular. You can clearly see this on your own forehead in a mirror. Notice that you can see pores and wrinkles clearly in a highlight, but they become much less noticeable when the light angle changes enough to move the highlight away. An image bump map or even procedural noise bump is extremely useful to get the broken-up highlights of realistic skin.

Finally, one last control allows simulation of a *different* kind of surface shading effect, called "rough diffuse" (often called by its inventors names, Oren and Nayar). This is a model of surface reflection that assumes that the surface is very very bumpy on a very tiny scale, and that light bounces around several times on these bumps before it reflects off. This is not a subsurface effect, just a result of light bouncing multiple times on the surface before leaving. This model is quite useful for rough, "dry" surfaces like rock, the Moon, and

especially clay (think of a flower pot). It can be combined with the layered subsurface simulation to approximate a material that has some surface interactions as well as subsurface ones. It can help with the appearance of skin, though the subsurface lighting model is much more important. There's no noticeable speed penalty for using Oren Nayar. (Layered Subsurface Scattering is slower than normal shading, but only by about 50, and is usually not noticeable).

A side note:

The graphics buzzword "subsurface scattering" has many vague definitions used differently by everyone. But there are two main techniques which are perhaps confused the most.

The first is "Layered Subsurface Scattering" which simulates reflections and interactions on and through surface layers. The classical reference is Pat Hanrahan's paper in SIGGRAPH 1993. This method is used by G2 for skin simulation. Examples are skin, leaves, and wax paper.

The second is "BSSRDF" light simulation which models the effect of light entering into a material at one location and exiting at a different location. There has been more modern research on this topic, especially by Henrik Jensen in 2001 and 2002. G2 uses this rendering method for its translucent effects on the "Scattering" tab. Examples are marble statues, a glass of milk, or a wax candle.

4.7 Subsurface Scattering

Subsurface Scattering is a complex name for an important lighting effect seen in *translucent* materials. Materials like plastics, colored glass, and especially wax show the distinctive "glowing" appearance from light that penetrates *into* the material.

G2's Scattering page allows you to create these kinds of surfaces, and even design and edit the effect interactively with the G2 preview.

There are two key visible scattering effects. The first is seen in transparent materials similar to colored glass or liquids. These materials are transparent enough that you can see an image *through* them. The material may be cloudy or colored, like tea or a glass paperweight. The interesting effect with these materials is that the opacity of the material depends on its *thickness*. You can see through a few millimeters of coffee at the bottom of a mug, but if you add more, soon the bottom of the mug will be obscured.

This transparency absorption effect is straightforward, though it's not built into LightWave. G2 allows transparent objects to vary their opacity based on this thickness.

The second effect is much more interesting (and complicated for the renderer). It simulates the way light penetrates into materials, bounces around inside (multiple times), and finally finally comes out somewhere else. This "subsurface scattering" is what gives wax its characteristic smooth glow. But other materials, including liquids, glass, and plastics share this distinctive effect. This subsurface scattering often occurs even in surfaces that are effectively opaque (you can't see anything through a wax candle, but a lot of light still scatters inside to make its surface glow when backlit.)

LightWave has its own "Translucency" control, but, despite the name, this is not actually used for true translucency. LightWave's feature is used to cast light and shadows onto the back of polygons (like a sheet of paper.) It is not a volumetric effect like subsurface scattering.

Apping in G2

G2's Shader panel has a "Scattering" page to define the attributes used by G2's subsurface renderer. The first two controls define the two kinds of effects. "Transparency Absorption" changes a transparent object's opacity, making it more opaque in areas where the material is thicker. This option is only useful when your object has surface transparency! In fact it's common to leave your surface at 100 transparent, and allow G2 to add the realistic thickness-based absorption. This changes only the transparency of material, *not* the way it reacts to lights. This is the control you'll use for colored glass. It's *unnecessary* (and pointless) for objects without surface transparency.

"Translucent Lighting" is the more important control, used for subsurface rendering to apply the effect of light bouncing through the interior. It makes a wax candle glow from its flame, and your ears shine red when they're backlit by the sun. Higher values create a stronger effect. Transparent materials (like glass) can and

should use both this control as well as the first Transparency Absorption setting.

When either of the two settings are on, the remaining gadgets are activated, allowing you to define the characteristics of the scattering material itself. The most obvious is the material's *interior* color. By default, the color is the same as your surface color, so your white candle will glow white. But ears may have a pink or brown surface color, even though the *interior* color is very red from blood, which is why the ears glow red when backlit by the sun.

By default, the internal material color is the same as the surface's. You can reduce the "Tint with Surface Color" control in order to define your own alternative color. Fractional values let you mix the surface color with your own alternative shade.

It's interesting to think about the color of a *volume* of material because it really is different than a surface. A surface is blue, or red, or whatever. But a volume can *change color* based on distance. You don't usually need to worry about complex multiple color absorption, but G2 supports it by allowing you to define a "Complementary Color" amount. When this value is 0, the object's interior is shaded the way you expect, as a solid hue. If you increase Complementary Color, the interior will allow more white light to penetrate through thinner volumes. Thicker volumes will become more and more deeply shaded.

This is the effect you see in a swimming pool. A foot of water looks transparent white, but as the water gets thicker, it becomes more and more distinctively blue. So you'd define a swimming pool's water with a deep blue color, but with a high Complementary Color. The opposite effect happens with apple juice. Even one millimeter of juice looks orange-yellow. This color does not change even in a thick jug of juice. So for juice, you'd leave Complementary Color at 0.

"Falloff Distance" defines how *opaque* the material is. It's a distance, a lot like the distance you define for LightWave's fog. The larger this distance is, the easier it is for light to penetrate into the material. So think of it as how *far* light can get through. Smaller numbers let less light through. If the value is too small, you'll get stronger effects on the surface, since the light can't get deep inside your object, so only the very outer layer of the model has an effect. A large falloff distance makes a more evenly lit volumetric effect.

"Internal Contrast" changes the absorption to make the effects vary more (or less) smoothly, creating more variation in the net effect.

All of these controls apply to Transparency Absorption. Translucent Lighting uses the *same* controls, but has three additional properties. When the renderer deals with lighting (as opposed to simple transparency), the way the light bends and bounces is very important. The most important setting is "Subsurface Scattering," which defines how "bouncy" the interior is; how easy it is for light to reflect and change directions. Small values tend to make the material glow in the area where you shine a light, and in the area on the other side where the light exits. As the value increases, the light is more likely to bounce and diffuse, so the lighting becomes more even and smooth everywhere. Glass tends to have small amounts of scattering, and wax has a lot.

"Nonlinear Scattering" is a term that controls how the scattering itself reacts to thickness. That is, as the light progresses through the interior, does it diffuse more and more in direction?

Finally, "Surface Refraction" adds an extra "bend" to the light direction as it exits. This changes how much the appearance depends on the surface normal.

These descriptions aren't amazingly useful because the right vocabulary for what they do requires fancy mathematical definition. But you have the huge advantage of *not needing to understand everything*. G2's real time preview will update and show you changes instantly! {The design of G2's renderer is by far the most complex part of G2's programming. Volumetric Subsurface Scattering is usually computed by a photon simulation, popularized and refined by Henrik Jensen. G2 uses this simulation method but extends it by storing the result in a way that can be reused and updated with to show changed properties. It's a variant of the technique that G2 uses for its main interactive preview. This means that changes to the scattering attributes can be accurately previewed without having to re-run the photon simulation. It does use extra memory for every light in your scene but it's a cheap price to pay for instant feedback.} The most important controls to experiment with are the last two (Falloff Distance and Internal Contrast) but you'll see instantly if other settings can help or hurt your look. {I understand the math, I wrote the code, I determined the controls and their names. But even I don't think about the meanings of each parameter any more. I just twist the dials to see what looks best. It's a great feeling! It's like how a painter doesn't have to worry about the type of

screws that hold his easel together, or even the type of wood in his paintbrush. It's a huge relief to forget about the details and use your brain to think about the image, not the software internals.}

Coffee is a good example. The color of coffee is black, right? Not really. In a thin layer of a millimeter or so, it is yellow!{Try it yourself with some coffee in a white cup.} Only a deep layer of coffee shows the distinctive brown. This is all due to the fact that really all materials *absorb* light. Different colors of light are absorbed at different rates. A thin layer of coffee is enough to absorb *bluish* light, but yellow can still get through. This makes the coffee look yellow. But coffee also absorbs *greenish* light. This absorption takes more coffee to become noticeable. But when the bluish and the greenish light are both absorbed, you're left with dark red.. which is the brown of coffee. This is unrelated to LightWave's own translucency control.

4.8 Specularity

Specular highlights are everywhere. Even LightWave 1.0 had specularity built into it, since it's so important for realistic appearances.

G2 adds extra control to surface specularity. A *lot* of extra control! But the specular effects update in the interactive preview like most other G2 features, so it's easy to see what effect the different settings give you.

By default, G2's specularity looks and acts very similarly to LightWave's.{But they're not identical. This is by design, since LightWave's specularity has an annoying fundamental flaw we didn't want to include with G2. This flaw causes plain LightWave highlights to show up on the back side of objects in error. G2's highlights don't do this, but that means they don't match LightWave exactly.} You can simply ignore the G2 Specularity options until you need them.

The most fundamental specularity improvement over LightWave's is that there are multiple *layers* of specularity. This means that instead of one highlight, you can have two or even three. This is common in many surfaces, the most common being car paint. Car paint usually has a broad, deeply colored highlight *as well* as a tight, glossy white highlight.

This kind of "double" highlight is usually caused by a layer of transparent wax or sealer. The base surface has a broad colored highlight from the underlying colored paint. The transparent layer on the surface adds the glossy white highlight. Since the top layer is transparent, you can see both highlights superimposed.

Multiple specularity layers are useful for designing any kind of surface, not just paint. Since each layer is independent, you can often combine them to make effects like a colored fringe around the edge of the highlight.

G2's "Specular" page of options defines all of the specularity settings. The first button at the top cycles through G2's three layers of specularity (Primary, Secondary, and Tertiary.) The strength of each layer can be boosted independently. By default, the second and third layers have a boost of 0%, disabling them.

Specular highlights are usually the same color as the light that makes them. If you want to *tint* this color to make, for example, a red highlight, you can pick this tint with the "Specular Color" control. The default is white, matching LightWave's. G2 can also use the object's surface color to tint the highlight using "Surface Color Tint." At 100%, highlights will be the same shade as the surface, very similar to LightWave's "Color Highlights" control.

The color tinting option is invaluable for matching real specularities because different surfaces have a variety of responses. Highlights from plastics are almost always white since the reflection is from the transparent plastic matrix and not the suspended color pigments. Materials like paper and wood often have deeply colored highlights even when lit with white light. Some materials such as metals often have odd color responses such as a bluish tint from copper and greenish from tin.

G2's specularity follows LightWave's Gloss settings. But G2's gloss can be changed independently for each specular layer with the "Gloss Boost" control. This allows different layers to have different glossiness settings, creating different sized highlights.

The "Gloss Bias" setting changes the edges of the highlight, sharpening or softening them.

Anisotropy is the effect of surfaces that have highlights that are *stretched* in one direction. The "Anisotropy" and "Anisotropy Direction" control how much effect this has on the specularity. The anisotropy effect itself is discussed

Retroreflection {Retro means reverse, or backwards. So retroreflection means to reflect back to the source.} is an interesting effect seen in some materials, especially road signs. Normally light reflects off of surfaces at an angle, like a mirror. These signs use either glass beads or carefully designed bumps on the surface to make light reflect back towards the source of the light. This causes the signs to appear very bright when your headlights shine on them, since the reflection heads *back* to you instead of off to the side.

G2's "Retroreflection" adds this effect of changing the angle of the reflected light. In practice, this tends to move highlights towards the outside rim of your object. This can create very interesting appearances. It's an easy way to add a highlight to the rim of objects.

G2's final specular control, "Respect Bump Map," enables the specularity to follow, ignore, or even boost the strength of the bump map applied to the surface. This independent control is especially useful when used on different layers. For example, wet pavement may have a bumpy broad, colored highlight. But the water on top is smooth. Two specular layers, one with 100% Respect Bump Map and one with 0%, would give your surface this kind of behavior. It's very cool!

4.9 Reflection

LightWave's built-in reflection raytracing has served it well, but has remained substantially unchanged since it was introduced ten years ago in LightWave 2.0. G2 adds a rather large number of additional reflection methods and abilities to LightWave's basic tools.

The G2 reflection options are controlled by the settings on the G2 Shader's "Reflection" page. The first three options basically produce modifications of strength or color of the reflections. Changes to these options are previewed in real time, since they use the rays that were fired when the preview was captured. The later controls dealing with blurry distributed reflection *do not update in the preview* because they require new rays to be sampled, which currently can't be done within LightWave Layout. These blurry reflection controls are one of the few G2 options that *don't* preview changes interactively.

The first of the basic options, "Backdrop/Map Strength," boosts (or reduces) the intensity of *background* reflection. These are the parts of the reflection that don't show a raytraced reflection of an object, but just show the default environment (or reflection map) instead. This is more useful than it seems at first because it allows you to balance the brightness of reflected objects versus reflected sky and ground. Quite often your background reflection of the sky is far too bright compared to the objects. This G2 option cuts that intensity down while leaving normal object reflections unchanged.

The next option, "Surface Color Tint," tints the reflections based on the surface color (or any other color you specify). Usually in LightWave, an object's reflections are uncolored, so a bar of gold would have untinted reflections. The Surface Color Tint option of G2 tints the reflection to the same shade as the surface color. If you use a 0% value, the surface color will be tinted by the shade you pick with the G2 color picker. Inbetween Tint values will mix the surface color with the color you pick.

Nonlinear Ray Reflections

Sometimes a reflection is washed out by the underlying surface brightness, making the reflection hard to see in some areas. A related problem occurs when you have raytraced reflections, but you don't have a full LightWave environment. Some of the reflected rays zoom off into the horizon and show up as plain black. That's not a problem except that it makes the rays that *do* hit objects very visible and exaggerates the contrast between areas that are showing something in their reflections and areas that aren't.

"Nonlinear Ray Reflection" is a simple tool for helping reduce both of these problems. It does not affect your reflections themselves. Instead, it affects the *rest* of your surface shading, reducing the brightness of the base surface whenever the reflection rays have hit something.

One example is a shiny floor surface that's reflecting a table. If you don't have an environment outside your scene, the reflection of the table will brighten the floor where you see the table, but the remainder of the floor will stay unchanged since it doesn't "see" anything to reflect. This makes the floor much brighter where you see the table, almost like that part of the floor is glowing. You can cut down this glow with Nonlinear Ray Reflection, which will reduce the shading of the surface in those areas to try to control the brightness.

Nonlinear Ray Reflection is also useful when dealing with composite shots. Imagine a project where you are putting a 3D duck onto the surface of a lake you have photographed. The lake itself may be modeled as a single flat polygon, and textured with LightWave's "Front Projection" mode to make it match the photo's pixels exactly. Because the photo doesn't need shading, you'd set the Luminous channel to 100%. Finally, you'd turn on LightWave Reflection in order to see the duck reflected in the water. All this works, except that the duck's reflection is *added* to the brightness of the water, and it becomes too bright. You want to dim the water's brightness only where the duck's reflected image is. This is done easily with Nonlinear Reflection just by turning it on and adjusting the strength interactively with the preview.

Blurry "Distributed" Reflections

The remaining reflection options control the direction and type of reflection, especially "blurry" reflections. Therefore changing them does not update G2's preview.

"Respect Bump Map" starts at 100%, but lower values can be used to make the reflections behave as if there were less (or no) bump mapping applied. This is very useful when you want to make a smooth reflective layer of water on a surface, but want the underlying material to appear bumpy like rock or some other material. This is similar to the way bump maps can be boosted for the Specular and Diffuse channels.

The remaining controls create *blurry* reflections by firing multiple rays instead of just one. This comes at a speed penalty (sometimes a substantial one!)

"Soft Reflection Spread" makes reflections become softer and hazy as opposed to the default mirror-like rays. Larger values create more blurring and "defocusing."

"Soft Reflection Quality" makes the reflections appear smoother and better, but at the expense of longer rendering. The quality can also be boosted for *all* surfaces with the Master panel's similar "Soft Reflection Quality" control.

"View Anisotropy" is an important real-world effect caused by the physics (and geometry) of light reflection. "Blurry" reflections are actually more like *streaks*, with a lot more blurring happening in the direction toward and away from your view. Think of a sunset over the ocean, with the low sun's reflection in the waves. The reflection of the sun forms a long vertical column of light. It does *not* show as a blurry disk. In fact there is nearly no sideways blurring, but enormous amounts of vertical smearing. G2's reflections also follow this behavior when using the "View Anisotropy" setting. (We call it View Anisotropy because there's no clear accepted term for this effect, every artist and researcher seems to call it something different!) The default of 50% is calibrated to match real world physics, but you can intensify (or weaken) the effect manually.

Plain "Anisotropy" also causes "streaks" in the blurred reflection, but these are customizable. These blurs might be caused by materials like brushed stainless steel, which has tiny scratches or grooves that make light reflect from them in one direction preferentially. This is exactly the same as Specular Anisotropy as described on page~ t{anisotropy}. The direction of the anisotropy is set with "Anisotropy Angle."

4.10 Anisotropy

G2 uses anisotropy in both reflection and specularity. Anisotropy{This is a common geeky word in computer graphics. Another is *isotropic* which is pretty useful, it means "the same in all directions." You can use this to describe a situation where direction doesn't matter. It's sort of like standing in the middle of a giant meadow of grass. No matter which direction you look, there's just more grass. The field is *isotropic*. But if you stand in a cornfield, there are clear, artificial directions caused by the planted rows. That's *anisotropic*, meaning having a preferred direction. To be honest, farmers don't really use this word much, it's kind of limited to physics and computer graphics geeks. I'm both types. Maybe I should have been a farmer instead. They make pretty good money. And it must be fun driving the giant tractor. Fresh air and sunshine. All the corn you can eat! Hum.. Ah, sorry, you can go back to the manual now, I apologize for wasting your time.}, a fancy name for describing the effect of a material having a "preferred" direction to reflect in. This preference is usually due to fine parallel grooves or scratches, which have the effect of spreading out reflection in just one direction. This can be clearly seen in many objects like wheel rims and hubcaps, stainless steel counters, and many fixtures such as doorknobs and railings. It can also be seen in materials which are made up of many parallel threads such as smoothed hair or fur.

G2 defines these preferred directions in the shader's "Extra" page. The definition is straightforward, with the scratches following the X, Y, or Z axis. The *type* of curve the scratches follow is either "Linear," "Cylindrical," or "Radial." Linear is the simplest, it's just a straight line. Think of a clean piece of smooth wood. Its wood fibers are anisotropic. If the wood is straight and smooth (not some tangled wood from a stump), the wood grain has linear anisotropy.

Cylindrical anisotropy is also common, where the scratches rotate around in a circle, like on a hubcap, wheel rim, or even a cooking pan.

Finally Radial anisotropy is the least common, and occurs when scratches all run away from a central point. I admit I cannot think of any good examples but surely they must exist. Crop circles, seen from the air?

The specular and reflection pages both allow you to *rotate* the anisotropy direction, making the highlights appear at an angle. For specularity, each specular layer can have its own angle rotation. Often cloth has *double* anisotropy with two specular layers at right angles. This comes from the two sets of woven threads, which are also at right angles. Silk, satin, and other shiny fabrics will have this kind of specularity.

4.11 Transparency

The sparse Transparency page allows you to control the strength of bump maps used for surface refraction with the "Respect Bump Map" control, similar to the ones used for diffuse, specular, and reflection.

The more interesting controls deal with "back-sided" polygons. Normally LightWave objects use only one side of each polygon, meaning that if you view them from one direction, they're solid, but viewed from the other direction, they're invisible. The reasoning for this is mostly efficiency, since back in the Good Old Amiga Days, it took time for every polygon to render. But the polygons on the *back* side of your object were never seen. So if LightWave skipped rendering them, you'd never notice yet rendering would be faster.

This same idea is used today. Computers are so fast that we don't notice the raw polygon rendering speed of LightWave any longer, but video cards are trying to render at hundreds of frames per second, and they can use the same "backside" trick to get faster rendering speed too.

Coming back to LightWave, these one-sided polygons have disadvantages, especially when dealing with transparent objects. In particular, you may really want to have the back-sided polygons render because of *refraction*. Rays that enter into a crystal ball bend when they enter and bend when they leave. But if the polygons on the back side are ignored, LightWave doesn't know that it should bend the ray on the way out. The net result is a crystal ball that looks wrong. But double-sided polygons don't work either since you can't assign different properties to each side of the polygon.

The proper solution to this problem is surprisingly complex. Newtek recommends that you copy every polygon of your glass object, paste them back onto the object and flip their direction. These new, doubled, flipped polygons are now called an "air" layer which is fully transparent and has an index of refraction of 1.0. This is really the only way to make proper glass objects in LightWave.

G2 makes this cumbersome step easier. Instead of modeling objects with special flipped layers, you can simply turn on LightWave's "Double Sided" option in the Surface Editor. While LightWave can't assign different properties for the back side of the polygon, G2 can.

The G2 "Zero Attributes of Backsided Polys" option turns off the backside of polygons to a featureless "invisible" appearance. Diffuse, Specularity, luminosity, reflection are all set to 0%. *Transparency* is set to 100%, and index of refraction is left unchanged. Similarly, the "Set Refraction Index of Backsided Polys" allows you to define the index of refraction of the backside, which you usually want to leave as 1.0.

This G2 option makes modeling glass easier. Instead of the Modeler steps, just make your surface Double Sided in the LightWave Surface Editor and turn on the two G2 backsided polygon options.

Your scene will have fewer polygons in it (since you don't need the extra copy of the polygons) and it's easier to set up. Unfortunately, LightWave isn't particularly efficient at raytracing doublesided polygons, so rendering speed is not improved. So the G2 backside option is more of a convenience than an amazing solution. However, if future LightWaves improve the performance of double sided polygons, this may turn into a superior method in speed as well as convenience.

4.12 Art Mode

The biggest advantage of G2's design is that it forms a framework for adding new kinds of effects to LightWave, automatically using the powerful preview system. This let us add and experiment with new shading styles, light types, and rendering methods.

G2's "Art Mode" is one of our "cool experiments" which allows a way of shading objects in a very different way than normal. We're used to thinking of the appearance of a surface as depending on light positions, shadows, diffuse surface properties, reflections, and so on. But why? Well, because it works and it simulates the way light acts in the real world.

But as an artist, there's an argument that you may want to step away from the physics of light and the dozens of surface parameters we use to define things like specularity and shading. As an artist, you might say "I know how I want it to look. I could even draw it. But how can I make my object appear that way?" G2's "Art Mode" lets you do exactly that, it allows you to *draw* your surface appearance and G2 will create your 3D objects which have that same "look."

This has a strong similarity to a method used in classical pen/charcoal/paint freehand art. Especially in schools, students are taught to preplan the appearance of their lighting and surfaces. The common planning technique is to make a sketch of a simple sphere with the same lighting and appearance you want on your real figure. This gives you a reference to compare to when you're trying to keep your light and shadow angles, highlight colors and midtones, and even surface texture consistent. Different artists call this reference sketch different names, but we use the term "Study Sphere" from one of our art instruction books.

The very simple idea of Art Mode is just like this traditional technique. You want to make your complex object (which could be anything, a boulder, statue, or giant robot), have different colors, lighting, and shading based on same areas (directions) on the study sphere image. If your study sphere shows a soft reddish tint on the top right, your complex object should also have a soft reddish tint on the parts that are facing up and right. The surface color and lighting and shadowing (everything!) is defined by the way your study sphere appears. This works as a guide in traditional art, but is enough to use for generating the full appearance in G2!

Notice that there is no need for defining lights, surface parameters, gradient distances, diffuse values, or even surface color or texture. It's a lot of fun to be brought back into a visual, not mathematical, way of making your object appear the way you like.

To use Art Mode in G2, you'll need a study sphere image. This is *just an image of a sphere*. The image can come from anywhere! It's often hand-drawn, but it can be processed in Photoshop, cropped from a digital photo, even originally rendered in LightWave and hand edited. And often you'll mix styles, for example by photographing a stone sphere in your garden but processing it in Photoshop to rebalance the colors, and airbrushing some changed details. Crop the image to center the sphere in the picture, and load it into LightWave.

On your 3D object's surface, turn on G2's "Shade with Lightless Art Mode" button in the Art Mode tab in the shader panel. Pick your study sphere image with the image picker. That's all you need to do! G2's preview shows Art Mode's effects immediately.

You can tweak two configuration options, allowing you to rotate and zoom into the study sphere image. This can be useful for tweaking the alignment of the effects on your object.

Art Mode is *fun* to play with. All of the normal LightWave ideas of lighting, shadows, and even surface texture are replaced with the simple study sphere idea. This is especially useful for making more organic, hand-drawn rendered images, with different styles like oil paint, dithered shading, custom anime "cel shading," ink and paint hatched shading, just about anything. Usually you won't use Art Mode for photoreal effects, since that's best handled by the "normal" idea of shading and lights and shadows. But as the name implies, it's a unique tool for *artistic* effects and new styles of images.

There aren't many rules for using Art Mode since you can experiment yourself and find new limits that even we haven't thought of. The effects of Art Mode work best on smoothly curved objects. Sharp bends and especially flat areas on your 3D object become boring since they are all oriented the same way and therefore become colored the same way. This tends to make unpleasant streaks or solid areas. So think of using Art Mode with statues, monsters, or organic shapes. Level landscapes, straight pipes, a flat desktop, and a big

city skyscraper won't turn out very well.

Animation with art mode is probably not very useful, since the appearance of your study sphere is based on one camera view. You're welcome to experiment but camera motion in particular looks strange since the lighting seems to "follow" the camera instead of being fixed in space.

For the study sphere images, we find that higher resolution images (500 pixels, say) work better, since there's enough detail in the output surface. If the study sphere image is low resolution, you'll see the effect of the original image's pixels in your final rendering. Similarly, try to avoid super-sharp details in the study sphere image. Soft and blurry is more effective than lots of buzzy, single isolated pixels.

Finally, the *rim* of the study sphere is especially useful to adjust. This rim becomes the rim of your 3D object too, so you can use it to customize the edge appearance. It's useful to draw an extra-wide rim around your study sphere, and then use G2's "zoom" control to interactively scale the study sphere to include more or less of that drawn rim to get the kind of appearance you like. A rim isn't a requirement, but for many non-photoreal appearances, it tends to define your object's contours and lets it stand out from the background.

4.13 Extra Controls

Several G2 controls are a bit lonely, stuck on the final "Extra" page of the G2 Shader panel. But despite their isolated location, they can be quite useful!

"User Comment" is is totally for your own use, perhaps for writing brief notes or names. Most importantly, your text will appear on the LightWave Surface Editor, in the description line for G2. This may be useful to label special surfaces with your own tag or note so you can recognize them later.

Anisotropy definition is stuck on the Extras page, but it at least has the honor of its own section of the manual [here](#).

"Rim Lighting" is a powerful option. {It deserves more than being exiled to the Extras page. It's cool! But it doesn't fit anywhere else.} In the real world, a light shining from *behind* an object will often create a thin bright edge. This rim lighting is often used by portrait photographers to highlight a person's profile.

G2's "Rim Lighting" increases the size of this lit rim around the outside of your object. This gives you even more control than a real life photographer, because the 3D object's rim can be extended as far as you like.

One problem with rim lighting is *shadows* because normally an object will shadow its own back side. You can work around the shadowing difficulty by using a different shadow opacity (on the G2 Boost page), or simply turn off the object's Self Shadowing in LightWave's Object Properties panel.

The final G2 option on the Extras page is "Terminator Sharpness." This increases the "hardness" of G2's shading transition between the front and back side of the object. This is similar to LightWave's Diffuse Sharpness, except G2's also affects specularly.

Chapter 5 Master Panel

G2's Master panel contains options that usually affect your entire scene globally, such as image processing. These settings are saved in your LightWave *scene* file, so different scenes each have their own independent values. This is unlike the G2 shader plugins, which are saved in object files, and therefore may affect more than one scene.

5.1 Image Processing

It's common to adjust the brightness or contrast of images to improve them. This is just as true for vacation snapshots of you standing in front of the world's largest cheese as it is for adjusting your masterpiece 3D render to look its best on your web page. It's completely acceptable to adjust your LightWave renders in Photoshop (or other programs) to improve them after rendering.

Most of these image adjustments are simple color and brightness changes. G2 adds the ability to apply these fundamental enhancements within LightWave itself. This actually gives you improved quality (because it works on high quality HDR colors, not low quality 24 bit), and is more convenient, since you don't need to launch Photoshop (or whatever program you use) to make the adjustments. G2 of course also allows you to view these adjustments in realtime with its preview system.

In G2's "Image Process" Master panel, you can adjust the fundamental "Gamma" and "Bright" of your output. Gamma is a kind of contrast control, increasing the differences between shades. Larger contrast will make dark colors darker and light colors lighter. Brightness makes all colors brighter or darker.

You can apply these adjustments to each RGB channel of your image independently, which is especially useful for matching the color tone of other images. In most occasions, you'll use only the *white* Bright and Gamma controls that affect all of the RGB channels equally, preserving your color hues.

Distance Desaturation

The Image Process panel also provides a special desaturation effect that can help make foreground objects stand out from the background by subtly (or not so subtly) removing color from distant parts of the image. This tends to make the objects near the camera stand out and feel "alive" compared to the blander background. This is somewhat similar to a kind of "fog" that sucks color away from surfaces instead of obscuring them.

"Distance Desaturation" sets the strength of this effect. At 100, all of the color is removed from distant objects, turning them into a black and white version of themselves. Smaller values will leave some of the original color so the background isn't totally colorless.

The amount of de-coloring (desaturation) is based on distance, similar to LightWave's fog. The "Clip Distance" is a minimum distance (from the camera) where no color is removed at all. After this minimum distance, color starts to fade (like fog) based on the "Falloff Distance." Smaller distances cause the color to fade off faster.

You can make specific surfaces immune to the color desaturation by setting the "Follow Master Desaturation" control on the Shader's Processing page to 0. This could be an artistic way to have your main character render in full color while the background remains black and white. With fractional strengths, it could be applied as a subtle effect.

5.2 Gamut Control

It's very common to have bright lights and surfaces in scenes. Unfortunately it's also common to make effects so bright that some pixels become too intense for your monitor to display. We're all so used to this that sometimes we forget that colors *can* be outside the classic range from 0 to 255. LightWave began storing these extended intensities with more precision and range in version 6, so it's now quite possible to handle brightnesses without any upper limit. The terms "floating-point color" and "High Definition Rendering" (HDR) describe this extended color range (and also the fact that the colors have more "digits" of precision.) You may think of a screen color as being from 0 to 255 using "old style" integer brightnesses or 0.0 to 1.0 using "new style" floating point colors.

But these extended ranges don't change your video card or monitor; They're still very limited by the maximum brightness they can show. So what should be done with pixels that are too bright? By far the most common solution is to just clip those intensities to the maximum that can be displayed. But this *sucks!* If we clip any intensity beyond 1.0 to be 1.0, a pixel of 2.0 brightness looks just like a pixel of 1.0 brightness. We've lost all feeling that one bright pixel is any different than another bright pixel. But even worse, we start losing *colors*. Imagine a deep red color with RGB (0.4, 0.1, 0.0) and make it ten times brighter, to (4.0, 1.0, 0.0). When this is displayed, it is clipped to (1.0, 1.0, 0.0) which is pure yellow! So even our color hues become mangled, not just our brightness. It simply sucks.

Unfortunately LightWave doesn't provide any solution to this problem except for exporting the render as a floating point image and adjusting it in some other program which has algorithms for managing the clipping. But even those programs are rare; I will bet only one in a hundred LightWave users have ever even tried to do this. {You professional studio guys with inhouse compositing systems can feel smug and superior now. But *only* if you export in full floating point!}

There is no perfect solution to this problem because in practice your final output *will* be clipped. (Even if you use a fancy output device like a film recorder that has a much higher dynamic range than a video monitor, there's still some limited range of displayable intensities.) So you need a tool that does this clipping intelligently and controllably. G2's "Gamut Control" tools allow you to do this from within LightWave.

G2's gamut{A "Gamut" is the range of colors that can be displayed by a screen or printer. This is used especially by professional color printers because some color shades are hard to reproduce. For 3D rendering we are usually concerned with brightness limits.} controls are an implementation of a standard solution to controlling the way clipping is applied. The basic idea is that instead of making a sharp cutoff at the maximum intensity, the transition is softened to become a smooth curve. Dim pixels are left at their original intensities, but bright pixels are smoothly "massaged" to display within the allowable range. G2 allows a wider range of intensities to be shown in the limited output range, and also avoids a sharp transition at the maximum brightness.

G2's controls

G2's gamut controls are found in the Master panel on their own page. They are straightforward to use since the G2 preview shows its effects immediately. Activate the gamut feature by turning on G2's "Gamut Rolloff" button. Intensities less than the G2 "Starting Intensity" limit won't be modified at all. A lower starting limit will cause more pixels to be modified (which is bad), but will give you more intensity "range" to squeeze the over-bright pixels into (which is good). The Starting Intensity lets you find a compromise between high accuracy of dim pixels and more range to display over-bright pixels without clipping them.

The "Rolloff Rate" defines just how those bright pixels are squeezed to fit into the remaining range. Larger values will allow even brighter maximum values to be smoothly displayed (good!). A value that's too high will waste some of your available output range (bad!).

Luckily setting the values is not too difficult since in practice you find good settings by eye using G2's preview. The preview's [Info window](#) also helps since it can show you the raw numerical full-range color values as well as the final clipped color of any pixel.

The unequal intensities of a clipped color cause the final color hue to change. G2's "Hue Preservation" control will allow you to keep that hue in your final output color even *after* clipping. However, this power, too, is limiting, since if you force the hue to remain unchanged, you reduce the available range to map bright pixels into. Like the other controls, you can find a good compromise between color hue accuracy and brightness accuracy by adjusting this value. The default of 50 works well for most situations.

5.3 Film Grain

Images recorded on film often contain very small miscolored dots, almost like dirt or dust. These spots are actually caused by clumps of chemicals or dye in the film itself. Different types of film have different sized clumps. Silver-based films have larger clumps than dye based film. "Faster" film, especially that used for movie film recording, has much more grain than the slower film used for still photography. The dirty artifacts these clumps cause are called "film grain."

Coincidentally, video cameras also have a similar kind of subtle "dirty" corruption due to the way their electronics record images, especially in low light. While this is not true "film grain" it looks and behaves quite similarly.

G2 can add artificial film grain effects to your rendered imagery. It may seem wrong to deliberately add noise to a rendered image, but this can help the image appear like it was filmed, and it can add a subtle but convincing realistic touch.

G2's Film Grain controls are located on the G2 Master panel. When you activate Film Grain, the G2 previews will immediately show the effect. Zooming into the preview can give you a closer view of the grain pattern if you wish. You can remove the grain effect from a preview by turning off the preview's "Img Pro" Output button to disable the preview's image processing.

By default, the grain pattern changes with each frame, since real film also has random grain on each frame. If you want the grain to remain fixed in place, you can uncheck the "Animated" button.

"Scale (pixels)" sets the size of the grain. Real grain is often very, very small, but you can adjust the size to match the behavior you wish. (The preview can help you decide the right size.) The scale doesn't need to be much smaller than a pixel, since grain smaller than that would start to become invisible. You can also make the grain automatically get bigger with higher pixel resolutions by turning on the "Auto Scale" button. This is useful for keeping the appearance of the grain consistent for different resolution tests. When Auto Scale is on, the grain size is based on a "natural" image width of 768 pixels. So if you used a Scale of 1.0 pixels, and rendered at $768 \times 2 = 1536$ pixels wide, the grain would become 2.0 pixels wide to compensate.

Different real film stocks have different grain effects. In real life, grain (and video noise) affects each color channel independently. Also, many grains become more visible when the underlying image becomes brighter. G2 allows you to set the "base" grain intensity for each RGB channel, which is how bright the grain is even in the black areas of the image. You can also add a *proportional* amount to this base level, as a percentage of the image brightness, causing grain to get brighter in bright image areas.

Real film grain tends to become more visible with intensity, but video noise tends to stay constant regardless of the displayed intensity. Obviously you can use any proportions you like to create the effect you wish.

There is also a "White" grain channel which is less physically realistic but might be useful for providing white colored variations, like dust or TV static.

Real films and video also share a common behavior in that very bright image areas tend to override and hide the grain effect. This is important especially for film, since the grain becomes stronger with brighter images, but then after some intensity limit the grain will start to *decrease* in visibility and effectively disappear. You can simulate this effect by turning on the "Saturated grain effect" button and specifying a starting "Intensity" and "Transition Width."

In practice, Film Grain is surprisingly effective at applying a subtle, attractive feel to your renders. But the key to proper grain effects is the word *subtle*. If you have any doubt about how much to use, use *less*. Still images can use quite strong grain without looking out of place, but animations should use less to prevent too noticeable (but realistic!) "buzzing."

Since grain is really a final step in output, it's useful to leave Film Grain turned off until you're adjusting its appearance or making your final render. It may be a bit too distracting while you're doing early lighting and surface adjustments.

5.4 Master Lighting

G2's Master panel provides a diverse set of "extra" lighting abilities for scenes. The "Master Lighting" page in the Master panel provides six controls for tweaking these global options.

The first is "Global Light Intensity" which mimics LightWave's control of the same name. Due to an oversight in LightWave's design, the LightWave control does not work with plugins, so G2 provides its own. This allows you to brighten or dim the effect of all lights in the scene simultaneously.

LightWave lights often use the realistic inverse-squared Falloff type. This LightWave feature is very powerful since lights in the real world all follow this behavior. However inverse-square lights can be troublesome to use because the lights often get *too* bright on surfaces very, very near the light. I mean really

really bright, a thousand times too bright! This effect is *also* natural but we usually don't see it because real world lights usually have area. We even add lampshades or light fixtures to soften the intensity close to the light bulb. And even a bare light bulb is "frosted" to even out the light intensity close to it. In LightWave, a point light doesn't have these protections so it's easy to get harsh super-bright lighting.

G2 can help tame this ugly ultra-bright lighting by limiting the brightness in those close-in areas while leaving distant illumination unchanged. G2's "Intensity Falloff Protection Distance" specifies the range (in meters) that has this smooth intensity limit applied. This does not affect lights with no LightWave falloff, only those with inverse or inverse-squared falloff.

A good strategy is to light your scene using the light intensities to give good lighting of *distant* areas. Then, when you're happy with the appearance, increase the G2 protection distance to "tame" the light intensity close-in. Since distant lighting remains unchanged, you only have to worry about the close areas. And, of course, you have the preview to adjust these settings interactively, so finding the right values is not difficult.

The next Master lighting option is "Shadowmap Transparency Support." This produces a kind of hybrid between a shadowmap and raytracing. It allows lights using LightWave shadowmaps to cast proper shadows from transparent and dissolved objects. The "softness" of shadowmaps is also removed (which can be both good and bad). Rendering speed is slower than shadowmaps alone, but usually faster than pure raytracing.

In a similar way, the "Trace Transparent Area Shadows" option controls G2's area light shadow computation. For efficiency, default area light shadows in G2 do not respect transparency. If you do want transparent objects to cast partial shadows (instead of fully opaque ones) you can enable this option and get the proper effect. The only disadvantage of this G2 option is slower rendering.

Moving to the next G2 options, the "Tint Ambient" color controls allow you to make LightWave's ambient light color change with angle. This can be very effective, especially for outdoor scenes. You could tint the top (Zenith) parts of your object bright white-yellow, but leave the underside (Nadir) a dimmer blue-grey. This is a lot like a "poor man's radiosity" effect, since you can increase global lighting realism, but with no speed penalty. {You can also tint ambient light per surface with the "Tint Ambient" control hidden on the G2 Shader panel's Lighting page.}

The final Master lighting control is "Global Illumination Contrast" which adjusts the radiosity and caustics effects applied to your scene. This change of contrast is very useful for tweaking the final radiosity appearance, especially for creating the proper balance between dark corners and fully lit areas. You'll often adjust this setting in conjunction with the Global Illumination boost controls to get the best overall settings.

5.5 Light Groups

It's not hard to manage objects and lights in a small scene. But some projects grow in complexity and you may start dealing with hundreds of lights at once. You may have 50 streetlights lining a road. Or 20 cars, each with its own two headlights and two taillights.

When you have huge numbers of lights like this, it's very difficult to make changes to many of them at once. LightWave Spreadsheet can help with many of these kind of bookkeeping changes, but these light adjustments are particularly common and it's still slow to use and edit the values in Spreadsheet.

G2's Light Group feature helps solve this common problem. The basic idea is simple. You choose a set of lights and define them as a group. Later, you can boost or dim the brightness of all of those lights simultaneously without affecting the other lights in the scene.

Light Groups are defined and controlled from G2's Master panel. The Light Group page allows you to switch between 8 separate groups by using the numbered buttons along the top of the page.

To define a new group, use the large item picker button (By default it says "None" since it has no lights picked.) The item picker allows you to pick any number of lights at once. Alternatively, you can pick one or more lights in Layout either directly with the mouse, or with the Scene Editor. After picking the lights, go to the G2 page and press the "Set" button and those lights will automatically define the new G2 Light Group. You'll see a small marker appear above the group's button to help show you that the group is now defined. You can type in your own text label to help you remember what the group is. A short note to yourself like "car interior" will help you a lot more later than the simple list of light names.

Once you've defined the Light Group, you can boost the effect of those lights on G2 surfaces with the "Boost" control. The effect shows up immediately in the G2 preview.

Each G2 surface can *independently* boost the effect of a group for just that surface. The G2 Shader panel lists the 8 light groups in its Lighting page. You can also instruct the surface to *ignore* light groups to make an exception for a surface. These settings are saved in your *surface* data. This may cause a problem if you want to use the same object in multiple scenes with different lighting setups. But G2 has object groups to deal with that situation too.

5.6 Object Groups, Too

Each light group has its own set of special *object* groups. Like light groups, an object group is just a set of objects. If you define an object group, you can then choose special intensity and shadow boosts that will apply only to the combination of lights and objects defined in the groups.

That's a little confusing, so here's an example. You have two foreground lights that shine on your main character. The lights look good on your character, but his motorcycle isn't quite bright enough. You want the motorcycle to become brighter, but *not* the character.

You can solve this by making a new G2 Light Group with your two foreground lights, and then pick the motorcycle as an object group. Then you boost the intensity of those lights *only* on the motorcycle, without affecting the character or anything else.

You can define up to 8 object groups, and each group has its own special boost. You can also boost shadow opacity for those lights and objects, which works in the same way the Boost page's [Shadow Opacity](#) behaves.

The original global light group boosts still work even when you also have an object group. This can be used to boost intensities for all objects *except* for an object group, by using the normal light group boost and then using the "inverse" value for the object group. For example, to brighten the effect of a light group on all objects *except* one special one, you would use a 200 Light Group boost, and compensate for it by using a 50 boost for the object. The net effect is to make the lights' strength double for all objects except the special one.

The Light Group features are very useful for studios and large projects, because it provides a way to tweak lighting for a specific object in a scene *without* changing the object file itself. This is really important for projects that use one object in multiple shots. If you changed the object's surface properties to brighten it for one scene, you would affect its appearance in *other* scenes as well, which is asking for trouble. G2 Light Group information is stored *per scene* so the special boosts don't affect other scenes.

Light Groups are *also* useful for the G2 preview, since the defined groups can be used to help choose which lights are shown in the preview window, as described [here](#).

5.7 Quality

The Master panel has several global quality settings on the Quality page.

"Area Light Shading Quality" controls the accuracy of shading from area lights. When an area light shines on a very glossy surface, or especially when you have a very large area light, the shading may show artifacts if the quality is too low. Higher values slow rendering speed. This quality setting *does* update in the G2 preview so you can easily judge when the quality is high enough.

Similarly "Area Light Shadow Quality" defines the quality of the *shadows* from an area light. These shadows are normally "noisy" since, like LightWave, G2 uses ray tracing to estimate the softened intensity caused by an object partially occluding the light. Higher quality values will cause G2 to use more rays to determine the shadow, making the shadow smoother and more accurate. This option does *not* update in the preview since it's used at render time to fire the shadow sample rays.

"Soft Reflection Quality" also affects G2's raytracing. In this case, it determines how many rays are used for soft reflection tracing, as described [here](#).

5.8 Preview

The Master panel's final page helps manage G2 preview windows. You can create a new preview with the Add New Preview button (though almost always it's more convenient to use the G2 Generic, "G2:Add New Preview", as described [here](#)).

"Refresh Previews" is also rarely used, since the preview windows almost always keep themselves properly updated.

The most useful controls are the display at the bottom which shows how much RAM the previews are using. This can get quite large if you use the cool animated "scrubbable" previews as described [here](#). You can manually force the previews to release their stored data with the "Free Previews" button.

Chapter 6 FAQ

A lot of problems you'll encounter are simple mistakes or misunderstandings. G2 is a very large, very complex system, and it takes time to become proficient with every part of it. From experience, we know there are extremely common questions and problems that everyone has, and we try to list them here. No matter how experienced you are, you may want to run through this list if you have a problem. Even *we* use this list! {It's embarrassing to waste 30 minutes thinking there's a G2 bug when the answer is in the FAQ.} It's a very very good idea to read this list completely even if you don't have any problems, because it will at least alert you to limitations, errors, and techniques you should know about. *This is the most important chapter in the manual to study.*

Question: Why doesn't my G2 Preview window update properly when I change something?

There are several possibilities. The most common one is that the part that isn't changing is not a G2 surface. The easiest way to check is to open the Info window and move the mouse over the area that is not updating. The Info window will explicitly print "G2 Surface" or "Not a G2 Surface." The G2 shader must be applied to the surface in order to update in the preview.

The preview might be showing some LightWave effect in *front* of the G2 surface. This might even be invisible, like a light fog, but it's enough for G2 to "lose control" over the pixel. Turn off the "LW Effects" button in the preview to see the underlying objects.

You may have restricted the preview to show only certain surface channels or lights, but your changes are to *different* channels or lights.

Are you really looking at the G2 preview window? No joke, sometimes even we look at the LightWave render output window, thinking it's the interactive G2 preview by mistake.

If you're moving, rotating, or scaling a light, LightWave doesn't record those changes unless you have "Dynamic Update" turned on in LightWave's Display Options. We also recommend using LightWave's "Auto Key" mode; otherwise you need to record your changes by setting a keyframe.

Question: I've cranked up the G2 luminosity (or any other channel) boost, but there's no change even when I render.

Boosts are multipliers of a base setting. If the attribute has a 0 intensity, doubling or tripling it still gives 0.

Question: I'm using specular anisotropy. My object has a weird line in the preview, but it renders fine.

This is an artifact in G2's preview that only occurs when using anisotropy. It's a compromise to save preview RAM. Storing the full anisotropic information would increase memory use significantly (perhaps 20MB extra!), so G2 uses a much more efficient method (which unfortunately has the one annoying artifact.)

Question: When I change some setting, I get a strange colored rim around some areas. It's OK when I render, and then the new preview is OK too.

These are probably edges of postprocessed effects like a lens flare or Sasquatch fibers. As a postprocess, these pixels don't update interactively, so they continue showing the originally rendered color. Rendering at higher resolution will shrink the rim of pixels, reducing the problem.

Question: I used Edge effects to make a stronger Specular value on my edges but I don't get a specular highlight there!

Specularity edge boost doesn't move a highlight to the edge, it makes highlights that are *at* the edge *brighter*. To *move* a highlight towards the edge, try using G2's specular "Retro Reflection."

Question: LightWave crashed when I used LightWave's Image Editor or an image sequence, with the G2 preview open.

This is uncommon but unfortunately possible. A LightWave bug can cause a crash when a plugin (like G2!) asks about an image color, but LightWave is busy using the image for its own purposes. Hopefully this will be fixed in future LightWave updates.

Question: How is G2 different when I use it with LightWave 6.5?}

G2 won't react to light distance falloff or LightWave light exclusion, since these were plugin features added to LightWave 7.

Question: How do I completely remove G2 from my scene?

Since G2 tries to add the Master and Pixel G2 plugins for convenience, the order you remove them is important.

Start by closing all G2 panels. Remove G2 from all your object surfaces by picking all your objects. (LightWave's Scene Editor lets you select all objects easily.) Next use the Generic plugin "G2:Remove from Objects" to strip off all of the G2 Shaders.

Next, visit LightWave's Master Plug-ins panel, and remove the G2 Master plugin. Finally, visit LightWave's Effects/Image Processing panel, and remove the G2 Pixel Filter plugin.

Question: LightWave's Shading Noise Reduction doesn't work with G2!

Shading Noise reduction is a useful feature in LightWave that applies a smart blur to the image. Unfortunately LightWave uses its own internal buffers for this effect, which are not available to plugins, including G2.

Question: Motion blur doesn't show up in previews.

G2 only stores the first rendering pass, so motion blur doesn't show up. Extra antialiasing passes are also ignored.

Question: Dithered motion blur leaves strange artifacts in the G2 preview!

LightWave doesn't inform plugins about the dithering, so G2 just decides LightWave is insane. But there's no need to render with Antialiasing or motion blur to use the G2 preview anyway.

Question: I'm running out of memory!

G2's previews can use a lot of RAM. The Master panel displays exactly how much is being used by the previews. Most computers will have no problems, *except* if you try to store an *animated* preview as described on page~\pageref{scrub_preview}. A sequence can use hundreds of megabytes.

The rest of G2 uses memory too (especially for subsurface scattering) but that use is usually very small compared to the preview's RAM.

Question: Volumetric lights, lens flares, Hypervoxels, and image filters either don't show up or don't update in the preview.

G2 can't make LightWave recompute these, so it just displays the effect from the last F9 render. Turn off the preview's "LW Effects" button to see the G2 surfaces underneath.

Question: In LightWave's Surface Editor "Edit by Scene" doesn't work. I can't change all the G2 surfaces at once.

LightWave 7 doesn't support changing settings for more than one surface at once, though the interface design implies that it can.

Question: I have two surfaces with the same name on two different objects. But I can't make them different.

This is part of LightWave's surface design. Unfortunately if you have cloned objects, or multi-layer objects, you can't give independent settings to the surfaces if they share the same name. This is true even without G2.

Question: G2 renders with strange shadow artifacts.

Most of G2's shadows are calculated right with LightWave's engine. Try the object without G2 and you'll probably see similar problems. Often this is caused by nonplanar polygons. Try triangularizing your object, Also merging points and unifying polygons can help fix damaged models. LightWave also has frequent problems with shadowing concave areas of object surfaces.

Question: Sometimes there are artifacts in my G2 preview that aren't in the render, especially around edges of objects or shadows.

This may be because of G2's subpixel interpolation. You can check by turning on the "Show Pixels" option of the preview in its Tools page.

Question: I've aborted my LightWave render and all I see is a red screen in G2's preview. Why can't I at least see the partial render?

Because LightWave never got a chance to tell G2 everything it needed about the scene. But if you click off "LW Effects" in the preview, you'll at least see the G2 surfaces that did render.

Question: I click on my model's face in the preview to edit her skin surface. But the wrong G2 panel opens!

G2 will open the panel of the G2 surface you click on. But that surface may be *transparent*, like a car's windshield. You may not see any effect from this transparent surface, so you may not even notice or remember the surface is even there. You can open any surface manually by using the G2 shader's "Next Surface" button repeatedly, or finding the surface in LightWave's Surface Editor and opening the G2 shader from there.

Question: Why don't I see anything happen when I adjust the Transparency Absorption under Scattering, even to 500%?

If your surface isn't at least partially transparent to begin with, you'll never see extra absorption being added.

Question: What's the difference between Gamma and Gamut?

Gamma is a kind of brightness contrast control. Gamut is a range of allowable brightnesses. They're unrelated but unfortunately *sound* similar.

Question: How is the G2 Area Light Quality related to LightWave's? Is a 40% G2 Area Light Quality equal to a LightWave Area Light Quality of 4?

There's no direct correspondence between the settings because the algorithms are different. G2 also splits area shading quality to be independent of area shadow quality.

Question: My scene uses a lot of polygons. Will that cause problems with the G2 preview?

Not at all. G2's preview has no speed or RAM penalty for scenes that use many polygons, since the preview takes a snapshot of pixels, not polygons.

Question: I'm using subsurface scattering and I'm getting weird bright spots or artifacts.

Subsurface scattering depends on LightWave raytracing as part of its rendering. LightWave raytracing is sensitive to non-planar polygons, and doubled points or polygons. Also, if your object has holes, they can cause surprising effects, like an eyesocket leaking a glowing beam of light through an object's interior.

Question: I'm boosting light intensities with Light Groups. But my lens flares and volumetric effects don't get brighter.

G2 light boosts only affect G2 surfaces.

Question: I've been given only two days to finish the lighting of a complex scene. I used G2's preview, so I finished everything in three hours. How should I use the rest of my time?

Most computer games are fun, but some clearly stand out as being great. For excitement, the popular *CounterStrike* mod for *Half-Life* is hard to beat. Some people prefer the straight out battle of *Unreal Tournament*, though. If you prefer a slower pace, *Civilization~III* is a classic that will easily fill any extra hours. {The latest nVidia GeForce cards can be justified to your boss, your wife, and are even tax deductible. It speeds up LightWave's interface; games are purely incidental, right? }

Question: When I try to open the G2 pixel panel, the Master panel opens instead.

There are no user controls or settings in G2's pixel filter. As a convenience, G2 will open the Master panel interface instead. The G2 pixel filter is used for applying the G2 image processing effects as well as capturing extra information that the G2 preview uses.

Question: I saved my scene but none of the G2 changes I made were there when I reloaded it.

G2 Shader settings are stored in your object files. You need to save your objects after editing them.

Question: I'm using Light Exclusion and some lights and shadows are behaving *really* oddly.

G2 allows light exclusion to be moderated to partial strength, like 50%. But if you change G2's "Respect Light Exclusion" to be *more* than 100%, then G2 will in fact start treating that light as not just dim, not just off, but actually as *negative*. This is unlikely to happen by accident, since you have to manually override the parameter's normal limits of 0 to 100%, but the power is available if you need it. But beware, it is easy to get confused once you go outside the normal limits. {This happened to us with a beta tester, and it took *us* a long time to understand what was happening, since the negative lights were making bright shadows and darkening the rest of the world.}

Question: I have Gamut Rolloff turned on, and set the Starting Intensity to 100%. The graph properly shows that there's no smooth rolloff. But my image is still being affected!

The Hue Preservation control always has an effect when Rolloff is on, regardless of Starting Intensity. Turn Hue Preservation to 0% to disable it too. (Or just turn off Gamut Rolloff.)

Question: I have multiple plugin shaders. What order should I put them in?

LightWave calls all surface plugins "shaders" but some are really texturing tools (which change surface attributes), and some are true shaders (which turn attributes into a final screen RGB color.) G2 is a shader. Some other plugins, like Sticky FP, or Surface Effectors, are texture tools. In general, you should not add more than one true shader to the plugin list.. you'll get strange results. So don't use more than one shader, and if you have other surface plugins, put G2 *last* in the list.