modo 601
User Guide

Copyright © 2004-2012, Luxology, LLC. All Rights Reserved.
Luxology, modo and the modo logo are registered trademarks of Luxology, LLC.

modo 601
Table of Contents

Getting Started

Initial Setup 2

- Installing modo 2
- Licensing modo 9
- System Requirements 11

Introduction 14

- What Is 3D? 14
- What Is modo? 18
- What's New in modo 601 19

Working With modo 24

- Beginner Tutorial 24
- Rendering Tutorial 44

The modo Interface

modo Controls 58

UI Conventions 58
Standard Tool Controls 69
Keyboard Shortcuts 73

modo Viewports 85

What is a Viewport? 85
3D (OpenGL) Viewport 86
UV Viewport 102
List & Info Viewports 108
Utility Viewports 135

Interface Layouts 178

Exploring the 601 Default Layout 178
Legacy Layouts 192
Customizing your Layout 200
Layout Controls 204

System and Files 209

File Commands 209
Edit Commands 211
Preferences - Input 213
Project Directories 219
Import Reference 220
Import/Export 222
Bitmap Image Formats 225

The modo Toolbox

Modeling primer 230

Modeling with a modo Mindset 230
Mousing Standards 234

modo Tool Reference 238
Creating Geometry 238
Transforming Geometry 267
Deforming Geometry 286
Duplicating Geometry 338
Edit Geometry 376
Vertex Component Tools 394
Edge Component Tools 401
Polygon Component Tools 409
Tool Modifiers 418
Action Centers 418
Falloffs 421
Selections 434
Selection Modes and Types 434
Selection Modifiers 442
Selection Sets 446
Vertex Maps 450
Working with Vertex Maps 450
Vertex Mapping 457
Working with UV Maps 475
UV Mapping 481
Precision 493
Symmetry 493
Snapping 494
Snaps Palette 503
Utilities 506
Work Plane 508
modo's Work Plane 508
Backdrops 512
Modeling with Images 512

Painting and Sculpting

Workflow 518
Working with the Tools 518
Add Image 522

Tool Options 526
Brush Tips 526
Inks 531
Tablet Nozzle 535

Sculpting Tools 537
Sculpting Brush Reference 537

Painting Tools 564
Painting Brush Reference 564

Hair Styling Tools 582
Hair Tools Reference 582
Hair Guides 591

Vertex Map Tools 594
Vertex Map Tools Reference 594

Particle Tools 598
Particle Tools Reference 598

Commands 612
Image Adjustment Commands 612
Sculpting Commands 613

Windows 615
Multi Resolution Palette 615
Color Theme Editor 618

Scene Layout

Workflow 624
Items vs. Components 624
Working with Items 626

Item Reference 630
Items Reference 630

Transforming Items 698
Centers and Pivots 698
Transform Item Tools 699
Transform Modifiers 716

Duplicating Items 717
Instances 717
Replicators 719

Freezing 723
Freezing Geometry 723

Rigging & Animation

Basics 728
Animation Overview 728
Working with the Timeline 733
Hierarchy 739

Animation Viewports 742
Track View 742
Schematic Viewport 744
Channels Viewport 749
Deformers Viewport 753

Channels 756
Channel Links 756
Add User Channels 759
Modifiers and Constraints 761
Applying Constraints 761
Constraints 763
Animation Modifiers 769
Setup & Tools 786
Animation Setup Tools 786
Skeleton Tool 793
Binding and Setup Mode 797
Pose Tool 799
Tools and Commands 802
Dynamics and Simulation 805
recoil Dynamics 805
Constraints 827
recoil Forces 856
recoil Falloffs 865
Deformers 874
Applying Deformers 874
Deformer List 877
Deformer Falloffs 902
Inverse Kinematics 902
Full Body IK Setup 902
Full Body IK 908
Weighting 912
Weight Container 912
Weighting Tools 914
Characters 919
Actors Actions & Poses 919

Shading, Lighting & Rendering

Applying Materials 924

Defining Polygon Tags 924
Polygon Set 929

Shader Tree 934

Shader Tree 934
Shader Tree Items 941
Advancing the Shader Tree 1617

Layers 1620

Layer Effects 1620
Layer Blend Modes 1623

Render Item 1628

Frame 1629
Settings 1635
Global Illumination 1643

Environments 1653

Environment Item 1653
Environment Material 1655
Image Based Lighting 1657

Lighting 1660

Light Types 1660
Light Material 1694

Volumes 1700

Blobs 1700
Render Sprite 1703
Volume 1703
Render Boolean 1704

Effects 1707

FX Item 1707

Camera 1710
Camera Item 1710
Rendering 1727
Render Output Items 1727
Render Commands 1744
Render Passes 1749
Post Render 1755
Render Display 1755
Tone Mapping 1765
Saving Images 1770
Baking 1775
Image Baking Workflow 1775

Frequently Asked Questions

Index

TIP: For the most up to date information, please reference the on-line documentation.

Getting Started

Users completely new to modo, or even 3D altogether might appreciate introductory nature of this section. The 'Initial Setup' guides user through a proper installation and the system requirements to ensure a healthy modo environment. 'Introduction', well, introduces modo and 3D at its most basic level with the 'Working With modo' section introducing a simple project that introduces a common modo workflow.

Initial Setup

Installing modo

If you've come this far, it's pretty likely you already have modo up and running. Still, there is important information users should know regarding installation. This page will go through the complete installation procedure for your chosen platform. modo is available for both Macintosh and Windows and is available by download only; both platforms use a straight forward installer that takes care of the applications installation. Aside from specifying an install location, and assigning appropriate file associations, there's not too much to it.

The product registration screen at Luxology.com

Luxology Account

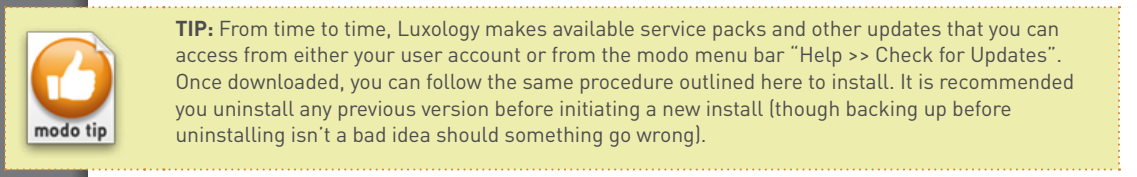
Before you can install modo, you first need to download it. modo program downloads and licensing are facilitated through our website. First and foremost you'll need an account there. If you purchased modo directly from Luxology, an account was created automatically during the purchase process. Use the same username and password you created when purchasing modo to log-in and gain access to the various installer downloads.

If you've purchased modo from a reseller, you'll need to create an account manually. In order to do so, you'll first need the 25 digit serial number from your reseller. It should have been e-mailed to you once the order process was complete (the timing of this varies by reseller) Once you receive that, visit Luxology.com and click on the 'Support' link and then 'Product Registration' sub tab. There you'll be presented with the above screen. You will need to click the 'Start Product Registration' button to continue.

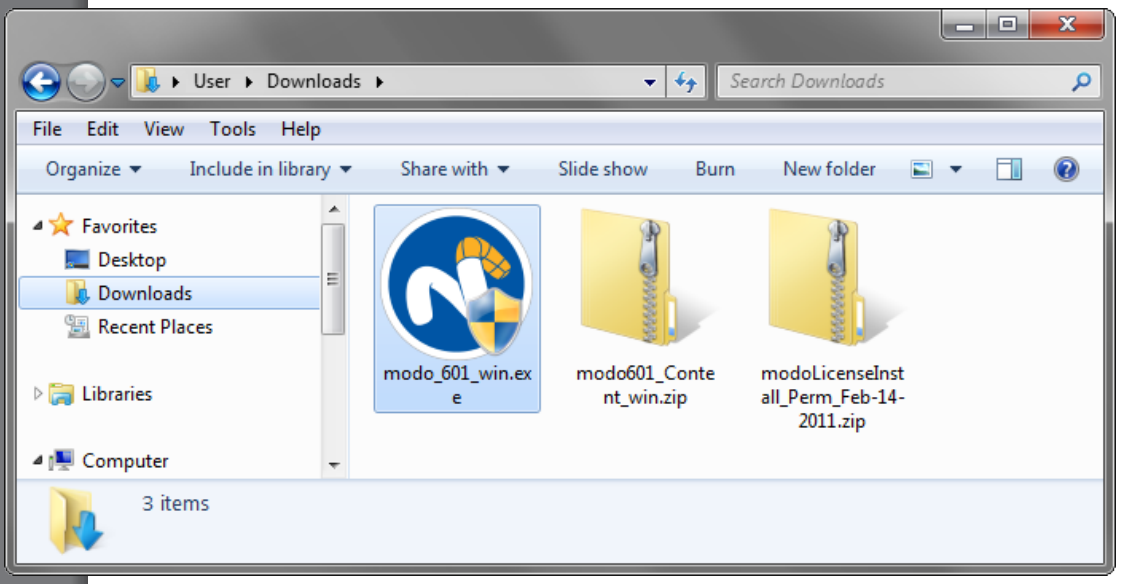
If you previously had an account (say for a product evaluation) the next screen will allow you to log in to that account or create a new user account altogether. Once logged in, you will be immediately presented with the page where you will enter the registration info provided by your reseller. Once you've gone through the account creation and registration process, you can access installer downloads from the account console. One thing to note is modo uses two installers for a 'complete' installation plus a license file download (more on that in the licensing modo section).

Once you have access to your console (*pictured right*) click on the text that says 'Registered Products' to access the various downloads available to you (*If you are returning to the Luxology website at a later time, you can always access your console by logging in and clicking your name in the upper right hand corner of the screen*). Within the registered product panel, there are 3 sections for downloading, one to download the application itself located under 'Install Software' (Step 1), next is the 'Install Content' (Step 2) where you download the content installer, which is optional, but contains the various preset libraries as well as numerous free objects and scenes to help get you started. It is highly recommended you install the content unless you have severely limited drive space available. The third and final section is where you download the modo license file (Step 3). By default,

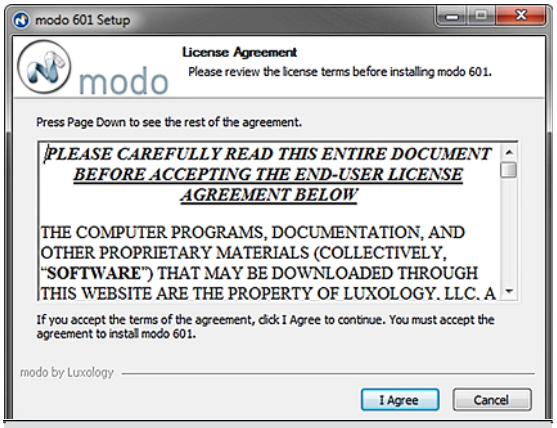
most browsers place these download into the systems 'Downloads' folder. For more info on the location of this folder, consult your operating systems documentation.



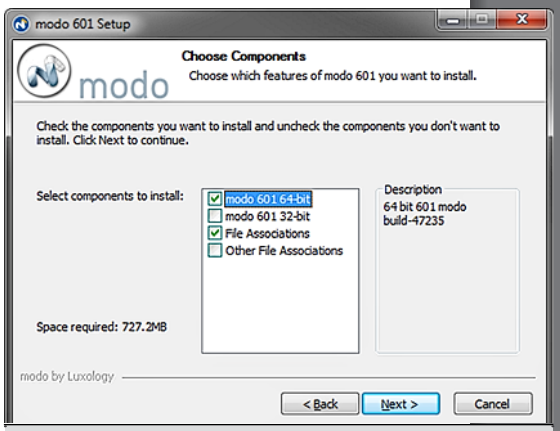
Installation Procedure for Windows



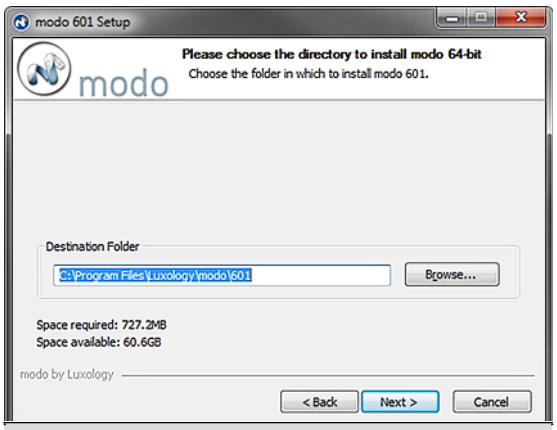
Once you've located your 'Downloads' folder, you can see the 3 files you just downloaded, in this case modo 601 for Windows. Installation order is important so the content can find the installed application, so first you will need to double-click the modo application executable file (.exe) to begin the installation procedure for modo itself. You may get a pop-up requestor asking if you want to allow the application to make changes to your system, click 'Yes' to continue. Immediately following, you will be presented with the InstallSheild Wizard screens-



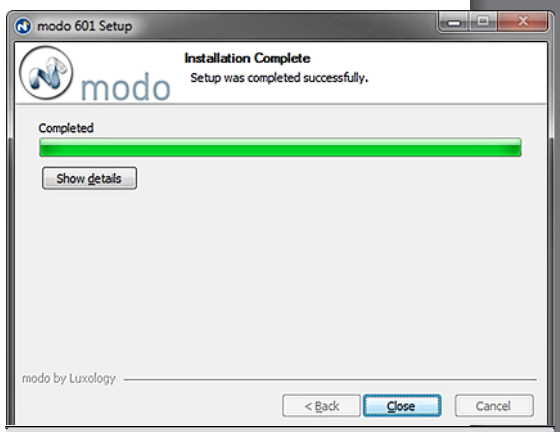
Screen 1 - First you'll be presented with the 'End User License Agreement' to continue select 'Accept' and click 'Next'



Screen 2 - Select which version you want to install and wether or not you want modo to define the file associations for you.

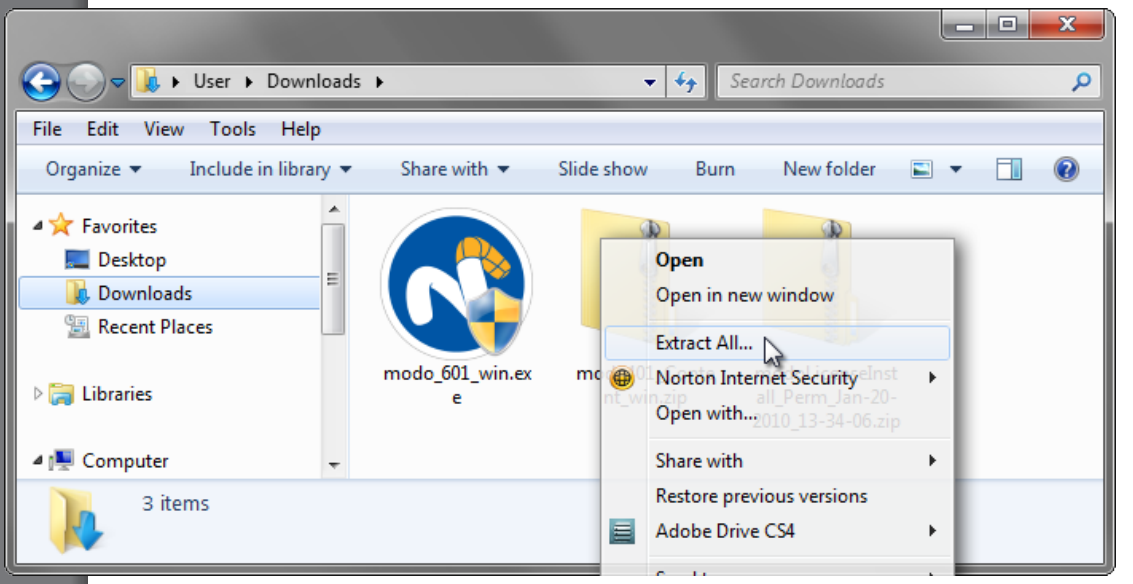


Screen 3 - Define the install location by clicking the 'Browse' button, or click 'Next' to accept the default and continue



Screen 4 - The installer will start the install process, showing progress. When completed press 'Close' to dismiss the installer.

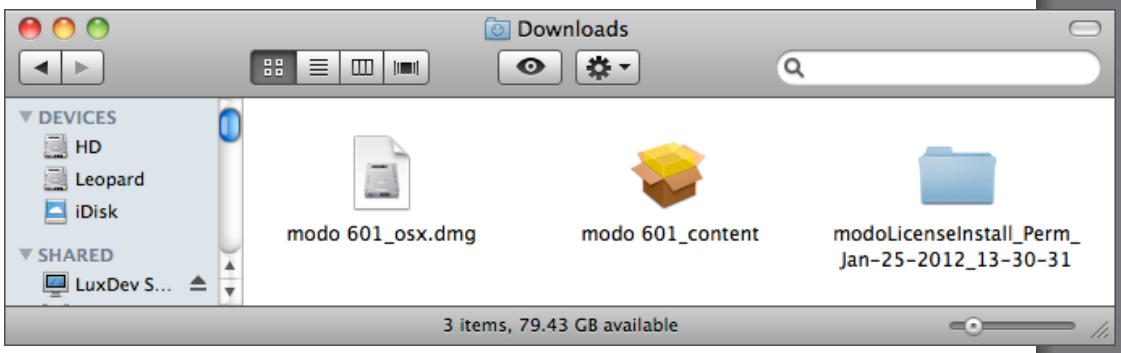
After the installation is complete, you can return to the downloads folder (Below) and RMB click on the Content and License files, and select the 'Extract All' option from the contextual menu that appears. Accepting the defaults will extract the contents to appropriately named sub folders.



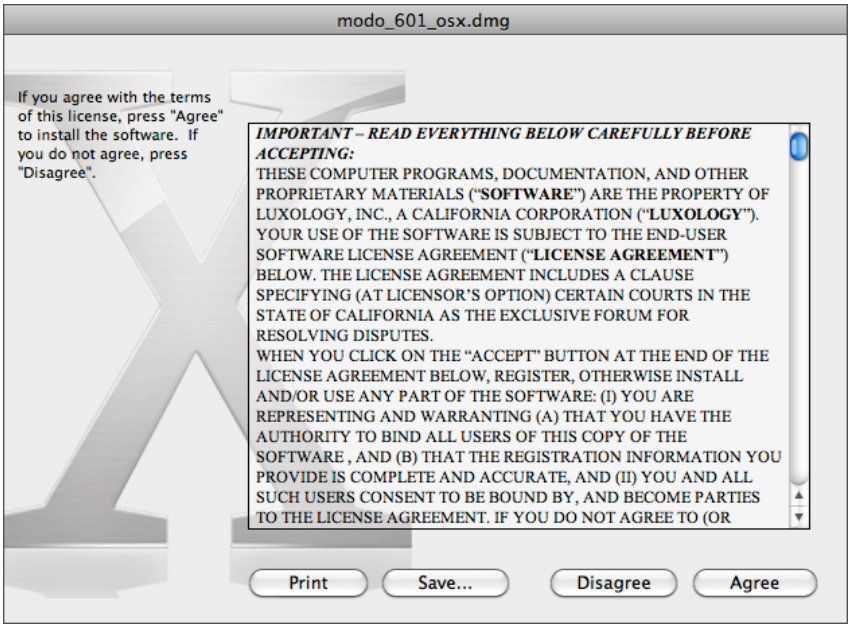
Within the extracted Content folder, there is a file named 'Setup.exe', double click this file to initiate the installation of the content. Follow the on-screen instructions to select the installation destination and click the 'Install' button to complete. Once finished, you are done installing modo, but it isn't ready just yet, see the Licensing modo page to find out how to properly license your copy of modo for use.

Installation Procedure for Macintosh OSX

Once you've located your 'Downloads' folder (shown below), you should find the three files, the DMG disc image file, the modo content package and the folder containing the license file. (*Note: If you are downloading with 'Safari', it's possible the installation will initiate automatically when the file transfer is complete.*)



Double-click the 'modo_XXX_osx.dmg' file, a small window will pop-up verifying the image, followed by the licensing screen below.



You may wish to save or print the 'License Agreement' from this screen. To continue with the installation, select the 'Agree' button, where you will be presented with the following installation screen.

To install modo, one only needs simply to click+drag the modo and the netrenderclient icons onto the 'Applications' folder icon. Your system will scan the file and add it to the folder where it will be immediately available. Consult your user manual should you wish to add modo to your 'Dock'.

Next you'll want to install the content, so return to the 'Downloads' folder and locate the modo content package.

Double click the package to initiate the installation procedure and follow the on-screen instructions; choosing the appropriate install location. Once completed, you are done installing modo, but it's still not quite ready for use, see the Licensing modo page to find out how to properly license your copy of modo.

TIP: Should you choose not to install the content, modo will ask for its directory location each time you open the application. If you receive the error message "*The Standard Content wasn't found. Some feature may not work properly without it. Would you like to visit the Luxology site to download it now?*" You can disable this message by clicking the 'In the Future' button and selecting 'Always No' from the small fly out menu.

Licensing modo

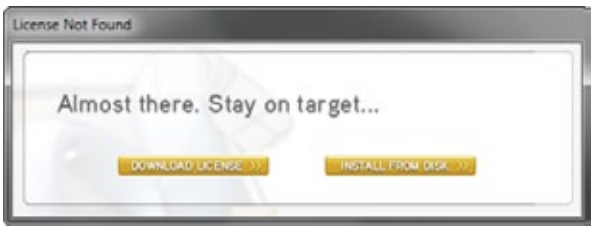
Once modo is installed on your system, one still needs to obtain a license it for it to work (unless you are simply installing render nodes on a networked system, this is the only case where no license is required). Luxology offeres a very generous and flexible licensing scheme, no dongles or extra hardware are necessary to run modo and the application is licensed to you, the user, regardless of your platform or location. The only limitation is modo can only be running as a single instance at any given time; modo can sense other open copies with identical license files on a network and will generate a popup message informing you of this case.

While there are a couple methods to obtaining and installing a license, the simplest method only requires one to open a freshly installed modo. The windows installer automatically adds a desktop icon. Mac users can access their modo installation from the 'Applications' folder (which can be opened from the 'Shift+Command+A' shorcut). Double click the applications icon to initiate modo; it will begin loading and after a moment the following screen appears presenting three option.

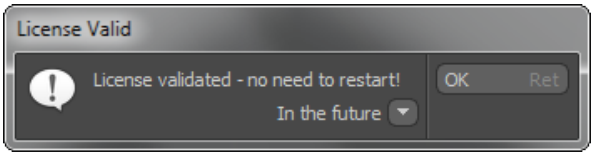
You will need to choose 'Get A License' if you wish to continue using modo. The following screen appears next.

Assuming you have already purchased modo, and downloaded your license from the previous 'Installing modo' page, you should choose the 'Install License' button. (The other options *'Buy modo!' and 'Try modo' will both take you to the appropriate page of our*

website where you can follow the online instruction.) The following screen appears next.



Select the 'Install from Disk' option to open a system file window; using this window, navigate to the location where you saved your downloaded license file (by default the 'Downloads' folder). Select the file and press the 'Open' button. If everything worked out OK, you'll see the following window popup.



You are now ready to go, you have a fully licensed and happy modo. :-)

Alternate Licensing Options

There are some other possible way to install a license for modo. For one, when you download your permanent license file from your account page on our website, within the License folder download is an executable file that will automatically install the license file, sans modo by simply double-clicking it (on Macintosh, the license executable is currently for OS9 only). Lastly, if you're the type that needs to know what's really going on, well, the license file (typically named modo followed by 5 digits, the build number, and the LIC extension -'modoXXXXX.lic') simply needs to be copied into the appropriate user folder. The default location of this folder is as follows-

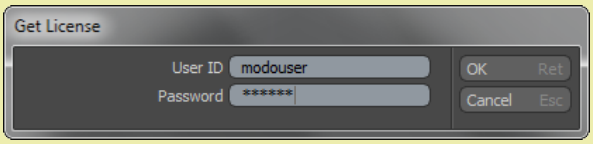
Windows XP
{Drive}\Documents and Settings\{login name}\Application Data\Luxology

Windows Vista/7
{Drive}\Users\{login name}\AppData\Roaming\Luxology

Mac OSX
{Drive}\Library\Application Support\Luxology\

Once the license file is located in this folder, that's it, you are all ready to go.

TIP: Power users will quickly realize there's no reason to download the license file from their account. Once you have a permanent license file available, you can click the 'Download License' button in modo (3rd step in the section above) and a window will ask for your login.



Simply enter your luxology.com 'User ID' and 'Password' and modo will automatically download and install the license. Easy, huh.

System Requirements

modo has been developed to ensure broad hardware and software support, still, the creation of 3D modeling and rendering can push the limits of even the most powerful computer. For the best overall experience when using modo, we recommend users systems comply with the minimum requirements outlined below. As in all things 3D, the more "horsepower" you have, so to speak, the better.

Windows® Minimum Requirements

- Windows 7, Vista and XP (32 and 64 bit versions)
- 2 GB available hard disk space (including full content installation)
- 2 GB of system RAM
- Accelerated Video Graphics adapter with OpenGL support capable of minimum 1280 x 800 resolution
- Mouse or compatible pointing device (pressure sensitive tablets supported)
- Pentium 4 processor or better
- Internet connection required for product activation
- Quicktime and web browser application necessary to view Inline Help system

Windows® Recommended Configuration

- Windows 7, Vista and XP (32 and 64 bit versions)
- Intel Xeon/Dual Core, Core 2 Duo/Quad, Core i7 and AMD Opteron/Phenom
- Accelerated Video Graphics adapter with 512MB of VRAM and OpenGL support
- 4 GB of system RAM

Macintosh® Minimum Requirements

- Mac® OS X 10.6 or later (32 bit & 64 bit versions)
- Macintosh with Intel Core 2 Duo processor or better
- 2 GB available hard disk space (for full content installation)
- 2 GB of RAM
- Accelerated Video Graphics adapter with OpenGL support capable of minimum 1280 x 800 resolution
- Mouse or compatible pointing device (pressure sensitive tablets supported)
- Internet connection required for product activation
- Quicktime and web browser app necessary to view Inline Help system

Macintosh® Recommended Configuration

- Mac OS X 10.6 or later
- Macintosh with Intel processor
- Accelerated Video Graphics adapter with 512MB of VRAM and OpenGL support
- 4 GB of system RAM

Online Documentation Requirements

It is recommended to have the latest version installed of your preferred browsing application. This documentation has been tested to perform properly under the most recent versions of the following browsers-

- Internet Explorer 8 or later
- Firefox 3.6 or later
- Safari 4 or later
- Chrome 4 or later
- Opera 10 or later

To view inline help videos Apple Quicktime is required. Also, while the documentation can be viewed locally from the host machine, an internet connection is required to view any user comments, rate pages and connect to any external links (most denoted by the icon).

TIP: Many systems (especially laptops) include what is called an 'integrated graphics adapter', the necessary component in a computer to display images on your screen. These systems, while economical, often share memory with the CPU and are designed more for power saving than for performance with their drivers often barely or very poorly supporting standards necessary to modo. Many issues in modo can be traced to the graphics adapters and their drivers. Luxology highly recommends the use of a discreet graphics adapter (often branded by ATI or Nvidia) for the best overall user experience.

Introduction

What Is 3D?

Many new modo users may have downloaded and installed our software with little or no background in 3D graphics, much less 3D at all, this section is meant as a primer for those interested in 3D graphics but with very little background or understanding on the subject overall. While not a complete definition by any stretch of the imagination, it has been stated as plainly and as simply as possible, providing new users with a basic introduction into the terminology many of us take for granted. 3D for all!

So what is this 3D stuff all about anyway?

3D seems to be *the* hot buzzword these days. Everything has got to be "3D", but what does "3D" mean really? Well, actually, it can mean multiple things; so first let's define what differentiates 3D from 2D by defining what "2D" actually is. A great example of this would be what you are presently looking at, your computer monitor (right?). Despite how things may look, your screen is a strictly two dimensional or "2D" surface, a flat plane. It has millions of tine colored squares on it called pixels (for Picture Elements). They lay across it horizontally and vertically in a grid formation. These two dimensions (horizontal and vertical) could be labeled as X and Y, respectively, as reference to geometry (the mathematical kind). With them, you can pinpoint any position on the screen by simply using two values, one represented by X, the up and down direction, and one for Y, the left and right direction. But, there is no "in-front-of" or "behind", it's just a smooth, flat surface all the way across. By adding one more dimension to that flat surface "Depth", suddenly everything changes. This depth, designated with the addition of a Z value added to the previous X and Y, can now pinpoint any location in space; you're no longer confined to the single plane, this is where 2D becomes 3D. If your monitor had all three dimensions, it would suddenly be possible to have a real 'in-front-of' and a 'behind' (as it likely will be one day, but alas, not quite yet) . This makes perfect sense because human vision is designed for "3D", our two eyes provide two slightly different views of the world that our brains then put together to provide our sense of depth. So the simplest definition of 3D is any medium or technology that introduces 'depth'.

INFO: In geometry, this system of defining positions in X,Y and Z space is known as Cartesian Coordinates named after its inventor, Rene Descartes, who was also known as 'Cartesius'. He was a 17th century mathematician, physicist and philosopher famous for saying "I think, therefore I am".

How did depth become a buzzword?

The term "3D" is still fairly broad, there are multiple different, yet oddly common meanings to the term (that seem to be used interchangeably, and often incorrectly); two in particular relate directly to what modo does.

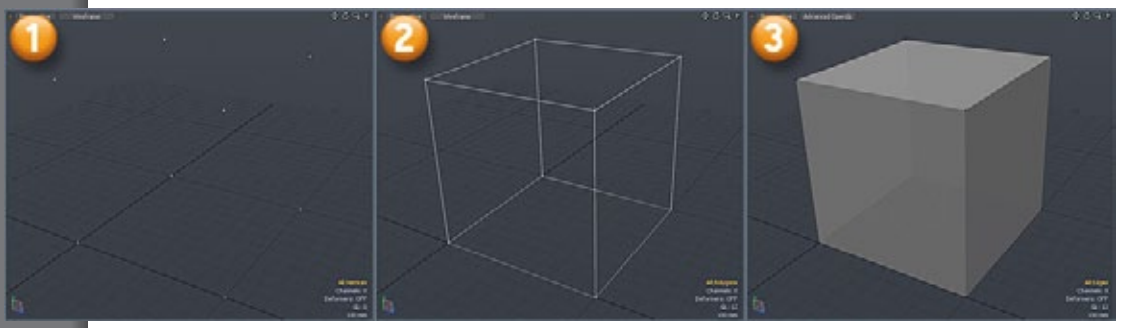
The first and most common is in the form of 3D stereoscopic images, a means of coaxing 3D depth from 2D images. Often referred to as '3D' when it is really called "Stereoscopy". The technique was initially introduced in the late 19th century (at the dawn of photography), the stereoscope (seen here on the right) was a early technology where two images, photographed slightly apart from each other, and observed through the viewer, produced a sense of 3D depth. The very same technique popularized by the View-Master system you likely had as a small child that is still captivating kids today. There have been other variations to the technology as well, such as overlaying the two images and then filtering how our eyes see them with glasses, referred to as 'Anaglyphic Stereo', many comic books and B-movies of the 1950s popularized this technique with their iconic red and blue paper glasses. Modern 3D cinema also combines two separate images (using two projectors aligned over the top of one another) onto the same screen that when, same as before, are filtered with glasses, but now they are 'polarized', a technique that capitalizes on the wavelength property of light. When done well, it produces a 3D image with all the depth and color of the world around us.

modo is capable of generating imagery very easily that can be used for these purposes, but there are no viewers or hardware one can use or purchase that displays the depth directly within modo itself. Users would need other external tools to combine images generated by modo to produce the final stereoscopic result.

modo and modern era 3D

The next most common meaning to the term "3D" (and the one we are actually most interested in), is in relation to images that are generated virtually, that is to say generated by means of using computers graphics software; think movies like 'Toy Story' . All 3D programs (like modo) utilize a method in which objects are modeled by defining their shape mathematically. In most cases, these virtual objects are defined using the previously mentioned X, Y and Z coordinates system, this is precisely what makes them 3D and why they are called such.

Let's explain it a little more. For simplicities sake, say you wanted to model a three dimensional cube, this is done by defining the coordinates of the eight corners, each would receive its own X,Y & Z value, designating their position in space. Each positions is referred to as a 'vertex' (plural: vertices); they have no size or volume by themselves, they are simply *positions*. Now, additional instruction are given to the software that tell which vertices are connected together by 'edges'. An edge is simply a line connecting two vertices together, and while it also has no volume, it does have length. To complete the cube, still more instruction are necessary to define what groupings of edges produce a 'polygon'. Like in geometry, a polygon in computer graphics is a plane defined by its edges and vertices that produces a surface. By virtue of how they are defined and arranged, the 6 polygons together, one for each side of the cube, produce the three dimensional cube shape.



In panel one, modo represents vertices as tiny white dots. Panel two illustrates the connecting edges between the vertex corners, with panel three illustrating the six polygon planes that make up the final cube shape. This object is considered a '3D model' as it is fully *three-dimensional* (taken from the XYZ dimensions that define it), just like a real box. Additionally, this 3D cube has mathematically perfect perspective, and it can be moved, rotated or scaled without losing any of its original properties. And don't worry, making a cube in modo is way easier than describing how one is built in 3D. It actually might be hard to believe, but if someone wanted to model something as

complex as a car, it is defined in exactly the same manner as this cube, its just several orders of magnitude more complex, and while the model itself is more complex, it's not really a lot more complex to make once you understand the tools and what they do.

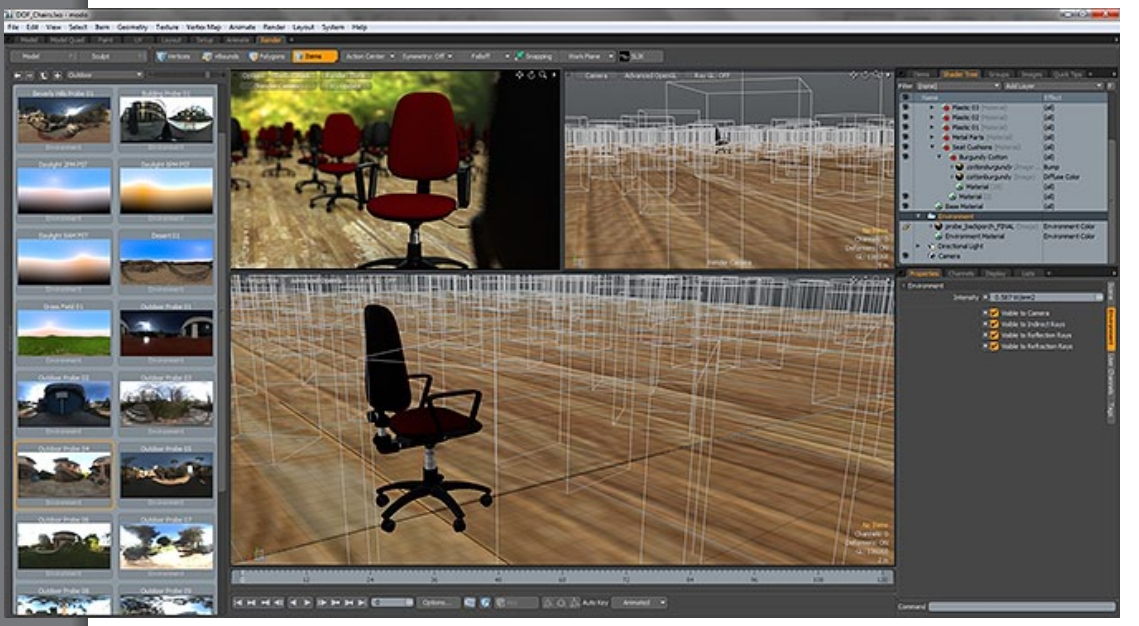
One aspect of creating these models that can be troublesome for users is the fact that, while dimensionally accurate, they are still displayed on a 2D screen, the world that exists inside the software is fully three dimensional, with all the depth that our real world has (I like to think of the monitor as a window into the applications virtual world), but the viewing medium can make it awkward to navigate. modo is designed to make this as painless and simple as possible, but there is a degree of familiarity one needs to acquire to feel comfortable working in this manner.

Going from a 3D model to a 3D image

Once created, these virtual collections of surfaces (be it a 'cube' or a 'car') are further processed by the software, using very complex mathematical algorithms that work in a similar fashion to how light acts in the real world and is captured by a camera in a process known as "rendering". The ultimate result of the rendering process is what is called a 'bitmap' image. A bitmaps is simply a collections of tiny individually colored squares, just like the 'pixels' on your monitor; when viewed together as a group produce a recognizable image. The greatest benefit about generating images virtually is users can control every single aspect of the image creating. Photographers are at the mercy of reality, but in 3D, images can be as stylized and artistic, or as photographically realistic as you want, it all depends on your determination and dedication.

Finally, I feel I should mention that even though the term mathematical has been tossed about quite a bit, the math-phobic shouldn't get too worried, working in "3D" is often times more art than science. The more you learn about 3D computer graphics, the more you'll have the power to generate and manipulate the polygons and pixels, producing results that are limited only by your imagination.

TIP: Are some of the terms above still a little confusing? You can easily find definitions for virtually any word using an online search engines such as 'Google' or 'Bing'. If you append the term " define: " to the front of your word (no space), you will receive a dictionary-like definition instead of links to pages with that word on it.



What Is modo?

modo is a 3D modeling, animation and rendering program that runs on Windows and Macintosh computer systems (If you're not familiar with what 3D means in this respect, I recommend you read the previous page "What is 3D?"). Created by Luxology, a company formed by the fusion of artists and scientists, modo is a tool that we ourselves would want to use. Designed as a next-generation application with a carefully considered interface and toolset, that works in unison to provide users with a fluid and intuitive workflow. We created modo with the understanding that our users would come from a broad spectrum of industries and disciplines, meaning modo would need to be flexible. modo can easily slide in alongside existing pipelines providing individual functionality, like modeling, sculpting or texture painting; supporting a wide variety of industry standard file and image formats allowing the bi-directional exchange of data. For others, modo can act as the entire pipeline itself allowing users to design, model animate and render entirely in the same environment.

In creating modo, we started with the users, and this has been at the core of all our policies as well. This is why we have the most flexible and generous licensing policy in the industry, we license modo to the user, not to a location, or some hardware or even a dongle or any other device that limits your ability to use the software. Wherever you are is where you can use modo, regardless of platform. All you'll need is an internet connection to download the installer from your account.

For users who employ modo as a component in a multi-application pipeline there are various file format options which allow modo to interoperate with most pipeline systems. The modo user interface also allows for the quick remapping of user controls so that as the artist bounces between modo and the pipeline application, the workflow is not impaired by having to make the mental switch and change things as basic as how to tumble a viewport or activate the 'Move' tool. The modo interface itself is so malleable it is effectively fluid in operation. This allows the user to either use a standard interface configuration and thus utilize modo in a specific focused mode, or to customize and adapt modo to their most demanding workflow requirements. These options represent a tremendous amount of research, design and implementation at the Luxology labs and the end result is an application that can be painlessly inserted into an existing pipeline. modo is an application designed to enable artists to produce stunning work more efficiently than ever.

The modo architecture enables a new kind of workflow altogether, a richer and much less linear workflow. The core fusion of the various technologies (modeling, sculpting, painting, rigging, animating, lighting, texturing and rendering) allow users to leverage all technologies literally at the same time. This deep integration also enables users to leverage rendering technology while painting or use geometry to create new images which can later be used as brushes to paint detail onto other models. This interweaving of technologies makes the modo experience much greater than the sum of its parts.

Enjoy it. We do.

What's New in modo 601

modo 601 has received many improvements across the board, big and small. Possibly being our biggest update yet. This page documents a lot of the changes, and links to the appropriate page of the documentation so users familiar with modo can get the information they need and start working with the new features right away.

Modeling

Primitive Symmetry - Primitives now created in Symmetry mode for Cube, Capsule, Cone, Cylinder, Ellipsoid, Sphere, Torus, Tube and Curve tools.

Primitive Poly Type - Defines what polygon type to create for Sphere, Ellipsoid, Capsule, Torus.

Primitive Poly Type Default - New Preference setting for default polygon type for Sphere Ellipsoid Capsule and Torus.

Tube Tool By Length - New option for the Tube tool to create segments by length.

Mesh Clean Up - New command cleans up many modeling problems that can cause render errors among other problems.

Fast Slice - A new Slice Tool option to speed up slicing complex geometry

Hold Slice - A new option for the Axis Slice tool that speeds up slicing of complex geometry.

Arc Tool - A new tool that creates pie and wedge shapes.

Weight Tool - Updated to work better with the new Deformers and Falloffs.

Soft Selection Tools - New tools that use the new soft falloff function Soft Move, Soft Rotate, Soft Scale and Soft Transform

Torus Tool - Updated options for 'Ring Radius', 'Cross Section Height' to simplify Torus creation -- mmmm, Donuts.

Slip UVs - New UV editing option added to the transform tools allowing greater flexibility when editing geometry.

Pixar Subdivisions now support Weighting on both edges and vertices.

Vertex Map Painting and the associated vertex map editing brushes.

Topo

Re-topology View - New viewport display options and tools make it easier than ever to do this now essential task.

Background Constraint: Point mode - A new constraint mode that keeps all vertices against the background.

Topology Pen - A new multi-function tool specifically designed to speed the re-topology process

Bridge Poly Lines - Now removes line polygons when bridging between edges.

Pen Tool - New options to make Triangles in Quad-Strips and a new Show Numbers option.

N-Gon to Quad - New command to creates a quad strip of polygons from an N-gon.

General

Three New Vertex Map types - Particle Dissolve, Particle Size and Transform Maps enhance new Volume functions.

Offset Work Plane Keyboard shortcut to offset work plane to mouse pointer position.

SDS level per type Independent levels available for regular Subdivision Surfaces and Catmull Clark.

PSUB Cage Mode Temporarily toggles all SDS on or off for any Mesh Item layer without losing of SDS definitions.

Alembic format support for Import and Export - Exchange animation data flawlessly with other packages.

Falloff

Soft Selection Falloff - A new type of falloff that modifies unselected geometry with a new falloff visualization method.

OpenGL

Camera Safe Area - New option for Safe Area camera viewport overlay.

Camera Stereo Volume Display - New display function to aide and enhance the creation of stereo images.

Overlay Weights - A new viewport display option to allow user to visualize the weight values in any display mode.

Show Weight Values - New option to display actual weight values for selected vertices.

Show as Silhouette - A new viewport display option to allow animators to better see shapes when animating characters.

Shaded Custom Items - Custom scene items, such as the Gear item are displayed properly in shaded views.

Shade Selected Replicators - Replicators can now be drawn as regular shaded items as well as bounding boxes.

Show Motion Paths - Animated elements can now display their motion as a visible per item path in 3D viewports.

Particle Tools

Particle Cloud - New item that creates clusters of particles (single vertices) that can be used with the new Volume options.

Particle Generator - A new surface particle generator that works like the Surface Generator but is evaluated much earlier.

RealFlow Particle Import - Option to import point cache data from RealFlow for rendering with the new volume functions.

Particle Painting tools and the associated particle sculpting brushes.

Rendering

Spectral Count - Additional user control over rendering of dispersion effects.

Contour Edge - New option to define a crease threshold to allow the drawing of interior lines along with previous exterior.

Photon Map Walkthrough - A walkthrough mode for Photon mapping providing jitter-free animating caustics.

Area Light - New options to make Area Lights visible to cameras and reflections.

Sky Portals - New light type that helps to focus light rays when rendering interiors that are lit using global illumination.

Clipping Plane - New rendering options that provide simple cut-away views for industrial and architectural renderings.

PSUB Render Level - New options to control Pixar Subdivision levels when rendering.

MatCaps - A flexible new Material type that allows an unlimited number of shading styles for 3D viewport display or rendering.

Diffuse Roughness - A new shading model that better represents highly diffuse materials such as brick and stone.

Texture Replicate - A powerful new texturing tools that allows wide areas to be textures with small textures and no seams.

Render Display - A completely reworked render display that provides a wide array of nondestructive image editing tools.

Preview Updates - New options added to the interactive preview allowing rendering at specific size and animation preview.

Maximum SSS Depth - New control that simulates solid cores (such as a bone in a finger) when rendering SSS.

Rounded Edges Shading - Instant smooth round bevels on faceted geometry with this new material option.

Flow Noise - New options added to the noise procedural make it easier to make organic animations.

Gradient - New 'Thickness' and 'Bumpless-Incident' input parameters for the Gradient Shader Tree item.

Hair Shader - A new shading model dedicated to the shading of realistic hair and fur.

Skin Shader - A new shading model dedicated to he shading of realistic skin.

Cel Shaders - Cel Edges and Cel Material are added to allow the rendering on non-photoreal elements.

Halftone Shader - Another non-photoreal rendering option that simulates printed halftones images.

New Render Outputs - Object Coordinates, World Coordinates, Shaded AA Samples, Shading Incidence.

Volumes

Render Booleans- New items type provides users the ability to render animated cutaways and trims per material.

Sprites - Fast Rendering Sprites can be used to simulate a number of volumetric type effects.

Blobs - metaball-like blobs that can be hypertextured and animated to produce an endless variety of interesting effects.

Volume Items - true volumetric rendering offers clouds, vapor and any other dusty, vaporous forms with ease.

Particles

Particle Painting Tools - Get control over Replicator placement with Particle Paint, also great for volume items.

Particle Cloud and Particle Generator - New item types provides additional options for Replicators and volume items

Particle Xform VMaps - Want to adjust Replicators on a per vertex basis, now you can, select the vertex and transform.

Import of RealFlow Cache data - Render complex particle simulations generated in RealFlow using Blobs or whatever you like.

Render Passes

Render Passes - This important rendering feature now allows endless variations on a scene that all render automatically.

Deformation and IK

Skeleton Tool - Add joints to a model faster than ever before, automatically creates the hierarchy as the bones are drawn.

Bind - Link joints to geometry in record time using the powerful 'Bind' option; creates Deformers and Weights in one go.

Setup action state - New mode defines the rest positions for Binding joints and IK.

Pose Tool - No need to set up an advanced IK rig to animate, simply grab the 'Pose' tool and start your posing.

Full Body IK - Joints and Skeletons just got a lot more advanced with modo's super easy, super powerful Full Body IK.

Animated Deformers - Plenty of animated deformer goodness because animation isn't all about joints and skeletons.

Deformer Falloffs - Provide additional control over deformers allowing lots of new possibilities, fully animated of course.

Replicator time offset - Legions of animated Replicators no longer have to leap in unison with time offset controls.

Local Transform Morph - New option allows users to create corrective Morphs shapes on deforming geometry.

Weight Container - Abstract rig weighting away from the geometry for more powerful and flexible rigging options.

Normalization - Multiple Weights are now averaged and normalized using the Normalization folder in the Deformers View.

Advanced Weighting Tools - Using the familiar painting workflow, users can intuitively adjust weighting in any view.

Dynamics

Recoil 2.0 - Oh, did we mention recoil is now fully integrated with the addition of new forces and Soft Bodies too.

Hundreds of additional refinements have also been integrated into the new modo 601 that were just too numerous to list here.

Working With modo

Beginner Tutorial

Modeling in 3D is most often the act of creating and modifying; users typically start with a basic shape and then utilize the *transform* or deform tools to shape the geometry into the intended form. In order to introduce new users to the modo, we're including a simple step-by-step tutorial walk through for modeling and rendering a coffee mug; simple enough to go through the various steps in detail so users can better understand modo's powerful tool set and unique workflow. It is expected that users are comfortable using a mouse, clicking and dragging, can navigate menus and have a basic familiarity with modo's interface. If not, it is recommended users read up on the 601 interface page of the documentation to get an idea where the various panels are located. It may also be extremely helpful for users to go over the User Interface Conventions page for a better idea on how users are expected to interact with modo. Along the way, I'll be explaining essential concepts as they become important. If you've

already covered modeling the mug, you may wish to skip to the Rendering part of the tutorial.

Getting Started

Before we begin, lets make sure we're on the same page. modo offers multiple interfaces by way of the tabs directly under the menu bar (referred to in these docs as the *interface tabs*), each with a focused toolset targeted toward the particular task. Modeling is best accomplished, not surprisingly, using the 'Model' or 'Model Quad' interface, both are identical except for the arrangement of the 3D viewports. The 'Model Quad' viewport may be comfortable for some users, but the 'Model' viewport offers a much larger view of the scene, and when combined with the 'Work Plane' and 'Action Centers' (which we will cover ahead), can be every bit as accurate. We'll be working in the 'Model' tab of the default interface which provides a large Perspective view of the so far empty scene. To ensure you are using the default interface, select the menu bar command "Layout > Layouts > 501 Default". Then left mouse button (LMB+click) on the 'Model' interface tab to open it, if it isn't already. You'll also want to start with a new empty scene, so if you haven't just started modo, press 'Ctrl+N' to create a new blank scene.

Now onto the mug, the best way to start a project is to know what it is you want to do, and that means gathering reference. Here's an image of the coffee mug we'll be modeling.

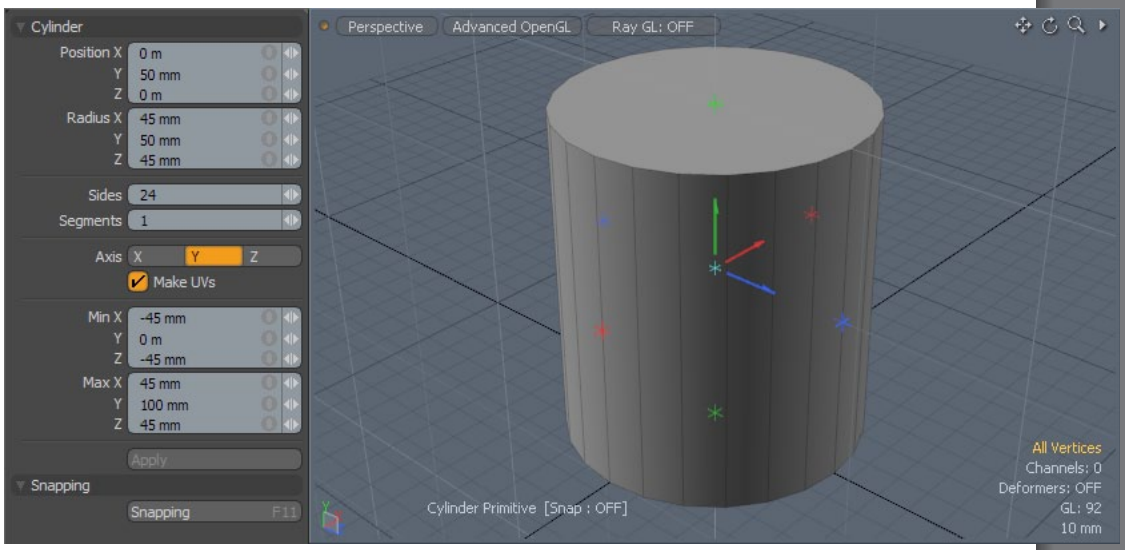
It will be easier to tackle the main body of the mug first, then add the handle later. As mentioned above, modeling is often the act of creating and then modifying, so that is exactly what we're going to do here. First let's examine the shape of the mug. It could be considered somewhat cone-shaped, but it would probably be easier in this case to

define a cylinder shape and modify that. So lets do that. LMB+click on the 'Cylinder' tool icon to select it, it's located in the 'Basic' sub-tab of the toolbox, located on the left side of the interface. When selected, its background will change to an orange color and its attributes will appear in the 'tool properties' viewport directly below the toolbox (pictured in the left panels further below).

Cylinder tool icon

You'll be using the mouse to create the cylinder in the 3D viewport view, but before you begin to draw you need to make sure the initial drawing orientation is correct. The default 3D viewport has two grids that users can use for reference, the dark grid is stationary and effectively represents the ground plane, within this grid are two darker lines that intersect at the world origin, the 0,0,0 position. The second light colored grid is called the 'Work Plane', this is the more important grid for modeling. It represents where actions occur in 3D space, when the user clicks in the viewport, the action originates at the intersection of the mouse click position *and* the Work Plane. The Work Plane also determines the orientation. As users spin and navigate in the viewport (hold down the 'Alt' key and LMB+click and drag in the viewport to tumble the view) you can see the Work Plane changes orientation to face one of the three axes. If you tumble the viewport so that the vantage point is facing downward toward the ground plane grid, the Work Plane will align itself to the ground, and set the initial orientation for our cylinder.

Once the Work Plane is positioned properly, hold down the 'Ctrl' key on the keyboard (to constrain all three axes of the cylinder tool to the same value when drawing) and then LMB+click at the origin position and drag outward. This action does two things, first it obviously places the geometry into the scene, the less obvious action is the tool is now in *interactive* mode; basically meaning that while the tool is still active users can use any of the available *handles* to modify the cylinder, as well as edit property values and see the shape update in real-time in the viewport. Handles are the controls that allow users to make specific modifications. Once a tool is inactive (referred to as *dropped*) interactive editing will no longer be possible, so you want to keep your tool in the *active* state until you have the shape the way you want it.



Lets go over the interactive handles, at the center of the cylinder is three arrows, for moving the position of cylinder, colored to correspond to the axis they move on, then there are small *plus* or star shaped handles for adjusting radius values individually. Holding the 'Ctrl' key when adjusting any of the handles will again constrain all values to be the same. Nearly all tools will have some type of handle specific to the function it provides.

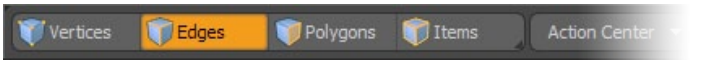
As an additional note, you may also want to pay particular attention to the orientation of these axes when modeling, as it will allow you to better follow the tutorial when a specific axis is referenced. You can easily follow my orientation by matching the axis widget in the lower right corner of the 3D viewport.

The measurements of the mug are 90mm wide by 100mm tall. Since the Cylinder is defined as a 'Radius' (half the diameter), the size we need is 45mmx50mm, adjust the handles or set the values into the input fields of the properties panel to match those above, LMB+click the red arrow handle and move the cylinder upward so it is resting on the ground, keeping particular note of the 'Min' setting which, in this case, the 'Y' value determines the lowest edge of the cylinder. The above value is at '0' ensuring that the mug will be resting on the ground when rendered. Additionally, the default tool will have multiple edge loops vertically, called 'Segments', in order for this turorial to work properly, you'll want to set the 'Segments value in the tool properties panel from the default value to '1'. Once you have the values the way you like, press the 'Q' key to drop the tool.

TIP: While the Work Plane was instrumental in determining the first action of the Cylinder tool, it was not a necessity to adjust the handles and make further edits while the tool was interactive. Users can freely tumble and zoom viewports, even when tools are active, subsequent movement of the Work Plane will not adversely affect the geometry.

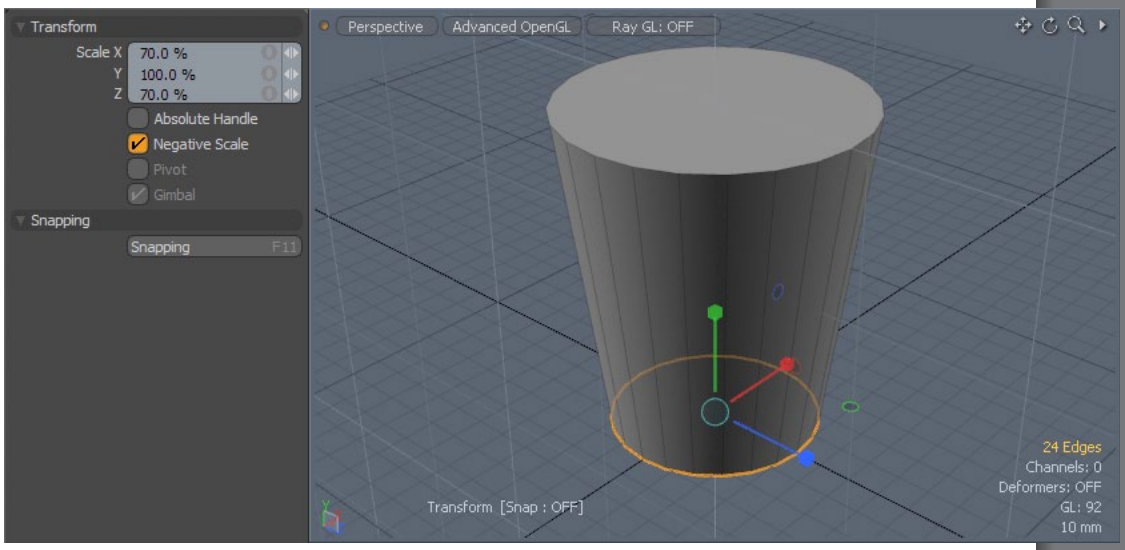
Editing the Form

For most editing that is applied to the model, you will not want it to apply to the entire mesh equally, therefore you need to tell modo what part it is you wish to edit. This is done through the application of a *selection*. modo offers several selection *modes*, these determine what type of geometric selections can be made, either Vertex, Edge or Polygon. If you are not familiar with what these *components* are, please read through the 'What is 3D' page of the documentation for an overview. modo references the current mode in the 'Modes' toolbar above the main 3D viewport. Users can easily toggle between these modes by pressing the 'Spacebar' to continuously cycle through the three main modes for modeling geometry. To make the next edit, we want to select some Edges, so press the Spacebar until the 'Edges' button is highlighted.



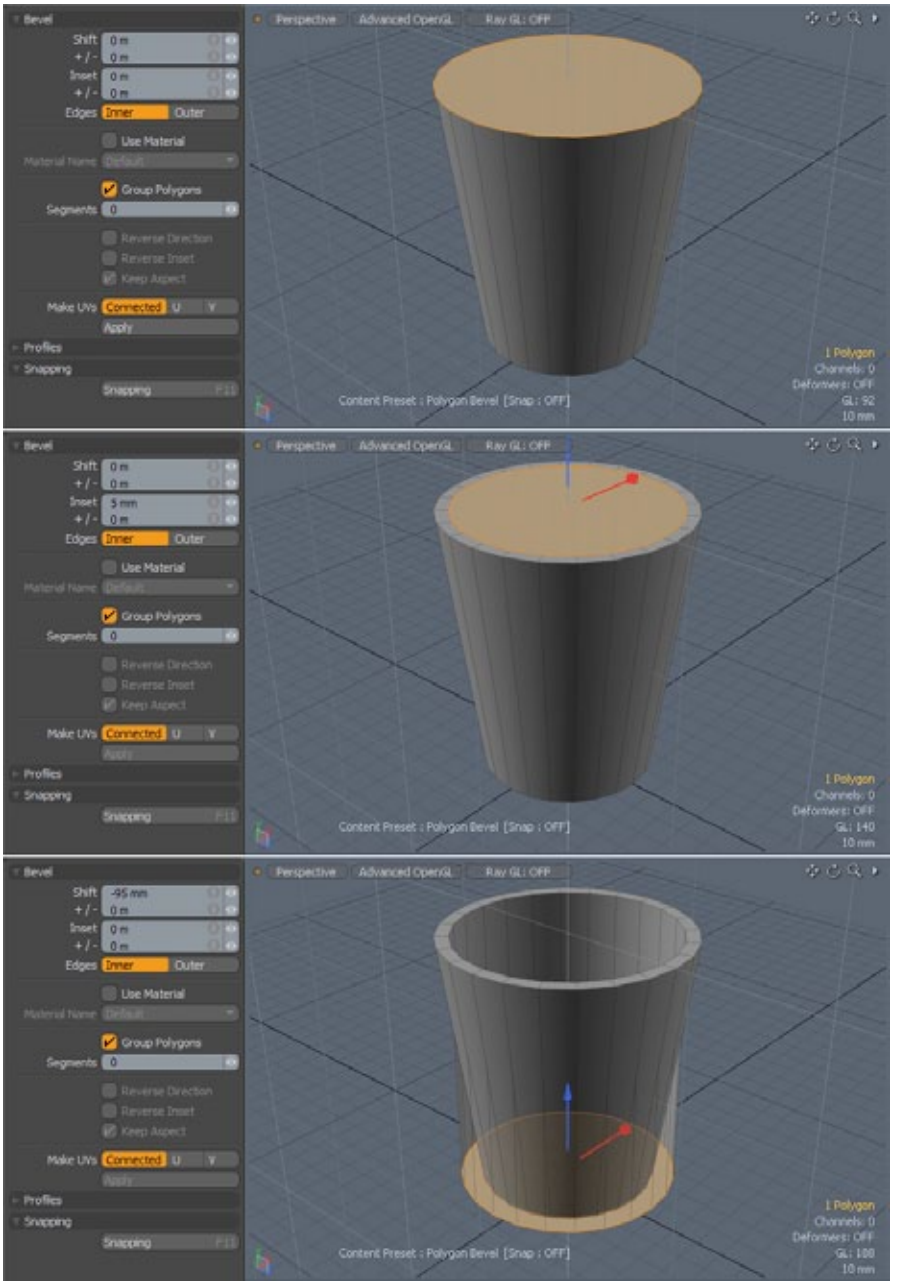
Now to make the selection. We want to select the entire bottom edge of the cylinder, modo has shortcuts to simplify these kinds of actions. In this case we need to only make a single initial selection. By default, modo makes what are called *paint* based selections, meaning when users LMB+click and drag over their geometry, modo will select all elements that are *brushed* over (sort of like painting). LMB+click and drag over one of the lower edges of the cylinder being careful to avoid the upright edges. The selected element will change to an orange color indicating its selected state. Next, press the 'L' key on the keyboard to select all the connected edges in the *loop*, this is called a loop selection, and is a handy way to edit continuous sections of elements.

With the selection set, we can now make the actual edit. Activate the 'Scale' tool, this can be done by pressing the icon in the interface, but being such a commonly used function, it has a keyboard shortcut 'R'. Press that key and the Scale tool handles should appear in the center of the selected ring of edges shown below here.



We want to scale the loop to 70%. To do that we could click on the ends of the red and blue handles separately setting both to 70% independently (as seen in the properties viewport above), but to simplify this operation the scale tool has an extra handle in the way of the floating circle. By LMB+clicking and dragging on the *green* circle, users can scale both the 'X' and 'Z' values simultaneously. Once that is applied, again press the 'Q' key to drop the scale tool and return to selection mode. The selection is no longer necessary either, that can be dropped by next pressing the 'Esc' key.

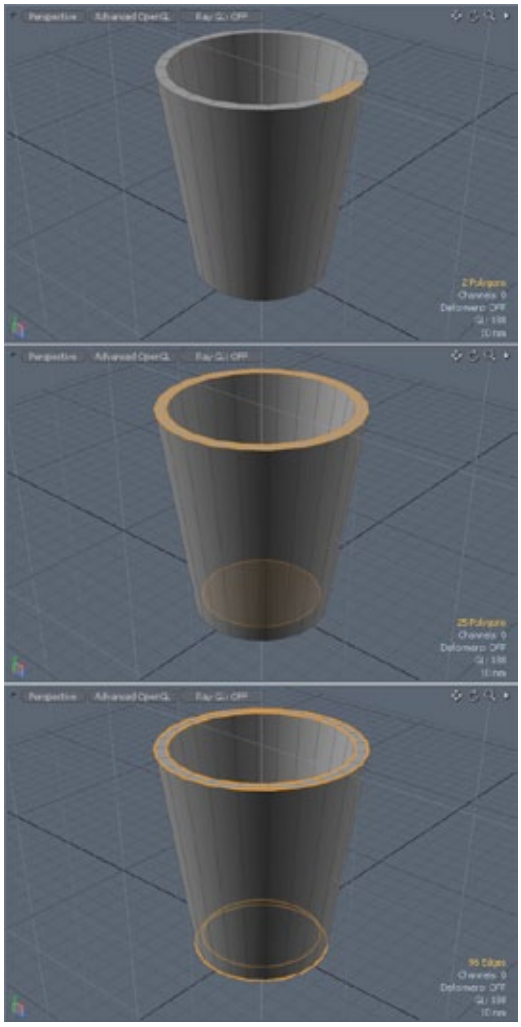
To make the interior of the mug, we'll need to again select some geometry and then edit it. In this case, we need to select the top polygon, so press the 'Spacebar' until the 'Polygons' selection mode is highlighted (like the 'Edges' mode was earlier). In order to know what particular element is being selected, modo has a function called pre-highlighting; when the mouse hovers over a particular element, its color will change indicating it is a *selectable* element. In this case, hover the mouse over the upper round polygon and simply LMB+click to select. The 'Polygon Bevel' tool is found in the 'Polygons' subtab of the toolbox. It can also be accessed from the 'Shift+B' keyboard shortcut. With the tool active, LMB+click in the 3D viewport to again enter interactive mode for the tool. The bevel tool has two handles, one to offset the polygon, and another to scale, we're simply going to scale the polygon (drawing the additional edges that will become the top of the mug), so LMB+drag over the red handle until the 'Inset' values read 5mm.



Next, it is necessary to bevel the same polygon again, but further editing of the handles will simply modify the initial bevel unless the tool is dropped and reactivated. Luckily, modo is designed to save steps, while the tool is still active, users can hold the 'Shift' key down and LMB+click anywhere in the 3D viewport (away from any tool handles) and reset the bevel tool allowing an additional application of another bevel without the need

to drop and reactivate the tool. This time LMB+click and drag the blue handle down to a value of '-95mm'. Press 'Q' now to drop the tool. Activate the 'Scale' tool again (by pressing 'R') and again scale the selection to 70%, just as before. (You might be wondering why the inset of the bevel wasn't used and instead replaced with a 'Scale'. Since the Bevel specifies the inset as a measurement, it was easier to keep the inside consistent to the outer rim by giving it the same *scale* value).

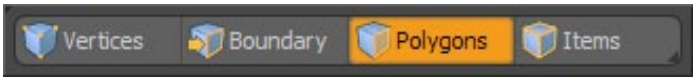
Indirect Selections



The mug in the photo has nice rounded edges, unlike this model so far, so we need to round them using the 'Edge Bevel' function. Since we only want to affect certain edges, we'll need to make a selection, but it will actually be a little difficult to get inside the cup to accurately select the edges necessary, so modo has way to indirectly make the selection we want. (Making sure you're still in 'Polygons' selection mode) select two adjoining polygon faces of the top rim. We want to select the entire loop, so once the initial selection is made, pressing 'L' again will select the connected faces in the loop, just like it did for the edges; the difference being that two polygons needed to tell modo which direction to run the loop. Users can add and subtract polygons from a selection easily by holding the 'Shift' and 'Ctrl' keys on the keyboard respectively, while making selection edits. So to add additional polygons to the selection, hold down 'Shift' and LMB+click over the interior polygon and them tumbling the viewport (hold down the 'Alt' key and drag in the viewport to rotate the view), hold down 'Shift' again and LMB+click over the bottom facing polygon.

The indirect part is the current Polygon selection defines the selection we want with its *border*, therefore we need to convert the polygons selection to the type we want. modo offers several ways to convert a selection of one type to that of another. Holding down

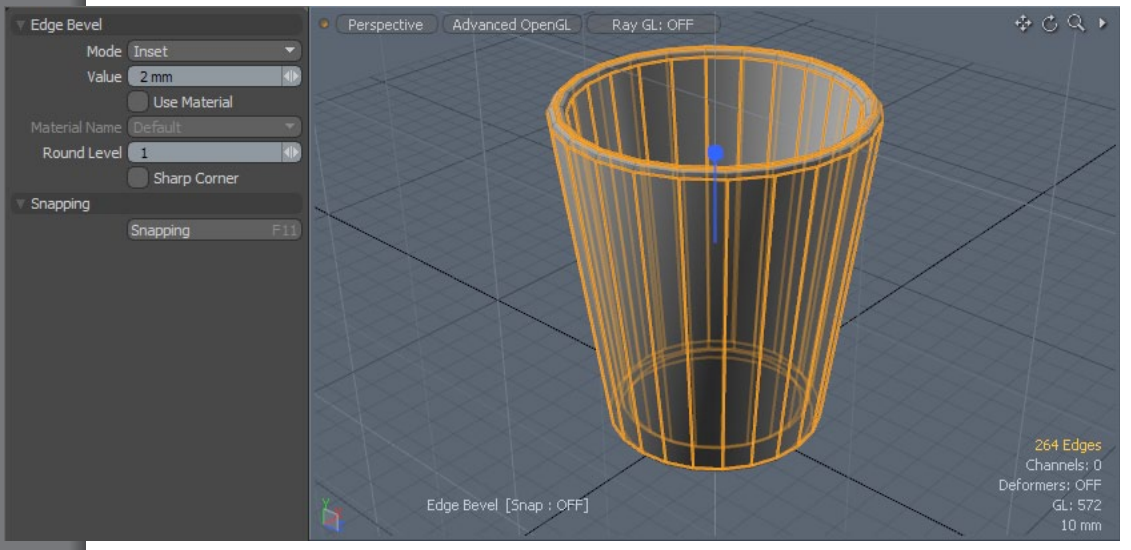
the 'Ctrl' key changes the 'Edges' selection mode button into a convert to 'Boundary' button. LMB+click the Boundary button to convert the selection, entering 'Edges' mode automatically. We now have just the border of the polygons selected.



Now with the selection we do want, switch to the 'Edge' subtab of the toolbox and select the 'Bevel' tool there, LMB+clicking in the 3D viewport to activate the tools interactive mode.

Edge Bevel tool icon

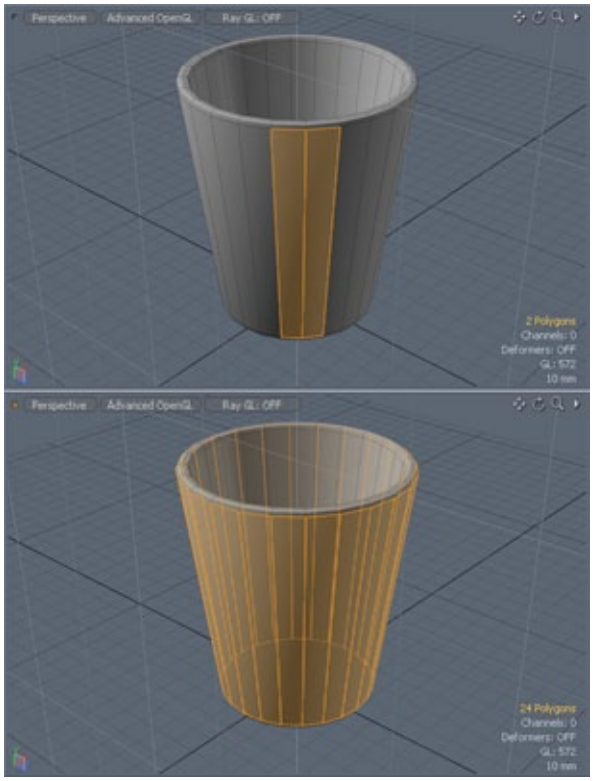
This is a case where it might be easier to simply assign the value in the Tool Properties viewport instead of dragging the tool handles. Since the tool is in interactive mode, we can make modifications to the values an see them update the viewport. We want a 2mm bevel value with a 'Round Level of '1' as illustrated below. That will give us nice rounded edges, just like in the original mug image. Once that is applied, that pretty much covers the mug body. You can press 'Q' to drop the tool and 'Esc' to drop the selection.



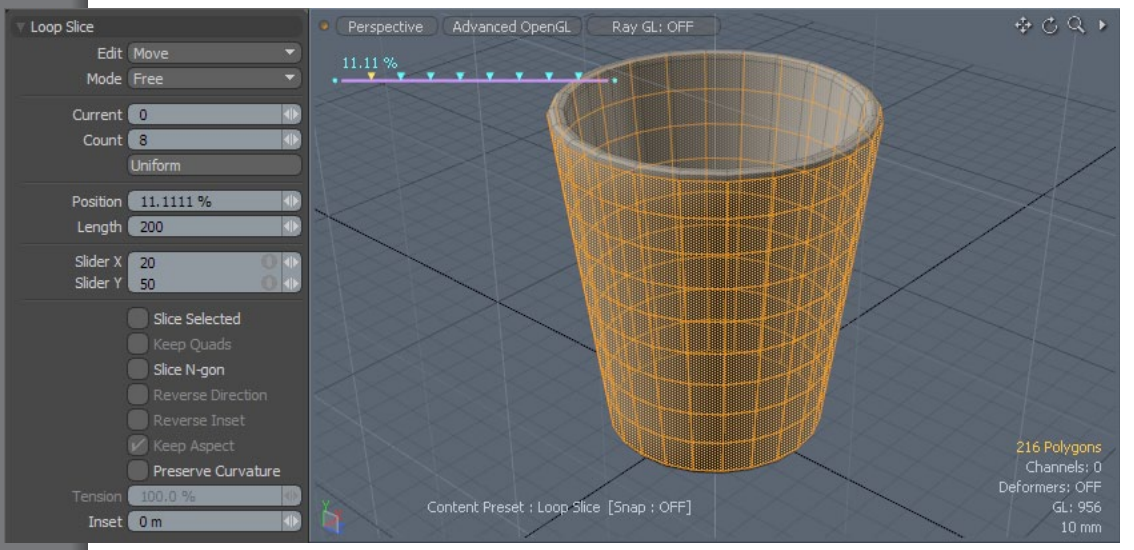
Thus far, we're gone over the Work Plane and its use, tool handles, some keyboard shortcuts, making selections, converting selections as well as the application of some pretty useful tools. All the basic building blocks of making more complex geometry, take a break if you like, making sure to save your object, or continue on.

Making the Handle

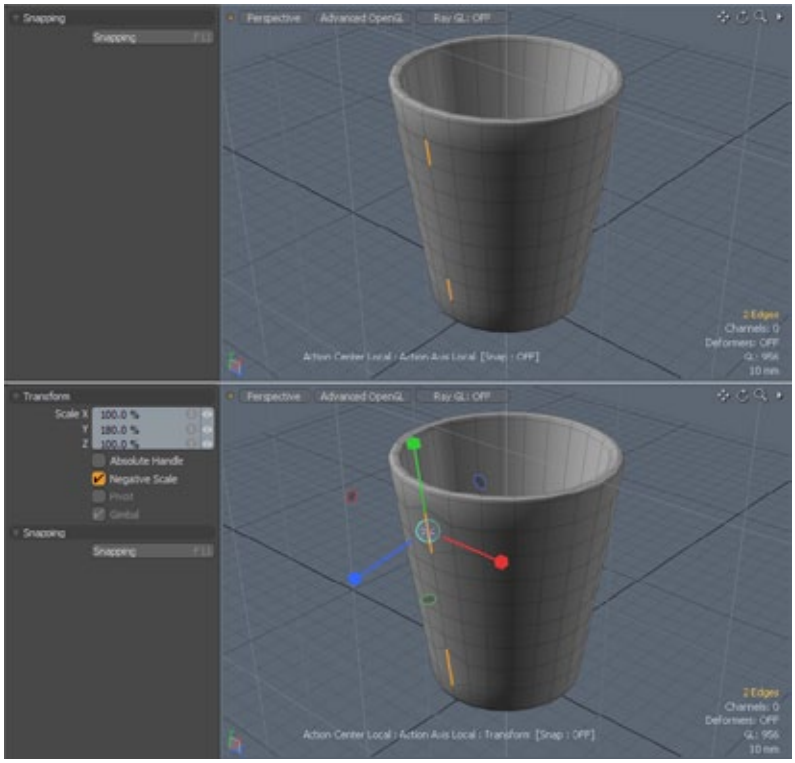
We want our object to be one continuous smooth surface, so in order to add the handle properly, we'll need to add some additional geometric edges to *grow* the handle from. We could've added subdivisions (additional polygon edges) earlier when we first created the cylinder, but that would've complicated further procedures, so modo allows users to add edges at any time. We'll do this by using the 'Loop Slice' tool. First we'll need to make the selection for our outer surface to apply the slice to, so again select two adjoining polygon faces, this time on the side of the mug and press 'L' to select the loop.



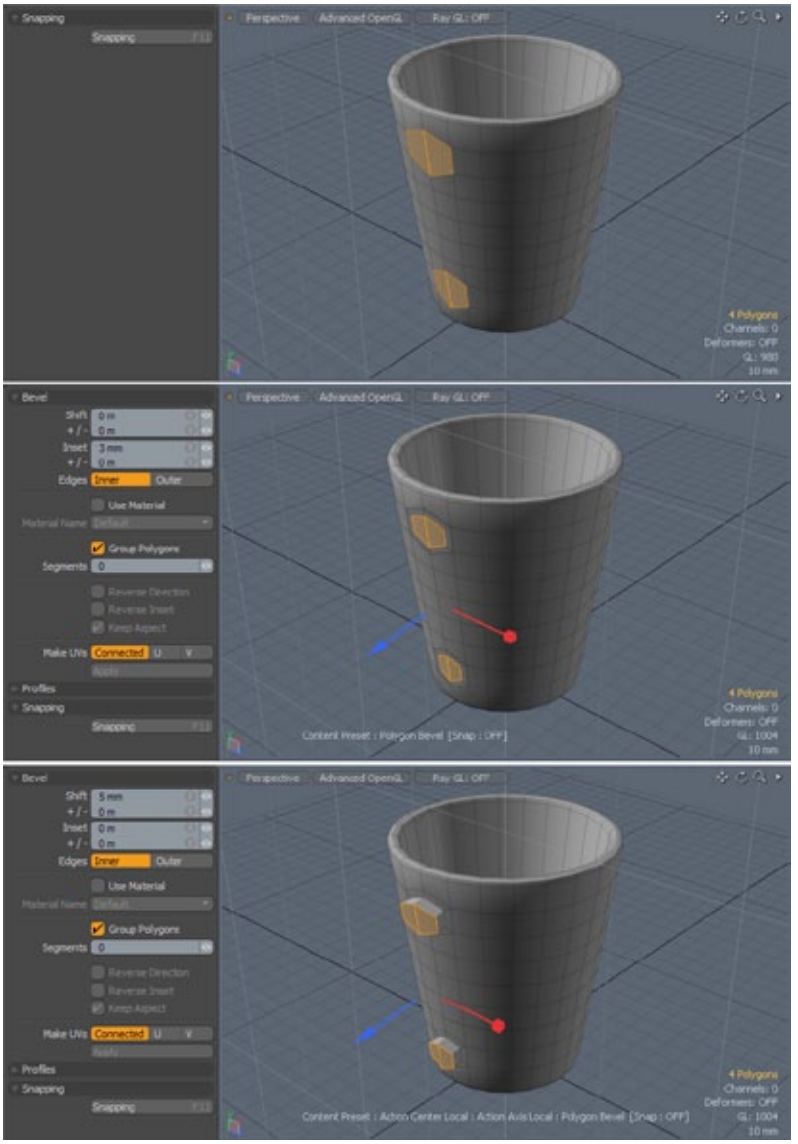
The 'Loop Slice' tool is found under the 'Mesh Edit' sub-tab of the toolbox. Users can also press 'Alt+C' to select the tool. LMB+click in the 3D viewport to activate interactive mode. This will draw a single slice halfway up the side of the mug, we actually want 8 slices, so in the tool properties panel, enter '8' into the 'Count' input field. The model should then look like the following. modo does all the hard math automatically, to make sure all the slices are evenly spaced. Drop the tool again and the selection and press the Spacebar to cycle into 'Edges' mode for the next operation. If any edges are already selected, press 'Esc' to drop them.



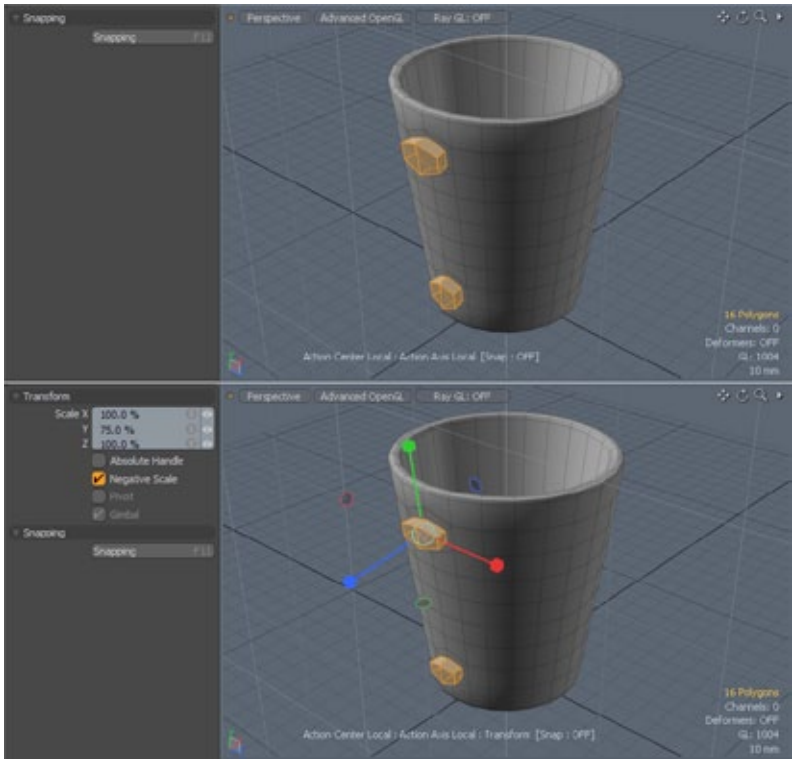
In order to give the handle its slightly rounded shape, we'll need to adjust the area where the handle attaches to the body. Select the two edges one span away from both the top and bottom of the mug, illustrated below (making sure they are the edges aligned to the side 'X' axis). Next we'll scale them both lengthwise. I don't want to do it as two separate operations, modo offers special editing modes that allow clusters of selections (or in this case single edges) to have their own independent centers.
To activate it, we must modify the 'Action Center' setting. These define where the operation originates from, i.e. where the tool handle is placed, and its axis direction (orientation). To the right of the modes tool bar is the 'Action Center' selector button. LMB+click to open the popup menu and select the 'Local' option. Next, press 'R' to activate the scale tool, and scale only the 'Y' (green) axis to 180% by LMB+clicking over the green handle and dragging the mouse upward. See how both scale from their own center in only one application, that's pretty cool.



The handle isn't quite as thick as these two shapes suggest, so let's bevel them inward somewhat to make a more reasonable sized handle shape. Toggle into 'Polygons' mode, and select the polygons flanking the edges you just scaled (using the methods you learned above). Activate the polygon Bevel tool; 'Shift-B', LMB+click in the 3D viewport (don't be alarmed if the tool handles doesn't appear in the correct location, it will still work correctly). Adjust the 'Inset' (red) handle to 3mm, like before, press 'Shift' and LMB+click somewhere in the 3D viewport to reset the bevel tool, then bevel outward 5mm and drop the tool.

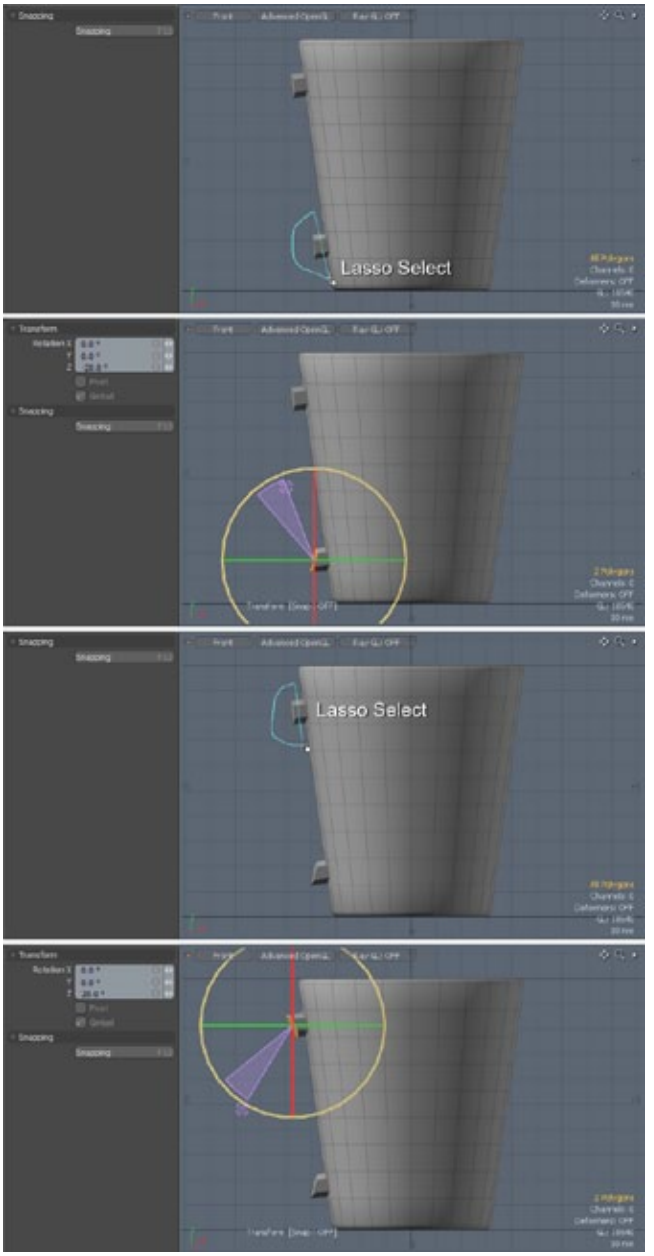


The height of each shape is a little tall, so lets scale it down some. I want to add to the selection here, and modo offers some shortcuts for doing just that. With the 4 polygons still selected, press the 'Shift+Up Arrow' key on the keyboard. (The arrow keys is sometimes in a cluster for all four directions, or as a toggle for the numeric keypad). This will grow the selection one step by selecting all adjoining polygons. Neat. Now, press 'R' to activate the 'Scale' tool and scale just the height of the handle buds to 75% on the 'Y' (green) axis ('Local' Action Center is still active from the earlier procedure, so you should be able to scale each one independently with the one action).



The next step is to connect the two buds to each other, modo will do this somewhat automatically, using the Bridge tool, literally growing a bridge of polygons between two selections. But for this to work, the polygon groups need to be facing somewhat toward each other (otherwise modo has no idea how to connect them). This will be the most complex task of modeling the mug, but it shouldn't take more than a few minutes. It's possible to make the selection we want in the 'Perspective' view, but it will be a little bit easier to see the results we want in one of the Orthographic views; this means there will be no perspective, giving us a good look at a single side of the mug.

In the top of the 3D viewport are three capsule shaped buttons (though not like the regular interface buttons for tools and such). LMB+clicking on these buttons open popup menus that provide users a means to customize the viewports display. We just want to adjust the vantage point, so where it says 'Perspective', LMB+click on the button and select the 'Front' view. If the rotation of your initial mug matched the one here, you should see the mug with the small handle buds off to the left side. If not, try one of the other views, such as 'Left' or 'Right'.

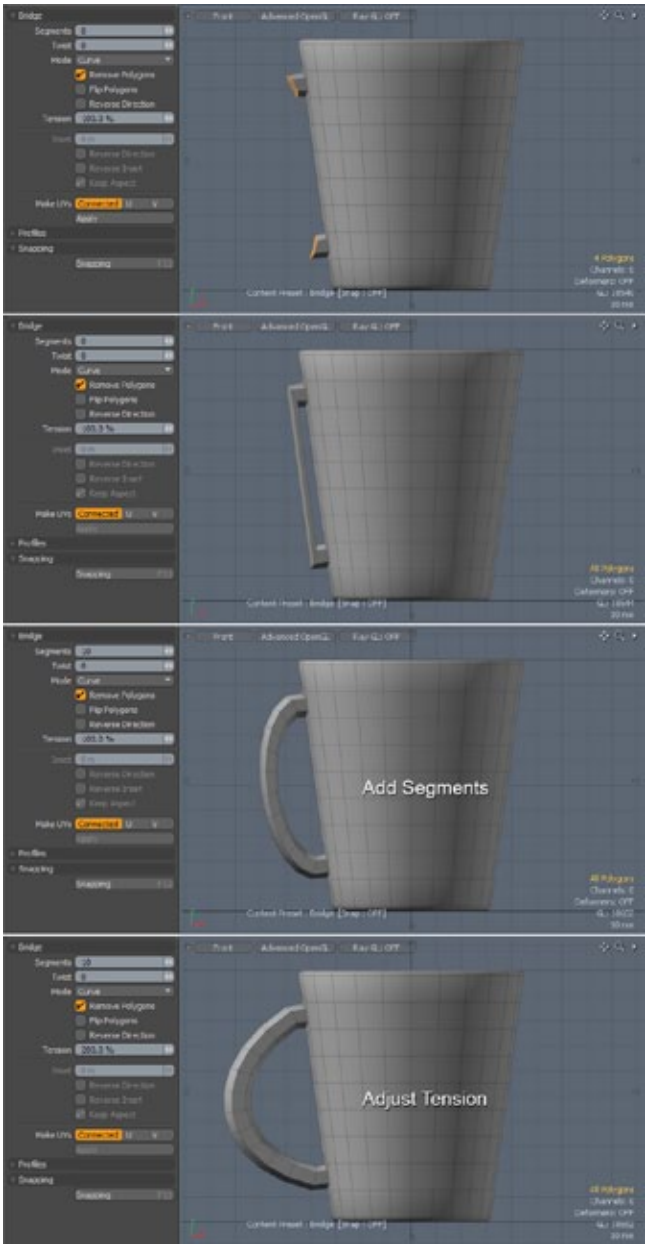


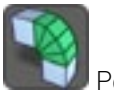
modo offers another way to select polygons beside the brush selection. It is called a lasso (or marquee) selection, where users draw a shape on the screen and all geometry that is contained entirely within the drawn area will be selected. This is helpful for when a lot of polygons need to be selected, or you don't have a good view of what is to be selected. To make the selection we need, first drop any existing selection first by pressing 'Esc'. Next we'll draw the lasso area. This is done by way of the middle mouse button (MMB). If you're using a mouse that does not have a middle mouse button or equivalent, then you will need to also switch the viewport into 'Wireframe' view. This can be done by selecting the viewport button that says 'Advanced OpenGL' and from the menu, select 'Wireframe'. I'll explain why. modo's behavior (by default) is to only select what users can see in the viewport. It makes sense that you would only want to edit what you see, however, this isn't always the behavior you want because sometimes you do want to edit those unseen or rear facing polygons, so there are two distinct methods for lasso selection geometry, the rights mouse button (RMB) lasso selects only what is seen, and the middle mouse button (MMB) selects through the model. When switching to wireframe mode, the rear facing polygons now

become visible so the lasso selection will select through the model, so for that particular mode the behavior is reverse; the MMB will only select the forward facing polygons. Now that you understand the different lasso options, lets make the selection already.

Pressing and holding the MMB, as you drag the mouse you will draw a line on the screen, this is the lasso that selects the polygons. You want to draw the shape just big enough to contain the end of the handle nub. With the two polygons now selected, press 'E' to activate the 'Rotate' tool, and using the round handle for the 'Z' axis (blue circle) rotate the polygons to face upward 20 degrees. Do this same action for the upper nub, but rotate the nub to face downward 20 degrees.

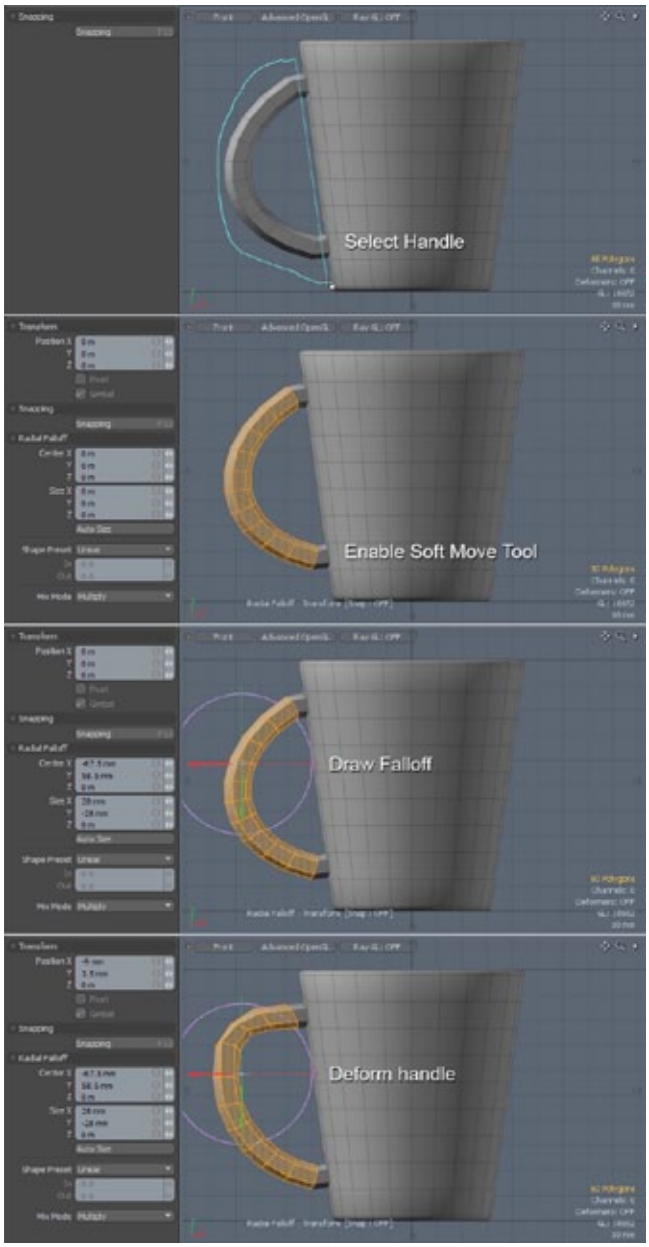
Now to apply the Bridge. Hold down 'Shift' to add to the existing polygon selection, so just the ends of both nubs are now selected. The Bridge tool works by connecting groups of polygons, so the lower group will connect with the upper group. Since they both have identical numbers of outside edges, it will match up very nicely. Activate the Bridge tool on the 'Polygon' subtab of the toolbox.



 Polygon Bridge Tool

LMB+click in the 3D viewport to enter *interactive* mode. A single set of polygons is drawn as a straight line between the groups. That's OK. In the Tool Properties panel, set the 'Segments' value to '10' to add some additional geometry for the bridge to work with, and then adjust the 'Tension' value to '200%' curving the handle further out.

That makes the geometry for the handle. But it doesn't match the shape of the reference image all that closely. modo has some tools specific to deforming a shape so we can twist the geometry that is there into something that more closely resembles the original image (we won't be matching it exactly, this is just a tutorial after all). modo applies a technique called a 'Falloff' to attenuate, or fade the affect any tool has on a selection. Normally when you move geometry, it all moves the same amount, as a single rigid piece, but the falloff lets you move one area more than another, with a soft fade between the two extremes. Users can apply a Falloff manually, but for the sake of simplicity here, we'll be using a tool that employs a falloff natively. First, select the new bridged polygons of the handle using the Lasso selection method. Then, under the 'Deform' subtab of the toolbox select the 'Soft Move' tool.

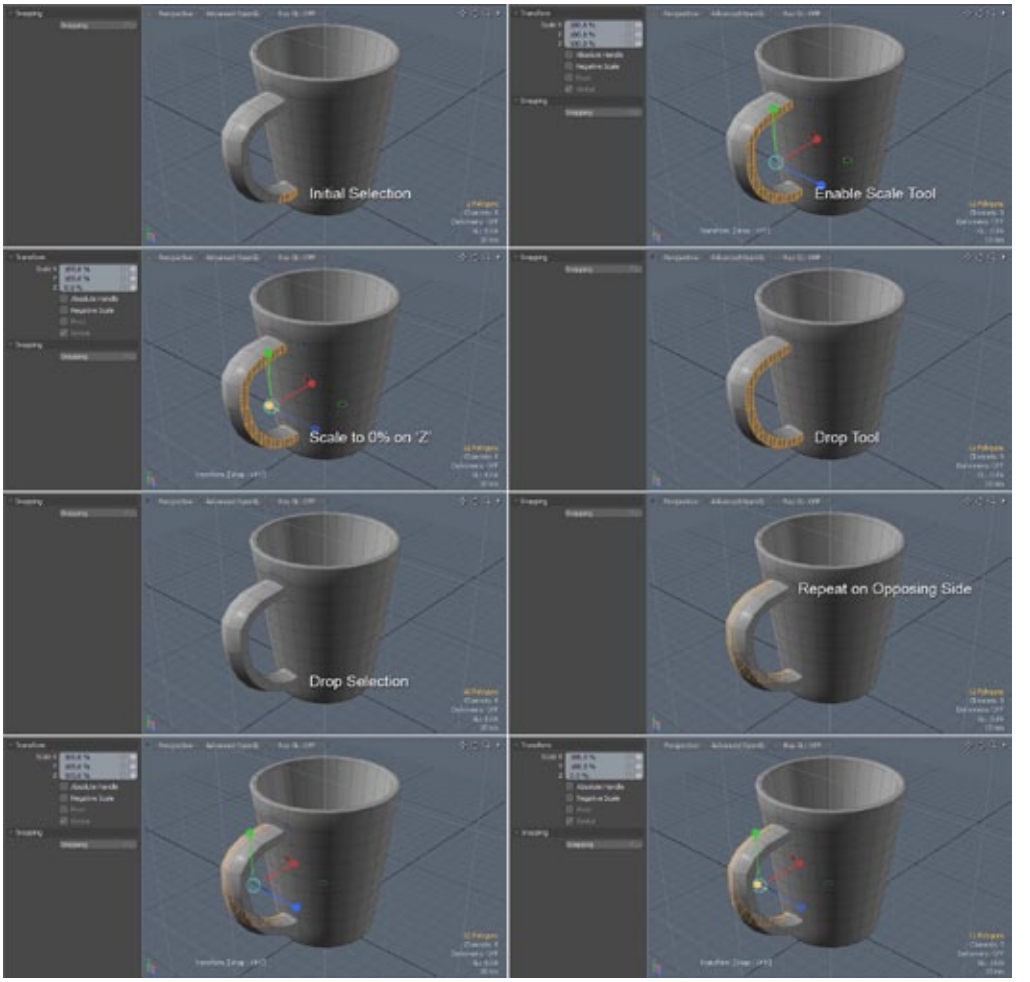


Soft Move tool

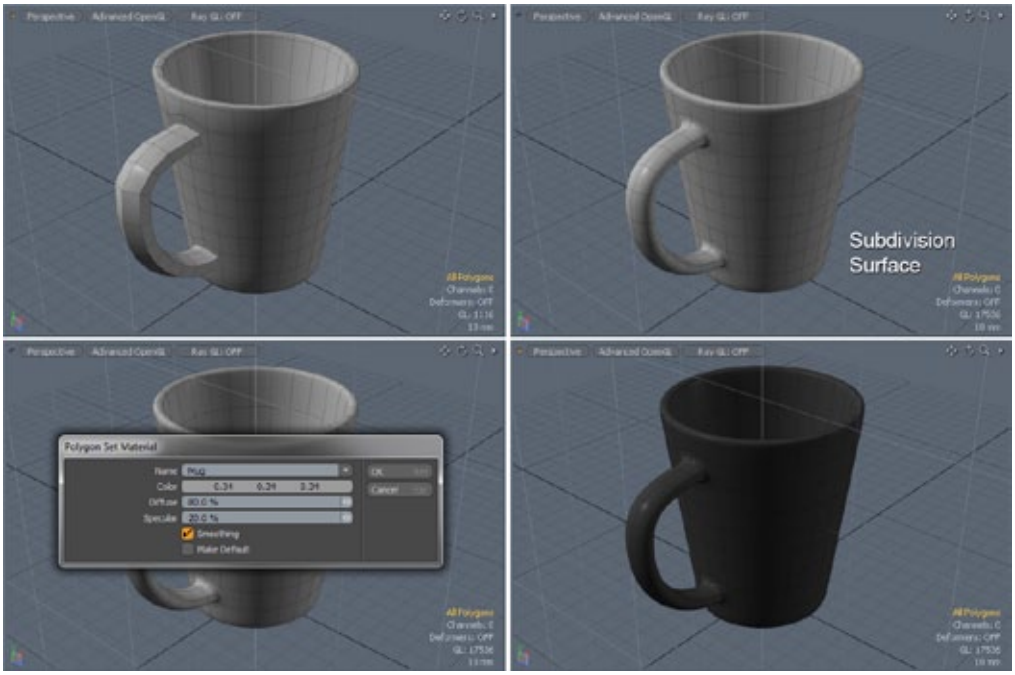
The 'Soft Move' tool requires two actions before it will deform the model properly. The first action will create the 'Falloff' and the second deforms the model. With the tool active RMB+click and drag in the upper left area of the handle itself to define the Falloff radius. Once the radius is created in the view, you can LMB+click and drag any of the handles to make subsequent adjustments (just like with a regular tool). Once you have the Falloff the appropriate shape (it doesn't have to be perfect) LMB+click and drag toward the upper left corner of the 3D viewport to deform the handle selection. One of the great benefits of a Falloff is it is live while the tool is active, so users can LMB+drag on the center handle of the falloff to reposition it and see in realtime the deformation change. You can move the Falloff and adjust the Transform amount till you have a shape you're happy with. Once finished, press 'Q' to drop the tool and keep the deformation.

Now the handle is a continuous extension of the body and almost completely finished. Next, switch from the 'Front' view back to the 'Perspective' view to make the last edit. The bottom of the handle is not as wide as the top, where in the original mug, the handle is an even width along its length. This can be fixed with the scale tool. Before

proceeding, select the 'Automatic' Action Center type, disabling the current 'Local' (otherwise the next step won't work as expected). This centers the action origin at the tool handles location and keeps the axis alignment upright. We'll also use another modo selection shortcut to make quick work of this edit. modo has the ability to figure out what you want to select next based on the last couple selections. Its called selecting the next plausible element. If you paint select just the first two polygons at the base of the handle and then press the 'Up Arrow' key, modo will continue selecting the next logical polygon in the loop. Each time the button is pressed it will add another element to the selection. Select the entire side of the handle this way and then activate the 'Scale' tool (R key). Scale the selection to 0% on just the 'Z' axis. This can be done easily using the interactive handle when the 'Negative Scale' option is disabled in the tools properties panel. Apply this operation to both sides of the handle. This will equalize and straighten them finishing off the mug.



The last operation is to apply Subdivision Surfaces smoothing to the model by pressing the 'Tab' key. This adds additional polygons to the model that are like *virtual* geometry controlled by the base low-resolution mesh. The new smooth surface looks much more like the original model reference, so all that is left is to apply a material tag for surfacing the mug in the next section. Press the 'M' key opening the 'Set Material' dialog. Assign a name to the mug, adjust the color to taste and press 'OK'. Ensure no polygons were selected or else the tag will only apply to the polygons that were, which is another good point to bring up. In modo there are implied selection, they are all covered in the modeling section of this documentation, but the most important one is that when no polygons (or any geometry components for that matter) are selected, then they are all considered selected, so any operation applied, applies to everything, hence why having no polygons selected when assigning the material applies to all the geometry. Once done, save your model.



In this second section, you're learned about slicing geometry, adding edges to use for additional modeling tasks, using some more advanced tools, like 'Bridge' and had some basic experience with using a falloff to deform the handle, also some additional shortcuts to use when selecting geometry and what an implied selection is. In the next section, we'll set up a simple scene and render out our mug into a finished image.

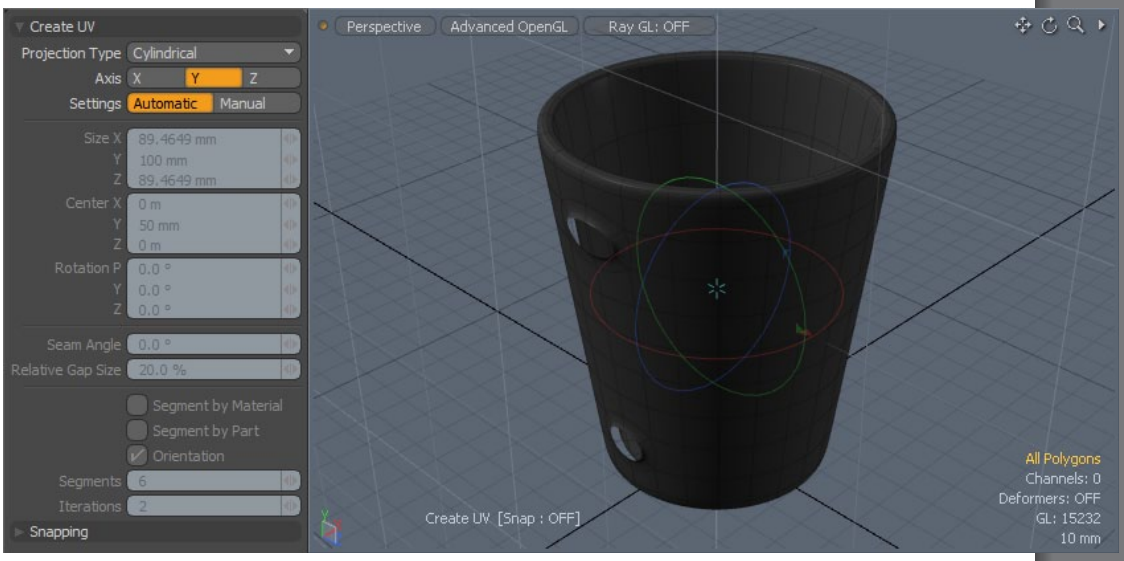
Rendering Tutorial

This is a continuation of the Coffee Mug Modeling tutorial. The information presented here builds on the lessons learned in that tutorial. In this section we'll be rendering the mug to produce a final image. With the Mug model open, make sure you're in the 'Model' interface tab and let's get started.

It would be simple enough to just press 'Render' and have an image of the coffee mug at this point, but it wouldn't be that interesting of an image, especially with the default settings. And while rendering a single object has its place, in most cases users will be using modo to render out a scene, where there are multiple objects, a background, and some distinct lighting. In this Rendering section we'll be assigning a UV map, adjusting the surface of the mug and cloning it, using a preset to make a basic scene, and then adjusting camera and render item settings to produce pleasing results.

Making a UV Map

The first thing we want to do is to apply a UV map for later surfacing. If you're not familiar with what a UV map is, don't worry, this will be a very simple process. UV maps are simply a way of assigning how a flat texture gets applied to a 3D surface at render time. To make things a little easier, I want to hide the handle first. Using the lasso selection method covered in the previous tutorial, from a side view, drag out a shape, selecting just the handle. It will be pretty tough to select the entire handle without also selecting some of the body polygons, so if you just select the main part of the handle, you can also use the grow selection shortcut- 'Shift+Up Arrow' to add the adjoining polys where the handle meets the body. Once the entire handle is selected, press 'H' to hide the geometry. Hiding geometry is a great way to concentrate on just a group of polygons in an object layer, and to ensure that geometry remains unmodified, as geometry that is hidden cannot be selected, or edited.

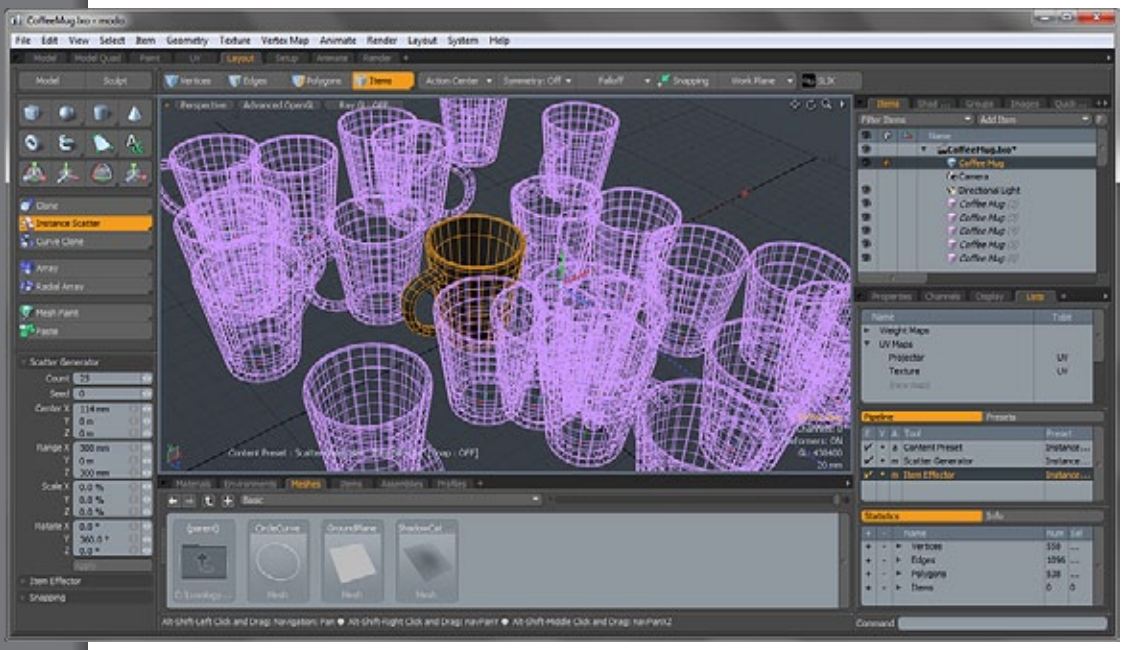


In the bottom subtab of the toolbox are some UV tools, not as many as in the 'UV' mapping interface tab, but enough to do what we need to do here. Select the 'UV Projection Tool' toward the top of the toolbox and LMB+click in the 3D viewport to activate the tool. In the tools properties panel, set the 'Projection Type' to 'Cylinder', making sure the axis is set 'Y', so the map will wrap around the cups height. Leave the 'Settings' on 'Automatic'; this sizes the cylinder volume to match that of the selected (or in this case visible) polygons outer bounding box volume. (A bounding box is simply a rectangular volume of space the represents an objects total height, width and depth). That's it, the map is made. Press 'Q' to drop the tool accepting the UV map values. The map you just created in this way is sort of like the equivalent of shrink-wrapping gift wrap around out 3D object, which also happens to be a cylinder, producing a perfect wrap. Next, we'll assign an image map that matches the logo on the original reference image. Press 'U' to unhide the handle, making it visible once again.

We're pretty much finished now with the 'Model' interface tab, so switch to the 'Layout' tab for this next section. We'll be focused on laying out the scene for rendering, and this tab has all the functions for doing just that. The Layout section is unique in that it has tools for cloning geometry, which is precisely what we are going to do next. Before we continue there is a rather important concept users must understand to use the tools to their best benefit. So far you've used the 'Polygons' and 'Edges' selection modes to modify the geometry, now while laying out the scene we'll be working in 'Items' selection mode. In 'Items' selection mode, users can only modify an items position, rotation and scale, collectively known as its transform values. The previous modes, 'Vertices, 'Edges' and 'Polygons', collectively known as the component editing

modes are strictly for editing the geometry itself. Two distinct operations take place in either the component or 'Items' modes (though their result will be identical, which often causes confusion), so they can be thought of in this way that each item layer in the 'Items List' is a container for geometry. While in Items mode, you transform the container, but while in any of the component modes, only the *contents* of that container can be edited.

The upper right viewport of the Layout interface has its own series of tabs and users can select the 'Items' tab to get a list of all the available items, or layers and they are sometimes referred to in the current scene. So, now while in 'Items' selection mode, make sure the mug layer is selected in the 'Items' list. LMB+click on the layer name (likely called 'Mesh', highlighting the layer. Now any operations applied to the scene will only affect that selected layer. The toolbox on the far left side of the screen has a row of buttons. Hidden underneath some of the buttons are alternate tools. These *hidden* tools are specified with the small gray arrow in the lower right corner of the button itself. LMB+click and hold on the button to open a context menu displaying all the alternate options. The actual tool we want is hidden under the 'Scatter Clone' button, so LMB+click and hold over that button to open the popup and then choose the 'Instance Scatter' button, enabling the tool.



LMB+clicking the 3D viewport will activate the tools interactive mode. You might not see anything in the viewport right away, aside from a tool handle, but if you press the 'A' key while hovering the mouse cursor over the 3D viewport to view all items in the

scene, you should see 8 coffee mugs scattered all around. In the tools property panel, you can adjust the 'Count' value to any number you like, I wanted quite a few mugs, so I set mine to 25. I also want all the mugs to rest on the ground, so the 'Range' for the vertical axis 'Y' was set at 0m. To randomly rotate the clones, I set the 'Y' rotate value to 360°and then adjusted the range on the 'X' and 'Z' axes to bring the mugs closer together. The Clone tool has no idea about intersections, so some mugs will overlap, we'll fix that in the next step. If you're not happy with the placement of the clones, you can adjust the 'Seed' value till you receive a result you are happy with. Once finished, drop the tool.

Now in addition to the original mug, you now have a number of pink wireframe clones. That pink color denotes that the duplicates are indeed *Instances*. Instances are virtual copies that reference the original mesh item. I chose them for two reasons, one, they contain no real geometry, just a reference to their originator, so they keep the scene file light, and second, any modifications made to the originator will propagate to the clones making it easy to modify them later on. Unlike other types of duplicated objects, such as Replicators, users have individual control over each Mug instance, so while still in 'Items' mode, users can LMB+click on any mug to select it, then press the 'W' key to transform the mugs position. If you LMB+click on the green circle handle, it will constrain the item to the ground plane, making it easy to modify each items position. Press 'Q' to drop the tool, select another Instance and repeat until you have all of the mugs positioned so they are no longer intersecting. Pressing 'Alt' and tumbling the viewport will help in getting a better view of the scene when making adjustments. 'Pressing 'Shift+A' will zoom the currently selected items to fill the entire viewport, making it a little easier to navigate around amongst the multiple items. I also added some rotation here and there and ended up with the results shown below.

The 'Items' list has now gotten a little over-filled with layers, so first I want to organize that a little. LMB+click on the top most Instance layer, and scroll to the bottom and hold the 'Shift' key, and LMB+click on the bottom most layer, selecting all the layers in between. Now press 'Ctrl+G' to group all the layers together into a 'Group Locator'. Toggling the small '▶' arrow icon preceding the name will hide all the layers in the list making it easier to navigate.

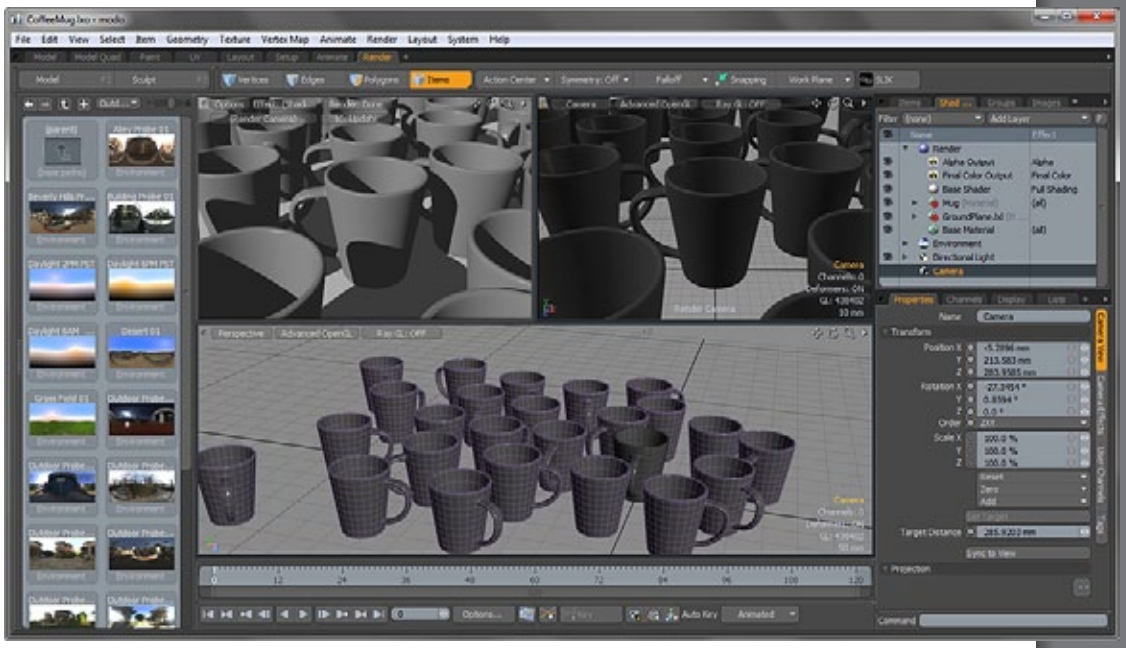
We need just one additional piece of geometry to fill out the scene and for that we'll use a 'Preset' mesh. In the 'Preset Browser' across the bottom of the screen, select the 'Meshes' tab and locate the 'Basic' folder, opening it. Inside you will find a few simple objects. LMB+double click on the one named 'Ground Plane'. This will serve as the ground in our scene, most importantly catching cast shadows, so the rendered mugs do not look like they are floating in space. Next, we'll move on to Surfacing and Rendering, make sure to save your scene before continuing.

Setting Up the Render

For the rendering portion of this tutorial, we'll close it out working in the 'Rendering' interface tab, so switch to that now. In the upper right viewport, expose the 'Shader Tree' panel, all the surfacing and render settings are set through the items in this list.

First, we want to position the camera. That can be done in the main viewport area. There are three 3D viewports to work with in the Rendering interface, a render

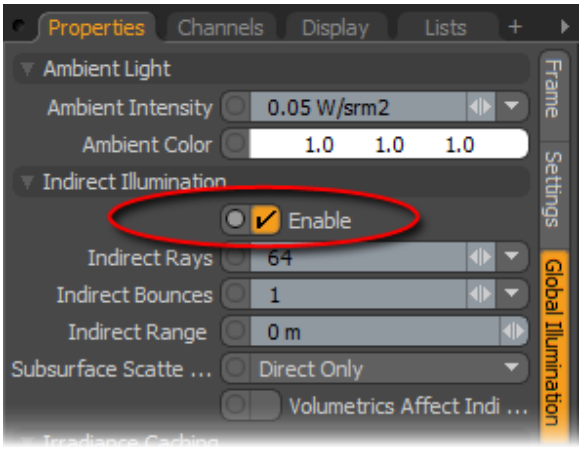
'Preview', giving a quick reduced quality sample of the final image, a Camera view that shows exactly what the camera will see geometry wise, and an advanced OpenGL 'Perspective' view similar to the one in the 'Model' interface. By LMB+clicking in the upper right, you can navigate around, using the standard viewport controls, 'Alt+LMB+click and drag' to rotate the view and 'Ctrl+Alt+LMB+click and drag' to zoom in and out. Although when doing these action in this viewport, users are actually moving the 'Camera' item, this is a great time saver, where there is no need to select the Camera item and then activate a transform tool. As you change the view, you will see the preview update in real time. I know I wanted my scene to center on my initial mug model, so I zoomed into that one quickly by selecting it and the pressing 'Shift+A', and then slowly zooming out, moving the camera backwards until I had the view how you wanted it. I also wanted a slightly wider angle of view than the default settings provide; this can also be adjusted interactively by holding the 'Ctrl+Alt' keys down and RMB+dragging in the viewport. Feel free to move any of the Mug items once the camera is set using the perspective viewport. Its a huge time saver to see how the rendered image will look while you're making scene adjustments. Once the Camera position is set, and all the items are in place we can get to the surfacing and rendering. Once you are happy, make sure to save your scene.



Surfacing the Mugs

Before we start surfacing the mug, I want to adjust the lights, enabling Global Illumination. Lighting plays such an important role in a rendered scene, I like to work

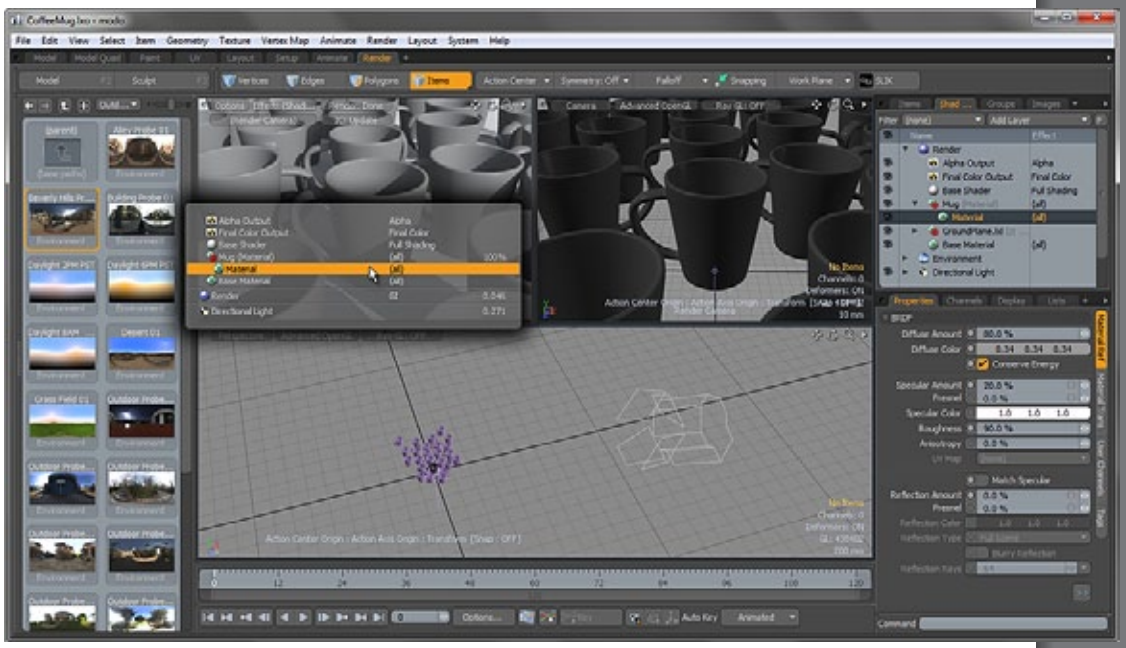
in a facsimile of the lighting I'll use for the final render when adjusting surfaces. In the Shader Tree are the various items or layers that represent the shading or rendering of our scene. Selecting one of the layers will display its attributes in the 'Properties' panel directly below. If it is not visible, select the 'Properties' tab, exposing the viewport. The very top item in the Shader Tree is the 'Render' item, this item holds the settings related to rendering the scene, such as frame size, Antialiasing and other quality related settings and also Global Illumination. Global Illumination is technique used in 3D graphics to mimic the effects of real world light bouncing around in a scene, producing extremely realistic results. On the right side of the 'Properties' form are the subtabs, select the 'Global Illumination' tab and then under the 'Indirect Illumination' options select the 'Enable' toggle, leaving the rest of the settings at default. Already you should notice a dramatic change in the 'Preview' viewport.



When enabling 'Global Illumination', modo lights the scene using the settings of the 'Environment' item. By default, this comes from a gradient applied to the 'Environment Material'. I prefer the look of Image Based Lighting, so I want to add an HDRI environment. This can be done easily using the presets modo provides. In the 'Preset' browser on the left side, under the "Environments > Outdoor" folder, LMB+double click on the 'Beverly Hills' preset (just like adding the ground mesh earlier) to add the Image to the Environment item. The lighting will again dramatically change for the scene in the render Preview producing a nice outdoor look to the rendered image, already improving it quite a bit from the initial settings. The lighting in the Beverly Hills image will be coming from the right hand side for my image (-X), so I also want to rotate the Directional light to try to match that. This can be accomplished most easily in the 'Perspective' view across the bottom. Press 'Ctrl+Alt' and dragging in the viewport to zoom out and then select the 'Directional Light' item in the viewport, but LMB+clicking over it when it highlights. With the item selected, modify the 'Action Center', selecting

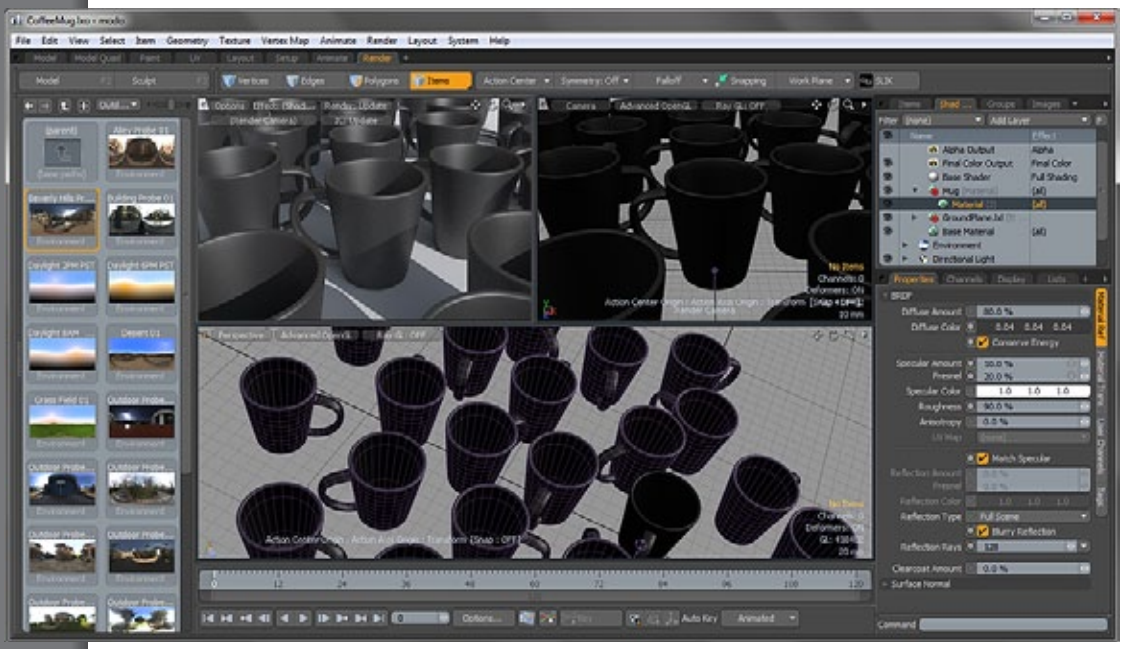
the 'Origin' option, allowing you to easily rotate the Light from the worlds center. Press the 'E' key to activate the rotate tool and then grabbing just the 'Y' axis handle (the Green' circle) rotate the light about 45° to the right.

Now onto the surfacing. Earlier in the modeling tutorial, a Material tag was applied to the model, named 'Mug'. This action also created what is called a Material Group. Not to go into too much detail in this tutorial, but this particular item creates a procedural mask that limits any surfing contained under the Group in the Shader Tree to just surfaces tagged with that name. It's covered more in-depth in the Shader Tree documentation. Each 'Material Group' mask contains a 'Material' item that controls most of the attributes of any surface- what color it is, how shiny it is, etc. We need to select this item in order to edit it in the 'Properties' panel. While its not that hard to select the proper Material item in the Shader Tree in this case, modo has a great feature to allow users to quickly see what layers are attributed to any part of the scene, by using a shortcut in the 'Preview' viewport. RMB+click over one of the Coffee Mugs in the Preview panel, opening a context menu with a list of all the layers that apply to the mug. In the menu, LMB+click on the 'Material' item, directly under the 'Mug' group, selecting it in the Shader Tree.



Looking over the attributes in the Properties viewport, I want to first adjust the color, as it is much lighter than the mug in the original image. You can LMB+click anywhere on the 'Diffuse Color' swatch opening the modo Color Picker. Adjust the value to a much darker gray color without going completely black. The original is also quite matte

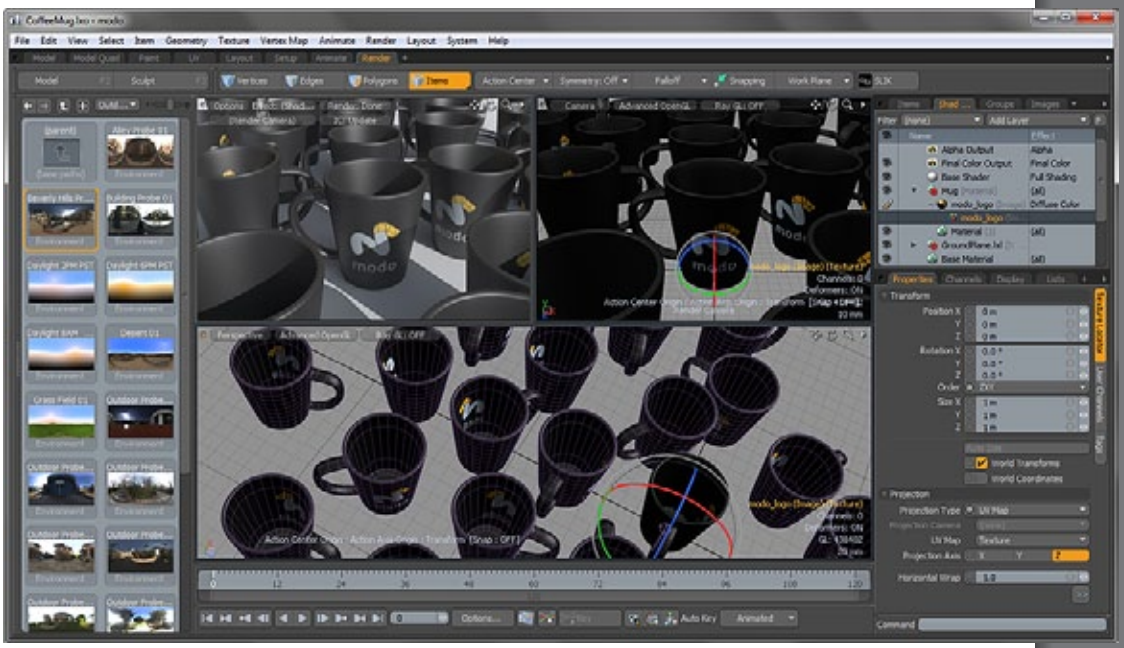
(as opposed to Shiny) so I will modify the 'Roughness' value to 90%, this softens the overall sheen, looking more like the original matte surface. I want to go for realism, so I also enable the following settings 'Conserve Energy', 'Match Specular' and 'Blurry Reflections', changing the 'Reflection Rays' setting to '128' and changing the 'Specular Amount' and 'Fresnel' settings to '10%' and '20%' respectively. These particular attributes are explained in detail in the 'Material' item documentation. They all happen to be render intensive as well, so if your system specs are modest, you may do best to keep them at the default disabled state. You can see now its much closer to the original mug.



Now we'll finally utilize that UV map we made earlier to apply an image to the mug. It's likely you don't have a texture with a modo logo handy, so grab one from this same section of the OnLine help and save it somewhere so you can apply it to the model. Using the 'Add Layer' button at the top of the Shader Tree, select the 'Load Image' option from the popup menu, navigate to where you saved the image to, and select 'Open' loading the image. The image should appear right away on the surface. If the image did not show up properly, it is probably not assigned to the UV map properly. To the left of the layer name is a small '+' icon, LMB+click on it to open the 'Texture Locator', this controls how the image maps onto the mug surface. Make sure the 'Projection Type' is set to the 'UV Map' option, and then, if there are more than one maps listed under the 'UV Map' option, select the alternate map.

The image we just applies is a transparent PNG; because of the layered approach of the

Shader Tree, the transparent edges will show the regular material color underneath out image instead of as making the surface transparent. However, the image is a little bit light. This is due to modo's gamma aware workflow. Most image editors apply gamma to an image when it is saved, an adjustment to the brightness of the image to make it more pleasing to the eye. modo also applies gamma to the rendered scene, so this double gamma makes the image appear too light. To de-gamma the image prior to rendering, select the Image Map layer in the Shader Tree, and enter the value 0.455 in the 'Gamma' value field. The issues surrounding gamma are beyond what can be covered in this tutorial, if you would like to read more, please see the 'Gamma' section of the 'Image Map' item.



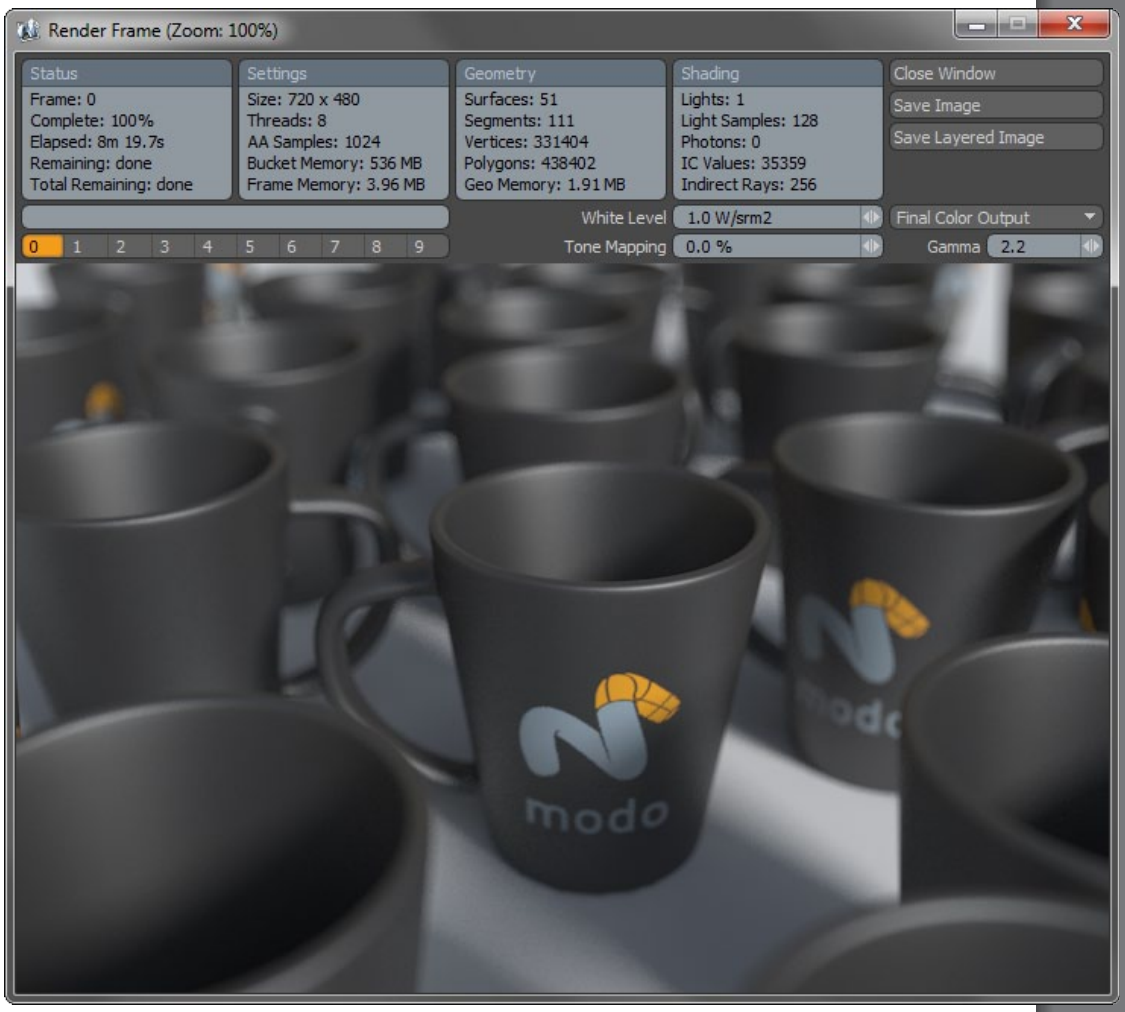
One problem you'll likely notice right away is the fact that the inside of the mug has a logo as well, since the mug in one continuous surface, it projects all the way into it, so the easiest way to remove the unwanted logo is to apply a *different* surface to the interior of the mug. This is where it turns into a good idea to have cloned using 'Instances', any changes made to the original now will propagate to all the clones. Press 'Q' to drop any tools and then 'Esc' to drop any selections. In the 'Perspective' viewport, RMB+click over any of the pink mugs and from the popup context menu that opens, select the 'Select Source of Instance' option. Press 'Shift+A' to zoom to that model. Make sure to switch to 'Polygons' selection mode, since we're now editing the model itself. Rotate the view to see the inside of the mug and LMB+click to select the base of the interior. Press 'Shift+Up Arrow' to grow the selection until in encompasses

the entire inside of the mug. Now to assign a new material tag, overwriting the previous. Press 'M' to open the 'Assign Material' dialog and name the surface 'Inside Mug', pressing OK to accept the rest of the default settings.

You can see in the 'Preview' that the insides of all the mugs are now back to the default values, but the logo is gone. I want to make the inside material match that of the exterior, so to make things easy on myself, I will first select the mug exterior 'Material' we just modified, RMB+click on it to open the context menu popup and select the 'Create Instance' option. There are now two 'Material' items in the group, but one has the name *italicized*. LMB+click on the italicized one and drag it right above the new material in the 'Inside Mug' group and release the mouse button to drop it. LMB+click to select the previous 'Material' in that group and press 'Delete', removing it from the tree as it is no longer necessary. Now the insides have no logo, the surfacing matches the exterior and since the interior mask contains an instance, any changes made to the original will propagate to the instance, so subsequent edits will effectively change both surfaces, another great time saver. Go back to the original 'Material' and make it pink, just to see, isn't that cool? You can see your progress by pressing 'F9' to render the scene. Feel free to make adjustments to the Shader Tree to get a feel for what does what if you like. Once you have the scene the way you like, make sure to save it.

The image look much better now, but I want to add a couple more effects to make it a little more realistic. Some of these settings may become quite render intensive, so if your machine specs are modest, or you don't like waiting for renders to complete you may want to proceed with caution. The first settings I want to adjust is that of the light. The shadow it is casting is a little harsh and that's is a dead giveaway that an image is computer generated. So to soften the shadows out, select the 'Directional Light' item in the Shader Tree and in the Properties viewport, change the 'Spread Angle' value to 10° and the 'Samples' setting to '128' making the light look a lot more realistic.

Now for the Camera, select the 'Camera' item in the 'Shader Tree' so we can edit its attributes. The shallow depth of field look is very popular, so lets enable that feature also. It is found in the 'Camera Effect' subtab of the Properties viewport. LMB+click the 'Enable' toggle under 'Depth of FIeld'; you can also select the 'Autofocus' button, this will cause modo to fire a single ray from the center of the camera and set the 'Focal Distance' at the first surface it encounters. Hitting 'F9' to render out the scene at this point gives me this result.



A pretty impressive image considering its just a basic coffee mug. This tutorial should've given you a good overview of working from start to finish in modo, and the steps that are required to complete project including setting up the scene, the difference between the component and Items selection modes and how to enable and

use modo's Global Illumination rendering engine. If you'd like to explore further, you may with to assign some preset surfaces to the Ground Plane, select a material in the Preset Browser and drag-and-drop it somewhere in the 3D viewport and dropping it onto the ground plane.

If the image looks too grainy for you, you can increase the Antialiasing level in the 'Render Item' under the 'Settings' subtab. For the 'Soft Reflections' and Soft Shadow options of the Material and Light items respectively, it can also help to increase the 'Samples values. Keep in mind that as the number of samples increase, image quality will increase, but so does the render time.

The modo Interface

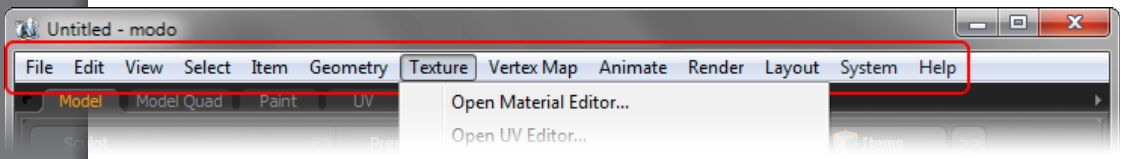
To get the most out of modo it is helpful to first understand its unique interface designed for maximum flexibility and fluid navigation. We believe in providing users with the most powerful tools, and this is exemplified by every viewport, panel and menu. The 'Controls' section provides an overview of interacting with modo's interface using the mouse, and subsequent sections cover the various windows within the application.

UI Conventions

modo follows a series of user interface (UI) conventions across the application that may be different from what some users are accustomed to, but provide a very intuitive and fluid platform for users to accomplish their *creative vision*. Understanding of these conventions is the first and most logical step in utilizing modo to its fullest potential.

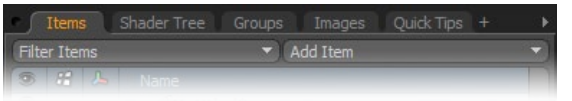
General User Interface Controls & Conventions

Menu Bar



The Menu Bar (highlighted above in red) provides users with quick access to virtually all of modo's core tools. LMB+clicking on any of the System Menu categories will activate a pull down menu with many entries and sub-categories for users to select from. The Menu Bar is actually composed from a standard modo Attribute Sheet and can be customized through the 'Form Editor' found in the 'System' section of the Menu bar. Menu item names that display another dialog are followed by ellipsis (three dots).

Tabs



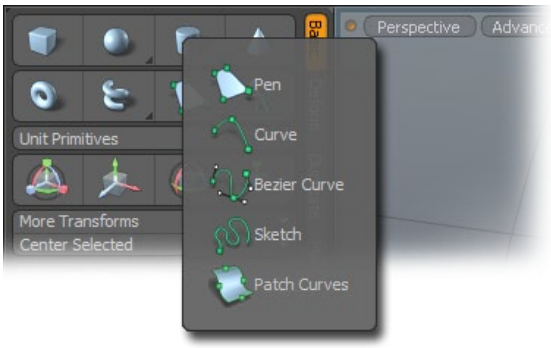
modo organizes certain aspects of the interface into what are known as 'Tabs', because of their remarkable similarities to the flaps of paper hanging from filing folders made for organize *paper* documents. 'Tabs' are a common and popular UI convention that help visually organize data. In modo they allow multiple *viewports* to exist in the same layout (or to organize multiple layouts) and are a handy way to quickly get access to certain facets of modo important only to a particular function. In some cases a viewports display is dependant on information *selected* in another viewport, which may be *hidden* under a different Tab (such as item selection displaying their attributes in the properties viewport). Tabs in modo do not auto-select, so users have the power to control what information they want displayed at a particular moment without the

surprise of viewports suddenly changing. User need only to simply LMB+click directly on a Tab to open it for display. When a Tab is active, the Title changes from a light gray to an orange color. Background tabs are a darker gray color.

Buttons

modo displays commands and state for many functions as 'buttons'. Buttons may exist as an icon, as an icon and text or as just text (and any available keyboard shortcut, if applicable). These buttons appear throughout the interface, but mainly in the toolbox down the left side of the default interface. Users may LMB+click on a button to select the function, when active the button highlights in *orange* indicating its selected (active) state. For certain buttons, like tools, they may be de-activated by LMB+clicking the tool button again or pressing the 'Q' or 'Esc' keys to *drop* the tool. For state buttons, such as the *mode* buttons, users cannot deactivate the state, but may choose an alternate buttons to change to a different state.

Button Popups



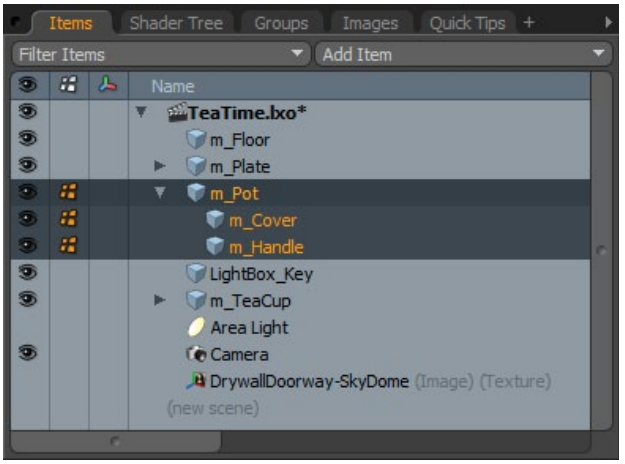
A Button Popup is a button that displays a secondary list or window. In the toolbox, these buttons are separated by a small gray arrow pointing toward the lower right hand corner. To open, users can RMB+click on a popup button to access this list and make selections from it. Alternately, users can LMB+click and hold to open the popup as well. The GL Viewport popups which allow for Viewport Type and Shading choices

are special cases; the options list is displayed when users RMB+click the buttons in the GL viewports. Popup transparency can be adjusted by going to the menu bar "System > Preferences > Display > OpenGL > Transparency > Popups" and changing the percentage there.

Alternate Buttons

Buttons can be set to perform alternate actions based on key qualifiers. As the qualifiers are pressed and released, the buttons change to show the alternate commands. Alternates are also listed on tool tips. An example of an alternate action based key qualifier is the Boundary command. When the 'Ctrl' key is pressed, the Edge button in the modo Modes toolbar displays 'Boundary' instead of 'Edges'. If the 'Alt' key is pressed instead, the Edges button displays 'Convert'. Another example is the use of the 'Ctrl' key with the Primitives. Pressing the 'Ctrl' key changes the display of the Primitive buttons, and clicking them will create a Unit Primitive of whatever type is pressed.

Item Selections



When working with any item or layer in modo, users will need to tell modo precisely which layer to work with, this is done by selecting the particular item. This can be done

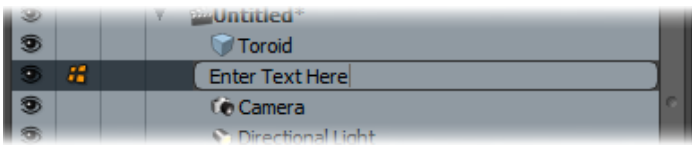
in the tree views of the 'Item List' and 'Shader tree', or in the list view of the 'Vertex Maps' viewport. To select a layer, users need only to LMB+click directly on the layer name. Selected layers are differentiated by the dark highlighting of the layer, with the layer name changing to orange. Selecting of multiple layers can be accomplished a variety of ways allowing for gang edits of multiple items. Users can select a series of items by LMB+clicking the top most item layer, and then pressing down on the "Shift' key and then LMB+clicking on the bottom most layer, selecting all the layers in-between. Non sequential layers can be selected by holding down the 'Ctrl' key and LMB+clicking to select the layers. Layers will automatically de-select when a new layer is selected, or individual layers can be removed from a selection by again pressing the 'Ctrl' key and LMB+clicking on the individual layers.

The first item selected will always be the primary selection, and it is separated from subsequent item selections by a slightly darker color and a slightly lighter layer name (it is subtle, but noticeable). This is an important function for recognizing ordered selections necessary for certain animation and item creation commands, such as when setting up Replicators.

Nesting & Hierarchies

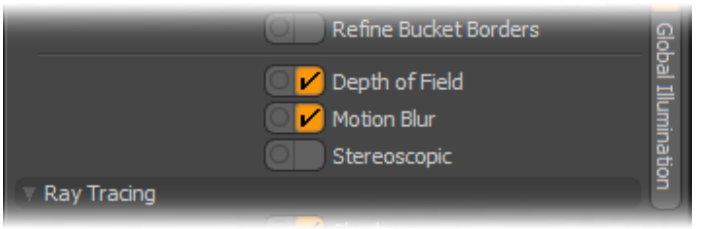
Entries in a tree may have sub-entries, which may be expanded by clicking on an arrow icon next to the parent . One column indicates the nesting, and sub-entries are indented relative to their parent. Entries may also have attributes, which may be expanded or contracted by clicking on a plus/minus (+/-) icon next to the entry with attributes. Attributes are also nested under their parent, but for clarity attributes cannot themselves have sub-entries. Double clicking on the parent item will select all the children. 'Shift'+LMB clicking the parent arrow will unfold/open all subsequent nested/child item layers simultaneously.

Inline editing



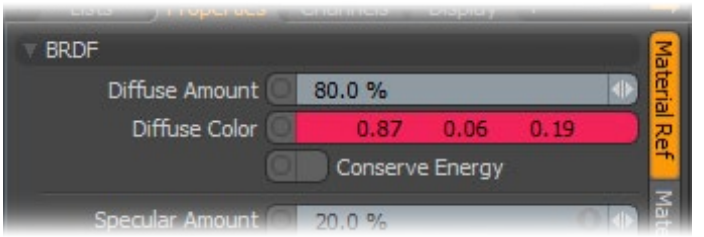
Text fields can be edited inline wherever possible. In particular, text fields in *trees* that users can alter, can be edited directly in place. To edit text users should LMB+click to select the text, pause, and then LMB+click again to convert the text to an inline edit field. Once converted, users can simply type in the new name and press 'Enter' to set the new name.

Boolean Buttons



Boolean buttons (aka *Checkboxes*) act as simple toggles to allow users to set the enable/disable state of an option. They appear as small gray squares when the state is set to *disabled*. Upon *enabling* the button by LMB+clicking on the square area, it becomes highlighted in orange and a black check mark appears in the square. LMB+clicking the button again toggles to the previous state.

Advanced Color Navigation



modo has unique color selection behavior. In any color swatch, three numbers are displayed, by default, each a floating point value between 0 and 1. Each value represent individual RGB color settings. Users can LMB+click and drag on any of the numbers to directly modify the color, which is displayed interactively in the background of the swatch as the mouse moves. Additionally, holding down a key will allow modifying all three values at once--

'Shift'+LMB+drag = Brightness

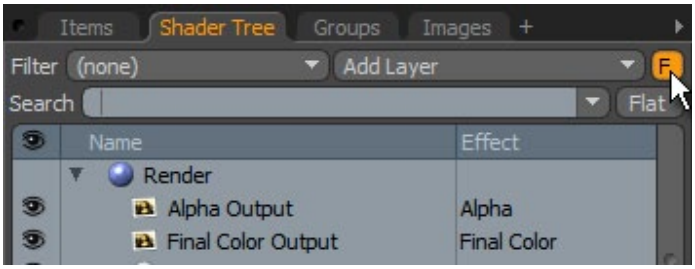
'Ctrl'-LMB+drag = Hue

'Alt'-LMB+drag = Saturation

Or, simply LMB+clicking anywhere on the color swatch itself will open the color picker. Pretty slick, huh?

TIP: The default floating point color system can be changed using the preferences dialog. In the menu bar select "System > Preferences..." and in the preferences viewport select 'Input > Accuracy and Units'. Here users can modify the 'Color Unit System' to 'Floating Point' (0 - 1), 'Percentage' (0% - 100%), 'Integer' (0-255), and 'Hexadecimal' (00 00 00 - ff ff ff).

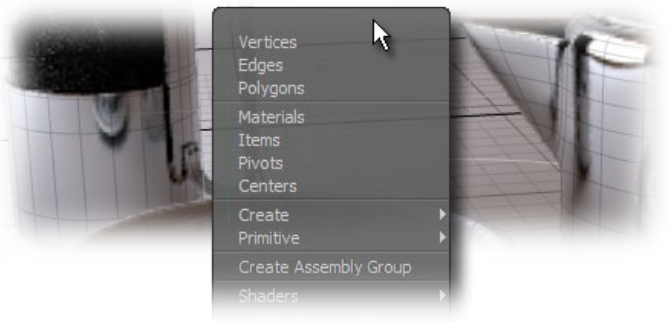
Simple String Search Filter



modo provides a function that allows users to use simple search strings to aide in locating layers in a list viewport, such as the 'Items' list or 'Shader Tree'. It is located toward the top right of the viewport as a small "F" button, where users can LMB+click on the button to open the filter input field. Text is then input onto the value field same as elsewhere by simply LMB+click on the input field to set focus and then typing on the keyboard. As users type strings of text, the viewport will update to display the results of the search. Users can then select a layer and LMB+click the "F" button again to hide the search options (if desired) returning the list view to normal, with the layer remaining selected. Users can select two modes for the search with the pop-up menu, either 'Simple Search' or 'Pattern Matching'.

When searching the in the Clips list (images viewport), the original filename is considered, for other viewports, the actual layer name is considered. When the 'Match Case' toggle is enabled, the search will consider the letter-case of the text. For users who prefer a flat display of the list results, the 'Flat' button can be toggled to allow easy view and selections of multiple layers.

Contextual Popups



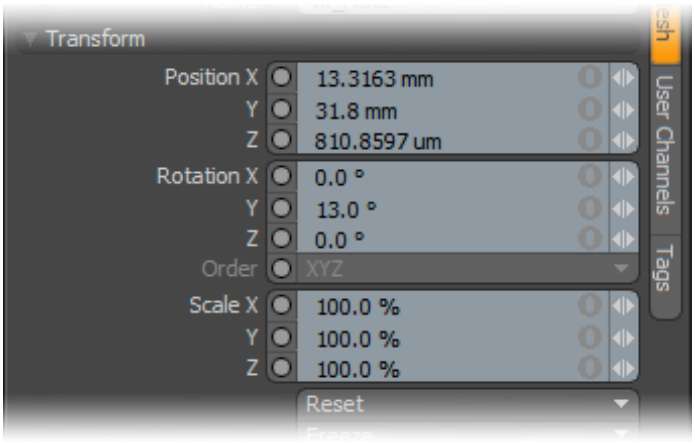
The various Viewport (including the 3D view and UV edit views) display popup option lists when users RMB+click within the viewport as well as in the header. Here users can apply functions to that particular viewport, activate certain tools, and affect selections within the viewport; all dependant on the mode, and over what element in the interface the particular RMB+click occurred. If there is no option available, then right-clicking does nothing; the menus only appear when the context is correct. For example, RMB+clicking on the Model tab of the modo 3D viewport window displays a list of options that can be chosen to change the viewport into something other than a modeling viewport and RMB+clicking in the viewport while in polygons mode offers a number of choices to affect the viewport and geometry selections.

Value Presets

Value Presets are the small white downward facing arrows that appear on the far right hand side of many numeric entry fields. By LMB+clicking on the arrow users can open

a mini popup window for selecting and saving often used values. The values presented within each property are specific to that particular property, meaning you will only see the F-Stop values in that particular field. Simply clicking on any of the values when highlighted will insert that particular value into the associated data field. Users can use the '(new)' command to create a preset, and the '(delete)' commands to remove any values from the popup. Once a value is created, it is saved to modo's config file.

Numeric Entry Fields and Minisliders



Numeric attributes can be edited directly inline by simply LMB+clicking directly on the value, converting it to an inline edit field, a second click sets the insertion point, then with the keyboard, users can enter a new value, then pressing 'Enter' to accept. Numeric properties can also be modified interactively via the 'Mini-sliders', the small white arrows that appear on the far right hand side of most numeric entry fields. By LMB+clicking and dragging on the Minisliders, users can quickly increase (dragging right) or decrease (dragging left) the current value in the field. Values can be *stepped* by simply clicking on the right or left side of the slider; click on the right to increase the value, click on the left to decrease it. The minisliders also have a system of detents that cause tiny pauses to numeric values as users drag over "nice numbers" such as 100% or 180°. Depending on the input the values are:

angles, major: 0, increments of +/- 90

angles, minor: increments of +/- 30 and +/- 45

percent, major: 0, increments of +/- 100%

percent, minor: increments of +/- 25%

distance, major: 0, powers of 10

distance, minor: nice grid increments

float, major: 0, powers of 10

float, minor: multiples of 2, 2.5, 5

This is very useful for quickly and accurately dragging out a value. Holding the 'Shift' key while using a Minisliders will increase the speed a value is incremented by a factor of 10, while the 'Ctrl' key will reduce the speed by a factor of 10.

Users can gang edit all three fields simultaneously as well, by LMB+Clicking the ⓪ to the left of the mini-slider enabling the feature.

The first click changes the field ⊖ 'Copy', identical value are applied to all like controls.

The second click changes to ⊕ 'Relative', value change is added to all like controls.

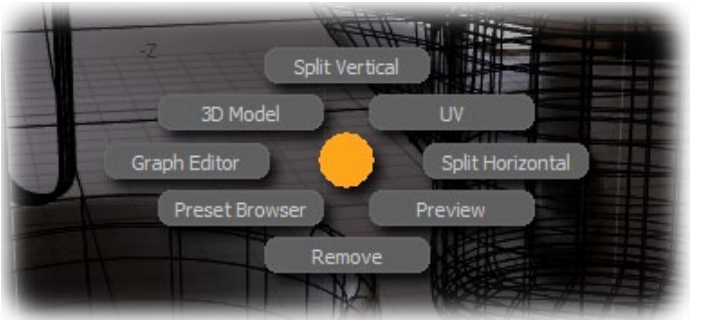
The third click changes to ⊛ 'Proportional' where value changes are applied proportionally to all like controls.

A fourth click returns the feature to ⓪ 'Independent', controls are edited independently.

When entering numeric values, users can use keyboard equivalents as well-- 'Ctrl'+'Alt'+'Enter' for Copy, 'Alt'+'Enter' for Relative and 'Ctrl'+'Enter' for Proportional.

TIP: Numeric entry fields allow users to enter simple equations and the value is computed automatically. This allows for very simple calculations without using a calculator. Example: Perhaps you wish to use the Clone tool to create 7 copies of the original mesh, each with a small amount of rotation so that by the seventh clone their total is 360 degrees. Users can simply enter 360/7 in the Rotation numeric entry field. When pressing 'Enter', modo will handle the calculation and the resulting 51.4 degrees will be in the field. Accepted symbols are / = divide, * = multiply, - = subtract and + = add.

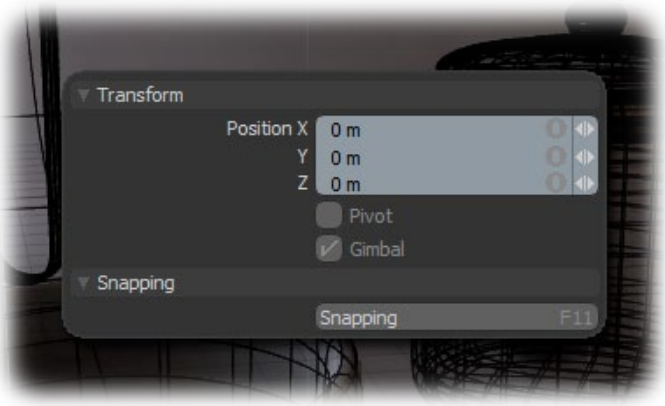
Pie Menus



Pie menus are quick contextual popup menus that appear using keystroke combinations. Menu items are laid out in a wheel so users can access them *gesturally* for speedier workflow. By default, when a key combination is invoked to recall a menu,

the menu opens instantly, Users can adjust this behavior with the preference setting "System > Preferences > Input > Remapping" with the 'Open Pies Immediately' toggle. When disabled, opening Pie Menus will require the keyboard key combination and a LMB+click over any viewport to open. User configured pie menus are created using the 'Forms Editor'. Note that pie menus are limited to only 8 items. The first 8 items that appear in the Forms Editor are used for the pie menus, including disabled items. The keyboard shortcuts page references the various default menus available.

Popover Window



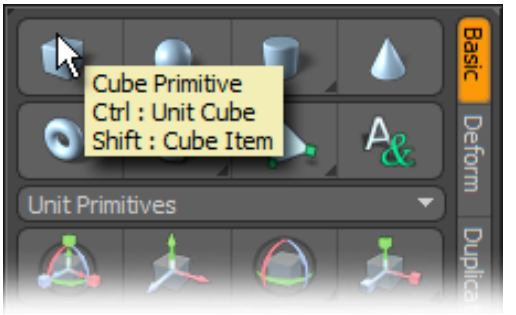
Any type of viewport or content may be opened as a popover. Once activated, interaction can happen normally in the window. Moving the mouse outside the popover bounds dismisses it. To see an example of a popover window, hold down ALT+Spacebar anywhere in the modo interface. The Quick Access popover window displays, showing tools for Transforms, Selection, Center & Axis, Falloff and Tool Properties. Popovers can speed your workflow by putting the tools directly under your fingertips, such as pressing 'K' in a viewport to display the tool properties within the viewport.

List Columns

To facilitate the display of various types of information in a list viewport, data is organized into columns. The default column sizes may not always be a proper fit for the data they contain. Users can resize column widths by hovering the mouse pointer on

the divider line until the cursor turns into two opposing arrows, then LMB+clicking the divider and dragging left or right will resize the column.

Tooltips



Hovering the mouse cursor over a button will display a Tooltip, a small text hint with the name of a function or tool, and keyboard equivalents if applicable. For certain commands, Tooltips also contain useful hints for using the function.

Hauling

Hauling is an off-handle click+drag action which controls one or two values of a tool directly in the viewport using the left/right, and possibly up/down motion of the mouse. Many tools have an advanced handle setting that displays hauling feedback in the form of lines drawn horizontally and vertically on the screen corresponding to the direction of mouse deltas. The lines are labeled, and will persist as long as hauling can be resumed. Clicking directly on a line highlights it and starts hauling for that parameter only.

Drag and Drop

modo supports Drag & Drop behavior for loading object files and bitmap images. Since modo is a multi-document system, each 3D object or project that users drag in will appear as its own document layer. When users drag and drop images, if dropping them onto geometry they will appear as a texture in a material. If you drag them into a 3D viewport, they will appear as a Backdrop Item. If you want to load a bitmap, but don't want to apply it to geometry or as a Backdrop Item, drag it into a non-3D viewport. Items can be dragged from the 3D views directly onto groups (where they become members of the group) or into the Schematic viewport. Channels can be dragged from the Channel List onto groups or into the Schematic.

Viewport Maximize & Restore

Viewport may be maximized or minimized within a group to temporarily expand the amount of space given to a specific viewport. This is very useful when working in a quad view modeler layout as it allows you to quickly maximize a particular viewport, then restore back to the previous quad view state. By default this is mapped to the '0' (zero) key on the numeric keypad. To use, hover the mouse pointer over the target window and press '0'. The target viewport will maximize to full size based on the group where it is embedded. Pressing '0' again while hovering over the same viewport will return the viewport to its previous state. This can also be useful when working in long Item Lists and Shader Trees and searching for particular layers.

Floating a Window

Viewports can be torn off and converted into floating windows, or new floating windows can be created which can be manipulated like the main window. To float any window, simply LMB+click the thumb (the circular indentation at the upper left hand corner of the window) and drag the position of the window. The window is removed from its position in the interface and floats when you do this. To dock the floating window, just reverse the process. Click the thumb and then move the cursor back to the newly desired position in the interface, by hovering over a viewport frame edge. A yellow line will display indicated the new location of the window. Releasing the mouse button will dock viewport. Floating windows are persistent between sessions. If users mistakenly mess up the interface, it can be reset using the menu bar command "Layout > Layouts > 601 Default". This will clean things up in a jiffy.

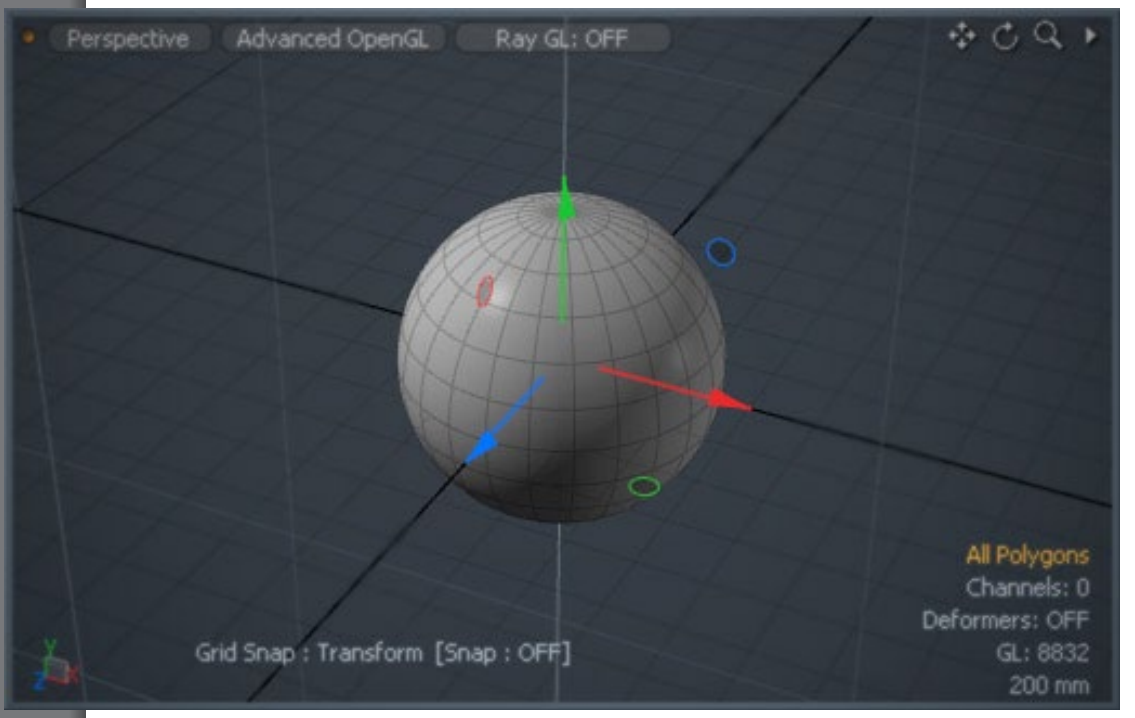
Context Help

User can access context sensitive help by pressing the 'F1' key or by choosing "Help > Activate Help" from the menu bar. The cursor changes into a question mark. Then clicking on a tool or UI element opens a page of related information will display in an HTML browser window. For complete help information, go to the menu bar and choose "Help > Documentation > Inline Help System". This displays the main index page of this documentation. If you still need help, you can visit the luxology forum (www.luxology.com) where users can directly post questions to be answered by expert users and helpful modo fans everywhere.

Standard Tool Controls

modo provides a series of universal functions for users to interact with when modifying and manipulating elements within a scene.

Tool Handles



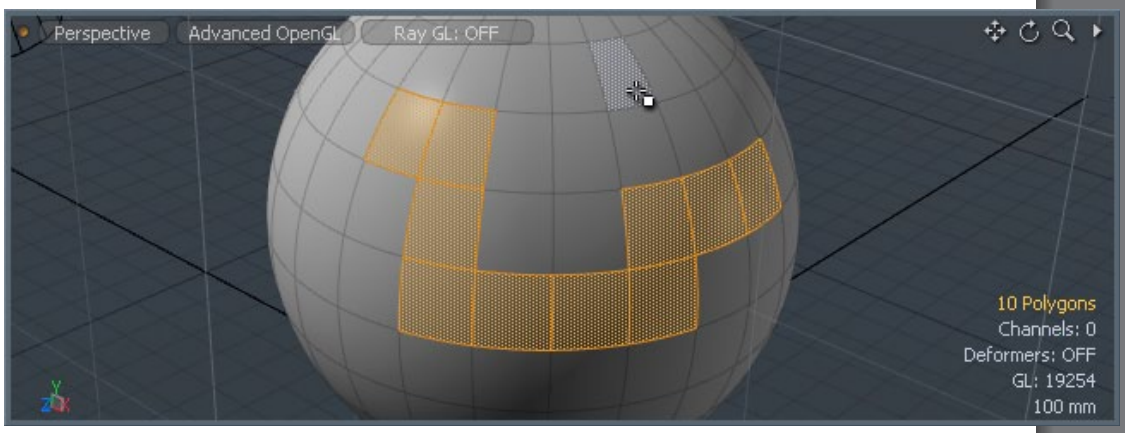
In modo, viewport manipulators or gizmos are referred to as 'Tool Handles'. They are used to interactively control modeling operations in the viewport similar to the way an artist works with sculpting or drawing tools. Handles have consistent color coding that acts as a quick indicator of the value that they will edit and/or the context in which they are going to perform. In the default color scheme, tools handles use red, green and blue when editing in 3D model space. When working in UV space the handles are drawn as mustard yellow and teal (blue-green). The colors are used to represent the major axes when applicable (such as with the Move or Scale tools).

In 3D space: Red = X, Green = Y, Blue = Z

In UV space: Teal = U, Yellow = V

Users can access different tool handle styles by going to the menu bar and choosing "View > Tool Handles > Draw Style". Here you will find options for 'Invisible', 'Basic', 'Advanced', and 'Default'. Use Invisible when you want to hide the Tool Handle entirely. Use Basic when you want the minimum viewport display for the handles. Use Advanced when you want more precision using the tool handles. Use Default to return to the factory preset. Use the minus "-" or equals "=" keys to make the tool handles larger or smaller while working.

Selections



In order to modify an element, users need to tell modo specifically what element they wish to manipulate by way of a selection. Selections are made via the left mouse button. modo uses a technique called pre-highliting to cue users what element will be selected when the user clicks the mouse. By default this pre-highlighting is a light blue color. Once the user clicks on an element, it changes to an orange color to signify its selected state. modo provides a number of function to aide in making selecting easier, found on the 'Selection Modifiers' page of the documentation. One important note to mention when in a component mode (vertex, edge, polygon), is that when no elements are selected, modo considers *everything* selected. This is not the case for other modes, where elements need to be explicitly selected in order to modify them.

Multiple Selections can be created through a variety of means. If users LMB+click to select something, then press and hold down the "Shift" key and click on additional components to add them to the selection. To remove a element (a vertex,edge,polygon or item) from a selection, press and hold down the "Ctrl" key and click on the selected element to remove it from the selection. Users can assign selection sets to recall a selection at a later time. Use the menu bar command "Select > Assign Selection Set" to set the set. Use the command "Select > Use Selection Set" to load the saved selection. Even better, users can use the Statistics tab on the Info & Statistics toolbar. LMB+click the plus button next to the named selection to activate the selection, or the minus button to subtract it from the viewport selection.

Key Toggles (aka *Sticky Keys*)

Hotkeys can be used as "soft toggles" to speed up workflow. For instance, if users wish to move vertices in a mesh, invoking the menu bar command "Edit > Transform > Move" choice, users may find it cumbersome to constantly drop the active tool, select

new vertices, and reselect the tool. Instead, can use the 'Key Toggles' functionality. For any tool that has a keyboard shortcut, such as the 'W' key used to activate the 'Move' tool, users can press and hold down the key, temporarily activating the tool, then use the mouse (or tablet Pen) in the 3D viewport to apply the adjustment (all while continuing to hold down the appropriate key). Once the adjustments are made, users can release the key, deactivating the tool without further mouse click or keypress necessary, making it very easy to make further selections.

Input Mapping

In modo, the primary input device is the mouse. To optimize the use of this wonderful device, and to create a consistent workflow, there are a number of rules for using the mouse depending on the action that one is working with. Let's begin by looking at the mouse behaviors when using a primitive tool.

LMB+click and drag = adjust handles (continue)

Shift+LMB+click and drag = adjust handles (reset)

RMB+click and drag = haul alternate parameters

MMB+haul = adjust handles (clone previous)

Transform tools adjust existing mesh elements, and typically have parameters that are hauled (clicking and dragging away from the tool handles with the tool active), or optionally set with handles. This includes all the basic transforms, most point pushers, and many vertex map tools.

LMB+click and drag = haul parameters (reset)

Shift+LMB+click and drag = haul parameters (continue)

RMB+haul = falloff setup parameters (if any)

MMB+click drag = adjust parameters (clone previous)

Non-transform edit tools adjust existing mesh elements, but they typically don't want to reset as their default mouse-down behavior. These may still work with falloffs. An important exception is when a tool of this type is used with the screen falloff, in which case they act like transforms with reset as the default.

LMB+haul = parameters & handles (continue)

Shift+LMB+click and drag = haul parameters (reset)

RMB+haul = falloff setup parameters (or alt parameters)

MMB+click drag = adjust parameters (clone previous)

The airbrush falloff tool is a special case when used with any other tool, transform or otherwise.

LMB+click and drag = paint weights (continue)

Shift+LMB+click and drag = paint inverse weights

Ctrl+LMB+click and drag = erase weights

RMB+haul = adjust airbrush size

MMB+haul = parameters of action tool

Apply

When the Tool pipe is inactive, an Apply button can be pressed to perform the action of the tool pipe with its current settings. This does not activate the pipe, and the Apply button is disabled when the pipe is active. The Apply Button, when available, will appear on the Tool Properties form. You can also right-click on the Tool Pipe entry and choose Apply Tool.

Reset Tool Attributes

When using a tool interactively or when using the numeric input fields for a tool, users can reset the values that have changed to defaults by choosing "Edit > Reset Tool Attributes", or press CTRL+D with the tool active for quick keyboard access.

Keyboard access

Press 'K' to open tool properties popover form.

Keyboard Shortcuts

Keyboard shortcuts can really streamline the user experience, by mapping specific functions to simple keyboard key presses. Any command in modo can have a keyboard shortcut assigned to it. These keyboard shortcuts can also be reassigned, customizing modo to a users specific needs. By also allowing the 'Ctrl', 'Shift' and 'Alt' key qualifiers (including combinations thereof), you have a wide array of keys to choose from.

Key mappings are identical on both Windows and Macintosh systems, with the exception of the basic Macintosh key commands 'Open' (Cmd+O), 'Save' (Cmd+S), 'Quit' (Cmd+Q), 'Cut' (Cmd+X), 'Copy' (Cmd+C), 'Paste' (Cmd+V) and 'Undo'/'Redo' (Cmd+Z). When using keyboard with reduced keys (such as on a laptop), refer to the menu bar equivalent or remap often used commands.

To assign a keyboard shortcut to a command, fire the command from the interface, then expand the Command History viewport, located in the lower right hand corner of the interface. Locate the command in the list (usually visible at the bottom of the history), then RMB+click and choose 'Map Command to Key'. You can also assign keyboard equivalents using the 'Input Editor' found on the 'System' menu on the menu bar. This is useful if you want to create Keyboard shortcuts for scripts or tools that don't actually appear in the default interface.

Default Keyboard Mappings include:

General/System

F1	=	Help (Click on interface element to open related help file)
Ctrl+N	=	New Scene
Ctrl+O	=	Open Saved Scene
Ctrl+I	=	Open Saved Image
Ctrl+W	=	Close Current Scene
Ctrl+Shift+W	=	Close All Open Scenes
Ctrl+S	=	Save Current Scene
Ctrl+Shift+S	=	Save Scene As...
Ctrl+Q	=	Quit modo
Alt+F12	=	Config File Save
Shift+F5	=	Run Script
Ctrl+Z	=	Undo
Ctrl+Shift+Z	=	Redo
Ctrl+X	=	Cut
Ctrl+C	=	Copy
Ctrl+V	=	Paste
Delete	=	Delete
Ctrl+R	=	Fire Previous Command Again
Alt+R	=	Fire Previous Command (Ignore UI interactions and selections)

Viewports

Navigation

Alt+LMB	=	Rotate view
Alt+MMB	=	Rotate viewport Z axis (bank) view
Alt+RMB.	=	'Flick' Rotate
Shift+Alt+LMB	=	Pan View
Shift+Alt+RMB	=	Up/Down Pan only
Ctrl+Alt+LMB	=	Zoom to mouse position
Ctrl+Alt+RMB	=	Box Zoom (drag box to zoom)
. (period)	=	Zoom In
Shift+.	=	Zoom In X2
, (comma)	=	Zoom Out
Shift+,	=	ZOom Out X2
G	=	Center view (Current mouse position to viewport center)
/	=	Turntable (Spinning view of scene)
Shift+/	=	Upright Turntable
Mouse Wheel Spin	=	Zoom View in & out to mouse pointer
A	=	Fit (Zoom to scene extents/all Items)
Shift+A	=	Fit Selected (Zooms to selected elements)
Ctrl+A	=	Align Selected (Aligns view to current selection)
Ctrl+Shift+A	=	Fit and Align Selected (Zooms and aligns view to current selection)

3D Viewport Options

Enter Values on *numeric* Keypad, hover mouse over target Viewport to select.

0	=	Open Viewport Options Popover
0 (zero)	=	Maximize/Minimize Viewport within Group
. (decimal)	=	Perspective
1	=	Top
Shift+1	=	Bottom
2	=	Front

Shift+2	=	Back
3	=	Right
Shift+3	=	Left
4	=	Shaded
5	=	Texture
7	=	Wireframe
8	=	Solid
9	=	Vertex Map
/	=	Wireframe None
Shift+/	=	Wireframe Light
Ctrl+/	=	Wireframe Dark
= (equals)	=	Toggle Viewport Smoothing

Tabbed Viewport

Ctrl+Shift+Home	=	Open First Tab
Ctrl+Shift+End	=	Open Last Tab
Ctrl+Shift+Page Up	=	Open Next Tab
Ctrl+Shift+Page Down	=	Open Last Tab

Item List

F	=	Locate Selected Item (layer)
Shift+LMB on ▶	=	Expand/Close All (also in Shader Tree)
' (apostrophe)	=	Swap Foreground/Background layers
Ctrl+D	=	Duplicate Selected Layer(s) (only in Items mode)
Shift+D	=	Instance Selected Layer(s) (only in Items mode)
N	=	New Mesh Item
L	=	New Locator Item

Channel List

I	=	Select Modifier Input
O	=	Select Modifier Output
C	=	Channel Haul On (only in 'Items' mode)

Floating Windows

F2	=	Modeling Toolbox
F3	=	Sculpt/Paint Toolbox
F4	=	Animate Toolbox
F5	=	Command History Viewport
F6	=	Presets Viewport
F7	=	Graph Editor
F8	=	Render Preview
F11	=	Snapping Popover
` (accent/tilde key)	=	Toggle Window Visibility

Pie Menus

Pie menus are quick contextual popup menus that appear using keystroke combinations. Menu items are laid out in a wheel so you can access them "gesturally" for speedier workflow. To open press the keyboard shortcut, then LMB+click in the 3D viewport to open. If you want the pie menu to appear immediately, go to "System > Preferences > Input Remapping" and *enable* 'Open Pies Immediately'.

Ctrl+`	=	GL Viewport Modify
Ctrl+1	=	GL Viewport Toggle
Ctrl+2	=	GL Viewport Style
Ctrl+Spacebar	=	GL Viewport Type
Ctrl+3	=	Open Popover
Ctrl+Tab	=	Modeling Tools
Alt+Q	=	Selection Mode/Transform Tools
Alt+`	=	Open Palette
Shift+`	=	Properties & Options
B	=	Brushes Pie Menu (only when a Brush tool is active)

Modeling

General Modeling

Q	=	Drops Active Tool
K	=	Tool Properties Popover
Esc	=	Escape current operation/Drop Active Selection
Spacebar	=	Cycle-through Component Modes
Shift+Spacebar	=	Toggle Component Mode to Item Mode (or Last Other Mode)
= (Equals Key)	=	Tool Handles Larger
- (Minus Key)	=	Tool Handle Smaller
Ctrl+D	=	Reset Tool Attributes
Shift+0 through 9	=	Select Layer 1-10
Alt+Spacebar	=	Quick Access Popover
M	=	Assign Material Tag
Tab	=	Toggle Subdivision Surfaces
Shift+Tab	=	Toggle Catmull-Clark Subdivision Surfaces
Num Pad +/-	=	Increase/Decrease Subdivision Level

Action Center/Axis

Alt+W	=	Origin
Alt+E	=	Pivot
Alt+A	=	Auto
Alt+S	=	Selection
Alt+D	=	Selection Center, Auto Axis
Alt+F	=	Screen
Alt+Z	=	Element
Alt+X	=	Local
Alt+G	=	Pivot Center, Parent Axis

Selection

Alt+Shift+A	=	Select All
Up Arrow	=	More (Select Next Plausible Element in Loop)
Shift+Up Arrow	=	Select Expand
Down Arrow	=	Less (Unselect Last Element)
Shift+Down Arrow	=	Shrink
Left Arrow	=	Select Previous Loop
Shift+Left Arrow	=	Select Previous Loop Add
Right Arrow	=	Select Next Loop
Shift+Right Arrow	=	Select Next Loop Add
L	=	Select Loop
Alt+L	=	Select Ring
]	=	Connected
[	=	Invert Selection
Shift+]	=	Close Loop
Shift+G	=	Select Between
Alt+1	=	Convert Selection to Vertices
Alt+2	=	Convert Selection to Edges
Alt+3	=	Convert Selection to Polygons
; (semi-colon)	=	Select Through Toggle

Hide/Lock Geometry

H	=	Hide Selected
Shift+H	=	Hide Unselected
Ctrl+H	=	Hide Invert (Toggle state)
U	=	Unhide
J	=	Lock Selected
Shift+J	=	Lock Unselected
Ctrl+J	=	Lock Invert (Toggle state)
I	=	Unlock

Work Plane

Home	=	Align to Geometry under Mouse Pointer
Shift+Home	=	Align to Selection (average of all selected)
Ctrl+Home	=	Align to Viewport
End	=	Reset Work Plane
Shift+End	=	Open Work Plane Popover
Page Up	=	Walk +
Page Down	=	Walk -
(num pad) *	=	Toggle Visibility
Alt+O	=	Offset Work Plane to Geometry position under Mouse Pointer

Snapping

X	=	Toggle Snap State (hold with tool active for temporary snap)
F11	=	Open Snapping Popover
Ctrl+Shift+T	=	Element Snap

Tools

Transform		
W	=	Move
E	=	Rotate
Alt+Shift+E	=	Planar Rotate
R	=	Scale
Shift+R	=	Uniform Scale
Y	=	Transform
Deform		
Ctrl+E	=	Bend
T	=	Element Move (aka Tweak)
Ctrl+T	=	Soft Drag
Shift+Y	=	Flex Tool

Shift+S = Smooth Tool

Duplicate

Shift+X = Extrude

Shift+L = Radial Sweep

Shift+Q = Grow Quad Row from Edge

Z = Extend Edge

Shift+V = Mirror Tool

Mesh Edit

Backspace = Remove

Ctrl+Backspace = Collapse

B = Bevel

Alt+B = Sketch Extrude

Ctrl+B = Create Polygon of Given Type from Vertex Selection

C = Edge Slice

Shift+C = Polygon Slice

Alt+C = Loop Slice

D = Subdivide Selected Geometry

Shift+D = Open Subdivide Options

F = Flip Polygon Normal

Ctrl+L = Split Face by Selected Vertices

Shift+T = Triple Polygons (convert to Triangles)

V = Spin Edge

Alt+V = Spin Quads

Curves

Shift+O = Make Open Curve

Ctrl+P = Make Closed Curve

Vertex Tools

Shift+W = Weight Tool

Shift+M = Morph Tool

Paint & Sculpt

B	=	Open Brush Tips Pie Menu (only when tool is active)
O	=	Open Large Color Picker Popover (hover over UI color picker)
S	=	Swap Foreground/Background Colors
E/R	=	Image Ink Rotate Left/Right
D/F	=	Image Ink Scale Up/Down
Shift+Z	=	Adjust Offset amount (sticky key, only while Brush tool is active)

Multi-Res Sculpting

Alt+(num pad) +	=	Subdivision Level Up
Alt+(num pad) -	=	Subdivision Level Down
Ctrl+(num pad) +	=	Current Level Up
Ctrl+(num pad) -	=	Current Level Down

Camera

Shift+Alt+LMB Drag	=	Move camera item
Shift+Alt+RMB Drag	=	Move Up/Down direction only
Alt+LMB Drag	=	Rotate around Target Distance
Alt+RMB Drag	=	Rotate around Camera center
Ctrl+MMB Drag	=	Rotate around 'Z' axis
Ctrl+Alt+LMB Drag	=	Move Forward/Backward only
Ctrl+Alt+RMB Drag	=	Adjust Focal Length
Roll Mouse Wheel	=	Move Forward/Backward only

Animation

Selection

Shift+Z	=	Select All Animated Items
Z	=	Select All Animated Channels

Hierarchy

Ctrl+G	=	Group Selected Items (under Group Locator)
P	=	Parent in Place
Ctrl+P	=	Parent
Shift+P	=	Unparent in Place
Ctrl+Shift+P	=	Unparent

Timeline

Users must be in 'Items' selection mode for these to work.

Shift+Up Arrow	=	Go to First Frame
Ctrl+Up Arrow	=	Go to First Keyframe
Ctrl+Left Arrow	=	Go to Previous Keyframe
Shift+Left Arrow	=	Go back One Frame
/	=	Start/Stop Animation Playback
Shift+Right Arrow	=	Go forward One Frame
Ctrl+Right Arrow	=	Go to Next Keyframe
Ctrl+Down Arrow	=	Go to Last Frame

Setting Keyframes

Shift-Y	=	Key Transforms
Shift-E	=	Key Rotations
Shift-W	=	Key Positions
Shift-R	=	Key Scale
S	=	Key Selected Channels
Shift-S	=	Key Selected Items

Graph Editor

A	=	Fit Selected
Ctrl+F	=	Fit Visible
Shift+F	=	Fit Visible Width
Alt+LMB	=	Pan View
Ctrl+Alt+LMB	=	Scale View
Alt+RMB	=	Scale Graph Size
Ctrl+Alt+RMB	=	Box Zoom (drag box to zoom)
Ctrl+Shift+V	=	Paste Insert Keyframes to Selected Channel
Ctrl+Alt+V	=	Paste Replace Keyframes to Selected Channels

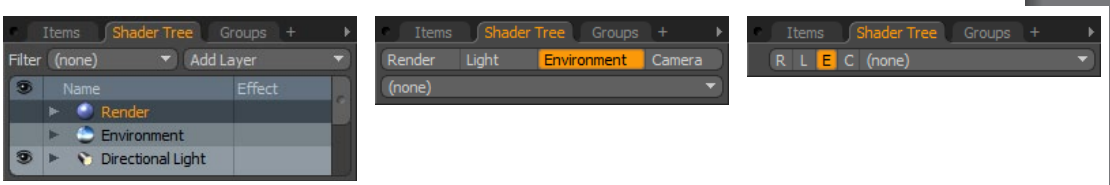
Render

F9	=	Render
Ctrl+F9	=	Render Visible
Shift+F9	=	Render Selected
F10	=	Render Current View

modo Viewports

What is a Viewport?

The modo interface is made up from a collection of frames called Viewports. A viewport is just an empty container whose contents is determined by the options of the viewport 'Widget' (covered fully on the 'Layout Controls' page of the documentation). Selecting the '3D Model View' option changes a viewport to a 3D openGL view with its own functions and settings, but it is still *just a viewport*. From the Widget, users can subsequently select other options, such as 'Shader Tree', and the 3D view will now become a Shader Tree viewport, which is the identical size as the previous 3D view. This is because users control the sizes of viewports by clicking and dragging on their edges (the small dividers between the frames). Viewports are dynamic and react to the size of the frame they exist in. They will even change how users interact with the contents to accommodate specific window sizes, illustrated here.



The same 'Shader Tree' viewport dynamically changes to accommodate different frame sizes allowing access on smaller screens.

There are several different viewports *types* modo uses, 'Forms' are the most common. Forms are generalized containers that can be either a button, a toolbox, or a menu. The content of these viewports can be modified via the 'Form Editor'. The next most common are the GL viewports, which display 3D content (named as such because they use the prominent open source graphic library named OpenGL) These viewports cannot be modified outside of the customization option available within the viewport itself (and the 'Preferences' settings). Other viewport categories include 'Data Lists' which contain lists of information regarding various modo items; these include the 'Item List' and the 'Shader Tree'. Users will interact with the various items in these views, but associated attributes for specific items are viewed in another frame called a 'Property' viewport. Property views are available for various function including Items, Render Settings and Tool Properties.

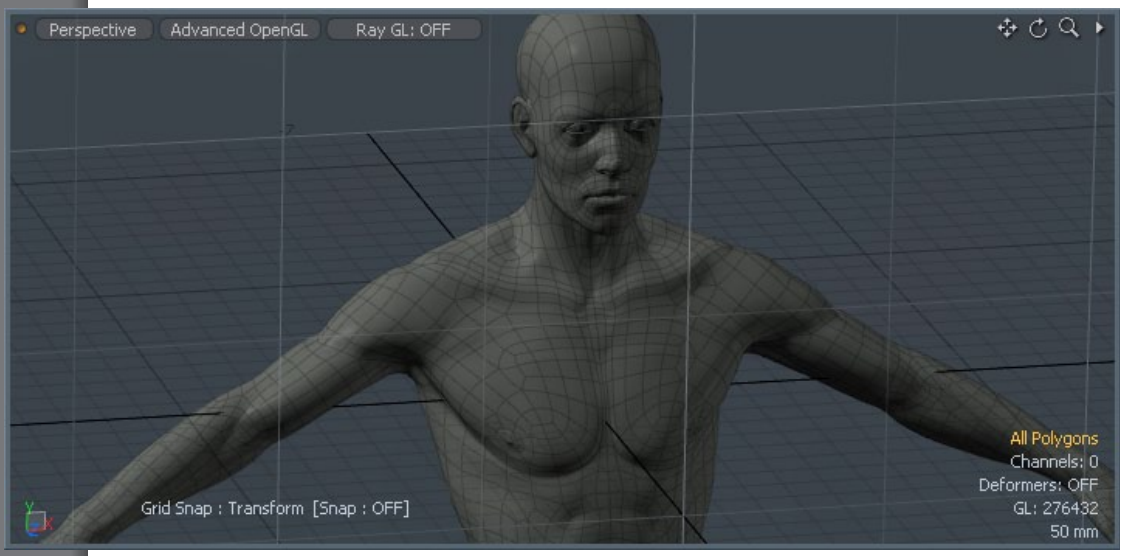
Viewports don't have to exist within the frame of the interface itself either, modo provides both 'Window' and 'Palette' viewports that float above the standard interface. These windows work in the same fashion as normal embedded frames, but their visibility can be toggled on and off via the "`" accent key (same as the ~ tilde key). The

only difference between a Window and Palette is that the windows retain the operating systems controls to individually move and minimize the viewports.

Viewports can be visualized like children's blocks, they can be stacked and arranged anyway the user sees fit, but better than blocks, they can also be scaled, divided and removed providing the most adaptable and customizable interface of any 3D application.

3D (OpenGL) Viewport

The 3D viewport is where users will likely spend their greatest amount of time when working in modo; it is the window into modo's virtual 3D world. It serves many purposes, allowing users to generate and manipulate geometry using any of modo's many tools; users can sculpt, paint, animate, deform, move elements about, and position cameras and lights. With so many jobs to do, it's important that it be flexible as well. There are numerous functions within the 3D viewport that allow users to customize what is seen and how it is seen, depending on the task at hand.



The default 601 interface opens with a large 3D GL viewport set to 'Perspective' view. If your view doesn't match, just select the default layout from the menu bar ("Layout > Layouts > 601 Default") and make sure to select the 'model' tab across the top, located directly below the menu bar.

First notice there is a dark grid moving in perspective off into the distance some, with another coarser light grid perpendicular to the dark one. The dark grid represents the

ground plane; the zero position of the y axis. Each square of that grid represents a fixed distance, since this grid is dynamic depending on the zoom level, a small display in the lower right corner gives users the "real world" equivalent measurements for each square, in the case above it is 50mm. The lighter grid represents the Work Plane; also dynamic depending on users rotational orientation to the world. While the ground plane is purely for reference purposes, the Work Plane's purpose is to show the user where in 3D space will objects get created. Think of it as a construction plane, dynamically adjusting to the most appropriate angle. Also within the 3D viewport, there are "heads-up" displays that give users pertinent information with regards to items visible within the viewport as well as guides to selections and the selected tool.

In the upper left corner of the window is a small dimple, called the Thumb, followed by three capsule shaped buttons. The Thumb will usually be a dark gray until a viewport becomes *active* in which case it will turn orange (To make any viewport 'active', simply click anywhere within the window frame). RMB+clicking on the thumb will present users with a series of options regarding the frame. Refer to the 'Layout Controls' section for further details regarding it. The left most 'capsule' button is meant for choosing the way the viewport sees the scene (detailed in the dedicated '3D Model View' section below) LMB+click and hold for a popup menu that allows the user to choose from the available options. The next button contains all the settings that will let users choose from a variety of scene display options meant to accommodate a wide variety needs. From here users can change the display style, helpful to many modeling and painting tasks, among others. Users can also change how items in the *background* are displayed (items that are *visible*, but *unselected* are considered background elements), and set the visibility of viewport elements. The viewport options are also available by hovering the mouse pointer anywhere over a 3D viewport and hitting the 'O' key. The third 'capsule' button allows users to enable the Ray GL in-viewport rendering preview option.

The upper right corner has several small icons. The first three let users manipulate the view itself, move, rotate and adjust the zoom level of the active viewport. LMB+clicking and dragging over any of the icons (also called 'hauling') will make the appropriate view adjustment providing an intuitive way for users new to 3D to manipulate the position of the viewports virtual 'camera', though most users will want to learn the viewport navigation controls for easier navigation (detailed in the navigation section below)
The tiny arrow icon to the right, called the 'Widget', when RMB+clicked will present a pop-up menu allowing users to select from the many various types of modo viewports. Selecting any of these items will convert that window into the chosen viewport. Don't select anything now, as viewport changes aren't an undoable action. See 'Customizing modo' for more information regarding viewports customization.

On the lower left, there is an XYZ axis gizmo, giving users an iconic visual reference to the orientation of the window as well as the current workplane orientation. For orthographic views, the icon will display the axis plane the viewport windows faces toward, or in perspective view, users can rotate the view and see the widget update in real time.

The lower right corner has several informational displays that dynamically update, displaying the current number of selected items, channels and deformers. The 'GL' display references the number of live polygons active in your viewport (including Sub-D surfaces) and the number at the bottom gives the real world size reference to the squares within the grid of the ground plane. The information displayed is controlled in the 'GL' section of the preferences.

Navigation

modo provides a number of functions to aide in the navigation of GL viewport allowing users a number of ways to *travel* around in 3D space. Most are invoked from a mouse button and keyboard combination. Shortcuts are based on the default interface values. Additional navigation options are available within the Preferences section.

Alt+LMB = Rotate view

Alt+MMB = Rotate viewport Z axis (bank) view

Alt+RMB = 'Flick' Rotate (also called Free Wheeling) view slows down when mouse is released.

Shift+Alt+LMB = Pan View

Shift+Alt+RMB = Up/Down Pan only

Ctrl+Alt+LMB = Zoom to mouse position

Ctrl+Alt+RMB = Box Zoom (drag box to zoom)

'.' & ',' = zoom in & out (easier to think of < and > key)

G = Center view (Current mouse position to viewport center)

/ = Turntable (Spinning view of scene)

Shift+/ = Upright Turntable

Mouse Wheel = Zoom View in & out to mouse pointer

A = Fit (Zoom to scene extents/all Items)

Shift+A = Fit Selected (Zooms to selected elements)

Ctrl+A = Align Selected (Aligns view to current selection)

Ctrl+Shift+A = Fit and Align Selected (Zooms and aligns view to current selection)

Visibility

modo provides a way for users to modify the visibility of object in the viewport directly. While generally users can toggle visibility of items in the Items list, that changes visibility of the entire layer. These commands allow the temporary toggling of visibility of only the selected element, including components, dependant on the selection mode. Once hidden, geometry will be un-editable, meaning it cannot be selected or moved. To apply, simply select the geometry component you wish to hide and invoke the menu bar command "View > Hide Selected" or press the 'H' key. To make the geometry visible again, invoke the "View > Unhide" command or press the 'U' key. When a file is saved and re-opened later, all hidden polygons will be visible.

H = Hide Selected (Hides all if nothing is explicitly selected)

Shift+H = Hide Unselected (Hides all the unselected Geometry)

Ctrl+H = Hide Invert (Toggles visibility, inverting the present state)

U = Unhide (Makes all hidden geometry visible)

Locking

Providing a similar function to *hiding*, Locking allows users to fix a selection of geometry so that is is immovable and unselectable, effectively un-editable, however it will still be fully visible. Unlike hiding, this locking function is limited to the component selection modes. To apply a similar function to item layers, use the locking ability of the 'Items List' or 'Assembly' viewport. To apply, simply select some geometry, and invoke the menu bar command "Edit > Lock Selected" or press the 'J' keyboard shortcut. Users can unlock any locked elements in an item layer with the "Edit . Unlock" command or by pressing 'I'.

J = Lock Selected (Locks all if nothing is explicitly selected)

Shift+J = Lock Unselected (Lock all of the unselected geometry)

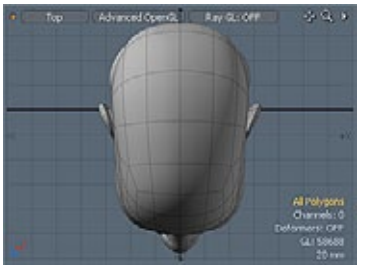
Ctrl+J = Lock Invert (Toggles locked state of all geometry)

I = Unlock (Makes all locked geometry editable)

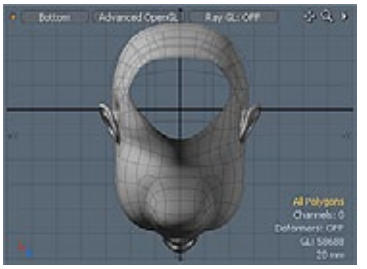
View Types

While viewports have no camera per se, modo does provide users the ability to change how a scene is viewed *through* the viewports. The options can be selected from the 'View Type' menu in the upper left corner of the viewport. LMB+click on the button will open the menu where users can choose alternate options, Notice the name changes for each different view below. The following Orthogonal views provide users viewpoints free from perspective distortion of all the cardinal viewing planes -Front, Back, Top, Bottom, Left and Right.

Top



Bottom



Back

Front

Left

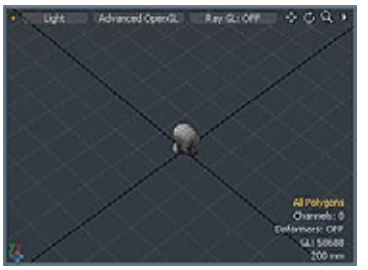
Right

Perspective views are similar to viewing the scene through a real-world camera, where the displays have perspective and a vanishing point. The amount of perspective distortion can be controlled by the "Preferences > Display > OpenGL" setting 'Flatness of Perspective' (This is for the perspective view only, the Camera and Light views are controlled by the items settings). Additionally user can view a scene from the Cameras point-of-view, mimicking the field of view settings of the item. When in the 'Camera' mode, typically translucent black bars appear over the frame to represent the rendered frames aspect ratio, for when it doesn't match that of the viewport frame. Finally, there is also the Light view that displays the scene from a particular lights point-of-view providing users an easy way to see what the light is pointed at and illuminating. It should be noted that when navigating in the 'Camera' or 'Light' views, users are actually moving the positions of the elements within the scene.

Perspective

Camera

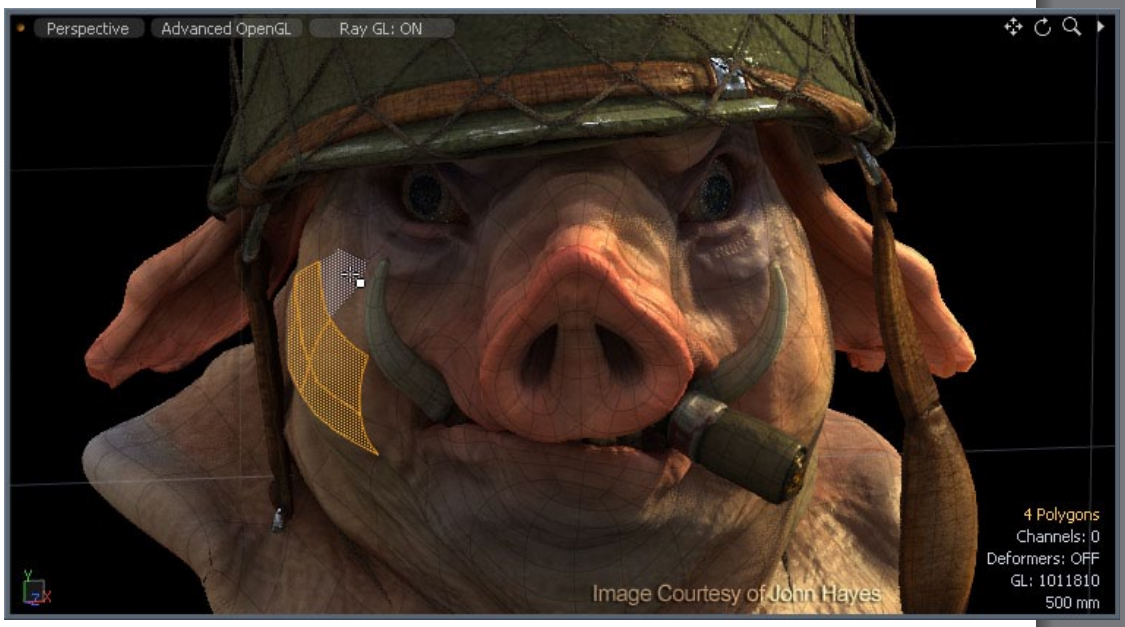
Light



Ray GL

The Ray GL viewport option leverages modo's render preview functionality overlaid with the standard viewport options to provide a hybrid display of the rendered scenes, providing users the ability to model and interact with the *rendered* scene directly. To use, simply LMB+click the Ray GL toggle button in the viewport, enabling the function.

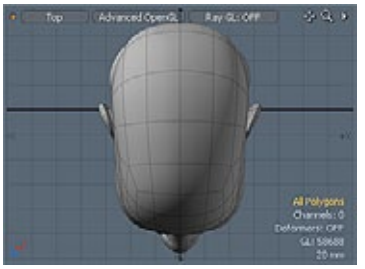
For simpler scenes, shading is near real-time, just like in the render preview, but for more complex scenes, there is a lag between edits where the system temporarily reverts back to the standard GL view. Users can control how Ray GL updates based on options available within the Preferences ("System > Preferences > Display > Ray GL"). Additionally, users can choose which elements are taken into account when rendering. Useful for say when you want to disable Fur rendering to speed up the preview when modeling a werewolf.



3D Viewport Styles

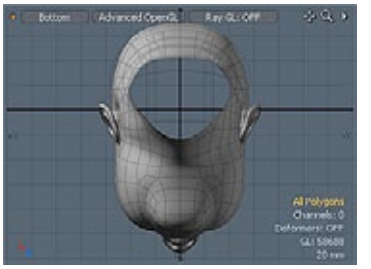
modo provides a number of ways to visualize a scene using viewport styles, these are basically, in-viewport *shaders* that display the geometry in specific ways making it easier to visualize certain aspects of a model or scene. Users may choose from the available options by using the 'Viewport Options' button within the viewport (the second button from the left on the top) by LMB+clicking to open the contextual menu and choosing from one of the available options.

Wireframe



In the Wireframe drawing style only vertices and the edges that connect them to make polygons are visible. The actual polygon faces are not drawn and as such you can see through the model to vertices and poly edges on the backside as well.

Solid



Solid shading mode renders the geometry without shading. This mode is most useful when combined with a wireframe overlay, resulting in a flat rendered "sketch" mode.

Vertex Map

Vertex Map shade option uses the currently selected vertex map to shade the model.
In the case of Vertex Color maps, the color of each vertex will be blended with the colors of the neighbor vertices. For weight maps positive values shade red while negative values shade blue. Any vertex with a value of 0 or no value at all will shade green. If Color and Weight type vertex maps are selected, the Color vertex map will take precedent.

Shaded

The Shaded style shows meshes with their basic material definitions such as Diffuse, Specular and Transparency. No image maps will appear in the Shaded style.

Texture

'Texture' style shows the mesh surface with only the most basic material attributes including any image map textures. In 'Texture' style RGBA images can be seen with their alpha blending them into the material beneath. However, limitations of standard Open GL force the image map itself to draw without shading (aka full bright). This is a reasonable view style for 3D painting or placing image decals onto a model. The Shaded Texture style can shade the image texture but can not use alpha channel. The Advanced OpenGL view will correctly alpha blend multiple image texture layers and shade them appropriately.

Shaded Texture

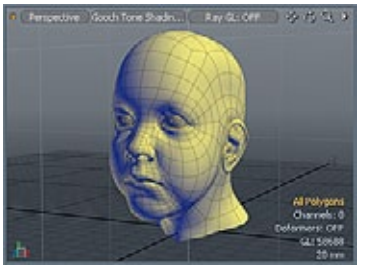
In the Shaded Texture view, image maps are visible as blended with the light shading effects. Due to OpenGL limitations this mode cannot show images with alpha channels so any RGBA image will appear as an opaque RGB image. To see alpha channels from the image users will need to use the 'Texture' or 'Advanced OpenGL' view styles.

Advanced OpenGL

"Advanced OpenGL" will allow users to see many more effects than the normal Texture view. With Advanced GL mode you can see the following types of image maps: Color, Bump, Specular, Luminous and Transparency. It will blend multiple layers of each (though Bump will only use the first layer). However, the total number you will actually see at a time is based on the resources of your graphics card. These resource limitations are per Shader Tree surface so if your card has 8 texture units, each Material can have up to 8 image maps.

Aside from simple alpha blending, each layer's opacity is respected as well as the base material settings - so you can, for instance, set the material to be 50% transparent, then create a clip and apply it as a transparency map and paint in areas of complete transparency or complete opacity.

Gooch Tone Shading



The "Gooch" shading mode is a vertex shading algorithm that displays the geometry using a technique known as tone shading. Rather than relying on the traditional light to dark shadow style shading, which can obscure parts of the model while working, tone mapping represents the curvature of the geometry, shading warm-to-cool where the warm areas are those that would traditionally be lit and the cool areas are those that would normally be in shadow. This shading mode allows you to check the surface of your model for proper contouring while still seeing the entire mesh evenly lit.

Cel Shading

The Cel Shading style creates a stepped shading on the model to appear as if it were a cartoon with a fixed number of shading zones.

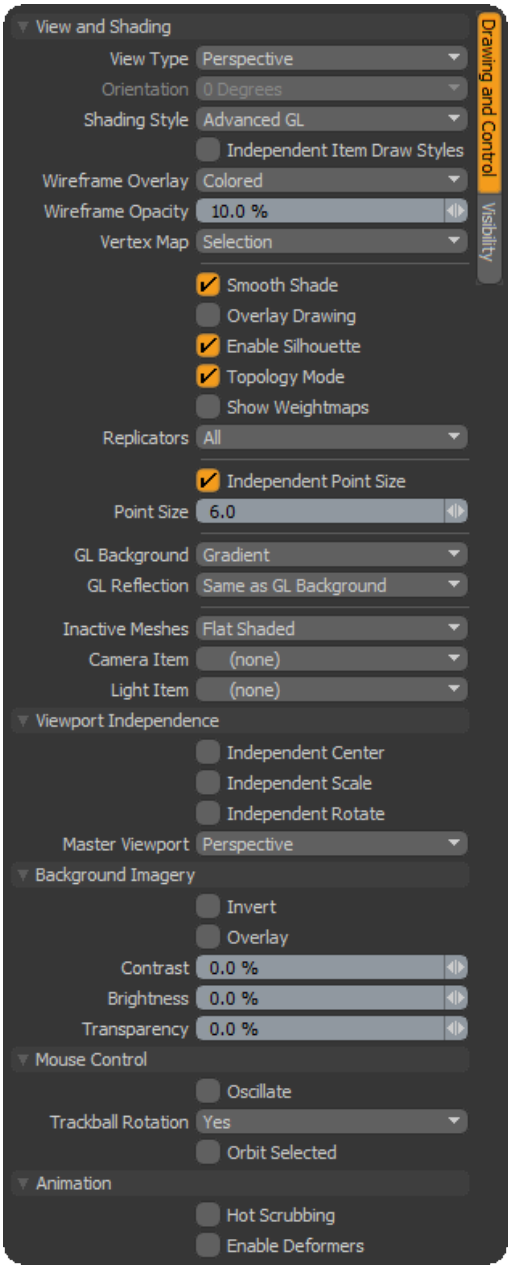
Reflection

The Reflection Shader displays your geometry as if it were fully reflective. This shader uses a cubic reflection map, and mixes in some of the original material properties with a small amount of shading to properly show object contours. This mode is very useful when performing visual review of a models surface as it assists in finding surface irregularities.

Viewport Options

Allowing users to further customize the display of elements within a viewport, modo provides a multitude of option available to control various aspects. The default interfaces apply varieties of these options for their custom workspaces, however, over the years they have been fined tuned to the degree that many users will rarely need to adjust them. Users can easily access the display menu options by hovering the mouse pointer over the target viewport and pressing the 'O' key to open the popover menu. Acting as a standard dialog, edits can be made normally. As soon as the mouse pointer moves away *from the menu*, it will disappear. Many options are also available from the 'Viewport Styles' menu.

Drawing and Control



View and Shading--

View Type: Users can choose from the various view types covered above -Top, Bottom, Front, Back, Left, Right, Perspective, Camera and Light.

Orientation: For the Orthogonal views only, users can control upright rotation angle in 90° increments.

Shading Style: Users can choose from the available view styles covered above -Wireframe, Solid, Shaded, Texture, Shaded Texture, Advanced OpenGL, Gooch, Cel and Reflection Shading.

Independent Item Draw Styles:

Wireframe Overlay: Determines the style of geometry wireframe display. Users can adjust the colors of the wireframe using the preference setting "System >Preferences" and then "Display > Colors". Once a color scheme is created in preferences and saved as a named preset, it must be applied to the active viewport with the menu bar command "View > Viewport Color Scheme".

None- No Wireframe display. (Shortcut = '/')

Uniform- By default this is a white wireframe outline. (Shortcut = Shift+/)

Colored- By default this is a black wireframe outline. (Shortcut = Ctrl+/)

Wireframe Opacity: Controls the opacity of the wireframe overlay display.

Vertex Map: Defines which type of vertex map is displayed when in the 'Vertex Map' viewport style-

Selection- Displays the most recent Vertex Map selection.

Weight- Favors Weight Maps for display.

Vertex Color- Favors Vertex Color Maps for display.

Smooth Shade: This option toggle overrides the material 'Smoothing' setting, only changing how smoothing is displayed in the viewport, not the rendered output. When disabled, individual polygons will shade flat, appearing faceted, when enabled, polygons will shade according to the material 'Smoothing' setting.

Overlay Drawing: For custom draw items, such as locators, cameras and lights (that have no inherent geometry) this option allows an X-Ray style of viewing, meaning users can see these elements *through* polygons when enabled.

Enable Silhouette: When the 'Enable Silhouette' option is *enabled* (the default state), any mesh item designated in its display properties as 'Show as Silhouette' will display as a flat solid shape with no interior details. This option should be thought of as a global enabler only, where the actual display of the silhouette itself is determined by the options of the items 'Display' properties.

Topology Mode: The 'Topology Mode' option is meant to aid in the process of *retopologizing* a mesh, sometimes also called simply 'topo'. This is the process of converting a high resolution mesh to a low resolution mesh by creating new polygons against the high resolution mesh set as a background element. Generally this process creates intersecting geometry that is difficult to see. *Enabling* the 'Topology Mode' option will force the drawing of vertices and edges to always draw *in front* of whatever background elements may be visible in the scene. Vertices will be enlarged for easier visibility, and edges will draw in green when in front of the background geometry and will draw purple where it intersects the background geometry and would otherwise have been hidden.

Show Weightmaps: When the 'Show Weightmaps' option is *enabled*, any selected weight map will be visible as an overlay to the shaded elements in the 3D viewport when displayed as 'Advanced OpenGL' for the viewport display style. This included the automatic weights associated to bones and deformers.

TIP: The 'Enable Silhouette', 'Topology Mode' and 'Show WeightMaps' options are all incompatible with each other and therefore cannot all be enabled at the same time. The 'Show Weightmaps' option will supercede the 'Show Silhouette' option and the 'Topology Mode' options will supercede both.

Replicators: The 'Replicators' options allow users to control the shaded display of any replicators in a scene. In previous versions of modo, Replicators were displayed only as bounding boxes. Users can now adjust this setting to view shaded Replicators never (None), or always (All) or only when a 'Replicator' item is active (Selected). Otherwise the Replicator clones will be displayed as bounding boxes.

Independent Point Size: The 'Independent Point Size' option is a per-viewport override for the display size of selected vertices. When users enable the 'Independent Point Size' option, they can then specify the actual display size with the 'Point Size' option.

Point Size: This option determines the display size of selected vertices when the 'Independent Point Size' option is *enabled*.

GL Background: Users can choose from several viewport specific environment background options-

None- Defaults to the flat background color as defined by the preferences.

Gradient- Displays gray shading from light above to dark below.

Environment- Displays the settings as determined by the 'Environment' item of the shader tree simulating the rendered background.

Image- Users can define a custom image for use as a viewport background. When this option is selected, a file requester will open, users can then select or navigate to the appropriate image.

GL Reflection: Users can choose from several viewport specific GL reflection options for controlling the display of reflective surfaces-

None- Displays no reflection, ignoring surface material settings.

Gradient- Displays a gray gradient shading on reflective surfaces.

Environment- Displays the settings as determined by the 'Environment' item of the shader tree simulating the rendered output.

Image- Users can define a custom image for use as viewport reflections. When this option is selected, a file requester will open, users can then select or navigate to the appropriate image.

Same as Environment- Reflection options are determined by the 'GL Background' settings.

Inactive Meshes: This option determines the Shading Style of Inactive or Background mesh items, defined as items that are visible in the 'Item List', but not selected (selected items are considered foreground and are denoted by highlighting in the item list). The options are-

Wireframe- Displays Inactive meshes as wireframe, only showing vertices and edges, but no solid polygons.

Flat Shaded- Displays inactive meshes as Flat Shaded, displaying only basic material attributes such as color, diffuse and specular.

Same as Active- Displays inactive meshes the same as active foreground meshes as determined by the viewport style.

Camera Item: In the case of multiple cameras in a scene, wherein the viewport is set to 'Camera' view type, this option determines which camera item controls the viewport. Selecting alternate cameras will change the viewport to that cameras point-of-view.

Light Item: In the case of multiple lights in a scene, wherein the viewport is set to 'Light' view type, this option determines which light item controls the viewport. Selecting an alternate light will change the viewport to that lights point-of-view.

Viewport Independence--

Independent Center/Scale/Rotate: To ease in viewport navigation, it is sometimes desirable to link/unlink grouped 3D viewports, so when changing the position, size and angle of the viewport view, associated viewports will move in sync with the active viewport. These three toggles allow users to control the syncing of the view properties 'Center', 'Scale' and 'Rotate'.

Master Viewport: Determines the linking of viewports, on a per viewport basis, options include 'Orthographic', which will link all associated orthographic viewports and 'Perspective', which links associated perspective views (does not include camera and light views).

Background Imagery--

For scenes using 'Front' type texture projection method, either from a 'Camera' or 'Light", the background imagery options control the display of the projected image in the viewport. To use, apply an image to the 'Environment' item in the shader tree, and in its associated Texture Locator, assign the 'Front' projection type, specify a camera or light if necessary.

I**nvert:** Inverts the RGB values of the projected image, producing a negative effect.

Overlay: The 'Overlay' option, when enabled draws the projected image on-top of the viewport, obscuring its contents. When set as such, changing the 'Transparency' setting below will allow users to see the viewports contents.

Contrast: This option adjusts the 'Contrast' of the projected viewport image, modifying the visual difference between light and dark values in the image. Positive values

increase the contrast, while negative values decrease it.

Brightness: This option adjusts the 'Brightness' of the projected viewport image. Positive values will brighten or lighten the image, while negative values will darken the image.

Transparency: The 'Transparency' option changes the opacity of the viewport projected image. When set to 'Overlay, this affect the visibility of items in the scene through the image, when overlay is disabled, this value reveals the settings of the 'Background' options.

Mouse Controls--

Oscillate: When in a 'Perspective' view type, users can enable the 'Oscillate' option, then using the 'Alt'+RMB button combination, users can drag the cursor moving the view, and let go of the mouse button and modo will continue to bounce back and forth oscillating over the previous mouse movement. A neat little trick to view a model in motion.

Trackball Rotation: Users are provided several option for the orbiting behavior when navigating the viewport. Trackball style navigation is a common way to rotate the view of a scene, as if the viewport itself was a giant trackball, allowing multiple axes rotation based on the mouse's viewport position. Disabling this option limits viewport rotations to only two axes. The options are--

No- This option disabled trackball style rotation in favor of standard upright orbiting.

Yes- This option enabled trackball rotation for the specific viewport.

Default- This option uses the preference setting, available in "System > Preferences" then "Display > OpenGL > Viewport Rotation"

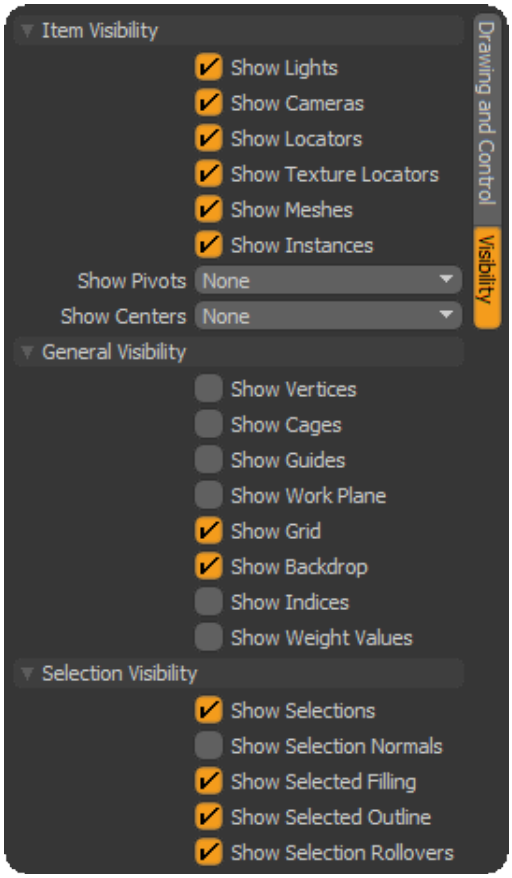
Orbit Selected: When this option is enabled, modo will automatically orbit around the selected element, using the selections bounding box center as the center of rotation.

Animation--

Hot Scrubbing: With 'Hot Scrubbing' enabled, animated items in the scene will update continuously when dragging the timeline, producing a smooth motion. When disabled, items will only update when mouse pauses, or isn't moving providing slightly faster GL performance.

Enable Deformers: The 'Enable Deformers' toggle allows viewports to display the results of animated morph and vertex map deformers in the viewport. This option is typically disabled for the modeling viewports, and enabled for animation viewports.

Visibility



Item Visibility--

Show Lights: This toggle determines the visibility of the light item representations in the current viewport.

Show Cameras: This toggle determines the visibility of the camera item representations in the current viewport.

Show Locators: This toggle determines the visibility of the locator item representations in the current viewport.

Show Texture Locators: This toggle determines the visibility of the texture locator item representations in the current viewport.

Show Meshes: This toggle determines the visibility of mesh items in the current viewport.

Show Instances: This toggle determines the visibility of instance items in the current viewport.

Show Pivots/Centers: Visibility for Pivot and Center representations can be determined by these criteria-

None- Only shows the Pivot/Center when in Pivots/Centers selection mode.

Selected- Shows the Pivot/Center when any item layer is selected in the Item List.

All- Always shows the Pivot/Center for all items.

General Visibility--

Show Vertices: Toggles the visibility of unselected Vertices (selected vertices are always visible unless 'Show Selection' is disabled).

Show Cages: Toggles the visibility of the Subdivision Surface Cage representation, this can be considered the original polygons that were smoothed.

Show Guides: Toggles the visibility of guide lines that draw between the Subdivision

limit surface and the cage geometry.

Show Work Plane: Toggles the visibility of the Work Plane in the current viewport. (keyboard shortcut = Num Pad ' * ' .

Show Grid: Toggles the visibility of the Ground plane grid at the 0 position of the up axis.

Show Backdrop: Toggles the visibility of any Backdrop items in the scene.

Show Indices: Toggles the visibility of the component index values display. Each number represents the particular component index. Visibility can be especially helpful when working with geometry constraints, where constraints require specific index values for input.

Show Weight Values: Toggles the visibility of weight map value display. A numeric value representing the current vertex weight amount will display next to each selected vertex in the 3D viewport. Weight Map usage is covered on the 'Working with Vertex Maps' page of the documentation.

Selection Visibility--

Show Selections: Toggles the visibility of selection highlighting.

Show Selection Normals: When enabled, this option displays normals for any selected polygon(s), displayed as a small dashed line pointing outward from the center of the polygon. Normals represent the facing direction for polygons.

Show Selected Filling: Toggles the visibility of the selection highlight overlay.

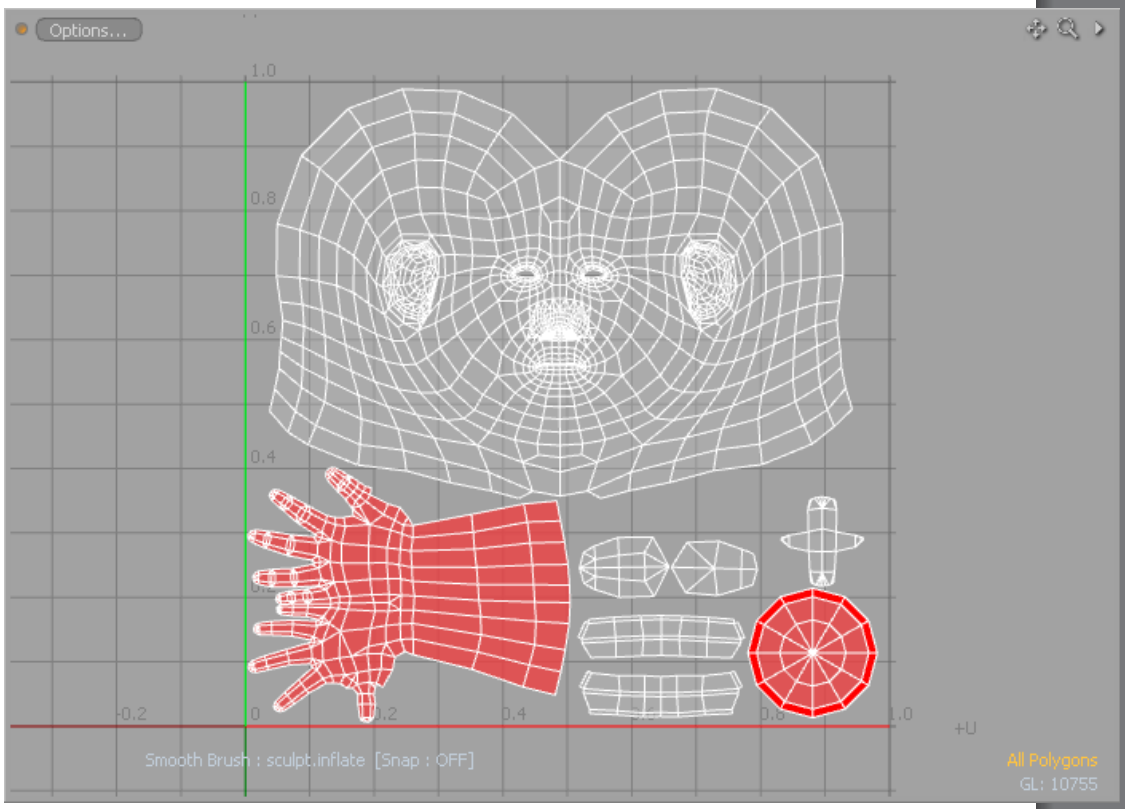
Show Selected Outline: Toggles the visibility of the selection highlighting outline.

Show Selection Rollovers: Toggles the visibility of selection rollover pre-highlighting.

UV Viewport

The UV viewport is much like the 3D GL Viewport, only instead of representing 3D space, it shows users the two dimensional view of UV space. UV maps are the flat 2D representations of a models 3D surface for the application of texture maps. Users not familiar with what a UV map is can explore them further on the Working with UV maps page of the modo tools documentation. The main toolbox and viewport for UV mapping is available in the 601 default interface under the 'UV' interface tab (found directly under the menu bar). If this Tab is not visible, users may select the default interface using the "Layout > Layouts > 601 Default" menu command. Users may also prefer

to try the configurations available under the "Layout > Layouts > UV Edit/UV Edit 102" menu, providing similar utility, with different configurations.



Compared to the 3D viewport, the UV viewport itself is rather sparse. The 3D grid is replaced by the 2D UV grid and flat background color. Relevant on-screen information overlays are still located toward the bottom of the viewport. The 'Thumb' and 'Widget' viewport controls are still available toward the top, covered on the 'Layout Controls' page of the documentation. The pan and zoom navigation widgets are located in the upper right corner, users can LMB+click on these icons to move and scale the contents of the viewport, making it easier to see certain areas for detailed editing. Lastly, user can LMB+click the 'Options' capsule shaped button toward the upper left corner to open a context menu with a variety of viewport customizing controls.

Navigation

Navigation of the UV view is very similar to the 3D views. Users can use the zoom and pan widgets to navigate, but most will find it far easier to use the mouse and keyboard combinations modo provides.

Ctrl+Alt+LMB = Zoom to mouse position

Alt+LMB = Pan view

Mouse Wheel = Zoom to mouse position

Shift+Alt+MMB = Pan Horizontally (U direction)

Shift+Alt+RMB = Pan Vertically (V direction)

Ctrl+Alt+RMB = Box Zoom (drag to define zoom area)

A = Zoom to geometry extents

Shift+A = Zoom to selection

'.' & ',' = zoom in & out (easier to think of < and > key)

G = Center current mouse position

Visibility

modo provides a way for users to modify the visibility of object in the viewport directly. While generally users can toggle visibility of items in the Items list, that changes visibility of the entire layer. These commands allow the temporary toggling of visibility of only the selected element, including components, dependant on the selection mode. Once hidden, geometry will be un-editable, meaning it cannot be selected or moved. To apply, simply select the geometry component you wish to hide and invoke the menu bar command "View > Hide Selected" or press the 'H' key. To make the geometry visible again, invoke the "View > Unhide" command or press the 'U' key. When a file is saved and re-opened later, all hidden polygons will be visible.

H = Hide Selected (Hides all if nothing is explicitly selected)

Shift+H = Hide Unselected (Hides all the unselected Geometry)

Ctrl+H = Hide Invert (Toggles visibility, inverting the present state)

U = Unhide (Makes all hidden geometry visible)

Locking

Providing a similar function to hiding, Locking allows users to fix a selection of geometry so that is is immovable and unselectable, effectively un-editable, however it will still be fully visible. Unlike hiding, this locking function is limited to the component selection modes. To apply, simply select some geometry, and invoke the menu bar

command "Edit > Lock Selected" or press the 'J' keyboard shortcut. Users can unlock any locked elements in an item layer with the "Edit . Unlock" command or by pressing 'I'.

J = Lock Selected (Locks all if nothing is explicitly selected)

Shift+J = Lock Unselected (Lock all of the unselected geometry)

Ctrl+J = Lock Invert (Toggles locked state of all geometry)

I = Unlock (Makes all locked geometry editable)

Viewport Options

Users can select from a variety of display customizing options from the viewport popover menu, by holding the mouse pointer over the viewport and pressing the 'O' key to open. Acting as a standard dialog, edits can be made normally. As soon as the mouse pointer moves *away from the menu*, it will disappear. Many of these display options are also available using the 'Option' context menu button in the upper left of the viewport.

Spans--

U/V High/Low Spans: Users can control the number of UV sections or spans displayed in the viewport using these controls. Each span represents a complete UV grid section. The default value produce the grid found most commonly in 3D programs representing, the -1 to 1 UV space coordinates. USers can place vertex data anywhere within UV space, this option simply determines the display extent of the UV grid.

Extents--

Minimum/Maximum U/V: Users can control the active area for UV mapping using the extents options.

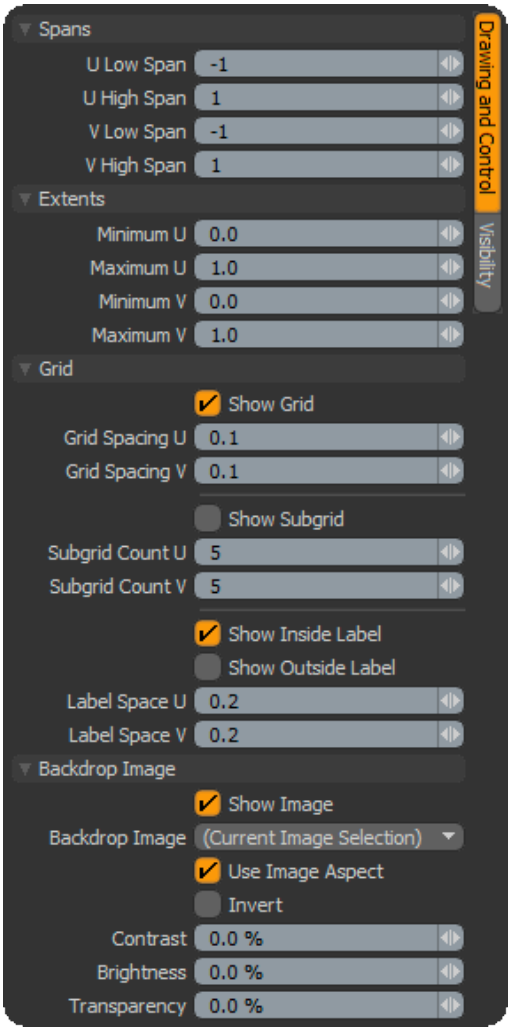
Grid--

Show Grid: This toggle determines the visibility of the UV grid in the viewport.

Grid Spacing U/V: Users can set the grid line spacing using these values, defined as decimals of the UV unit. The default value of 0.1 draws grid lines every 10th of a unit.

Show Subgrid: For detailed maps, the standard grid may not have enough subdivisions to be visible when zoomed in, users can enable this setting to display a lighter secondary grid.

Subgrid Count U/V: These options determine the number of divisions between each 'Grid' subdivision.



Show Inside Label: This option toggles the visibility of grid value labels inside the grid area.

Show Outside Label: This option toggles the visibility of grid value labels outside the grid area.

Label Space U/V: These values determine the division level for the grid labels. Typically this value will match or be a division of the 'Grid Spacing' value.

Backdrop Image--

Show Image: This option allows users to toggle the visibility of a background reference image map. The particular image displayed is determined by the 'Backdrop Image' settings.

Backdrop Image: The 'Backdrop Image' option offers users the ability to select from a list of loaded images (those that are available in the Clips/Image list) or when the option is set as "Current Image Selection", whatever the current image selected. Images can be selected in the 'Images' tab or within the 'Shader Tree'.

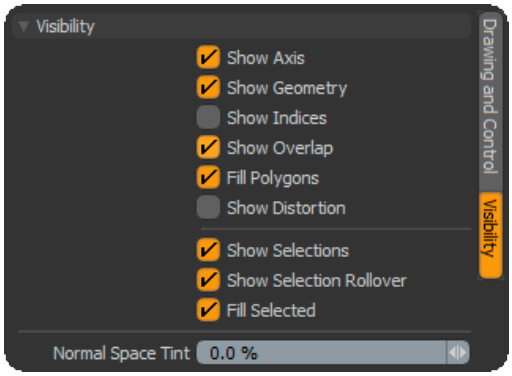
Use Image Aspect: It is a common misconception that UV mapped images need to be square (with equal height and width). Regardless of aspect or resolution, images will render and display correctly. When users enable the 'Use Image Aspect' option, the UV window will be scaled horizontally to math the aspect ratio of the selected image making it easier for users to edit UV maps for non square aspect ratios.

Invert: This command will simply invert the RGB color values of the Backdrop Image producing a negative effect.

Contrast: Contrast is the apparent visual different between light and dark values in an image. Values greater than 0% will increase the contrast of the backdrop image, while values lower than 0% will decrease it.

Brightness: Brightness is the degree or amount of light or luminosity in an image. Values greater than 0% will increase the brightness of the backdrop image, while values lower than 0% will decrease it.

Transparency: Transparency determines the opacity of the backdrop image. Values greater than 0% will increase the transparency, ramping toward 100% where the image would become completely invisible.



Visibility--

Show Axis: This option toggles visibility of the colored axis display at the UV origin.

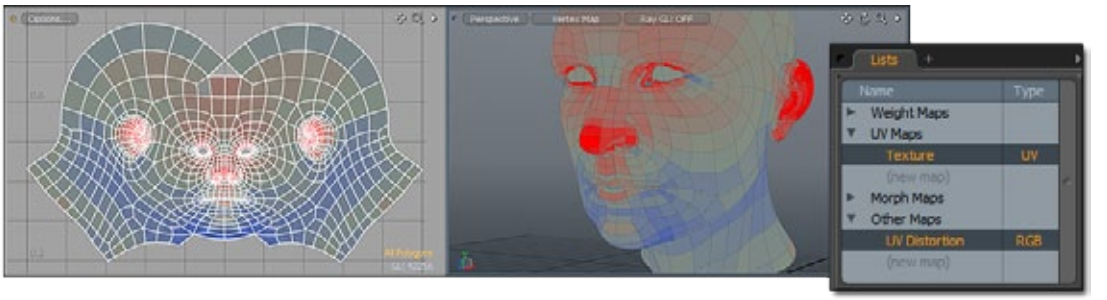
Show Geometry: This option toggles visibility of all geometry in the viewport.

Show Indices: This option toggles visibility of vertex index value for selected items.

Show Overlap: When the area of a polygon sits over the area of another, enabling this option will display a red highlighting to signal users of the overlapping state. Users will want to avoid overlapping when baking as it will cause rendering errors, but often times for texturing game models it is desirable.

Fill Polygons: This option toggles visibility of the faint fill coloring given to polygon faces.

Show Distortion: Two main goals when UV mapping are to eliminate or reduce seams, and to also minimize distortion from stretching. Unfortunately, these two are not mutually exclusive. Distortion is a fact of life with UV mapping. Enabling this option will display a color overlay on the geometry fading toward red when polygons are distorted and smaller relative to other polygons in the map, and fading toward blue when they are distorted and larger relative to other polygons. Polygons that are the closest relative size to others in the map with the least distortion will display a middle gray/green color. Users can also display the distortion colors in the 3D viewport by selecting the 'Vertex Map' viewport style and then selecting the 'UV Distortion' named vertex map generated under the 'Other Maps' heading of the vertex map 'Lists' viewport.



When 'Show Distortions' is enabled, colors overlay onto the geometry signifying distorted areas of the UV map.

Show Selections: This option toggles visibility of highlighted polygon selections, by default an orange color.

Show Selection Rollovers: This option toggles visibility of selection rollover pre-highlighting.

Fill Selected: This option toggles visibility of the orange fill color for selected polygons.

Normal Space Tint: The 'Normal Space Tint' option controls the amount of tint coloration applied to the Normal Space area of the UV viewport, identifying the central or main UV area of 0 to 1. This is where textures originate, outside of this area they simply tile.

List & Info Viewports

The List and Info viewports provide important functionality to modo, providing an intuitive way for users to select, arrange and group the various elements that make up a scene. Outside of simply selecting items, viewports like the 'Item List' provide important functions for setting up hierarchies for animation, and the 'Shader Tree' is the necessary component for assigning surfacing attributes.

List & Info Viewports

Item List

Channels

Clip Browser

Info & Statistics

Shader Tree

Image List

Properties

Command History

Vertex Map List

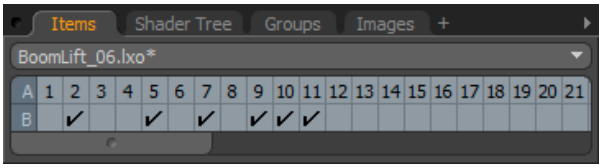
Item List

The Item List has a number of functions. First, and most obviously, it lists all of the locator type items within your project (For more info about *Locators* and lists see 'Item List versus Shader Tree' section below). In the Item list users can create, edit and organize modo's various Item layers- Meshes, Locators, Cameras, the various Light types, Texture Locators and Backdrop Items. The functionality of the Item List includes, simple selection of the Items in the list, creation and deletion of Items and entire scene files, re-ordering items in the list, creating parent/child hierarchies, setting visibility, edit item properties and creating item groups for general organization.

Item List modes

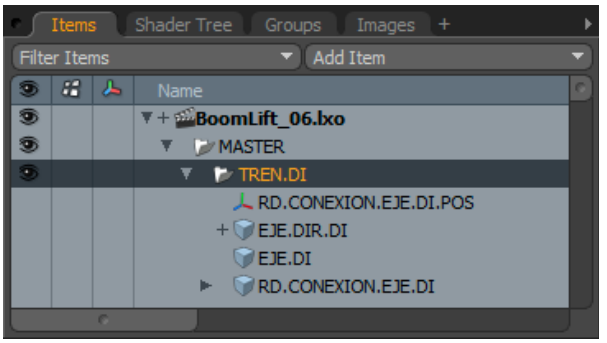
The Item List has three UI modes depending on the size of the viewport. When the Item List is fully expanded it shows all Items and acts as a complete project manager. When the viewport gets collapsed it converts to a Mesh List allowing a very compact user interface for quickly selecting the various meshes in your project. The Mesh List mode has two sizes, medium and small depending on the amount of screen space available to the viewport. This allows users to compress the list into a very small space on the monitor, making room for other UI elements. The viewport can be resized by LMB+clicking on any frame edge and dragging, the viewport will automatically change UI style depending on the amount of space available.

The small view first shows the current project popup, this popup allows users to select any of the currently loaded scenes. Directly to the right of that is the left/right navigational widget that is used to move forward and backward through large lists of mesh layers. The small cells to the right of that represent the mesh layers within a scene file. The top row is for Active mesh items (also called foreground objects) and the bottom cell indicates an inactive (but visible) mesh item (also called background objects). Ghosted dots in the top row indicate the mesh item contains geometry. White dots indicate that a mesh item exists but contains no geometry (an empty layer).



The medium mode functions exactly as the compact mode, with the only significant difference being the layout of options. In this viewport configuration, the file pop-up button is found at the top of the viewport. The layer "cells" are found just underneath

that. In this case there are 3 rows, the topmost row shows the layer number, the next row is the foreground row and beneath that is the background row. Finally, at the bottom of the row is the layer navigation slider bar that allows you to drag to the right or left to navigate to layers that do not fit in the mesh list in this configuration.



The full view can also be called "tree view" as the layers are displayed in a tree structure from top to bottom. It is called a tree because in this view you can also see the hierarchy of the layers if any has been set. In most cases the mesh layers are in a "flat" hierarchy indicating that there is no parent child relationship between any of the layers. However, if a layer has been parented to another, that child will appear underneath and indented slightly to the right of the parent. The parent layer will then have an expand/contract widget to the left so that users can easily collapse entire hierarchies quickly. To set a layer to be parented to another layer, simply drag the intended child layer on top of the intended parent and drop. The result is that the child layer will show up underneath and indented to the right slightly. To move it back to the same level, simply drag and drop the child layer back into the list and drop it between two layers, rather than on top of one.

Items in the list can only be modified when selected, by LMB+clicking on the name of any layer in the list. Multiple items may be edited at once as well. Non sequential selection can be created by holding down the 'Ctrl' key and LMB+clicking multiple layers. Sequential items can be selected by first LMB+clicking on the top most item and then 'Shift'+LMB clicking the bottom most item, all items in between will be selected automatically.

When selecting layers in this view, notice that there are some combinations of font style and colors that indicate various states of the layer. Any selected layer(s) row will appear darker than the others with the layer name text highlighted in orange. Selected layers are considered *foreground* layers. Item layers that are visible but not selected are considered *background* layers, and show their names as plain black text. Mesh items that have been created, but contain no data have their text drawn in light gray.

At the top of the list is the 'Scene' item ( ), representing the scene file itself in **bold**

text styling. Below it are the various items layers that make up the scene -mesh items, lights, cameras and so on. Visibility of various layers is controlled via the left most column. LMB-clicking on the eye icon ( ) will toggle visibility of items in the 3D viewport, as well as when rendered (unless specified differently by the Display viewport or the items properties panel)

The next column displays icons signifying the current editing *mode* of a selected layer or layers, this is important because the current mode affect the layer differently. The icon denotes an 'Item' mode transformations. The icon denotes mesh items can be edited directly as components, such as Vertices, Edges or Polygons. Component versus Item transforms are an important concept to understand in modo. If you think of mesh items as containers, item mode transforms are applied to the entire container, while component transforms modify the contents of the container only. Nearly any item can have an item level transform applied - position, scale and rotation. Only mesh items can be edited in component mode, using any of the tools modo provides. Locators, Cameras, Lights and Backdrop items can only be edited in Item mode.

Set Reference System

The third column, when LMB+clicked will toggle an item between world space and *local* space, displaying the icon next to an item layer when in local mode. When in local mode, an items *center point* temporarily becomes the origin. This can make certain modeling tasks easier. For instance, if one modeled a cars tire by radial sweep around the origin, and then in item mode, the tire is positioned at the axle of a car, if you needed to further modify the tire, you could place the item in local mode and re-sweep the tire around the origin. When done, return to world space without ever having to move a single item. This action may also be accomplished by invoking the menu bar command "Item > Set Reference System"

Auto Visibility

There are two very different workflows between modeling and layout/rendering processes. When modeling you often want to focus on a single mesh item (often referred to as a layer) whereas the layout process demands a complete view of the project. modo offers a solution for both via 'Auto Visibility'. By default this option is *off*. The Auto Visibility mode will automatically turn visibility *on* for the currently selected item and *hide* all deselected items. In this way you can easily set the focus to a specific mesh item. Once the item is visibly isolated you can use the 'Shift' or 'Ctrl' key to add to the selection. If you want to have one layer active but another layer visible and inactive you can 'Ctrl'+LMB click on that Items visibility column to override the Auto Visibility setting at that time. To access the Auto Visibility option you can either RMB click on the viewport tab or header, or you can use the Viewport Widget and navigate to the Viewport Settings.

Creating Parent/Child Hierarchies with the Item List

Parent/Child relationships amongst items have several benefits. If you have experience with animation, you are likely familiar with the power of these types of hierarchies. Since setting them up in modo is so simple this is a first step at rigging your project for animation. Still, even if you have no intention of animating your scene, there are other advantages to creating these hierarchies. The two main areas of advantage are gang-manipulations and organization.

Creating hierarchies in modo is a simple drag and drop operation. In the Item list, drag and drop one item onto another and that item becomes the child item. That's really all there is to it. Well, there are some nuances. The default behavior for parenting in the item list is to parent in place when you drag and drop items. The control key is used to parent without maintaining current transforms of child items. This is the same in the 3D views where the shortcut 'P' will parent in place, 'Ctrl+P' performs normal parenting. Also 'Shift+P' unparents in place, 'Ctrl+Shift+P' unparents.

Managing Items in the Item List

As your projects become more complex you may find a long flat list of Items becoming unwieldy. Parenting items to a 'Group Locator' will help organize a project. Different from a 'Group', Group Locators are a special kind of Locator item as they do not draw anything in the 3D views removing additional scene clutter, and can be easily collapsed which keeps the item list clean and manageable. Group locators have an icon that looks like a folder in the Item List to distinguish them from regular Locator items. In all other aspects they act the same as a normal locator, they can be moved and rotated in the 3D views for example.

To create a Group Locator users can choose "Add Item > Group Locator" from the button atop the Item List viewport. Once created, users can select some items and begin to drag and drop them into a hierarchy to organize the list. If you have several items that you want to quickly parent to a Group Locator, you can select them all by holding down the 'Ctrl' key and LMB clicking on individual items in the list; once selected, press 'Ctrl-G' to automatically create a new Group Locator, with all selected items parented in one step.

Manipulating Items in a Hierarchy

It is important to note that there are some ramifications to editing item layers in 'Items' mode when a hierarchy is established. (The following section is only relevant when the selection mode is set to '*Items*'.) For instance, when translating (*moving*) only the parent item, the children will receive the same amount of translation, remaining in the same relative position to their Parent item as it moves. However, when selecting the parent

AND child items, the effect of the transform will be doubled on the child, it will be inheriting the parents transform AND since it is also selected, it will be moving as well. If there are Item level transforms applied to the Parent item before the relationship is defined, the Child will move to match the Parents transforms, this is called 'Parent in Place'; the transform amount of the child will be adjusted so the Item remains in the same relative position to its parent (it's moving but it won't look like its moving). If this behavior is undesirable, the solution to maintain the child's current position during parenting or un-parenting the item. Simply hold the 'Ctrl' key down when dragging and dropping the child to or from the parent item. This will retain the child items Transform values such that the item does not move from its original local positioning, but will move in world space as it has inherited its parents transformation values.

TIP: Often when you have created hierarchies it may be arduous to expand the various parent items to dig down to the level where you can see the item you are looking for. If you change to 'Items' selection mode and click on the item in 3D view (where it is most likely visible in plain site) you can press the 'F' key to find the item in the Items List. The list will auto-expand any necessary group locators or parents to reveal your item. You can also quickly expand/contract an entire hierarchy branch with a single action. Holding 'Shift' when clicking to expand or contract a hierarchy will expand/contract the current item as well as all items down the branch. Fancy.

The Item Context Menu

There are several options available via the Item List context menu. Simply RMB click on any item to open the menu.

Rename: Allows users to rename the selected item.

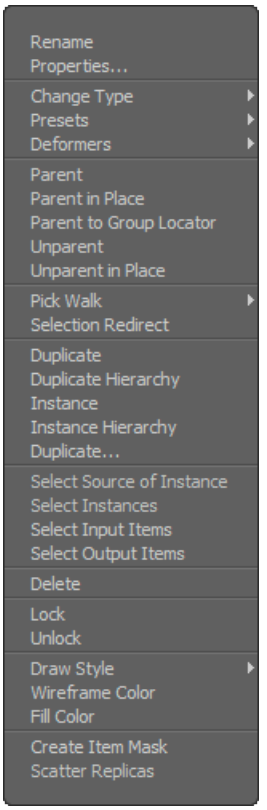
Properties: Mesh items have options for renaming, setting the subdivision and spline patch levels, adjusting the curve refinement angle, visibility and toggling between smooth SDS UVs or linear UVs. All other items only have the option to set their name as the other options are irrelevant to non-mesh items.

Change Type: With this option you can quickly convert an item into another item type. For instance, you can convert an Instance mesh into a real mesh so that you can edit the item at the component level. Another example is to convert a Directional Light into an Area Light. It should be noted that certain parameters would be lost if you were to switch back to the original item type. Common parameters should be preserved.

Presets: This option will allow you to load, apply and delete preset parameters to your objects.

Deformers: This option will allow you to add any of modo's various deformers. The 'Add MDD' option will allow you to apply an MDD file, useful for animating objects in other

packages and applying the the animation as a deformation for rendering in modo. The 'Add Morph' will allow you to animate a mesh items morphs over time. Finally, the 'Add Vertex Map' will allow you to add animated deformations to a mesh item based on a weight map.



Parent: This allows you to easily create a Parent-Child hierarchy of several items. To invoke, select multiple items within the item list and right click on any of them, select 'Parent' from the popup menu and the top most item in the item list will become the 'Parent', while all the subsequent items will become 'Children'.

Parent to Group Locator: This creates a Group Locator item and parents all currently selected items to it. The same can also be achieved with 'Control-G' shortcut.

Unparent: An easy way to remove Parent/Child relationships of items within any hierarchy.

Pick Walk: Pick Walking is used to make it easier to navigate complex hierarchies, as it will allow the user to easily create selection relationships up and down a hierarchy chain.

Clear Selection Redirect: Selection Redirects are useful to aid in selecting specific items in a 3D viewport, as an item is selected in a 3D viewport such as a locator, that selection can be redirected to another object that would be difficult to individually select otherwise allowing it to be manipulated. The 'Clear Selection Redirect' command is an easy way to remove a selection redirect.

Duplicate: This command will create a duplicate of the current item.

Duplicate Hierarchy: This command will create a duplicate of the current parent item and all child items while retaining the overall relationships of the hierarchy.

Instance: The Instance command is only relevant for Mesh items. The result is a duplicate of the current mesh that is simply a reference to the original mesh. Instances can be transformed at the item level and can have their own materials and textures applied to them via Item level masks. However, they can not be edited at the mesh level as they derive all mesh information from the original mesh item. The advantage is that all mesh changes made to the original mesh item will propagate to its instances.

Instance Hierarchy: The 'Instance Hierarchy' command will create instanced versions

of a parent items and all child items, while retaining the overall relationships of the hierarchy.

Delete: This will delete the currently selected item(s) layer.

Lock/Unlock: Users can use these commands to quickly and easily lock and unlock items from accidental manipulation, eliminating the need to do so manually through the Assembly viewport. Once an item is locked, a small padlock icon ( ) appears next to the layer name. Once an item is locked, no transformation modifications may be made. To make a change, users much first unlock the item.

Draw Style: Will allow you to set a specific draw style for an item regardless of the 3D viewports display setting. Useful for when you have items within a scene that you wish to 'fix' the way they draw in the viewport, essentially mixing view styles within one window; i.e. wireframe mixed with shaded mixed with weight shade.

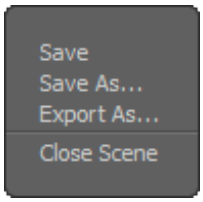
Wireframe Color: Allows you to change the default display color of an unselected item when displayed as a wireframe. Useful for differentiating multiple items in a complex scene.

Fill Color: Allows you to change the display color of an object in any GL viewport, independent of its surface attributes.

Create Item Mask: This option will create a mask group in the Shader Tree with the current Item as the filter. This allows you to quickly add a material/texture mask at the mesh item level so that you can isolate material changes to that specific item. Especially useful for applying surface attributes to an instanced object. If your original item is, say for instance red, you could change the color of an instance of the item to blue by using an item mask, while adding no additional real geometry to the scene.

Scatter Replicas: The 'Scatter Replicas' command only applies to Replicators and act as an easy way to apply a Surface Generator to an item allowing finer control over duplicated variations through that shader tree.

Project Item Context Menu



RMB clicking on the file name itself in the Item list will open the file context menu. This is a very convenient place to perform routine tasks.

Save: Saves the scene loaded in memory to disc overwriting the previous saved file

Save As: Save the scene currently loaded in memory to a user specified name and location.

Export As: Exports the entire scene to a variety of different formats.

Close Scene: Removes the scene and all related files, like images, MDD and IES light files from memory. Defaults to a blank new scene if no other scenes are loaded.

Item List versus Shader Tree--

You may have already noticed that some modo items appear in both the Item List and the Shader Tree. The Item List is generally used to select *Locator* type items. Locator type items are items that can be selected and edited directly in the 3D viewport. There are items that are not Locator type. For instance, materials, shaders and texture layers are also items but are not directly editable in the 3D viewport. These items are accessed via the Shader Tree rather than the Item list. There are some crossover items that appear in both the Item List and the Shader Tree. This is because they may share common linked items such as a texture layer and its associated locator item (texture positions are set by Texture Locators which can be manipulated in 3D), or items that can be modified by textures (Lights have a material and can be textured within the Shader Tree).

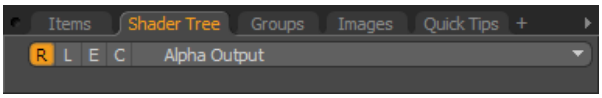
Shader Tree

The Shader Tree is the main interface for adding materials and textures to surfaces in your scene. modo uses a layered approach to texturing, allowing a simple to use, yet very powerful method for assigning textures exactly where you want them, as well as creating complex looks for surfaces with ease. The visual impact on your scene is from top to bottom; an image layer with 100% opacity that is placed above a procedural checker pattern will obscure the checkers in the rendered frame. Changing the image layers transparency would gradually reveal the checkered layer below.

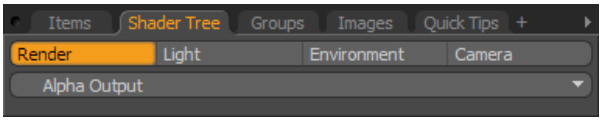
The viewport itself, in the default configuration, appears in the upper right viewport group. Click its tab to make the viewport visible. Whenever a layer is selected within the viewport, its associated settings will appear in the 'Properties' viewport found directly below. If the properties are not visible, one may need to also click its tab to reveal the viewport.

Shader Tree UI--

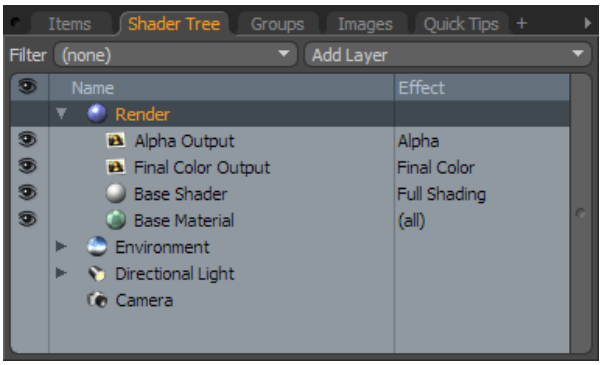
The viewport has 3 UI modes, 'Compact', 'Small' and 'Full' . Depending on your amount of screen real estate, you can adjust the viewport by dragging any edge; it will dynamically scale giving you the appropriate set of options based on the viewports size.



In 'Compact' mode, the first four buttons represent 'R' for the render item, 'L' for lights, 'E' for environment and 'C' for camera. The bar on the right, when clicked on will pop open a contextual menu allowing the user to choose individual layers specific to each 'R', 'L', 'E; and 'C' setting. Additional layers can be added using the menu bar commands 'Texture > Add Shader Tree Layer > (choose type)'



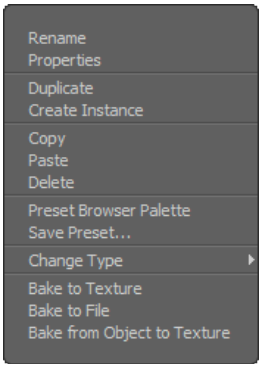
In 'Small' mode, the 'Render', 'Light, 'Environment' and 'Camera' buttons become full size and the layer selector appears below.



In 'Full' mode, each item is represented as a layer with a corresponding icon. Clicking on any item will select it, allowing the user to edit the various properties, multiple layers can be selected using the Shift and Ctrl key modifiers. Layers can also be dragged and dropped within the viewport, changing the layer order which in turn affect how they layers feed into the render outputs, They can also be dragged into material groups for better organizing scenes with many surfaces.

Some additional controls appear above the viewport as well. The 'Filter' option will automatically show/hide various type of layers. The 'Add Layer' button will allow the user to add additional layers to the shader tree. The 'eye' icon column allows the user to enable/disable the effect of a specific layer without removing it; simply click the eye to toggle. The 'Effect' column allows the user to specify in what way the layer contributes to the rendered image, RMB-click on the effect name to open a context

menu with many additional options. Any layer name, once selected, can be RMB-clicked providing a context menu with several options-



Rename: Allows the user to assign a new name to the specific layer.

Properties: Opens the 'Properties' viewport with the selected layers corresponding settings.

Duplicate: This command will create a new layer with identical settings. Changes in the original item are not reflected in the duplicate layer, nor are modification to the duplicate reflect in its original layer.

Create Instance: Creates a virtual duplicate of a layer that references the original, changes to one layer propagate to all instances.

Copy/Paste/Delete: Entire layers, as well as the settings of individual layers themselves can be copied, pasted and/or deleted using these commands. When pasting values from one layer to another, the layers must be of the same type, i.e. material to material.

Preset Browsers Palette: This command opens a 'Preset Browser' palette.

Save Preset: Any layer in the shader tree can act as a preset. When one selects 'Save Preset...' a file save dialog is presented asking where you'd like to save your preset. If you want the preset to show up automatically in the existing set, you'll need to navigate to the directory where your presets are stored and save within that directory, one can also create a new directory for saving presets, but make sure to use the '(add path)' command so you can see them within the browser viewport. Often a surface made up of multiple texture layers, in that case make sure to create the preset from the material group () , as that will include all associated texture layers within the preset. Don't forget that Fur is made up from texture layers so making hair presets is just as easy as making surface presets.

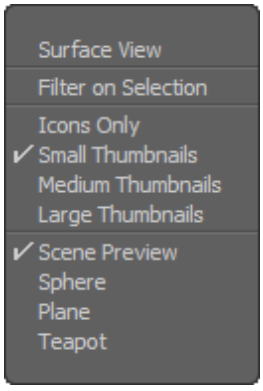
Change Type: This command offers the ability to convert a layer of one type, such as an image map to another type such as a procedural. Associated 'Texture Locators' will retain their settings if applicable.

Bake to Texture: This command offers the ability to generate a texture map image of a layer. Baking textures from inside the Shader Tree is useful for baking a single 'Effect' such as Color or Diffuse into an image whereas the 'Bake' command from the Render menu bakes all light and texture contributors into a final shaded image. It is reasonable to think of the Shader Tree level of baking as similar to *flattening* layers in Photoshop.

Only layers of the same "Effect" setting underneath the texture layer that is triggering the bake will contribute to the baked layer. Once "baked" this texture can remain in the Shader Tree as a pre-computed result of all contributors which results in a savings of render time and provides a real-time view of the effect. The 'Bake to Texture' command requires a UV map on the baked surface as well as a blank image map to bake into.

Bake to File: The 'Bake to File' option works in an identical manner to the 'Bake to Texture' function, however instead of simply baking to the render display viewport, a file requestor is opened and the render is saved to disc rather than into memory.

Bake From Object to Texture: This command offers the ability to use the UV map of a low polygon count model to bake a texture map from a higher resolution model. Maps can be used for any of the various 'effect' including diffuse color and displacement. Like 'Bake to Texture, a UV map is required and a blank image map to bake into. Both surfaces must have different material name and cannot share the same UV map. When calculating a displacement map, the 'distance setting' from the popup requestor should be used as the 'Displacement Distance' setting in the material dialog box when applied.



There are also a few viewport specific options available by RMB-clicking on the 'Shader Tree' tab itself and choosing 'Viewport Settings'

Surface View: When enabled, this setting changes the hierarchy of the Shader tree layers to a surface centric version that may be more familiar to LightWave users where surface are organized by their associated mesh items.

Filter on Selection: When enabled, this setting will only display the various surfaces corresponding to any selected mesh items.

Icons/Thumbnails: The various Icons/Thumbnail settings offer options to allow the display of small icons representing the individual material groups. 'Icons Only' disables the Thumbnails, while the other options render thumbnail previews of various sizes.

Preview Type: This group of options only affect the viewport when thumbnails are enabled. The various options change the way the thumbnails themselves are generated.

Info: For a full overview of working with the layers of the Shader Tree, please reference the Shader Tree page of the Shading, Lighting and Rendering section.

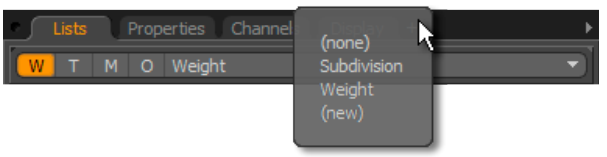
Vertex Map List

The Vertex Map List view lists vertex maps for the current mesh in groups: UV, Weight, Morph, and Other. The "Other" list includes the type of the vertex map as well as just the name. The list of maps available is the union of the named maps in all the *active* (selected) meshes.

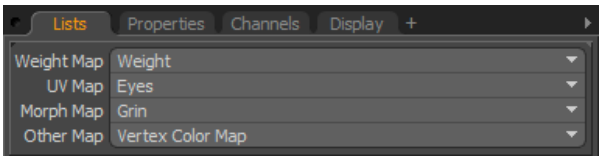
From within the Vertex Map list viewport you can select (new map) regardless of the viewport size. If the viewport is compressed such that you are presented with a single or multiple pop-ups, you will find the (new map) option inside the pop-up. If you accidentally choose (new map) from the wrong map type pop-up, you can simply change the map type on the new map requester that appears when creating a new map. When the viewport is expanded to its full size, expanding any of the vertex map types in the list will present you with the (new map) option. Again, selecting this option in the list will open the Create New Vertex Map requestor.

When working in the vertex map viewport, the current selection is maintained even when selecting new mesh layers, if each layer contains a vertex map with the same name. For instance, if you have a mesh layer with a texture map named "SteelMap" selected in the vertex map list, changing layers to another mesh that also has a texture map with the same name will automatically select that vertex map. This is very handy when working on objects that have been broken up against multiple layers.

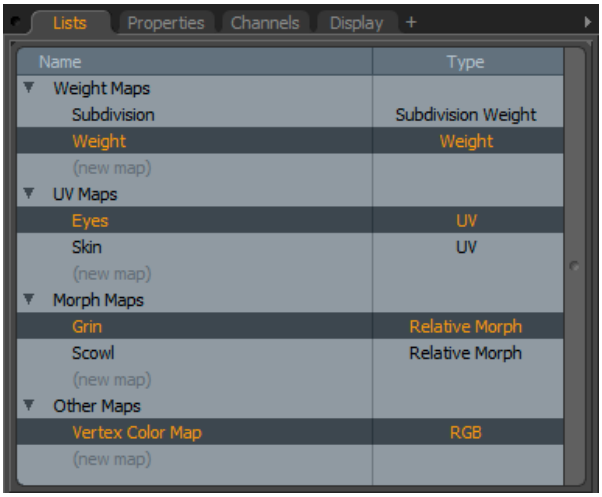
When the Vertex Map List viewport's size is changed, it will automatically change its appearance so that the viewport is more appropriately displayed for the space available. The viewport has 3 size settings: Compact, Medium, and Expanded.



Compact: The viewport is minimally displayed as a list of four buttons W,T,M, and O (Weights, Textures, Morphs, Other) next to a pop-up list. The four buttons allows you to quickly set a filter for the pop-up list. If W is selected, the pop-up list will display all Weight Vertex Maps in the current mesh layer. To see the list of Texture maps, you would click on the T button and so forth. The "Other" vertex map list contains color vertex maps as well as Pick maps or custom vertex maps added by 3rd party extensions or foreign data formats.

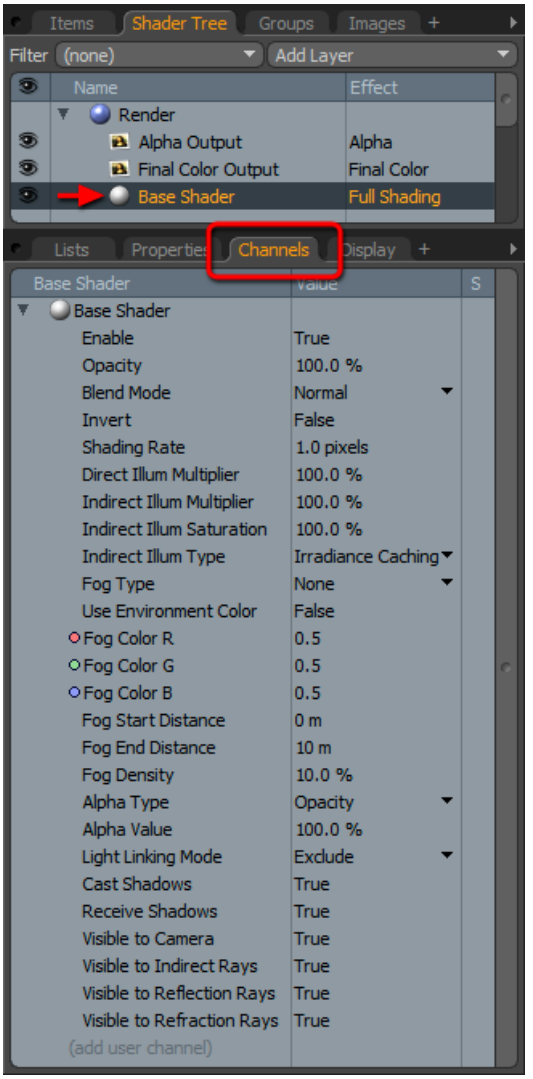


Medium: The viewport is displayed as four pop-ups stacked vertically. The type of vertex map is clearly labeled next to each pop-up.



Expanded: The viewport is displayed as a tree with four categories with expand/contract widgets on the left side of the text. In this view it is possible to view and select multiple vertex maps from the four vertex map groupings.

Channels



The Channel viewport provides an exhaustive window into the soul of the modo animation system. All Channels of *any* item can be viewed in the Channel Viewport. Channels are simply any of the animateable attributes of an item (the identical attributes found in the Properties viewport!). The Channels viewport provides a direct method for selecting channels as well as creating, navigating and managing any keyframes on said channels. There are three columns on the Channel viewport; Channels, Value and Status. Obviously the first column displays the channel names for the currently selected Item, the second column displays the value of the channel at the current scene time and the Status (S) column shows the state of the channel be it animated, static or actually on a keyframe. This is communicated via the same *dot* method used on the Channel keyframing control.

Interacting with Channels--

Same as the 'Properties' viewport, the Channels viewport updates to reflect the current item selection. This can be items within the item list, the shader tree, or even the Groups palette. Once selected, all the animateable attributes appear in the Channels viewport. To affect an attributes value, the user simple needs to LMB+click directly over the current value in the 'Value' column. For toggles (boolean values), this will change the value field between true/false. For attributes with various properties, a menu will open presenting all the possible options. For numerical values, users can use the click+pause action to activate the field to enter a new numerical value, pressing enter to set it, or simply clicking away form the field.

Setting Keyframes--

Users can set animation keyframes in all the usual ways using the keyframe buttons, the 'S' keyboard shortcut, in the 'Properties' viewport, or within the 'Graph Editor'. To set a keyframe for a value directly in the Channels viewport, users can RMB+click on the target value and choose the 'Add Key' option from the context menu. Users may also remove keyframes using this methodology. When any attribute has a keyframe, colored circles will appear next to the value in the 'S' (Status) column referencing the attributes current state. The possible states are--

None- Channels is at the default value

Constant- Channel has a constant value

Animated- No keyframe at the current time position

Animated- Keyframe at the current time position

Mixed- Channels have differing values or state

 Driven- Channel is driven by an external source

 Undefined- Channel is not defined

An arrow over the dot represents that channel as driving another for a channel link.

 Driver- Channel is linked to another

A gear icon represent a channel that is being controlled by another for a channel link

 Driven- Channel is linked to another

Users can automatically select input/output items by selecting the appropriate channels and pressing ' i ' to select *input*, and 'o' to select *output*.

Contextual Menu--

RMB+clicking in the Channels viewport opens up the contextual menu with options for modifying the keyframes of the various channels.

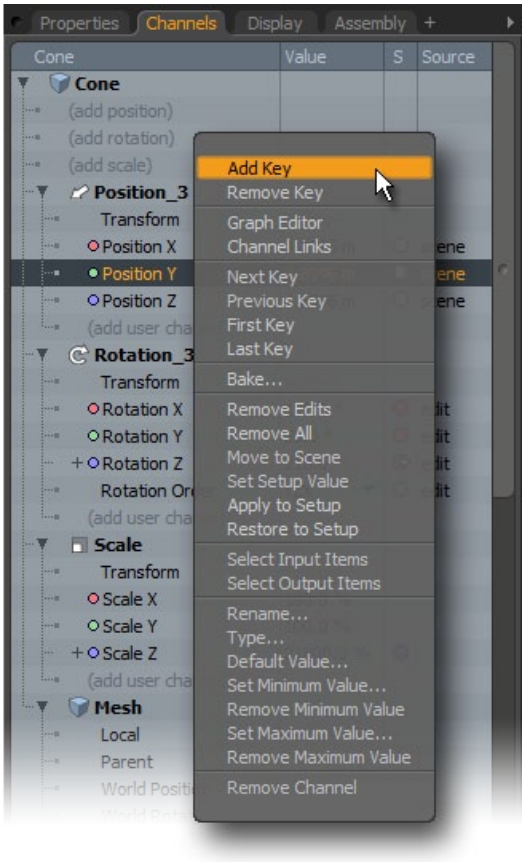
Add Key: Sets a keyframe at the current time with the current value.

Remove Key: Removes the keyframe at the current time.

Graph Editor: Opens the Graph Editor window.

Channel Links: Opens the Channels Links palette

Next/Previous/First/Last Key: Moves the timeline to the appropriate time.



Bake: Bakes, or *freezes* the current animation to individual keys at every frame. Useful in exporting animations of driven items.

Remove Animation: Removes all keyframes, retaining the value of the initial key.

Remove All: Removes all keyframes and returns to the default value.

Select Input Items: For driven channels, this option selects the driver item or channel.

Select Output Items: For driven channels, this option selects any items or channels being driven by the current channel.

The following options are relevant only to user created channels.--

Rename: Opens a dialog box to Rename the selected user channel.

Type: Opens a dialog box allowing users to modify the type of user channel.

Default Value: Opens a dialog box allowing users to specify a default value for the user channel.

Set Minimum Value: Opens a dialog box allowing users to set a minimum value for the user channel.

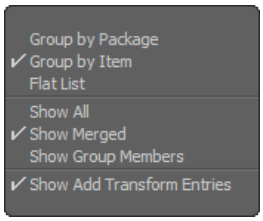
Remove Minimum Value: Removes any minimum value, if specified.

Set Maximum Value: Opens a dialog box allowing users to set a maximum value for the user channel.

Remove Maximum Value: Removes any maximum value, if specified.

Remove Channel: Deletes the selected user channel.

Context Menu--



RMB+clicking on the 'Channels' tab will open the viewport menu. Select the 'Viewport Settings' to open the Channels viewport options menu.

Grouping: The first three options define how channels are grouped in the channel list

Group By Package- Organizes channels by package, sub-groupings of related elements by individual item.

Group By Item- Organizes channels strictly by item.

Flat List- Channels listed as straight list with no grouping.

Show All:

Show Merged:

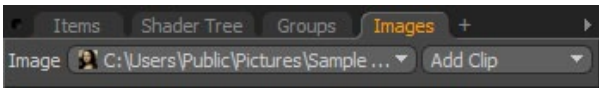
Show Group Members:

Show Add Transform Entries: Displays the 'Add Position/Rotation/Scale' entries to the channel list. These options provide a shortcut to users for adding additional transform channels, but for smaller monitors removing them can save some screen real-estate.

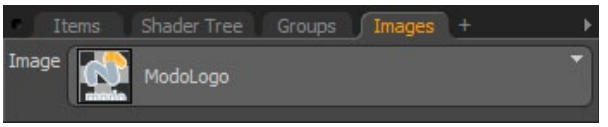
Image List

The Clip List (also known as Image List) viewport provides direct access to all images loaded into modo organized by scene. Different sizes of the viewport will display the information differently. In the smallest *compact view*, users can select particular images and add images to the list using the 'Add Clip' button. Hovering the mouse over the edge of the frame and LMB+click dragging the frame resizes it reveals larger views. The small list only displays the current image, but when clicked opens a gallery browser for navigating and adding images. At full size the list view shows the image by name, includes an icon of the image for visual reference and has data about the resolution and format of the image. Users can also select a 'Gallery' view from the viewport widget, choosing the 'Viewport Settings' menu. In the case of a viewport *tab*, the Options menu is available by RMB+click on the tab name itself. The menu also offers some additional options for the sizing of the icons while in gallery mode. Additionally, the gallery mode has a slider across the top of the viewport where users can resize the icon images interactively.

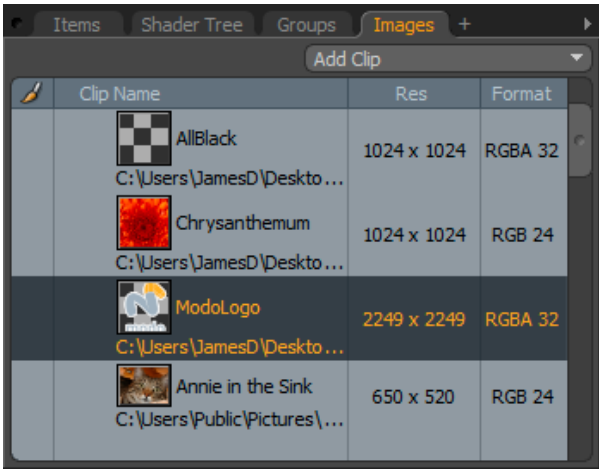
Clip List Compact



Clip List Small



Clip List Full Size



Clip List Gallery

The 'Add Clip' button provides users commands for not only loading still image and image sequences, but for creating blanks new images/sequences, saving users a trip to an external image editor. Selecting any of the commands opens an OS specific dialog box where users can navigate to a folder and either load or save the image, depending on the operation chosen.

Double clicking on an image in the viewer will open it in a Preview window set to 'Clips' so you can view the image at a larger resolution. RMB+clicking on an image provides the image item context menu with a variety of options--

Rename: Renames the current image. This new name only affect the reference name in modo and does not rename the file itself.

Delete: Removes the image from the scene. If the image is applied to a surface, modo will popup a requester stating that the images is in use and asking if you are sure you want to remove the image.

Reload: This command reloads the image as saved to disk. Useful for when edits made in an external app don't trigger modo to reload it automatically.

Replace as Still: Will replace all instances of the current image/sequence in the scene with a new still image loaded from disc.

Replace as Sequence: Will replace all instances of the current image/sequence in the scene with a new image sequence loaded from disc.

Save: Saves the current image to disc, overwriting the previous version. Use this command to save any changes made to images using modo's painting and sculpting tools.

Save As: Saves the current still image to a new location and/or name.

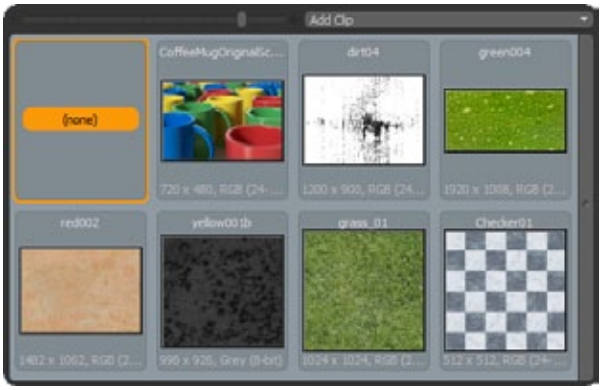
Save Sequence As: Saves the current image sequence to a new location and/or name.

Preview

Users can see a preview of any image in modo by simply LMB+double click on any image in the image list/clips browser. This opens an 'Image Preview' pane. Users can move around the image using the navigation widgets in the upper right corner. Simply 'LMB+Click drag to operate. Alternately users can use the keyboard shortcuts; 'Alt+LMB click drag' to pan, and 'Shift+Alt+LMB click drag' to zoom. To close the palette, users can click the OS specific window close mechanism, or by LMB+double click on the initial image again.

Clip Browser

The 'Clip Browser' is a quick and easy way to view and select images loaded into modo for use in a variety of situations. Typically users load images and sequences using OS specific dialogs. When re-using images presently loaded, users can browse through them here. Working like a combination 'Preset Browser' and 'Image List' view, the Clip Browser allows users to scale the icons using the slider in the upper left hand corner. LMB+click and drag to the right to increase their size, and left to make them smaller. Users can access the Clip Browser in the Shader Tree 'Add Layer' pop up menu ""Add Layer > Image Map > Use Clip Browser". It can also be found in the 'Paint' interface layout tab for selecting images for 'Image Ink' and 'Brush Tips'. TO select an image for use, simply LMB+click on the image icon dismissing the viewport.



Add Clip

In the upper right hand corner of the viewport, LMB+clicking on the 'Add Clip' option offers some options to load additional images and sequences into modo.

Load Image

Selecting the 'Load Image' command opens an OS specific dialog box where users can browse and select images saved to the system drive(s). modo accepts a variety of bitmap formats covered on the 'Image Formats' page of the documentation. Once an image is selected and the user presses 'Open' to load it, the image is utilized as if it were selected in the browser, dismissing the window.

Load Sequence

Selecting the 'Load Sequence' command opens an OS specific dialog box where users can browse and select still image formats saved with sequential numbering for use as animated clips. modo accepts a variety of bitmap formats covered on the 'Image Formats' page of the documentation. Once the first image in the sequence is selected and the user presses 'Open' to load it, the sequence is utilized as if it were selected in the browser, dismissing the window.

New Image

Selecting the 'New Image' command opens an OS specific dialog box where users first browse for a location and designate a name for the image file, selecting the format with the 'Save as Type' options. modo writes a variety of bitmap formats covered on the 'Image Formats' page of the documentation. Once an image location and name is defined and the user presses 'Save' to create, modo opens the 'Create New Image' dialog where the users defines the images pixel size and bit depth and fill color if

desired. Once the user selects 'OK', the image is created and applied as if it were selected in the browser, dismissing the window.

Blank images are useful for painting, image based sculpting and baking. Once the user has modified the image, it will need to be saved. Users can RMB+click on the image in the Images List to open the context menu and choose the 'Save' option, or in the menu bar, use the command "File > Save Image" option.

TIP: To save an image as 16bit, users must enable the 'Floating Point' option or images will default to 8 bit regardless of the chosen image format.

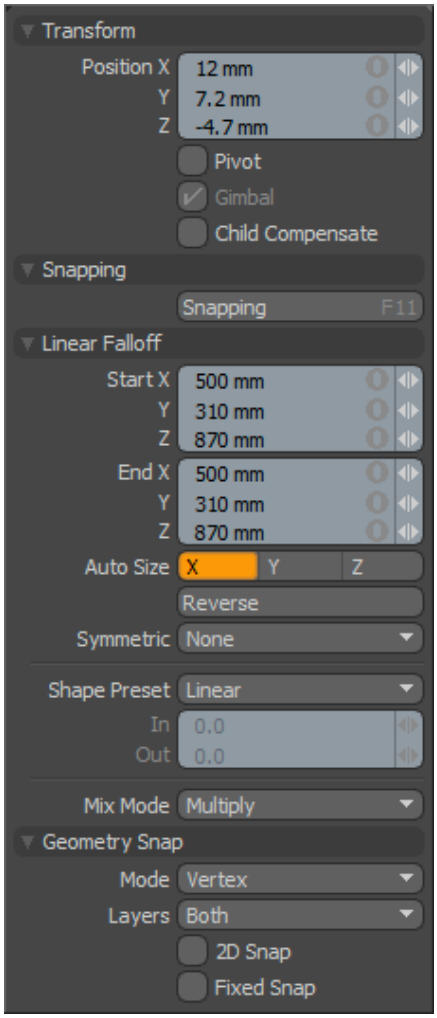
New Sequence

Selecting the 'New Sequence' command opens an OS specific dialog box where users first browse for a location and designate a name for the image file, selecting the format with the 'Save as Type' options. modo generates Sequences as consecutively numbered still image files, writing a variety of bitmap formats covered on the 'Image Formats' page of the documentation. Once a sequence location and name is defined and the user presses 'Save' to create, modo opens the 'Create New Sequence' dialog where the users defines the initial images pixel size and bit depth and fill color if desired. Once the user selects 'OK', the image is created and applied as if it were selected in the browser, dismissing the window.

Blank images are useful for painting, image based sculpting and baking. Once the user has modified the image, it will need to be saved. Users can RMB+click on the image in the Images List to open the context menu and choose the 'Save' option, or in the menu bar, use the command "File > Save File". The length of the sequence is determined by the number of frames the users creates while in modo. For example, when painting a sequence, when the current frame is incremented (press 'Shift+Right Arrow Key to move forward), modo duplicates the last frame. Users can then add to the image frame. Each new increment will duplicate the frame in its current state. Users can also return to further edit previous frames. modo will only create the length of frames that the users specifically generated by incrementing the timeline. For a 300 frame sequence, if only 3 frames are actually generated and saved, then when rendered, modo will simply repeat the 3rd frame until the last frame of the rendered sequence. Once the user has modified the sequence, it will need to be saved. Users can RMB+click on the image in the Images List to open the context menu and choose the 'Save Sequence' option, or in the menu bar, use the command "File > Save Sequence" option.

Properties Viewport

Toolbox Properties

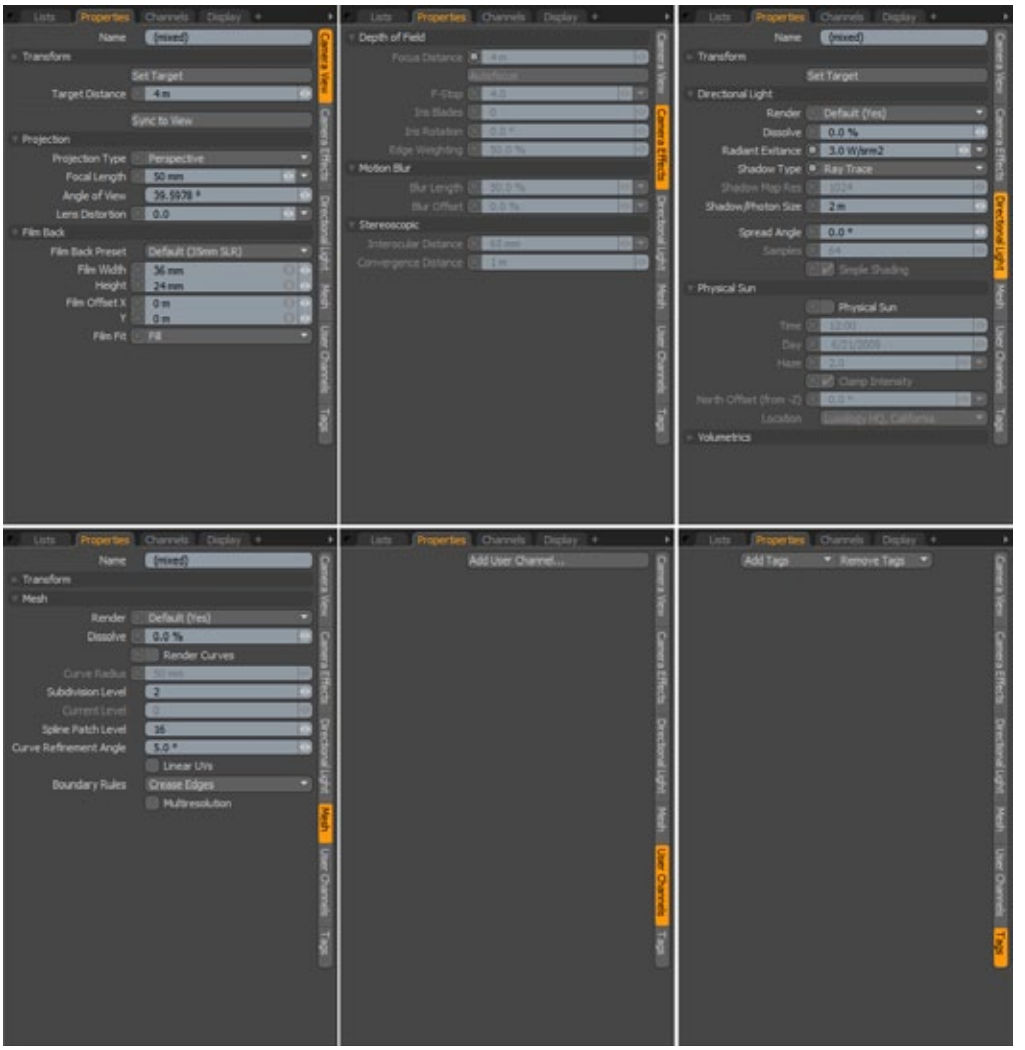


Properties viewports are the chameleons of the viewport family as they are basically empty frames until an item or tool is selected. Once selected, all the editable attributes related to that item are listed in the panel where users can make changes to the options available and, if applicable, see them update in the viewport or render previews.

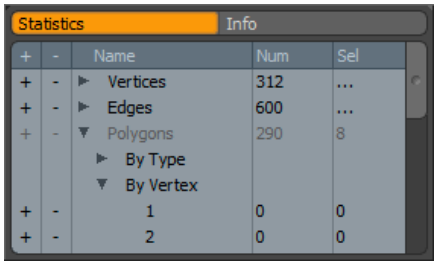
The main properties panels are those of the toolbox and item properties. In the default modeling interface, the toolbox properties panel appears on the left side of the interface under the toolbox itself. Upon loading, the viewport is empty, but as tools are selected, the attributes appear in the panel. Additionally, when falloffs, actions centers and snapping modes are enabled, additional section of the properties panel will begin to fill presenting all the editable settings related to the active tools and functions.

This is the same case with the item properties, located on the right hand side of the default interface. Unlike the tool properties, it is a tab, so users may need to select the particular 'Properties' tab to make it visible. Upon loading modo, the panel will appear empty, but selecting items populates the panel with the various attributes related to the selected items. Unlike the tool properties where the various additional attributes stack up, the Item properties adds additional sub-tabs down the right hand edge of the panel that relate to all selected/active item layers. Users can select from the available sub-tabs by simply clicking on the named button.

Items Properties Tab



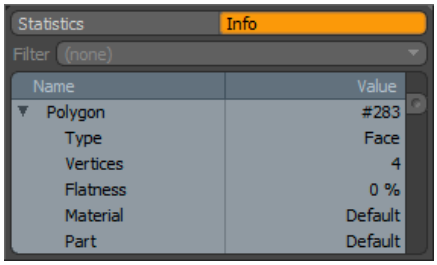
Info & Statistics



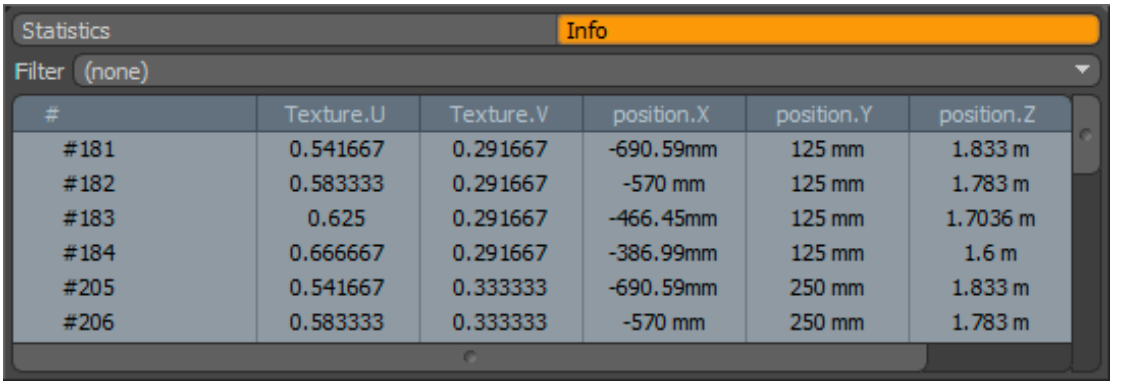
The Info and Statistics viewport provides information about the currently selected/active geometry in the scene. It also provides users with powerful selection and precise editing functionality. The viewport is made up of two tabs: a 'Statistics' tab, and an 'Info' tab.

The Statistics tab is an integral part of the modo workflow, giving users a wide range of tools to use for selection. This is a very powerful alternative to manually selecting vertices, edges, polygons and materials in the 3D viewport using a mouse (or tablet). Selection are made via the +/- columns. Simply navigate to the appropriate statistic, such as polygons by number of vertices, and press the + to select (add) and - to deselect (subtract). When working with the various components (vertex, edge or polygon) make sure to also be in the appropriate mode.

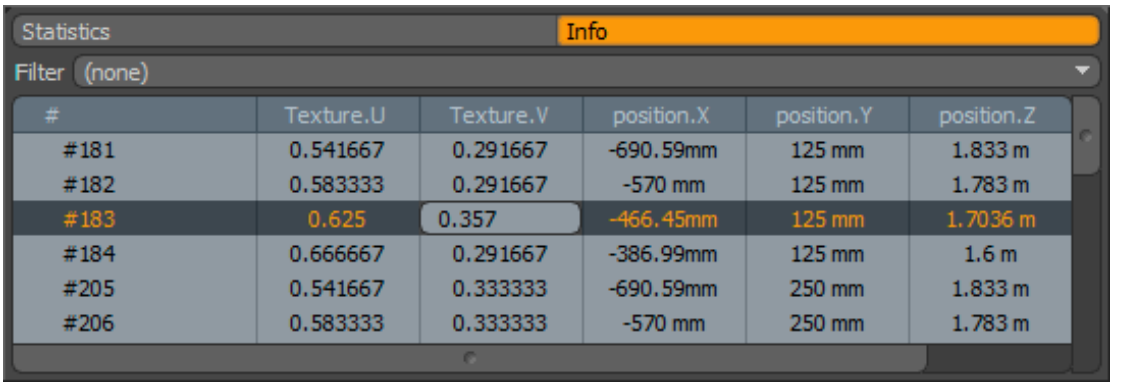
The Statistics tab also displays total count and number selected of all vertices, edges, polygons and materials in the scene. You can use this for very fast selection of the desired components in your scene. Expanding the entries in the Name column by pressing the preceding arrow icons, gives users access to numerous methods for geometrical selection. Some of these include the ability to select vertices by the number of edges or polygons they connect, by vertex map, or by named selection set. For Edge Selection you can also select based on boundary. For polygon selection you can select polygons by type, number of vertices, by named part or selection set, or by material.



The Info tab displays vertex indices, subdivision weights, texture UVs and position in X,Y and Z. The information displayed will change depending on the current mode. The Info tab allows inline editing of the displayed values for precise numerical editing. When the view is stretched wide horizontally it reorganizes into a spreadsheet view. Click on any cel you wish to edit and type in new values. The filter option provides a means for only displaying a certain type of information simplifying the list when editing large numbers of values.



When opened wide, the Info display changes into a spreadsheet view.



Simply LMB+click a values to numerically enter a new value. This also works for multiple selections

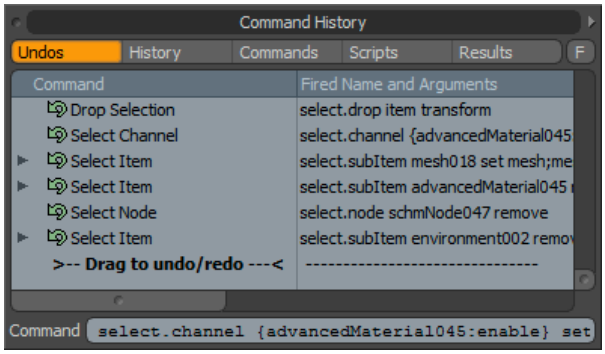
Command History

modo keeps a listing of every command used during a session in the Command History viewport. This view is useful for quick undo/redo operations, managing macros, quickly assigning keyboard shortcuts to specific commands and many other utility functions. To access the Command History, look at the bottom right-hand corner of the interface where there is a single Command field, then drag the divider upwards that is just above that panel until the Command History Header is displayed.

There are five tabs in the Command History panel: Undos, History, Commands, Scripts and Results. By default the Undos list is active. Within this list are sections for Command, Fired Name and Arguments and Description.

You can drag within the Command list to undo/redo. Locate the line that reads "Drag

to undo/redo" and then simply drag it up to undo many commands at once. By right-clicking any command item you can quickly Map that command to a Keystroke.



To control the maximum number of available undos, go to the System menu and choose Preferences > Input > Remapping then adjust the value in the Max Undo Level field using the mini-slider. In the same form, the Group Selection Cmds checkbox controls the way undo deals with selection operations. As a default this is turned on, and it provides the behavior that groups selection operations into a single undo. Turning this off, provides the ability to step back through the selection of polygons, edges, or vertices at a more granular level.

Clicking on the Commands tab will give you a complete list of all the commands available within modo.

Clicking on the Scripts button within the Command History will give you access to all recorded macros stored in the config. Here you can Edit or Run individual lines in these scripts.

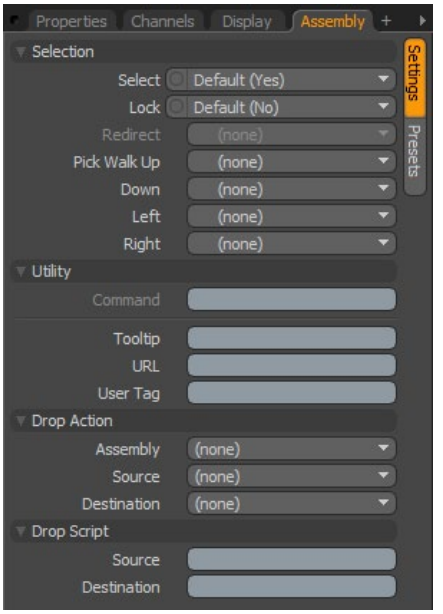
Utility Viewports

Utility viewports serve a number of purposes in modo, providing a way for users to define and select color schemes in the 'Color Picker', edit the color values of a gradient in the 'Gradient Editor', or simply display the rendered image when pressing the 'F9' key. These viewports are found throughout modo, depending on the particular interface and utility.

Utility Viewports			
Assembly	Gradient Editor	Preset Browser	Tool Pipe
Color Picker	Graph Editor	Preview	
Display	Groups	Render Display	

Assembly Viewport

The 'Assembly' viewport provides functionality for selection locking and for 'Pick Walking' up and down a hierarchy with the keyboard arrow keys; options that are essential to streamlined animation workflows. The Utility section is related to items once saved as an Assembly Preset and defines how the item reacts to being placed from the preset browser into a given scene.



Selection--

Select: Controls whether or not items are selectable within any 3D viewport, 'Yes' enables direct item selection, 'No' disables it. Items designated 'No' can still be selected via the 'Item List'.

Lock: When 'Lock' is enabled by selecting 'Yes', items will be locked from the application of any type of item transform.

Redirect: When an item is selected in a 3D viewport, specifying an item for 'Redirect' will trigger modo to automatically select the alternate item instead. Helpful in easing item selections in complex hierarchies, locators with special Display properties can be used visualize selectable items in the viewport that redirect to the actual item you wish to modify.

Pick Walk: The arrow keypad can be used as a way to 'walk' selections through a hierarchy. By default, parent items are selected by hitting 'up', and child items are selected by hitting 'down'. The 'Pick Walk' function allows users to specify which items specifically are selected by any of the 4 keys, Up, Down, Left and Right. Say for instance you when you reach the bottom of a hierarchy, the foot, you could specify by hitting the left or right arrows the opposing foot would be selected, and by hitting down the head would be selected, an so forth.

Utility--

Command: The 'Command' option specifies a command to fire when the item is clicked on in the 3D view. This can be any of modo's commands. There are two commands that are especially useful when setting up assemblies:

----- item.channelHaul
This will select any user channels on the item and activate the Channel Haul tool.

----- item.channelPopover
This displays a popover form containing controls for any user channels on the item. If used in conjunction with selection direction this can also be used to display the channels belonging to a specific Group.

Tooltip: Tooltips are the small blocks of text that appear, if assigned, when a mouse point momentarily hovers over an item. These can be notes or reminders to what an item is intended to do in a scene but tooltip's can also be useful for an assembly author to provide some contextual instructions on how to work with an item in an assembly.

URL: If a custom URL is assigned to an item, Using the 'F1' help feature and clicking on the item in the 3D viewport will open that particular location. Useful for users to add custom documentation to an item.

User Tag: This is a text string that can be used to identify particular items within an assembly or scene. These tags can be read by scripts with the 'item.userTag' command. An example of use might be to identify items to be deleted or hidden when an assembly script has finished etc.

Drop Action--

Assemblies offer the user the ability to create a rig using modo's many animation modifiers and then save the setup as a Preset for easy application elsewhere. In the following descriptions the item being dragged is referred to as the 'source' item and the item that has received the drop is referred to as the 'destination' item.

Assembly: Users can choose from a series of actions that will be performed when the preset is initially dropped into the scene from the Preset Browser. The possible actions are as follows-

Parent- Source item is made a child of the Destination item.

Parent in Place- Source item is parented as above, but with 'Compensation' applied.

Match- Source item is modified to match position scale and rotation of destination.

Match Position- Source item is modified to match the position only of the destination item.

Match Rotation- Dropped item is modified to match the rotation only of the destination item.

Match Scale- Dropped item is modified to match the scale only of the destination item.

Insert- Inserts the source item into the hierarchy of the destination item. The item

will be positioned and orientated to match the destination item and will become the parent of the destination item.

Insert at Parent- Similar to 'Insert' but the source item will match it's position and orientation to the destination item's parent (if a parent item is present).

Insert in Place- The source item will be inserted into the destination item's hierarchy as it's parent but will retain it's current position and orientation.

Source: The Source Drop Action will be performed when the assembly item is dropped onto a scene item. Same action options as 'Assembly'.

Destination: As above, but action will be performed when a scene item is dropped onto the assembly item. Same action options as 'Assembly'.

Drop Script--

Additional to the above actions, scripts can be assigned to run when certain actions occur.

Source: The assigned script (omitting the @) will run when this item is dropped onto another in the scene. The script gets the source item and destination item passed as arguments. If multiple items are dropped then the script is called for each item in turn.

Destination: Same as above but script will run when any item in the scene is dropped onto the item it's assigned to. As with the Source Drop Script the source and destination items are passed to the script as arguments.

Assembly Presets

Assembly Presets tab--

Assemblies once saved to the proper destination will appear in the Assembly preset tab. The Assembly preset browser works the same as the standard preset browser.

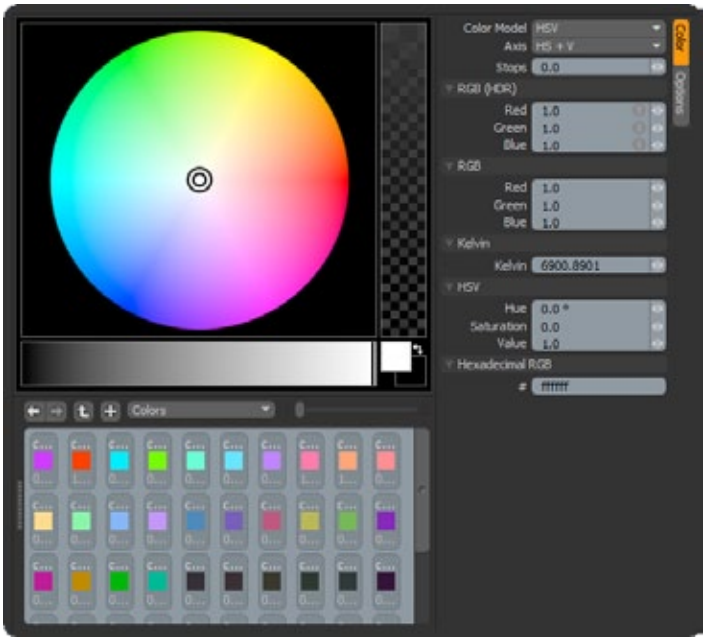
To save an Assembly Preset, just select the root item of your setup in the 'Item List', or all the items you wish to save, and go to "File > Save Assembly..." Any scripts used by the preset need to be saved into the root script location so they can be found automatically.

Color Picker

Color is an integral function when painting and surfacing for 3D. Color sets the mood, leads a viewers eye, and provide a sense of cohesion to a scene. modo provides a dedicated window for editing colors and creating schemes for scenes, that can then be easily applied when surfacing, setting up lights and any other color related operation. The color picker opens whenever a user LMB+clicks on a color swatch in modo. Being non-modal, the popup disappears as soon as the user moves their mouse cursor away from it.

The viewport is dominated by the color picker with its color wheel on the left, the color options on the right, joined by a preset browser across the bottom. Combined with its many options, the color wheel is a full-featured color scheme generator, with options for creating a variety of automatically coordinated schemes based of various rules users can adjust.

The wheel itself is straight forward, users only need to LMB click over the color they wish to choose, while LMB clicking and dragging, users will update the value in real-time, releasing the mouse pointer when the proper color is arrived at will select it. Depending on the color model, the bottom color strip will update accordingly, but users can LMB+click and drag in that window as well. The right hand side of the color picker offers users the ability to define alpha (transparency) value, with the box in the lower right hand corner displaying the current color swatch. Users may RMB+click the color swatch to Save the color as a preset. Automatic preset saving can also be accomplished by dragging and dropping the swatch into the preset window. For aspects of modo that require two colors, such as when working with procedural textures, users can simultaneously define both colors by clicking the arrow to swap foreground/background colors, define color and clicking again to swap back. The properties panel on the far right side controls the display of the picker, and allows users to define numerical values as well set the scheme rules, and save presets to the browser on the next page.



Color Model: This option allows users to choose the color model that suits their needs best, there are three current options-

HSV-- Standing for 'Hue, Saturation and Value'; it is considered the most intuitive color model, users can adjust colors by their 'Hue', the apparent color itself, saturation, which controls the intensity of the color, and value, which controls how light or dark the color is. This is also the only model that is displayed as the familiar color wheel.

Kelvin-- The Kelvin scale is based on thermal dynamics, and considers colors as temperatures. This single axis scale is useful in describing the color of light sources, as the hotter they burn, the brighter, and more white they become. The goes from warm colors, reds and oranges, through yellow to white, and then extending to a slight blue as it increases to the maximum value.

RGB-- Standing for Red, Green and Blue, the model itself is based on the additive property of light, and is the primary model for electronic displays and sensors (such as your monitor and digital camera). If pure versions of each color are combined, they produce white light. While it is a common model, its not very intuitive, as users often don't have a good idea about what will happen if they reduce 'Red' by 50%. Still, it is the most common format for colors when exchanging data between applications.

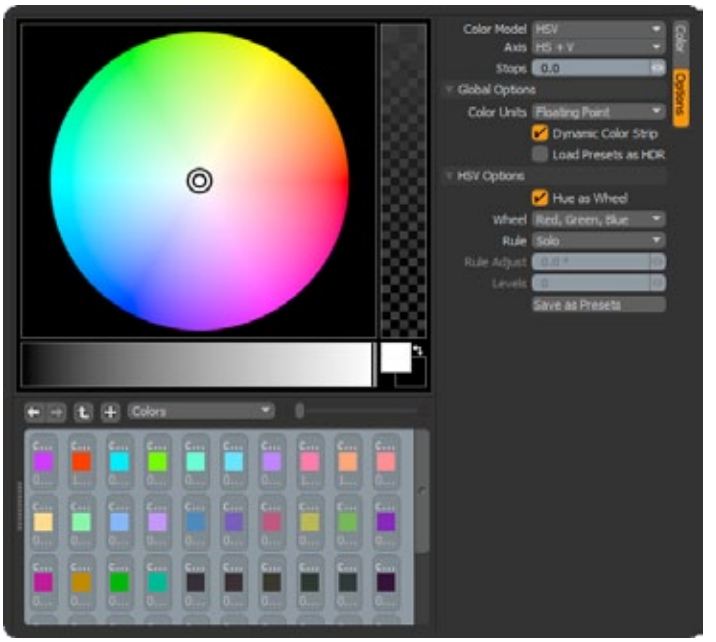
Axis: Since the color picker is two dimensional, and the various color spaces are multi dimensional, the additional axes options modify how users interact with the picker providing additional options for customizing the picker and color strip to suit the users particular needs.

Stops: This option is related the f-stops, familiar to photographers. It is intended to aid in previewing and selecting HDR (High Dynamic Range) colors, whose brightness goes beyond what typical computer monitors are capable of displaying. At the default 1.0 value, the color picker works like any other, maxing out at pure white. Changing the stops value to 1.4 allows users to define colors whose brightness is twice that of the 1.0 setting. A stop of 2.0 would again be twice as bright (or four times brighter than 1.0). These values work the same as those in photography, where each additional stop value lets in twice as much light as the previous. The standard photographic progression of stops is 1, 1.4, 2, 2.8, 4, 5.8, 8, 11, 16, & 22. It should be noted that the 'Stops' value only changes how modo displays the colors in the picker and allows users the ability to choose over bright values in the color wheel, and doesn't change the color value itself.

Models--

The five following sections, 'RGB(HDR)', 'RGB', 'Kelvin', 'HSV' and 'Hexadecimal RGB' are simply numeric value input fields for displaying and entering colors using alternate color models. Displayed values will always represent the current foreground color and will update accordingly as the users interact with the picker. All models, except 'RGB (HDR) are not capable of defining or displaying HDR colors, and will clip at the maximum values for the selected model (showing whatever the equivalent color is); the maximum value will be different for each of the various 'Color Units' options as defined by the user.

Global Options--



Color Units: The color units option changes the global preference for how color values are entered into modo color fields. There are four options-

Floating Point: For this model, standard color values exist between 0 and 1, with HDR colors values being those beyond 1.0

Percentage: Color values are defined as percentages, similar to Floating point values, where up to 100% would be normal color range, and values beyond 100% extend into the HDR colors.

Integer: This is the standard format for most image editing applications, where standard color values exist in the 0 to 255 range.

Hexadecimal: Hexadecimal colors are the standard for defining color on the web as they are natural values for computers to understand, similar to binary, where the two values represent 255 different levels.

Dynamic Color Strip: When this option is checked, the Color Strip bar will update dynamically showing the the color selection as a component of the bar itself. When unchecked, the bar will simply show its single color axis, as defined by the 'Axis' option.

Load Presets as HDR: When a value is saved, its tagged as either a LDR (low dynamic range) color or HDR (high dynamic range) color. Enabling the 'Load Presets as HDR'

will allow users to only load HRD color presets from the browser, effectively hiding all the LDR colors from view (they are still there, only hidden) This is an easy way to determine if colors are HRD or LDR in a given scheme.

HSV Options--

Hue as Wheel: When the 'Color Model Options' of 'HSV' is selected, enabling the 'Hue as Wheel' option will display the picker as a continuous wheel. When displayed as such, several options related to making rule based color schemes becomes available.

Wheel: This option defines the type of HSV wheel to display, users can choose the standard 'Red Green Blue' (RGB) option common to many image editing applications, or the more familiar and artist friendly 'Red Yellow Blue' version. Unlike the RGB version, the RYB version works like the color wheel many artists learn in art school, where red & green, blue & orange, and yellow & purple oppose each other on the wheel.

Rule: modo allows multiple options for generating automatic schemes based on common color rules-

Solo-- The default mode, only a single color is defined.

Complementary-- Two colors whose positions oppose each other on the color wheel.

Analogous-- Three colors that contain a similar hue, appearing close to each other on the color wheel.

Triadic-- A three color scheme, where colors appear equidistant from each other on the color wheel.

Tetradic-- A four color scheme, where colors appear equidistant from each other on the color wheel (dual complementary).

Compound-- A five color scheme, with a main color and two split complementary colors.

Tints--Tints appear as graduated value (brightness) variations of the foreground color toward white.

Shades-- Shades appear as graduated value (brightness) variations of the foreground color toward black.

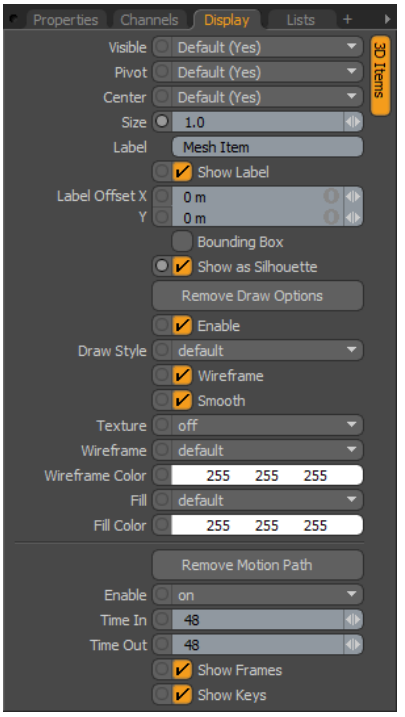
Rule Adjust: Users can adjust the angle of degrees between scheme rules that adjust the scheme to suit their own taste.

Levels: This option becomes active for the 'Tints' and Shades' rules, defining how many steps will be in the scheme.

Save as Presets: Pressing this button will save the current foreground color into the 'Presets > Colors' folder of the preset content. If the default content hasn't been installed, users will need to manually define a directory and add it to the preset browser using the 'Add Path..' option. When defining color schemes using the rule options, all the color defined in the swatch patch will be added to the preset browser simultaneously.

Display Viewport

The display viewport offers users the ability to control how individual items appear within a 3D viewport over-riding the viewport style itself. To modify any of the item display properties, simply select an item within a viewport, or the item List, and open the display viewport panel. Depending on the type of item layer selected, slightly different options will appear. All the various options are covered below.



3D Mesh Items Display Options

Visible: Sets visibility for the 3D item itself, 'Default' respects viewport settings, 'Yes' set the item as visible always, 'No' disables visibility entirely.

Pivot: Sets visibility for the 3D item pivot, 'Default' respects viewport settings, 'Yes' set the pivot as always visible, 'No' disables pivot visibility entirely.

Center: Sets visibility for the 3D item center, 'Default' respects viewport settings, 'Yes' set the center as always visible, 'No' disables center visibility entirely.

Size: Multiplier that adjust the display size of the items representations, for camera, light & locator icons. Does not affect the representations of mesh layers.

Label: Typing in text for the Label will display the text positioned next to the 3D item in the viewport for easy identification. It will also add two additional properties to control the display of the label in the viewport -'Show Label' and 'Label Offset'.

Show Label: This toggle will temporarily enable/disable the display of the label in the viewport without losing settings, only available when a label is specified.

Label Offset: This value specifies the distance away from the locator where the label will appear, only available when a label is specified.

Bounding Box: When *enabled*, mesh items will display as a simple wireframe box around the geometry's total volume instead of displaying the geometry itself.

Show as Silhouette: When *enabled*, item will draw only as a solid shape with no interior details. For the purposes of animation, it can be beneficial to view certain elements as a silhouette to ensure that key poses are expressing certain information. Works in conjunction with the 'Enable Silhouette' setting of the Viewport display options which must be *enabled* (the default state) for the silhouette to be visible.

TIP: The 'Show as Sihouette' display option is only applicable to the 'Advanced OpenGL' viewport display mode.

Add/Remove Draw Options: The Draw Options open/remove additional properties for adjusting how items display in the 3D viewports, when selected additional properties appear.

Enable: The 'Enable' checkbox will toggle the effect of the Draw Options without losing values.

Draw Style: Allows user to choose a specific draw style for an individual mesh item, 'Default' respects viewport settings.

Wireframe: Disabling the 'Wireframe' checkbox will disable the wireframe overlay on 3D mesh items.

Smooth: Disabling the 'Smooth' checkbox will disable the smoothing applied to polygon surfaces, making polygons appear faceted.

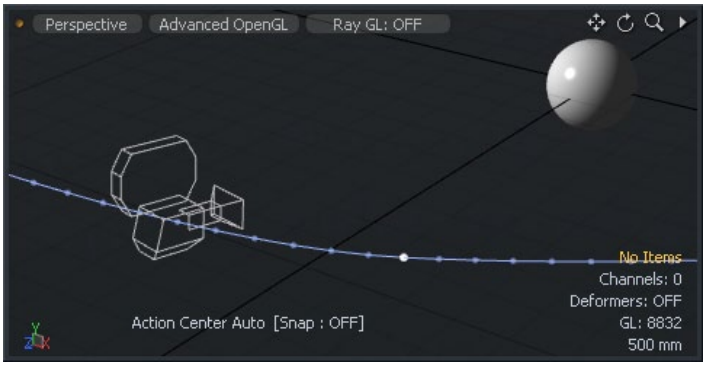
Texture: Controls the display of texture layers on a Mesh Item. 'Off' disables texture display, 'Texture' draws the texture without shading; this can make it easier to see a texture for painting, and 'Shaded Texture' takes into account the shading of the surface to which the texture is applied.

Wireframe: Adjusts draw color of wireframe overlay of the 3D item in the 3D viewport.

Wireframe Color: When 'Wireframe' is set to 'Custom', the 'Wireframe Color' sets the color of the wireframe.

Fill: If 'Draw Style' is set to 'Solid', this value controls the color of the 3D items fill.

Fill Color: When 'Fill Color' is set to 'Custom', this value will specify the RGB color of the fill of the 'Solid' in the 3D viewport.



Add/Remove Motion Path: To display the motion path for specific items within a scene (shown above for the 'Camera' item for illustrative purposes), users can select the 'Add Motion Path' option revealing additional properties controlling the display of the path in the 3D GL viewport.

Enable: Provides users a means to toggle the display of the path 'On' or 'Off', or to have the path only visible when the item is 'Selected'.

Time In/Out: 'Time In' determines the number of frames before the current time in the timeline that are displayed by the curve and 'Time Out' determines the number of frames after the current time that are displayed.

Show Frames: When *enabled*, displays the position of the element at each frame as a small dot along the path. Can be useful to see how the element moves in time, but can also get busy for some types of motion and can therefore be disabled.

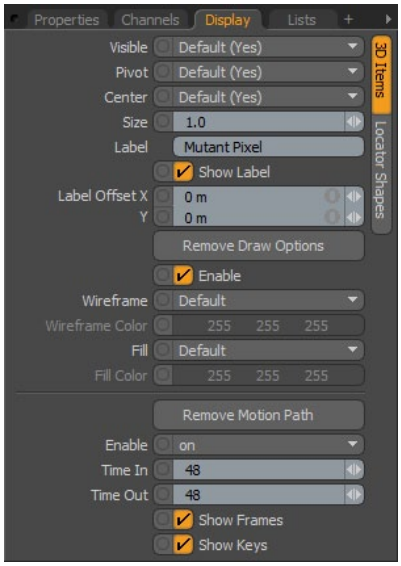
Show Keys: When *enabled*, displays the position of the element at each keyframe as a white dot along the path.

Locator Display Values

Locators are instrumental in setting up controllers and modifiers for animation, therefore controller have their own specific set of draw options increasing their utility even further.

Visible: Sets visibility for the 3D item itself, 'Default' respects viewport settings, 'Yes' set the item as visible always, 'No' disables visibility entirely.

Pivot: Sets visibility for the 3D item pivot, 'Default' respects viewport settings, 'Yes' set the pivot as always visible, 'No' disables pivot visibility entirely.



Center: Sets visibility for the 3D item center, 'Default' respects viewport settings, 'Yes' set the center as always visible, 'No' disables center visibility entirely.

Size: Multiplier that adjust the display size of the locator.

Label: Typing in text for the Label will display the text positioned next to the locator in the viewport for easy identification. It will also add two additional properties to control the display of the label in the viewport -'Show Label' and 'Label Offset'.

Show Label: This toggle will temporarily enable/disable the display of the label in the viewport without losing settings.

Label Offset: This value specifies the distance away from the locator where the label will appear

Add/Remove Draw Options: The Draw Options control the look of the locator in 3D viewport's, used mostly when Custom Locator Shapes are enabled.

Enable: The 'Enable' checkbox will toggle the effect of the Draw Options without losing values.

Wireframe: Adjusts draw color of locator in the 3D viewport.

Wireframe Color: When 'Wireframe' is set to 'Custom', the 'Wireframe Color' sets the color of the wireframe part of the locator.

Fill: If Custom Draw shape is set to 'Solid', this value controls the color of the fill itself.

Fill Color: When 'Fill Color' is set to 'Custom', this value will specify the RGB color of the fill of the 'Solid' in the 3D viewport.

Add/Remove Motion Path: To display the motion path for specific items within a scene, users can select the 'Add Motion Path' option revealing additional properties controlling the display of the path in the 3D GL viewport.

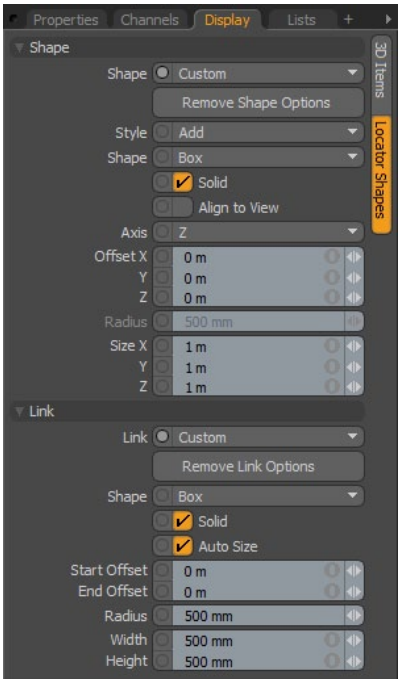
Enable: Provides users a means to toggle the display of the path 'On' or 'Off', or to have the path only visible when the item is 'Selected'.

Time In/Out: 'Time In' determines the number of frames before the current time in

the timeline that are displayed by the curve and 'Time Out' determines the number of frames after the current time that are displayed.

Show Frames: When *enabled,* displays the position of the element at each frame as a small dot along the path. Can be useful to see how the element moves in time, but can also get busy for some types of motion and can therefore be disabled.

Show Keys: When *enabled*, displays the position of the element at each keyframe as a white dot along the path.



Locator Shapes

Locator Shapes allow the user to change the default draw style of the locators in the 3D viewport, since locators have so many uses, it may be necessary to made each unique for easy identification and selection.

Shape--

Shape: Choose between the 'Default' cross style, or choose 'Custom' to enable additional properties for locator display customization.

Remove Shape Options: This button removes the additional Shape options reverting to the 'Default' cross representation.

Style: Controls whether the Shape options draw in Addition to the default value (add) of if they replace its display (Replace).

Shape: Choose from multiple primitive shape options for the locators representation.

Solid: Toggles between a filled or wireframe representation of the shape.

Align to View: Forces the representation to align with the viewport's camera.

Axis: Chooses the Axis for display of the shape representation.

Offset: Adjust the offset values to position the locator representation.

Radius: Adjust the size of circular locator representations.

Size: Adjust the size of rectangular locator representations.

Link--

modo can automatically draw a non-rendering element between an item and its child (Links always draw to child items in hierarchies). Links are a useful in creating visual representation of of item associations in complex hierarchies.

Link: Choose between the default value 'None', or a simple 'Line', or choose 'Custom' to enable additional properties for link display customization.

Remove Link Options: This button removes the additional Link options reverting to the default 'None'.

Shape: Choose from multiple primitive shape options for the link representation.

Solid: Toggles between a filled or wireframe representation of the shape.

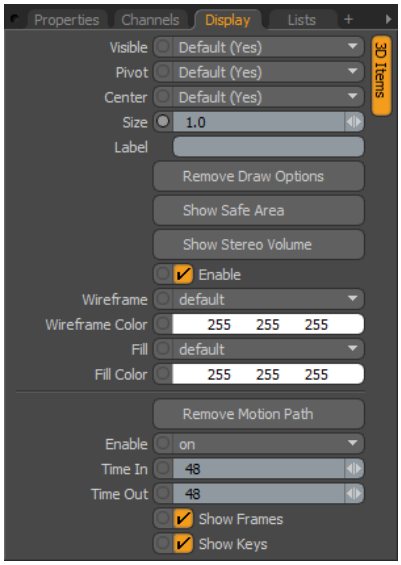
Auto Size: Dynamically scales the link shape depending on the distance between the two items.

Start Offset: Offsets the start position of the link representation.

End Offset: Offsets the end position of the link representation.

Radius: Adjust the size of circular link representations.

Width/Height: Adjust the size of rectangular link representations.



Camera Display Options

Visible: Sets visibility for the 3D item itself, 'Default' respects viewport settings, 'Yes' set the item as visible always, 'No' disables visibility entirely.

Pivot: Sets visibility for the 3D item pivot, 'Default' respects viewport settings, 'Yes' set the pivot as always visible, 'No' disables pivot visibility entirely.

Center: Sets visibility for the 3D item center, 'Default' respects viewport settings, 'Yes' set the center as always visible, 'No' disables center visibility entirely.

Size: Multiplier that adjust the display size of the locator.

Label: Typing in text for the Label will display the text positioned next to the locator in the viewport for easy identification. It will also add two additional properties to control the display of the label in the viewport -'Show Label' and 'Label Offset'.

Add/Remove Draw Options: The Draw Options control the look of the locator in 3D viewport's, used mostly to define a custom wireframe color for the Camera item.

Wireframe: Adjusts draw color of locator in the 3D viewport.

Wireframe Color: When 'Wireframe' is set to 'Custom', the 'Wireframe Color' sets the color of the wireframe part of the locator.

Fill: If Custom Draw shape is set to 'Solid', this value controls the color of the fill itself.

Fill Color: When 'Fill Color' is set to 'Custom', this value will specify the RGB color of the fill of the 'Solid' in the 3D viewport.

Show Safe Area: Users can *enable* this option to display safe area guidelines as an overlay to any 3D viewport with a view-type set as 'Camera'. Note that the target 'Camera' item must be visible under the viewport visibility options for the overlay to display. The options to control the placement of the guides can be found in the 'Camera' item 'Properties' only when the 'Show Safe Area' option is enabled.

Show Stereo Volume: This option enables the display of the Stereo Volume guides in the 3D viewport to give users a understanding of the amount of parallax in a given stereo scene. When *enabled*, a series of options to control the size and opacity of the guides can be found in the 'Camera' items 'Properties' panel only when the 'Show Safe Area' option is enabled.

Add/Remove Motion Path: To display the motion path for specific items within a scene, users can select the 'Add Motion Path' option revealing additional properties controlling the display of the path in the 3D GL viewport.

Enable: Provides users a means to toggle the display of the path 'On' or 'Off', or to have the path only visible when the item is 'Selected'.

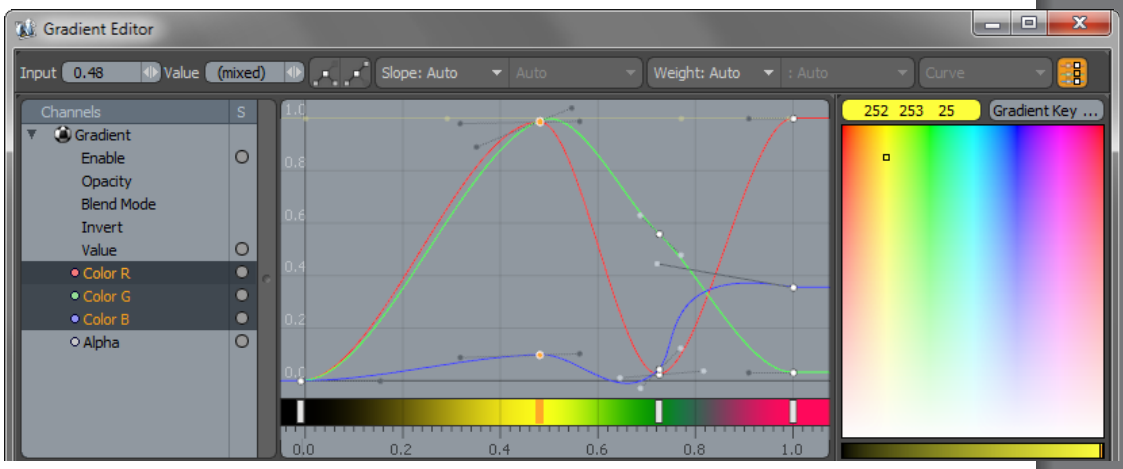
Time In/Out: 'Time In' determines the number of frames before the current time in the timeline that are displayed by the curve and 'Time Out' determines the number of frames after the current time that are displayed.

Show Frames: When *enabled*, displays the position of the element at each frame as a small dot along the path. Can be useful to see how the element moves in time, but can also get busy for some types of motion and can therefore be disabled.

Show Keys: When *enabled*, displays the position of the element at each keyframe as a white dot along the path.

Gradient Editor

The Gradient Editor lets you control Gradient textures in modo. Gradients are useful in modifying values gradually across a certain parameter such as length or angle. Once you have added a Gradient texture in the Shader Tree and indicated the effect it will have on the material, you will need to select the input parameter on the 'Gradient Texture' properties panel, this defines how the gradient modulates the surface is is affecting. By clicking the 'Edit Gradient' check box one can open the Gradient Editor window. The 'Gradient Editor' is in many ways similar to the graph editor, the difference being the graph editor modifies values over time.



Channels

From within the channel section, users can select the value they wish to modulate. For diffuse color this would be the 'Color R,G and B channels, for specular amount, this would be the 'Value' channel. Once selected, a line representing the value appears in the graphing window.

Key Creation

Keys can be added to the currently selected curve(s) by MMB clicking either on the curve itself or anywhere on the background.

Setting Colors for Keys

The color of selected keys can be set or adjusted by either clicking and dragging on a Color Picker viewport or from the RMB context menu. Selecting 'Key Color' from this menu will open the Color Picker from which a color may be chosen. In both cases additional keys may be created on associated curves at the key Input value.

Deleting Keys

Keys can be deleted from curves in two ways:

To delete only the currently selected keys press the 'Delete' key.

To delete the currently selected keys and keys at the same Input value on associated curves select 'Delete Key' from the RMB context menu.

Selection

LMB+click will select any key under the cursor, holding the 'Shift' key while selecting adds to the selection, and holding the 'Ctrl' key removes a key from the selection set. Clicking on the background deselects all keys.

LMB+click drag on the background starts box marquis selection, drag out a box to enclose the keys you want to select and release the mouse button. All keys within the box will become selected, any outside the box will be deselected. Holding down the 'Shift' key adds the keys within the box to the selection while the 'Ctrl' key removes them.

LMB+click on a curve in the background (*dimmed*) will select the curve. Curves can be added to the selection by holding down the 'Shift' key and clicking on a curve and can be removed from the selection by holding down the 'Ctrl' key.

Curves may also be selected from the menu at the top left of the viewport, this menu lists all the curves associated with the currently selected Gradient item(s).

Editing

LMB+click drag on a key selects the key if it's not already selected and allows quick editing of both Input & Output values at the same time. If there are several keys already selected and you click drag on one of them then all the selected keys will be modified.

MMB & Drag on a key works in a similar way except that editing is constrained to the Output value. Similarly RMB & Drag constrains the edits just to the Input Parameter value.

MMB+click drag vertically on the background adjusts the Output value of all selected keys.

RMB+click drag horizontally on the background adjusts the Input value of all selected keys.

'Ctrl'+MMB+click drag on the background provides constrained editing of Input &

Output values for selected keys based on the initial direction of movement.

LMB+double click on a key selects all the keys on the curve.

Context Menu

The Gradient Editor context menu is accessed by RMB click & release. If the cursor is over an un-selected key when the menu is opened the key will be added to the selection. Operations performed from the context menu will affect all the currently selected keys.

Navigation

Panning & Zooming behavior in the Gradient Editor is similar to the 3D viewports with a few additions:

Alt-MMB Drag adjusts the start value of the Input (horizontal) range displayed in the editor, while Alt-RMB Drag adjusts the end value of the Input Range.

Alt-Shift-MMB zooms the Output Range displayed in the editor, while Alt-Shift-RMB zooms the Input Range. In both cases the zoom is centered on the initial cursor location.

Alt-Control-RMB provides box zooming.

Mouse Wheel - Zooms in and out around the cursor location.

0 - Hover mouse pointer over color picker to open Color Options Dialog.

Graph Editor

Animating in 3D, in some ways can be very similar to the art of stop motion animation, where you build a model, place it in a scene, light the scene for atmosphere and then animate the model to bring it to life. However, unlike stop motion animation where the animator must move the model a tiny amount and then take a picture for every frame, in 3D animation, one can easily skip multiples of frames and lock in a fixed value, called a keyframe. Between those skipped frames the computer interpolates from one value to the other. The Graph Editor controls how the system interpolates those in-between values.

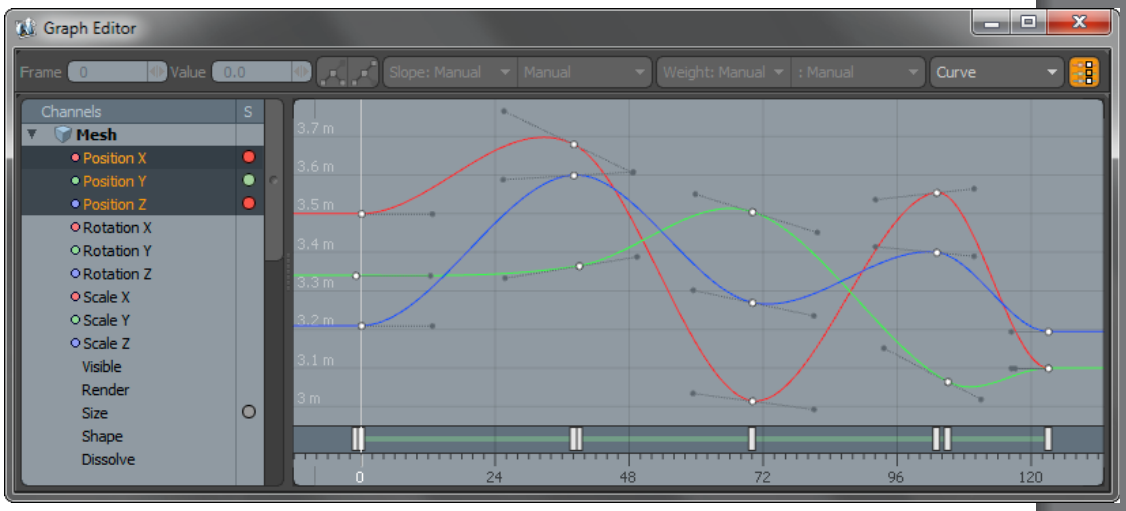
Not only for transforms, such as position, scale and rotate, nearly any property of any item can be animated in this way, by keyframing the values of an items channel. Channels are simply the animateable properties of any item, gathered together in list form. Generally, users interact with channels within the channel viewport, but since channels are so essential to the graph editor, the very same channels appear when

an item is selected in the left most column. Channels can be selected individually with a LMB click over the channel name. Multiple channels can be edited at the same time as well. Non-sequential channels can be selected using a Ctrl-LMB click on the various channel names, or sequentially with the shift key modifier and clicking the top and bottom most channels. When a particular channel is selected, its animation curve (as it is usually referred to, regardless of its actual shape) is displayed in the graph area. If no keyframes keyframes are defined, it will appear as a flat dotted line. Once a keyframe is created, it will appear solid.

The graph area is where the actual values are manipulated. In this graph, left to right always represents time, but up and down changes depending on the channel type selected. The line that connect the individual keyframes represents the values changing over time. By default, modo uses a smooth curve to interpolate between keyframes values, providing a nice ease in/ease out type of effect; but there will be times when one will want a very different interpolation style or simply finer control of it. Animation curves can only be changed by manipulating the keyframes and are very similar to working with bezier curves in the modeling viewport. Keyframes are represented as tiny dots along the line -white for selected channels and line colored for unselected channels. Keys frames with broken tangency will display in a magenta color. Existing keyframes can be edited by first selecting the key you wish to edit. Click over the dot to select, or LMB drag a marquee around one or more dots. Once selected, the dot will turn orange with a white outline. New keyframes can be created by MMB clicking. Values and Frame position can be changed by LMB-click dragging over any selected keyframe and moving it to the new frame/value position.

Navigating the graph area view itself is just like navigating any of modo's other viewports, Ctrl+Alt+LMB drag to scale the window, Alt+LMB drag to scroll. All keyframed values can be quickly brought in to view with the 'A' (view all) shortcut key or Shift+A (view selected).

A selected keyframes position in time, 'Frame' and 'Value' appear in the upper left hand corner where new values can be entered manually via the keyboard. You may also have noticed that the various buttons across the top become available when a keyframes is selected.



Break Slope: Breaks the curve handles at the tangency point allowing a sharp corner interpolation across the keyframe. When tangency points are broken, there is the opportunity to adjust both 'In' and Out' parameters for the keyframe.

Unify Slope: Unifies the curve handles at the tangency point forcing a smooth interpolation across the keyframe.

Slope: The slope controls lets the user specify the animation curves interpolation type on a per keyframe basis. If the tangency is broken, the left button controls the incoming value, and the right controls the outgoing. If the tangency is unified, only the left slope control will be enabled.

Manual- User can manually adjust curve using the handles. Clicking the dots at the end of the curve will edit their length and position.

Auto- modo automatically angles the curve handles to produce a smooth ease-in/ease-out interpolation.

Linear In- Points the incoming curve handle toward the previous keyframes value for a linear type effect.

Linear Out- Points the outgoing curve handle toward the next keyframe value for a linear type effect.

Flat- Flattens curve handles horizontally for the smoothest in/out interpolation.

Auto Flat- Adjust the curve handles based on timeline position.

Stepped- Adjust curve so value will jump at keyframe position.

Weight: Adjusts the length of the curve handles to control the curves length across

the keyframe, longer handles will 'ease' a keyframes value to a greater degree than a shorter handle. If the tangency is broken, the left button controls the incoming value, and the right controls the outgoing. If the tangency is unified, only the left weight control will be enabled.

Manual- User can manually drag out handles length to set the weight of the curve

Auto- Locks the handles length automatically adjusting it for smooth interpolation based on keyframes distance apart.

Curve: Adjusts the animation curve type for the entire channel. 'Curve' is the default behavior, 'Linear' interpolates as a straight line between keyframe values, and 'Stepped' will keep values constant until the next keyframe position in the timeline, jumping to the new value.

Link Keys: When the 'Link Keys' button is active (its background will change to orange), associated keyframes will remain linked together in time, meaning that, for properties like Position, Rotation and Scale that each have associated X,Y and Z values, editing a single axis value, by moving it forward or backwards in time, will keep all the keyframe positions linked together as if all 3 were selected. When inactive, only the selected channels will be modified.

Contextual Menu--

When RMB+clicking on a key value opens a context menu with a number of options for working with the keyframe data.

Key: Sets a keyframe at the 'current time' as specified by the timeline.

Manual/AutoLinear/Flat/AutoFlat/Stepped: These adjust the slope of the selected keyframe(s), same as when adjusted with the slope menu.

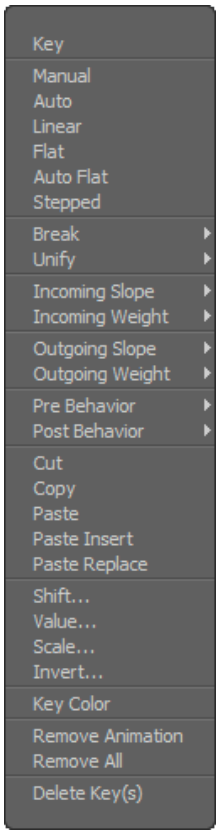
Break: Several options are offered to break the tangency of the keyframe's slope. Same as menu above.

Unify: Several options are offered to unify the tangency of a keyframe's 'broken' slope. Same as menu above

Incoming Slope: Adjusts the incoming slope of the keyframe. Same as menu above

Incoming Weight: Adjusts the incoming weight of the keyframe. Same as menu above

Outgoing Slope: When tangency for a key is broken, this adjusts the outgoing slope of the keyframe.



Outgoing Weight: When tangency for a key is broken, this adjusts the outgoing weight of the keyframe.

Pre Behavior: Sets the channels behavior prior to the first keyframe in a sequence.

Constant- The value of the first keyframe is maintained, with the outgoing slope smoothly interpolating toward the next keyframe value.

Stop- The Value of the first keyframe is maintained, with a linear outgoing slope producing an abrupt transition.

Linear- Incoming slope will linearly match the vector direction of the first keyframe.

Repeat- Repeats the entire sequence of keyframes.

Oscillate- Repeats the entire sequence of keyframes, alternating between backwards and forwards for a ping-pong like effect.

Offset Repeat- Repeats a sequence of keyframes offsetting each sequence by the value of the last key.

Reset- Sets the incoming area outside of the keyframe sequence to a value of '0'.

Post Behavior: Sets the channels behavior after any sequence of keyframes.

Constant- The value of the last keyframe is maintained, with the incoming slope smoothly interpolating toward the last keyframe value.

Stop- The Value of the last keyframe is maintained, with a linear incoming slope producing an abrupt transition.

Linear- Outgoing slope will linearly match the vector direction of the last keyframe.

Repeat- Repeats the entire sequence of keyframes.

Oscillate- Repeats the entire sequence of keyframes, alternating between backwards and forwards for a ping-pong like effect.

Offset Repeat- Repeats a sequence of keyframes offsetting each sequence by the value of the first key.

Reset- Sets the outgoing area outside of the keyframe sequence to a value of '0'.

Cut/Copy/Paste/Paste Insert/Paste Replace: Options for managing keyframes through a clipboard. Users can select a number of keys, 'Copy' them and then 'Paste' the values elsewhere in time. 'Cut' removes keys to clipboard, 'Copy' duplicates them

to the clipboard. 'Paste' merges the keys into the destination channel, 'Paste Insert' inserts the keys, moving existing keys to make room, and 'Paste Replace' overwrites the keyframes across the range of the copied keys.

Shift...: This command offers the user a pop up window with options for shifting or moving the time of a selection of keyframes.

Value...: This command offers the user a pop up window with options for setting a specific value on a sequence of keyframes.

Scale...: This command offers the user a pop up window with options for scaling the value of a selection of keyframes.

Invert...: This command offers the user a pop up window with options for inverting the time position of a sequence of keyframes.

Key Color: Opens the systems color palette allowing the user to specify a color value for a keyframe position.

Remove Animation: Removes all keyframes, keeping the incoming value of the first key on each channel.

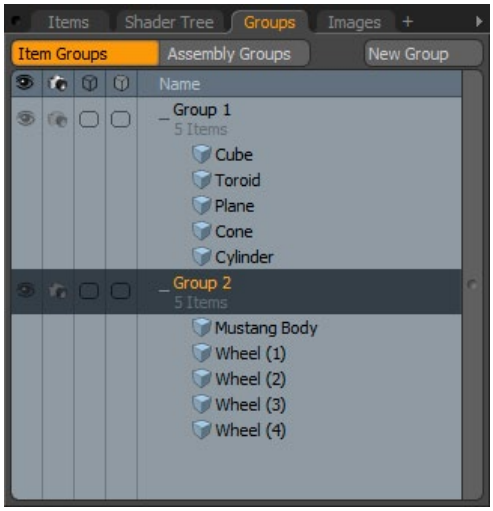
Remove All: Removes all keyframes resetting all channels values back to default '0'.

Delete Key(s): Deletes the currently selected keyframe(s).

The timeline across the bottom is functionally identical to the timeline in the Animation layout. LMB clicking and dragging will change the current time. As you drag the current time value around, you can see the effect of the animation curve in the 3D viewport. Handles appear in the timeline wherever a keyframe appears, and can be LMB-dragged in time forwards or backwards modifying the position of the keyframes for selected channels.

Groups Viewport

The Groups viewport is a great workflow streamlining tool allowing users the ability to easily control multiple items and/or channels within a scene, and also provides a number of functionalities to speed up common tasks, such as setting keyframes for complex hierarchies. Groups also add functionality to 'Replicators' and the 'Light Linking' feature found in modo. Not to be confused with 'Group' Items in the Item List which are a purely organizational construct and Shader Tree 'Groups' which define masks for surface shading as well as to organize the many Shader Tree layers.



To work with groups, one will first need to create a new group using the 'New Group' button on the top of the viewport. When a new group is created it will use the default name 'Group' which can be easily changed by clicking to highlight the name and then pause clicking to edit. Groups can be made up from any number of scene items, including mesh items, cameras, lights, and even locators. Channels can also be part of a group to speed up animation workflows. When in 'Item' mode, items can be added to a group by drag and drop, simply select the item in the 3D viewport you wish to add and click-drag it to the group window and release the mouse button over the group you wish to add the items to. On orange highlight appears in the list to denote where the item will land when dropped. Channels can be added in the same fashion from the 'Channels' viewport.

The viewport itself has a number of controls. The columns on the left allow the user to make changes that affect all items in the Group. The first column affects visibility for items within the group essentially the same way the eye column toggles visibility in the 'Item List'. The second column toggles render visibility; whether or not items appear in rendered output. Note that both these settings can be overridden by the same settings on the individual item's properties forms. For example, if a user has a group containing 20 items, of which 5 were set to 'Yes' for visibility in individual items properties, then toggling the groups visibility would only toggle 15, as the other 5 items settings would override the group settings. Setting the 5 individual items back to 'Default' would return the items back to following the group settings.

The third column when clicked will present a color picker that one can use to change the wireframe color for all items within the group and the fourth column will change the solid display fill color for all items in the group. Changing the display colors of items can be an easy way to visually tell which items are members of which groups in scenes with many items. Individual item settings can be adjusted within the 'Display' viewport when items are selected.

Preceding the group names and item and channel lists are little +/- toggles that unfold/fold list visibility. When the lists are unfolded, individual items can be selected and their respective properties or channel settings can be viewed in the appropriate viewport.

Channels in Groups--

Groups can contain any number of individual channels from any combination of items in the scene. These Individual channels can also be members as many different groups the user wishes. If the visibility of a group containing channels is turned off, then those channels will be hidden in the various channel lists, i.e. the main Channel List viewport, Graph Editor and Channel Links viewport, though Properties are not affected by this setting. If a group containing channels is locked, then the channels cannot be edited anywhere, including the Channel list, Properties viewport or interactively in a 3D viewport with the transform tools.

Group Parenting--

Groups can be parented together, similar to 3D items, just drag a child group and drop onto a parent item, working the same way parenting is performed in the Item List. Instead of inheriting transforms from their parents though, they inherit the group settings from their parent groups. Groups do not need to have any member items or channels either. They can be used as organizational tools. An example would be multiple groups each containing many items. If parented to an empty group and the visibility is switched off on that group, then all the child groups and hence their member items will be hidden in the 3D views.

Group visibility and render channels can also override in the same way as their counterparts on mesh items. So in the above example setting one of the child group's visibility channel to 'Yes' will override the parent setting making all members of the group visible. Conversely if the parent group visibility is set to show member items, then setting a child's visibility channel to 'No' will override that and hide it's members. All other children of the parent group will still get their settings from the parent group. This also applies to the groups render, selection and locking settings available by the RMB context menu.

Group Drawing Settings--

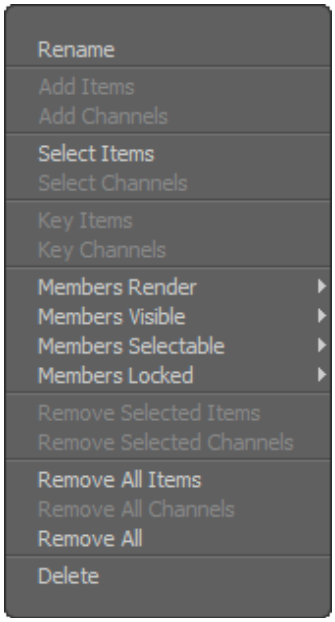
The 'Wireframe' and 'Fill' columns of the groups viewport allow easy access to basic display settings, however groups also support the full custom drawing settings available for 3D items. If you look in the 'Display' viewport with a group selected you'll see an "Add Group Draw Options" button which when clicked will display the custom drawing channels used for 3D items (if you've already set a wire or fill color for the group the custom drawing channels will already be available). The difference with setting these channels on a group as opposed to individual items is that the group settings will apply themselves to all items within the group.

Group custom drawing is layered from parent groups to child groups, so settings on parent groups will override any settings on child groups. In order for settings on child

groups to be visible custom drawing on any parent groups needs to be disabled in the Display properties viewport by switching off the 'Enable' channel. It is not currently possible for child group custom drawing to override parent settings in the same way as the visibility settings for example.

For all settings the order in which the group appears in the groups viewport is also taken into account. Groups are searched from the top down, when an item is found in a group hierarchy modo looks at the settings on the group containing the item, if that setting is 'Default' it looks up the group hierarchy for a non-default setting. If one is found then that setting is used, otherwise it continues to look through the groups in the viewport from the top down.

Since multiple groups can contain the same items this can be used to very quickly change settings, for example, if two groups contain the same items, one is set to lock member items and make them unselectable and has it's custom drawing set to a dark color to indicate they can't be changed. The other group has the inverse settings applied. Just by dragging either group above the other in the viewport will change the settings applied to the items in the groups, e.g. if the locked group is topmost then all the items will be drawn dark and will be uneditable and not selectable, otherwise the items will be drawn in the other color and can be edited and selected.



Contextual Menu--

RMB clicking on a group will present the user with a number of options as well.

Rename: Will present the user with a popup dialog box that allows the user to rename the highlighted group.

Add Items/Channels: Will add all the currently selected items from the item list or 3D viewport, or channels from the channels viewport.

Select Items/Channels: Will item mode select all the associated items or channels within the group.

Key Items/Channels: Will add a keyframe at the current timeline position for all associated items or channels within the group.

Members Render: Sets if group members are visible in rendered images: "Default" respects item list visibility settings (or parent group settings if applicable), 'Yes' setting enables member items visibility in rendered images, while 'No' disables member items render visibility. When used in conjunction with

group parenting, child group settings will override parent settings.

Members Visible: Sets if group members are visible in 3D viewports: "Default" respects item list visibility settings (or parent group settings if applicable), 'Yes' setting enables member items visibility in 3D viewports, while 'No' disables member items visibility in 3D viewports. When used in conjunction with group parenting, child group settings will override parent settings.

Members Selectable: Sets if group members are selectable in 3D viewports: "Default" respects item list settings (or parent group settings if applicable), 'Yes' setting enables member items as selectable in 3D viewports, while 'No' disables selecting of member items in 3D viewports. When used in conjunction with group parenting, child group settings will override parent settings.

Members Locked: Sets if group members are locked from property changes: "Default" respects item list settings (or parent group settings if applicable), 'Yes' setting enables member property modifications, while 'No' disables member items property modifications. When used in conjunction with group parenting, child group settings will override parent settings.

Remove Selected Items: Users can select an item in the 3D viewport or ctrl-select multiple items and use the 'Remove Selected Items' command to remove selected items from a group.

Remove Selected Channels: Users can select a channel in the channel viewport or ctrl-select multiple channels and use the 'Remove Selected Channels' command to remove the selected channels from a group.

Remove All Items: Will remove all items from within the group.

Remove All Channels: Will remove all channels from the group.

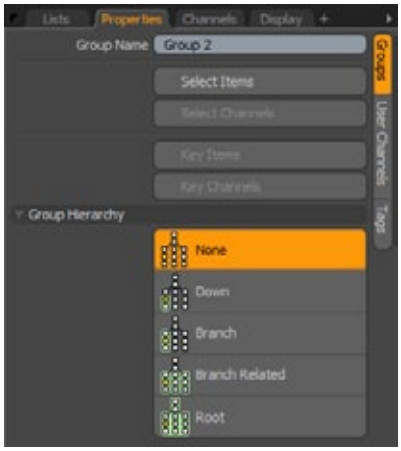
Remove All: Will remove everything, all items and channels from the group.

Delete: Will delete the group, removing it and its contents from the group viewport.

Keyframing and Selecting Group Members--

Groups can also be used to help with keyframing workflow as well as with item selections. On the Group Properties viewport there are buttons to key items and channels within the selected group (also available via the right click context menu). When keying group items only channels belonging to the items that are already keyed will be affected. Keying channels within a group will add a key at the current time to each member channel, regardless of whether they have been keyed already or not. This becomes more powerful with parented groups and the use of the group Hierarchy

buttons also displayed on the Group Properties form. By default only the contents of the currently selected groups will be keyed (or selected), but this behavior can be changed with the Hierarchy type buttons:



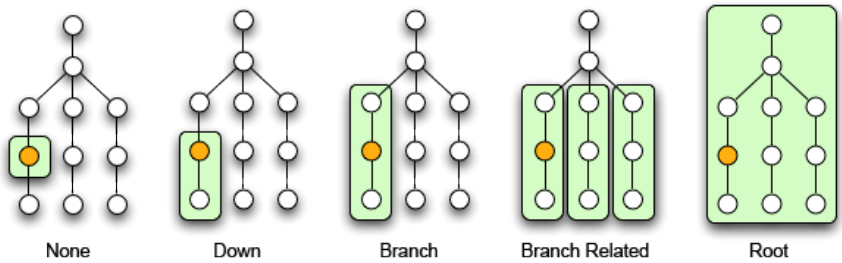
None: The default, only keys the contents of the current group(s).

Down: Keys the contents of the current groups and all child groups.

Branch: Keys the contents of the hierarchy branch that the current group resides in

Branch Related: As above but will also key the branches related to the current one, i.e. branches that share a common parent. (An example might be fingers on a hand.)

Root: Keys the contents of the entire hierarchy that the group belongs to.



Light Linking--

When a Group contains one or more light items (directional, spot, point or any of the other light types), they can also be used to control the lights overall affect on shading on a surface by surface basis using a Shader in the shader tree with the Llight Linking function. Please reference the Shader Item properties for further information.

Replicators--

Multiple items can act as base Prototypes for Replicators when set into a Group, allowing for greater variety in replicated objects. To use multiple mesh items as prototypes for a single replicator, first create a group that contains all the different mesh layers, once a Replicator has been created, the Group name will appear in the replicators 'Prototype' drop down list. When specified, the replicator will randomly grab

objects from the group for each 'replica'. Please reference the Replicators page of the documentation for further information.

Preset Browser

The 'Preset Browser' is a library of pre-defined objects, surfaces and settings organized and categorized for quick visual browsing, greatly speeding up scene creation. Presets are applied to the scene using simple drag and drop operations in the case of materials and objects. For profile presets, both 2D and 3D, one simply needs to select the profile within the browser before activating the modeling tool. Users can also easily create their own library of commonly used objects and settings for streamlined application to their own scenes.

Navigating the Browser--

There are a number of functions for navigating through the preset library itself. First off you want to make sure that you have the content folder installed and that the Preset Browser is pointed to it on your hard drive. You can add the path to the browser using the '(add path)' function found in the directory tree or the small directory bar across the top. The default assets are typically located in the "[drive]\[modo folder]\content\Assets\" folder where you installed the content folder included with modo.

Depending on your amount of screen real estate there are several methods for browsing the folders associated with presets, one is the Directories tree, the left most column in the above image. If it's not there, you can make it visible by LMB-click drag to the right of the inside edge of the left side of the viewport, there is a tiny row of squares there, you'll know you're on the right spot when the cursor changes to two opposing arrows. In the case of the material browser, there are a number of folders you can see now that organize associated presets. The small triangles to the left of the folder names will expand/collapse any sub folder and clicking the folders themselves will display their contents in the main browsing window. The far right

column with the 'X' icon over it will display an 'X' next to added preset directories, clicking it will allow the user to remove these directories from the tree. If you prefer not to use the Directories tree, you can collapse it by LMB-click on the right hand edge of it and dragging to the left until it disappears. One can also navigate through the various folders using the iconic buttons in the upper left of the window -they are forward, backward, parent and New directory respectively. RMB-click on either the forward or backward buttons to get a context menu of available folder names. If the Dir Browser is hidden and the viewport is set to not display thumbnails, a popup containing the path hierarchy is displayed between the navigation buttons and the zoom slider. Changing thumbnail modes is done via Viewport Settings, which can always, never or conditionally (depending on if the Dir Browser is shown) show folders as thumbnails.

The zoom slider across the top of the viewport allows for smoothly scaling the size of the icons in the browser. When the number of items in the window exceeds the amount of space available, the scroll bar down the right side can be used to scroll through numerous selections.

Applying Presets--

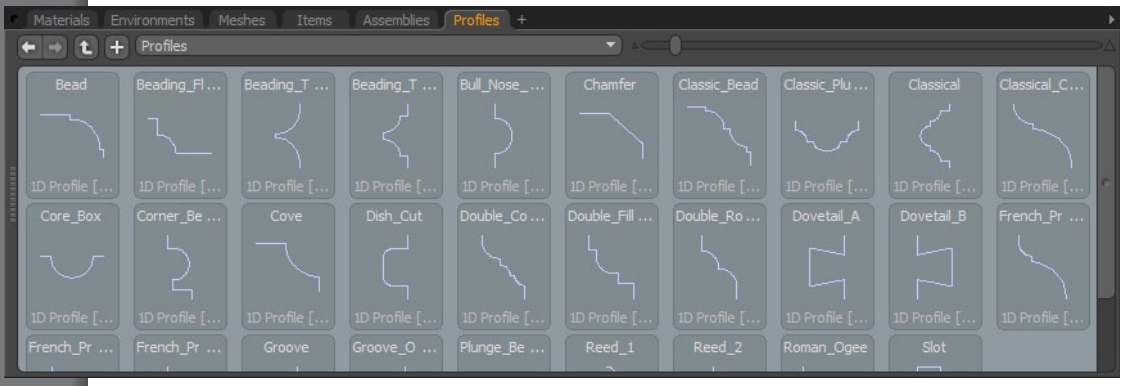
Materials: Applying material presets couldn't be easier. You can select a number of polygons and either LMB-click drag onto the selection or just double click the material preset itself and the polygons will be tagged with the appropriate surface name and applied in the shader tree. If you've already tagged all your surfaces using the 'm' keyboard shortcut, that's even better, you can just drag and drop any of the material presets onto the surface and the appropriate material will be updated in the shader tree. For presets with associated image maps that require UVs, if no UVs are present, the material will default to cubic projection. Presets can also be dragged and dropped directly onto group items in the Shader Tree itself.

Environments: Environment presets can be added to a scene by simply double clicking on the preset of your choice. This automatically adds it to the environment item of the

shader tree. If you don't have a preview window visible, you may need to render the scene to see the effect of an environment preset.

Meshes: Mesh presets can be applied to a scene just as easily as any other type of preset. If a preset is dragged onto an empty part of the scene, it will automatically position itself appropriately in relation to the workplane becoming its own mesh item layer. If the preset is dragged over geometry (for example a light fixture onto a wall) the item will align itself to the polygon it was dropped onto and becomes part of the same layer. If one simply double clicks the preset, the item will be created at the origin (0,0,0 position). Once inserted, a mesh preset is just like any other geometry in a scene and becomes fully editable.

Profiles: Profile presets work with many modeling tools and can be simply applied by first selecting the profile you wish to use and then invoking the appropriate tool, say for instance the bevel tool. With the tool in interactive mode, you can see the effects of the profile added to the bevel as you drag it out. Selecting any other profile will update the bevel profile instantly allowing users to try out different settings before dropping the tool.



Render Settings: modo provides users the ability to save the 'Render Item' settings, such as Frame Size and Resolution, Antialiasing, and the Global Illumination settings as a 'Render Preset', making it easy for users to define commonly used default values and then apply them quickly to new scenes, without the need to change multiples of settings manually. Camera Items may also be included in these types of presets. To apply a "Render Preset' users need only to LMB+double click over the target preset in the viewport to apply it to the currently active scene. Once applied, the Render Items updates reflecting the new values. An additional 'Camera' item defined by the preset may be added as well.

Creating your own Presets--

There are probably a number of items you create for every scene, a ground plane, a certain environment gradient, or maybe a specific material. Creating your own presets is a great way to streamline a workflow and speed up scene creation eliminating repetitive task so you can concentrate on the important stuff.

Materials: Any layer in the shader tree can act as a preset. If you RMB-click on a layer, and select 'Save Preset...'' from the drop down menu where you will be presented with a file save dialog asking where you'd like to save your preset. If you want the preset to show up automatically in the existing set, you'll need to navigate to the directory where your presets are stored and save within that directory, one can also create a new directory for saving presets, but make sure to use the '(add path)' command so you can see them within the browser viewport. I'm sure quite often you will have a surface made up of multiple texture layers, in that case you will want to make sure to create the preset from the material group ( ) , as that will include all associated texture layers within the preset. Don't forget that Fur is made up from texture layers so making hair presets is just as easy as making surface presets.

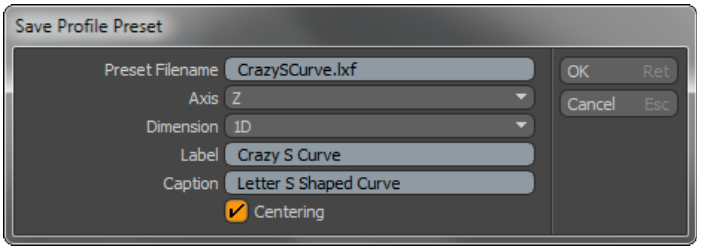
Environments: Environment Materials can also be saved as presets by RMB-clicking on the environment material layer and selecting 'Save Preset...' Just as with materials, you will be presented with a file save dialog asking where you want to save your preset. If you want the preset to show up automatically in the existing set, you'll need to navigate to the directory where your presets are stored and save within that directory, one can also create a new directory for saving presets, but make sure to use the '(add path)' command so you can see them within the browser viewport.

Meshes: Mesh presets require a small amount of setup for them to work properly. Mesh presets align themselves based on their relationship to modo's origin when they are created, therefore it is recommended that one *create* mesh presets at modo's origin point. You'll also want them resting on the ground, so they are facing the Y up (positive) direction. That way when you create say a light fixture that you are applying to a series of walls, each one will be properly aligned to the wall in the proper orientation when placed. Note that item transforms are not considered in presets, so make sure you make your alignment changes at the component level, otherwise use the Freeze command in the items properties tab. Once your model is properly positioned, one only needs to RMB-click on the mesh item layer in the item list viewport and select 'Save Preset...' where you will be presented with a file save dialog asking where you'd like to save your preset. If you want the preset to show up automatically in the existing set, you'll need to navigate to the directory where your presets are stored and save within that directory, one can also create a new directory for saving presets, but make sure to use the '(add path)' command so you can see them within the browser viewport.

Render Presets: Users can easily save the settings of the Render Items, which contains core rendering attributes such as 'Frame', AA and ray trace 'Settings' and 'Global Illumination' settings, by simply LMB+clicking on 'Render Item', the top-most item of the 'Shader Tree', to select, then RMB+Clicking on the item to open the context menu and selecting the 'Save Preset' option. Additionally, users may select the 'Render Item' and invoke the menu bar command "File > Save Item Preset". Either way opens an OS specific 'File Save' dialog. If you want the preset to show up automatically in the existing set, you'll need to navigate to the directory where your presets are stored and save within that directory structure; one can also create a new directory for saving presets, but make sure to use the '(add path)' command so you can see them within the browser viewport.

If a 'Camera' item is also selected when the preset is saved, then that camera and its associated settings will become part of preset and when applied will be added to the scene in addition to any existing cameras. Useful to set-up automatic DOF and motion-blur cameras.

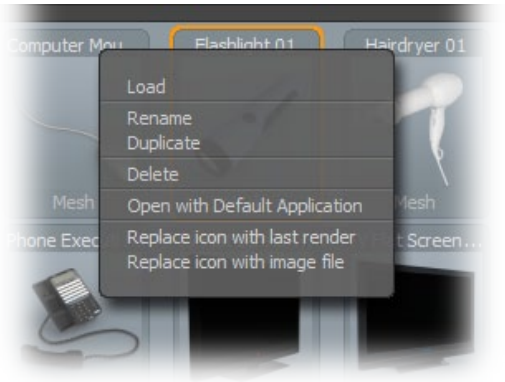
Profiles: Profiles are based off of a curve drawn in modo. You can create them in a number of ways, including the curve tool, bezier curve tool or the sketch tool. Once you have your profile curve defined, select it and in the menu bar select the command 'Geometry > Profile > Save...) where you will be presented with a file save dialog asking where you'd like to save your preset. If you want the preset to show up automatically in the existing set, you'll need to navigate to the directory where your presets are stored and save within that directory, one can also create a new directory for saving presets, but make sure to use the '(add path)' command so you can see them within the browser viewport. Once you click 'Save' you will be presented with this dialog box...



The 'Preset Filename' will contain the name you specified in the previous file save dialog, so it shouldn't be necessary to edit it here, 'Axis' specifies the plane which the profile was created on, either X, Y or Z. 'Dimension' specifies to modo whether the drawn profile is '1D' or '2D'. 'Label' and 'Caption' are pretty self explanatory, and will appear above and below the profile respectively in the preset library window. The 'Centering' toggle will automatically center the curve when enabled (if it was created away from the origin). Click OK to finalize saving of the profile.

Customizing the view--

By RMB-clicking on a preset icon, users are presented several options for managing presets as well as customizing the icon display.



Load: Will load the currently selected preset into the scene, for mesh presets, this will create a new item centered at modo's origin. For materials, you need to select a material group in the shader tree before the material preset will load in this way.

Rename: Allows the user to apply a new label to the preset, replacing any previous labels.

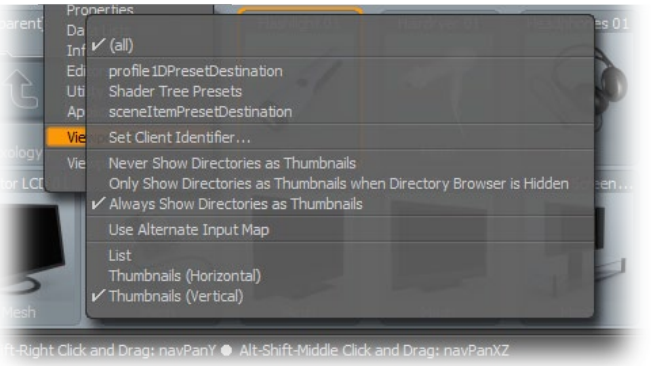
Duplicate: This command will pop up a dialog box requesting a new filename for a new duplicate preset file.

Delete: Will remove the currently selected preset. Note that this also removes the file from the hard drive.

Replace icon with last render: Will embed last 'F9' render into currently selected preset item.

Replace icon with image file: Will embed a saved image file into the currently selected preset item.

By RMB-clicking on the preset browsers tab and selecting viewport settings, the user is offered some options to customize the way presets display within the window.



The 'Set Client Identifier' option allows a Preset Browser to have its own unique selection. This is most useful when configuring a preset browser to be used with a specific tool; for example, this is how the Polygon Bevel tool knows which of many possible Preset Browser viewports to use (in the case of Polygon Bevel, the browser viewport is embedded right in the Tool Properties viewport). Note that each tool must be specially coded to look for a specific client identifier, and thus this feature is only really useful to tool developers (which, at the moment, is only us).

The next three options control how directories themselves (folders) display within the viewport and are self explanatory.

"Use Alternate Input Map" tells the browser to use a different input map from its default. In a standard configuration, the default input map applies presets on a double-click of a thumbnail, while the alternate input map applies a preset on a single click, mimicking the behavior of the 360 application.

Finally, the last three options deal with how the presets are displayed, List is a strict listing of the object names, and thumbnails can be displayed cascading horizontally or vertically.

Adding Your Own Preset Tab--

If you wish to further customize the tabbed Presets Browser to your personal workflow, such as creating a repository for project specific presets, adding your own preset tab to the workspace is a relatively easy task. First, navigate to the preset browser itself, to the right of the 'Profiles' tab in the default configuration, there is a little 'plus' sign (+) for adding additional tabs, LMB-click on it. Within the pop-up, select 'Data Lists > Presets Browser' and a new tab will be added to the right side of the existing tabs. Now, RMB-click on the tab where it says 'Preset Browser' and in that pop-up, select 'Viewport Controls > Rename Tab' and type the name you'd like to use in the window.

From there you can display a custom folder of presets by navigating with the standard controls to your preferred directory.

Working with Replicators--

Mesh presets are fully compatible with Replicators, though are not automatically applied as such. To use a preset mesh as a replicator, double click the mesh object in the Preset viewport so it will be created as its own mesh item at the origin. Then, in 'item mode', select the Point Source and the newly created preset item mesh and create the replicator in the usual way --menu bar "Item > Add Replicator" Please reference the Replicator item for more information on working with replicators.

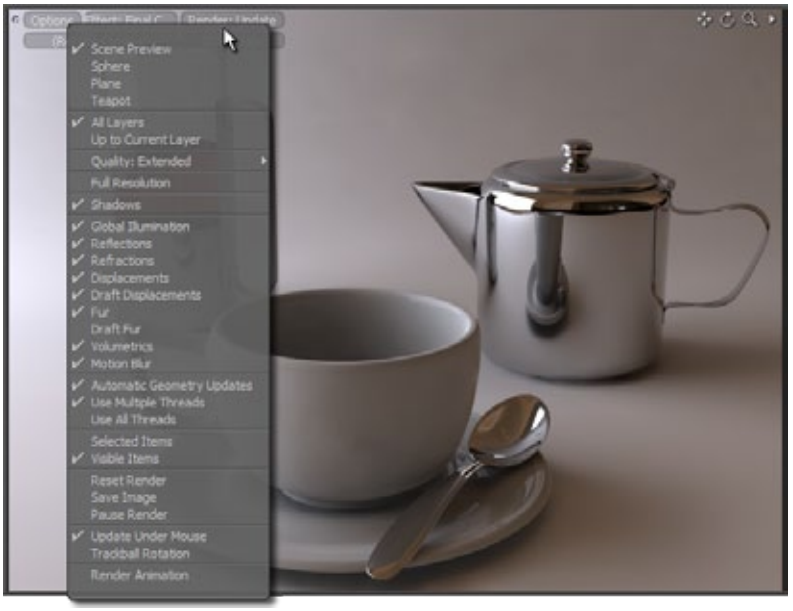
Preview Viewport

The Preview viewport uses the modo render engine in a progressive refinement mode that allows users to see all changes in a scene with a fully rendered image. This viewport is thread aware so that when it is not the current viewport it will not interrupt interactivity from other views. This allows the user to work on their project without performance degradation yet anytime there are "free cycles" the Preview viewport will begin a progressive render starting at low quality and filling it in to higher quality as time permits. In the Preview viewport users can view changes to the geometry, materials, textures, cameras, lights and even render settings such as ray tracing and global illumination.

In traditional systems the OpenGL viewport gives a proxy view of the project in real-time and a separate offline render engine (if available) is summoned periodically to generate a full quality image. This offline rendering is a modal effect which requires the artist to completely turn over control of the system to the render engine until the image is complete. This "start and stop" process is a complete workflow killer. Even waiting

a several seconds to review a change takes the artist out of their creative flow. The Preview window enables the artist to simply pause to glance at the progress of their project and then continue the process of refining their vision resulting in a much more fluid and art conducive process.

The Preview viewport has three preview "modes" and several options for how much of the project to render interactively and which channels to view. The Types are 'Auto', 'Shaders' and 'Clips'. The Shaders mode tells the Preview to perform a full project evaluation or render whereas the Clips mode only allows the Preview to display images or texture maps that are loaded and selected inside modo's Image List. 'Auto' allows the Preview to display the most recently selected Clip or render item.



The Preview options provide extensive opportunities to optimize the performance of the viewport by limiting or focusing the rendered data. The first option allows users to choose between a full scene render or a sample primitive including Sphere, Plane and Teapot. These will give lightning fast results and are also very useful for creating material/texture presets before you have any geometry in a project.

The next option, 'All Layers' or 'Up to Current Layer' allows you to determine how much of the Shader Tree the Preview will display. In the default mode all layers are rendered giving the most accurate view of the final image. If you want to focus on a specific texture or material layer for editing you should use the 'Up to Current Layer' setting. In this mode the Preview will only display the currently selected texture or material layer and any that are beneath it in the Shader Tree. This can yield both improved

performance as well as improving the focus when editing a particular layer.

The next option allows users to choose the Quality of the preview image, providing options for 'Draft', which throttles some of the most render intensive settings to provide a more interactive view, 'Final Render Quality' which uses the settings of the Shader Tree to determine quality and finally 'Extended Refinement Passes' allows the Preview render to continue to add refinement to the image even when the maximum number of visible pixels has been achieved. In this case the Preview will begin to add antialiasing effects to the image. It should be noted that the direct rendering in modo (Render command) will generally yield better results in a shorter overall render time.

The 'Full Resolution' option allows users to render a preview at the same resolution as specified in the Render Items 'Frame' options. When this option is enabled, the 'Alt+mouse combination' shortcuts will navigate the image itself, rather than the viewport. However, the viewport navigation widgets will still operate as normal.

There are several visibility toggles for various Effects. They include 'Shadows', 'Global Illumination', 'Reflections', 'Refractions', 'Displacements', 'Fur' (with an option for "Draft" or low res on fur and displacements), 'Volumetric' and 'Motion Blur'. By default these are all set on. Disabling specific effects will typically improve performance. The Draft options will force modo to use a smaller amount of micro-polygon tessellation in the Preview viewport. Micro-polygon displacement can generate millions of polygons at the time of render and as such has a significant setup time requirement making it difficult to Preview this effect interactively. Draft options will reduce the total amount of polys generated improving performance. Also, once the mesh has been tessellated the Preview can display lighting, camera and shading changes very rapidly. Only mesh edits will force the tessellation to occur again.

'Automatic Geometry Updates' indicates to the Preview to constantly keep the mesh data in sync with the current state of the mesh. Deactivating this setting will "lock" the mesh in its' current state and only allow material, lighting and camera edits.

'Use Multiple Threads' will allow Preview to render with multiple cores or processors. However, since the idea behind Preview is to allow it to cooperate with the modeling, lighting and painting processes it is recommended to leave this option off so that the other cores or processors are available for interactive work. 'Use All Threads' will maximize the thread use of the viewport providing the best overall performance for the preview window.

The 'Selected' or 'Visible' option allows you to chose which items will appear in the Preview render. The default Visible setting will render all item in the project that have their visibility setting on. To remove an item from Preview in this mode you simply click the eyeball icon in the Item List. When this setting is on, Selected Items the Preview

will only render the currently selected items. This is a very convenient way to isolate certain item(s) for Preview.

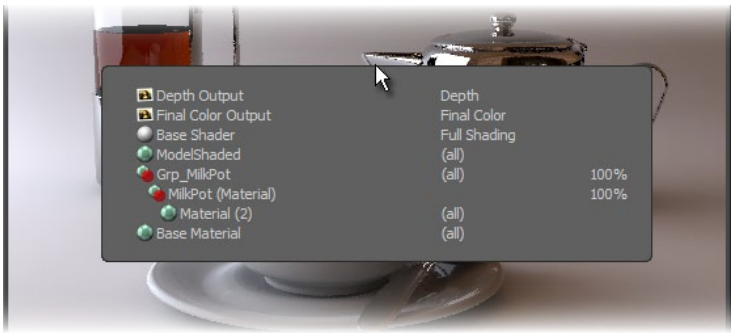
'Reset View' will simply reset the render state including refreshing the mesh data. 'Save View' will save the current state of Preview to an image file. 'Pause View' stops rendering and leaves the view in the state at which the view was paused.

'Update Under Mouse' provides users a means to control, to a degree, where modo should spend its efforts when updating a fresh preview. Enabling this feature will allow users to simply hover the mouse pointer over the preview viewport in the area of greatest interest (wiggling the mouse pointer over the target area will help). modo will then concentrate additional effort in this area to update faster. When disabled, the available CPU cycles will be evenly distributed across the entire image.

The 'Render Animation' option uses the Preview renderer to render animated sequences, and it works the same as using the "Render > Render Animation" command with two exceptions: 1) You can only render the currently active pass (as opposed to specifying a 'Pass Group'), 2) You specify how long you want the whole render to take. It will evenly divide this time up between each frame, and for each frame will progressively refine until time is up (or the frame finishes, which can definitely happen if you're in Final or Draft mode).

Below the Preview options button is a Camera pop-up selector allowing the user to easily select the current camera for rendering, without the need to specify it specifically in the Render Item. Directly next to the options is an 'Effect Button' that allows the user to define a specific rendering channel for previewing, allowing for a number of diagnostic possibilities, such as previewing the amount of Fresnel reflectivity on a surface. Additionally, users can toggle the visibility of a selected layers contribution to a scene, visualized as an semi-transparent overlay by enabling the "Current Layer Overlay" option. Once enabled it displays the current texture as an overlay on top of the render, visualizing the raw value unaffected by things like 'Blending Mode', 'Opacity' or shading. So if you have an image map on the Subsurface Color channel, it will display the raw color of the map. For 'Groups' it display the group mask opacity. This is very handy to debug texturing and see what layer's doing what.

Visually Selecting Shader Tree Layers--



To make it easier to find a particular surface, users may RMB-click over any area of the preview image itself, opening a popover menu displaying all shader tree layers that contributed to the pixel directly under the mouse pointer when the RMB was pressed. Clicking on any of the layers in the dialog will select them in the shader tree.

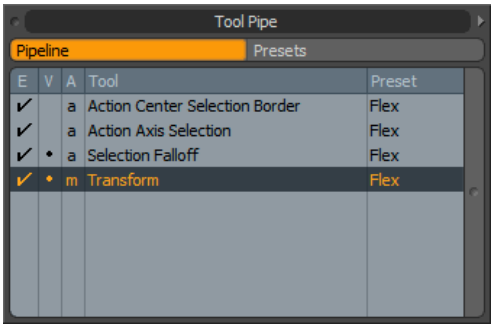
Navigating in the Preview--

Navigating the camera around in the Preview window works just like any other 3D GL viewport, with the additional ability to define a rectangular area by pressing Ctrl+Alt+RMB to define a box area for zooming. You may wish to zoom in for a closer look without affecting your cameras position, it is possible if your mouse has a middle scroll wheel, first, press and hold the shift key on the keyboard, then place your cursor over the center point of where you'd like to zoom toward and roll the wheel. To reset the view, use the Render Done/Hold button at the top of the viewport.

Render Display

Please see Post Render / Render Display in Shading, Lighting and Rendering Chapter.

Tool Pipe



The 'Tool Pipe' is a very powerful component inside modo that allows the combination of tools *and* tool modifiers. It is also useful as a simple view to the currently active tool combination. There are many settings available through the Tool Pipe interface via context menus on the specific tool pipe components. The Tool Pipe also acts as a list that shows all the tools in the current preset combination, with the *action* tool at the bottom. The selected tool in the list is the haul tool, and selecting it again asserts the haul state if it has been cleared.

Within the Tool Pipe, tools can be made visible or invisible, enabled or disabled. They can also be removed, renamed, or reset. Tools required by downstream tools cannot be disabled. The attributes are shown as sub-tree lines.

E column = checkmark denotes if entry is enabled

V column = controls visibility; RMB+click for 'Invisible'- no handle, Basic' for Basic Handle, 'Advanced' - advanced handles provide additional controls (if applicable) and 'Preference' defaults to the user Preference setting.

A column = determines if tool is 'A' auto-activated or 'M' manually activated. With auto-activated tools users can instantly start hauling in the viewport when selecting the tool, with manual, users must first click in the GL viewport to activate hauling for a tool.

Tool column = Displays the tool or functions name.

Preset = Displays the preset to which the function belongs to.

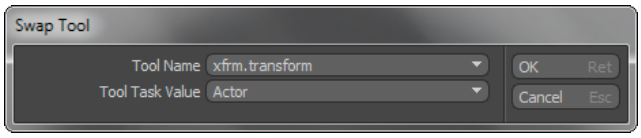
Working with Tool Pipe Presets--

There are several options from the Tool Pipes RMB context menu.

Apply Tool: Users can fire any tool by right-clicking on it in the Tool Pipe and choose the 'Apply Tool' command. This is a good alternative to using the tool interactively in the 3D Model View.

Reset Tool: Will revert all the tool's attributes back to their default state.

Remove Tool: Will simply remove the selected tool from the Tool Pipe.



Swap Tool: When 'Swap Tool' is invoked, a dialog window pops open allowing users to replace the existing tool with any other tool or function available to modo.

Save Preset: Allows the current state of the Tool Pipe to be captured as a preset for loading during later sessions. Many of the default modo tools are simply tool presets such as Element Move which combines Element falloff and the Element Action Center with the Move tool.

Load Preset: To reload presets users can use this context menu, create a custom button on the interface or assign the preset to a keyboard shortcut.

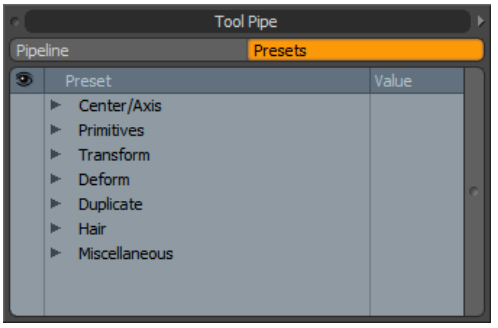
Auto Activate: This is a useful toggle. When active, a tool will become active immediately upon clicking its button or pressing the hotkey. All of the transform tools in modo are set to Auto Activate and this is why their handles appear immediately. Geometry creation tools such as the Cube or Bevel tools are not Auto-Activated so they require an initial click in the 3D viewport by the user. This allows the user the opportunity to set values on the Property viewport prior to activating or simply applying the tool.

Lock: This option will keep the current state of the tool properties as the default setting each time the tool is reactivated. For instance, if you use the Cube tool to drag out a 1 meter cube and then activate the Lock flag, whenever you activate the Cube tool the

numeric properties will be setup for a 1 meter cube. Without Lock on each time you complete the Cube tool will set the values for the next activation.

Auto-Drop: This option is most useful for painting style tools such as Mesh Paint or 3D painting operators. With most tools when you release the mouse the tool is still active and will allow you to edit the current values. For instance, if you activate Bevel you can click and drag repeatedly in the 3D view to edit the same Bevel settings. If you Shift-click you will start a new bevel. If the same were true of Mesh Paint each time you clicked the previous mesh placement would jump to your mouse rather than creating a new mesh. With Auto Drop active each new stroke is a new Mesh Paint. If you set the Bevel tool to Auto Drop you could quickly click and drag multiple times to create complex bevels.

Select Through: This is a highly useful option for a specific workflow where users do not want to drop tools to change their current selection. In the default mode modo uses any click away from a tools handles to "haul" the primary values. If you set the tool to Select Through you will no longer have "hauling" behavior but will get selection events instead.



Tool Pipe Presets--

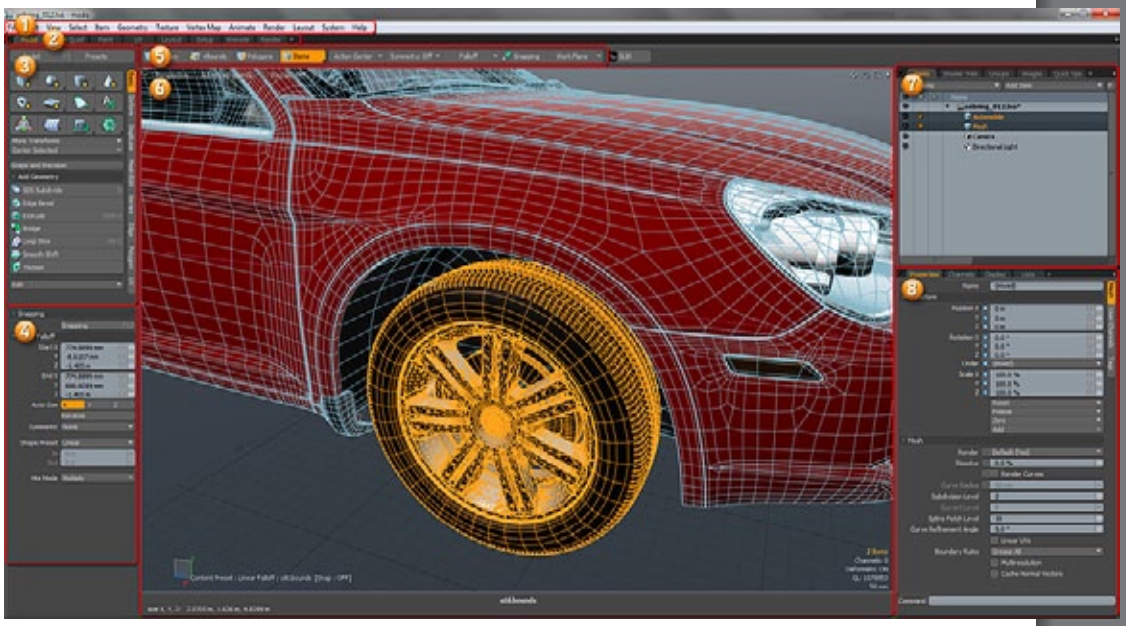
The 'Tool Pipe' presets view simply displays all the available presets organized by tool task.

Interface Layouts

Exploring the 601 Default Layout

When you first open up modo after installing it, this is the screen you're presented with. Think of it as the cockpit for the modo 3D rocket; designed and refined for the smoothest user experience. If the viewport you see doesn't match for some reason, one can easily restore the default interface by selecting from the menu bar 'Layout >

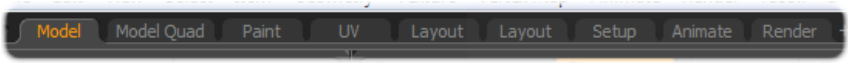
Layouts > 501 Default Layout'. The main interface is organized into 3 main sections. The left side column for tools and commands, the center viewport is where you will likely spend the greatest amount of time using modo and lastly, the right hand side houses information regarding the scene, its many items and their respective properties. Mouse over the image below to see the different sections--



Section 1 -- Application Menu Bar

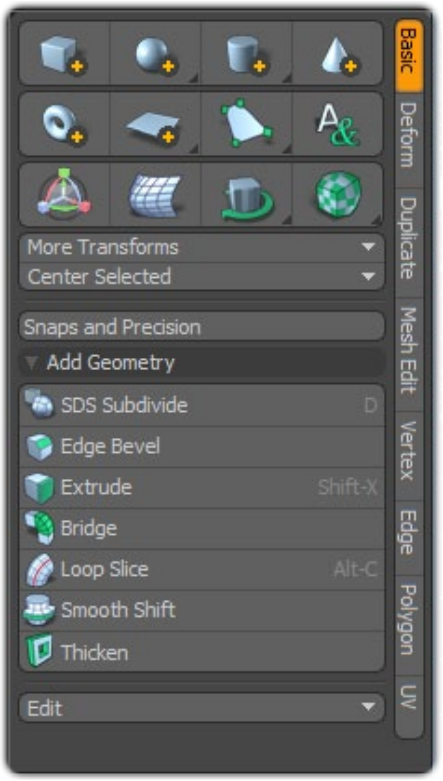
Virtually any function of modo is available from the menu bar, organized by their respective application. LMB-clicking on any of the menu bar items will reveal the menu itself, a list of commands that sometimes activates a tool that requires further action within the 3D viewport (Section 6), or sometimes initiates a popup window requesting additional input. Most of the functions within the menu bar are repeated elsewhere within the application or are available via modo's many keyboard shortcuts (listed to the right of the menu item). This functional duplication is meant to enhance overall workflow reducing the number of clicks necessary to accomplish a specific task, exemplified by the various image loaders found throughout the interface.

Section 2 -- Interface Layout Tabs



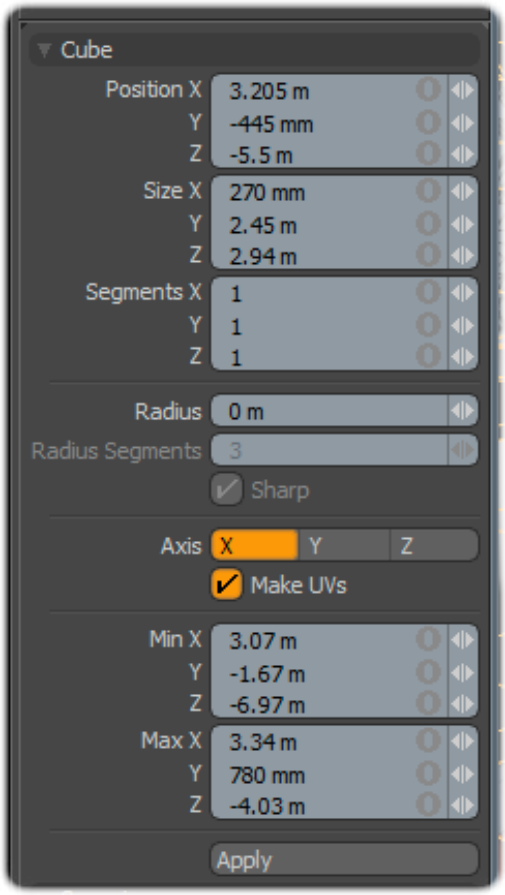
Directly below the menu bar is a series of click-able tabs that open separate interface layouts, each dedicated to the particular task named by the tab. While nearly all layout configurations share the same basic structure, toolbox on left, viewport in center and info and properties on the right, each tab features the tools specific to the task at hand, the modeling tab has the tools for creating and manipulating your geometry, the painting tab features the brush tools for sculpting and texturing your models and so on. Think of each tab as its own workroom geared toward streamlining the workflow of that particular function. The additional tabs are covered below.

Section 3 -- The Toolbox



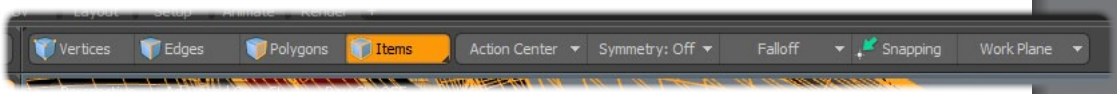
The toolbox section resides in the upper area of the left hand side of the screen. Nearly every interface tab has its own particular dedicated toolbox, illustrated here with the 'Model' tabs toolbox. The toolbox itself will house the various tools and commands. Down the right side of the toolbox are a series of rotated tabs or sub-sections (referred to as sub-tabs) that organize the various tools by their function. Clicking on any of the tool icons will activate the tool for use within the 3D viewport. One can easily tell which tool is active by the orange highlighting that appears behind it. Most tools function interactively with a left mouse button (LMB) click dragging within the 3D viewport (section 6). Some tools can be modified by control/command, alt/option and shift keys on the keyboard with some tools support additional function for the middle and right mouse buttons. Depending on the tools function, properties appear for it in the section directly below where settings can be modified and applied. A tool can be dropped (deactivated) by pressing 'Q' or the space bar, essentially freezing the current state or by selecting another tool. Buttons to quickly access alternate toolboxes are available just above the toolbox.

Section 4 -- Tool Properties



Directly below the toolbox are the properties dedicated to the currently active tool, in this case the 'Cube' tool. Tool properties will only appear when a tool has been activated and will change depending on which tool is selected. For most tools, one can interactively LMB-click drag in the viewport, and see the properties themselves update in realtime, conversely, users can directly edit the properties by LMB-clicking over a value in the light gray area and typing in a new value; though you wont see it update in the viewport until you hit the 'Enter' key. The 'Tab' key can be used as well, to jump between value input fields eliminating the need to click over each field. Once the user is satisfied with the settings they want, one simply needs to drop the tool by hitting the space bar to 'freeze' those particular settings. Users can also activate a tool, and enter values directly into the properties panel but will need to press the 'Apply' button to see their effect, losing any interactive editing functionality. By clicking in a 3D viewport while a tool is active, you are placing modo into an interactive mode, where the 'Apply' becomes unnecessary.

Section 5 -- modo modes

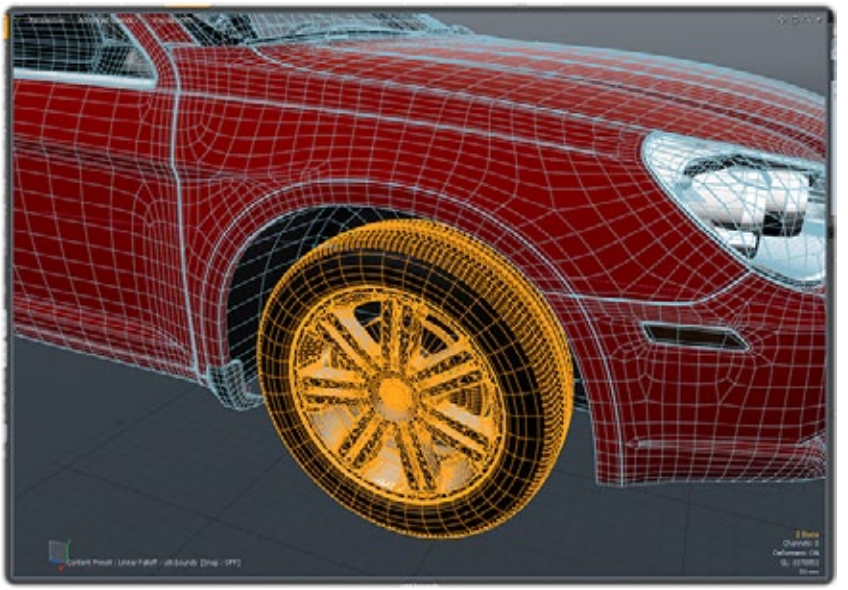


The modes section appears under the menu bar and above the 3D viewport. The first three buttons from the left are the component editing modes, an important distinction in modo. These modes set how one selects and works directly with geometry for manipulation --pushing, pulling and modify polygons. Users can easily toggle between any of these modes by pressing the space bar, continuously looping through the three.

The fourth button, 'Items' mode is for editing items. It can be somewhat confusing as the same transform tools modify both, but if you think of each item layer as a container, the component modes edit what's inside the container, and item mode edits the container itself, its much easier to understand. Why a separate 'Item' mode you might ask? Well, for one, when animating in modo, you are transforming items (the containers), individual vertices are not directly animateable (to do this, one would use an animated deformer, but that's another section). Also, replicators, and instances work by duplicating the *container*.

If you LMB-click and hold over the Items mode button, a popup reveals some additional modes for selecting surfaces within the Shader Tree (the heart of modo's surfacing and shading system) and for selecting an items center and pivot points. The rest of the buttons are modifiers for tools, changing their basic behaviors. 'Action Centers' are useful in that they can automatically adjust a tools axis and orientation. 'Symmetry', does exactly what its name implies, duplicating manipulations across an axis. 'Falloff' can control the affect a tool has on a selection based on an interactively applied distance. 'Snapping' can ease certain workflows that require levels of precision and lastly the 'Work Plane' is an interactive and modifiable virtual *workbench* within modo that can simplify many modeling tasks among other things. The documentation has dedicated sections that further explain each of these functions, but none are strictly necessary to start working in modo.

Section 6 -- 3D Viewport



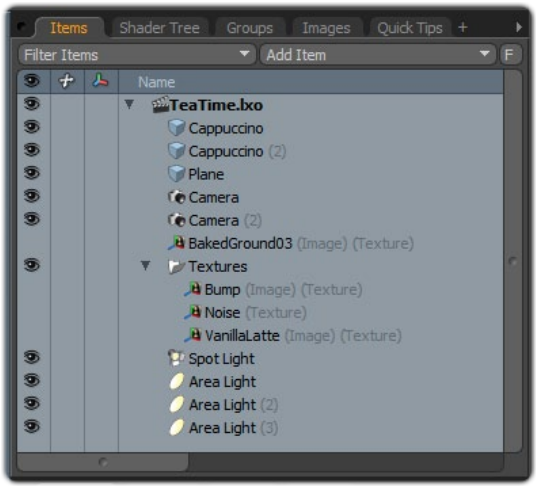
The 3D viewport is where you will likely spend the greatest amount of time working in modo. It is your virtual window into modo's 3D world. One should immediately notice the dark grid going off into the distance, this plane is purely for reference, denoting the zero position of the 'Y' axis. Think of this as the ground plane. As items move up, they will increase their 'Y' position value (sometimes referred to as the 'up' axis). The 'X' values move from left to right, and the 'Z' axis (depth) moves from front to back. (For users used to an alternate up axis, this setting can be modified in the 'System > Preferences') The darkest lines represent zero for the X and Z planes, and where all 3 intersect is the origin --'0,0,0' on all 3 axes; Item level transforms all take place *from* this position.

Navigation of the viewport can take place using the 3 icons in the upper right hand corner, LMB-click dragging over the the four arrows icon will move the viewpoint position, same thing for the rotation icon rotating the view and the magnifying glass icon scaling it. Of course there are also keyboard shortcuts for viewport navigation, making for easier navigation. Pressing both the shift and alt/option keys and then LMB-click dragging over the viewport will move the view position around left and right and up and down, the middle mouse button (MMB) will move the view forwards and backwards while the right mouse button will restrict the view change to just up and down. To rotate the viewpoint, press the alt/option key alone and LMB-click drag over the viewport; MMB-click dragging will restrict the rotation around the 'Y' axis, and the RMB will *flick* the view i.e. it will slowly stop when the button is released instead of directly stopping. Zooming can be done in the viewport by scrolling the mouse wheel over the viewport, or by ctrl/command-alt/option LMB-click dragging over the viewport; RMB-click dragging will define a rectangular area that when the button is released, modo instantly zooms to.

If you've been trying out the keyboard shortcuts for navigating the viewport, you likely noticed the lighter grid, and how it would interactively change position based on the current viewpoints position. This is the workplane, think of it as the virtual desktop within the viewport. If one were to select the 'Make Box' tool and LMB-click drag over the viewport, the first plane of the box would originate upon the work plane itself. Drop the tool by hitting the space bar and rotate your viewpoint until the Work Plane changes position and then activate the cube tool again and draw another square, this time it originates from the new workplane position. So based on your viewports position, the workplane always tries to face the user, and appears within the viewport as reference to where new items will be created. For users used to working in a 4 viewport type system where one can easily place items in 3D space by moving it around in each viewport, the Work Plane helps by allowing users to work in a single large perspective viewport, and easily position items within the scene without surprises as to where items will be created. There are several functions for modifying the Work Planes position and how it

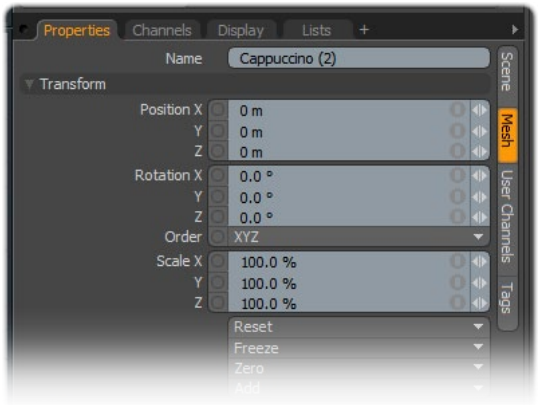
adapts to the viewport covered more fully in the dedicated Work Plane section.

Section 7 -- Items List, Shader Tree, Groups and Images



The top right area of the interface basically lists all the various items that make up a modo scene, these lists are important to the overall modo workflow, as such, they appear in all the various layout sections (*interface tabs from section 2*) with some variation. The tabs across the top of this section choose basically which items are listed; grouped by how modo sees them. The 'Items' list shows all the items that make up the scene itself --a geometry layer is an item (the container), a camera is an item, lights are items. Within the list, items can be grouped together by parenting for organization or for animation hierarchies. Selecting the various items will display their properties within a properties viewport along with numerous other functions covered in the Items List page. The 'Shader Tree' viewport organizes all the surfaces within the scene and their respective items, such as materials, shaders, textures and groups. When rendering a scene, it is the shader tree that controls how the final rendered image appears, covered fully in the rendering and shading section of the docs. The 'Groups' viewport is useful for creating groupings of items in a scene, making it easier to select and manipulate multiple items, as well as for controlling lighting (light linking), making multi-item replicators and for working with animated scenes when setting keyframes. The 'Images' tab lists all the external images currently loaded into modo and offers functionality for working with them. Note that images aren't saved in the modo file itself, so they'll need to be available each time a scene is loaded. Lastly, the 'Quick Tips' viewport offers some shortcuts to videos and such for quickly getting up to speed in modo.

Section 8 -- Properties, Channels, Display and Lists



Directly below the item and shader tree lists section, is, drum roll please, more lists! As scenes fill up with items and textures, UV maps and such, a serious amount of information is created along with it, and modo offers many ways of viewing and manipulating all that info. In this particular gang of lists, the 'Properties' is the most important, users will most likely have this one open most of the time.

The first tab across the top is the main item 'Properties' tab. Every item in modo has a series of attributes associated with it, and this is the viewport where users can edit them. Selecting any item will display the properties specific to it, dynamically changing as different types of items are selected. Next over, the 'Channels' tab is a more concentrated version, displaying every animateable attribute associated with an item as a continuous list. Lastly, the 'Display' viewport has settings specific to how items are displayed in the 3D viewport.

The next tab is the Channels viewport. Basically, every animateable attribute of an item is listed on the Channels list for easy view, selection and manipulation. The channel list is covered more in depth in the 'Channels' page of the documentation. The 'Display' tab controls the visibility and display of items within the 3D viewport, also more fully covered on its own page.

The next tab 'Lists' is actually three mini viewports in one. The top section lists all the associated 'Vertex Maps'. A vertex map is an instruction associated with a vertex to tell it do a certain thing, such as move a specific offset distance (morphmap), specify a 2D position on an image (UV) or simply a value that is fed into another function such as weight maps do. To see the vertex map associated to an item, the item(s) must be selected (highlighted in the item list). From this window, users can create, rename and delete any of the various maps.

Next down is the 'Pipeline' viewport, also known as the tool pipe; the heart of the modo

toolbox. A tool can be made up from a single function like the make cube tool. Or a tool can be a combination of functions rolled together, such as a transform tool with a certain falloff and action center (flex tool!). The tool pipe is where a user can make custom tool combinations. 'Presets' contains the various pre-made tool combinations included in the various tool boxes that install with modo.

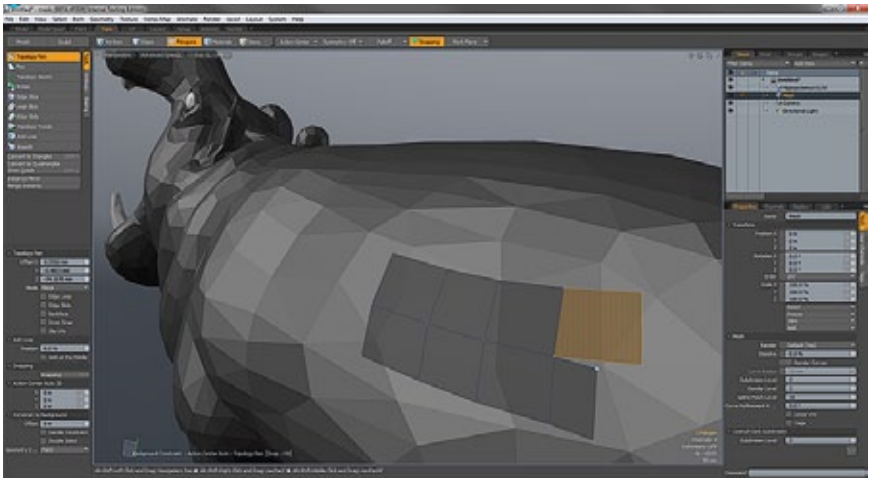
'Statistics' is precisely that, a listing of scene statistics -the total number of polygons, how many of them are triangles or have more than 5 sides. More than just a list, users can use the little +/- signs that precede each line for selecting/deselecting geometry in the scene based on the specific criteria. Lastly, the 'Info' tab will give you all the information associated with a selection. Select an item, you get info on that item. Select some geometry and you'll get info on all the associated vertices and map values as well as applied surfaces and selection sets associated with it, among other things.

Model Quad

The next *interface* tab (*section 2 from above*) is identical to the base modeling tab covered above, except the the 3D viewport is split into a *quad* view, displaying three orthographic views with a perspective view. This method of working may be more familiar to some users allowing them to precisely place object within the scene by making adjusting in any of the four views simultaneously.

Topo Tab

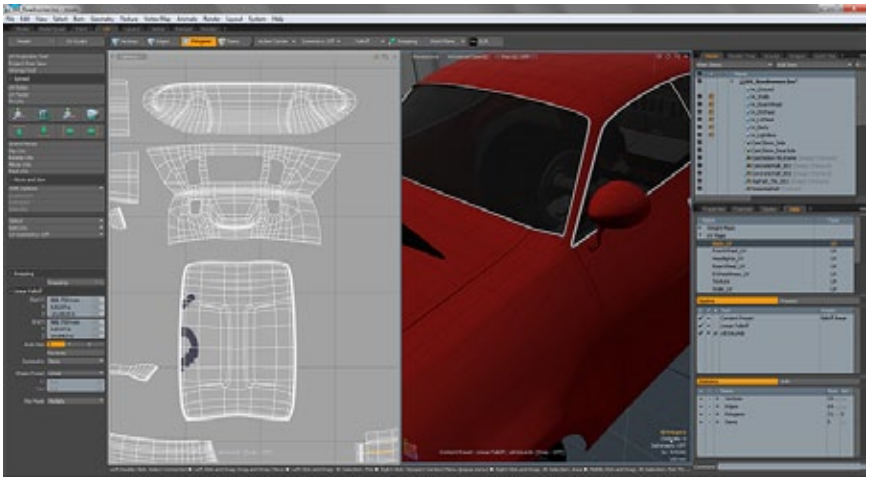
The 'Topo' interface tab, or retopology as the case may be contains all the retopology related tools. Retopology is the act of taking very dense (high resolution) mesh and using its surface in the background as a constraining surface to build a clean lower resolution mesh. This is a common task when creating game assets where low polygon counts are a requirement, also when cleaning up scanned mesh data for animation or when simply making it easier to change the topology and polygon flow of an existing model. The toolbox contains tools that automatically conform to any item in the background, with a special openGL drawing mode to make it easier to see what is being drawn in the foreground over what is located in the background.



Paint Tab

The 'Paint' interface tab focuses on painting and sculpting. The left hand tool box is separated into the various tools and controls for painting textures and sculpting meshes. Below the 3D viewport are introduced an preset browser for viewing brushes and preset tool options. Users can cal also directly access a color picker and see the details of the current brush tip in the tip view window.

UV Editor

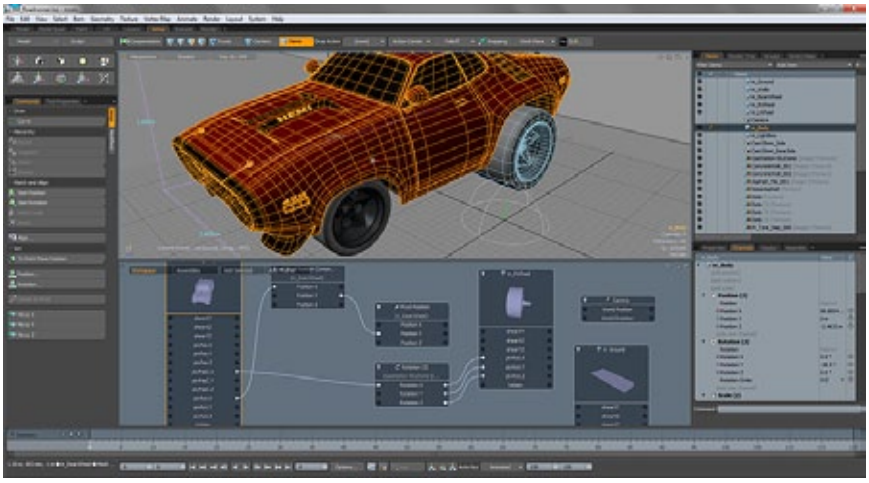


The UV editor interface splits the 3D viewport vertically adding a UV view on the left. Users can select from the various maps in the Vertex Map list on the right hand side. The toolbox focuses on the various UV creation and editing tools.

Layout Editor

The Layout Editor is a useful interface to block out scenes, set up Replicators and Instances, and generally design the layout of any scene. The Preset Browser dominates the area below the 3D viewport, providing users effortless access to the many preset assets provided with modo for a streamlined workflow. Users can simply drag and drop from the preset viewport into the 3D viewport to populate a scene with anything from environment maps, assemblies, mesh items and surfacing presets mimicking a large variety of surfaces.

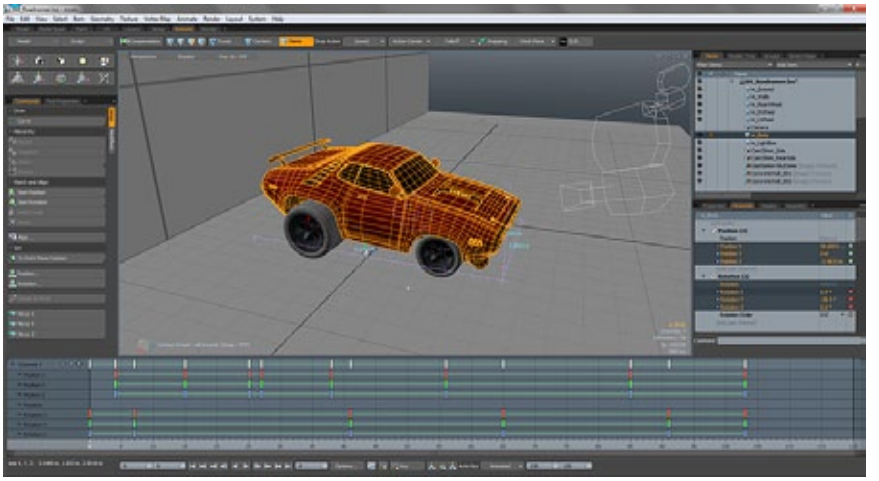
Setup Editor



The Setup layout is mainly designed for rigging using the Schematic Editor. Users

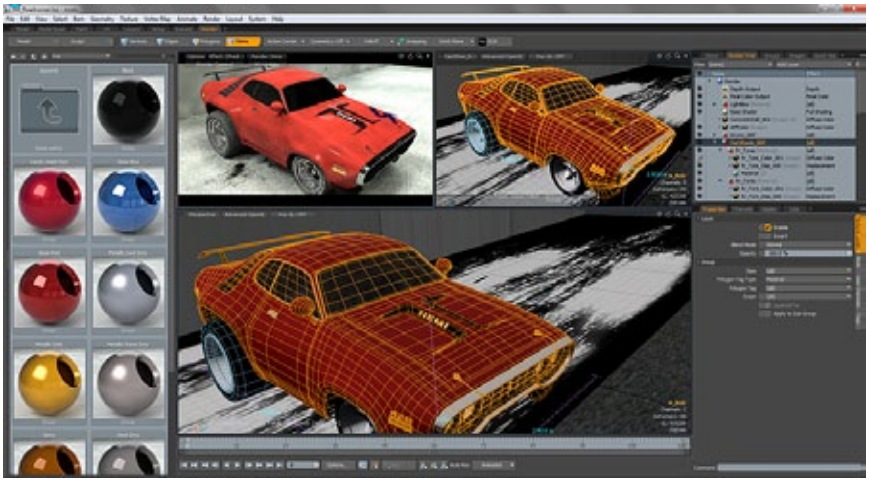
can easily select items in the Items List and Shader Tree and grab their associated channels in the channel list and drag them into the schematic view for node linking. The toolbox also offers command related to channel links and modifiers, offering functionality similar to that of the Animate interface tab (*below*).

Animate



The Animate tab focuses on the setup and animation of elements in a scene. The toolbox is similar to that of the 'Setup' interface allowing users to define hierarchies, channel links and assign modifiers. The viewport introduces the concept of time to the scene via the timeline. Hidden just above the timeline is the Track View designed to allow users to easily re-time animated elements. To open, simply drag the top frame edge of the timeline upward revealing the viewport. Timeline playback controls run across the bottom of the interface with additional controls for setting keyframes and controlling the auto keying feature.

Render



The Render Layout provides streamlined access to the main options users need when focusing on the task of rendering a project. On the right hand side is the Shader Tree and Render Properties so users can create and edit the various material and texture items and settings. The left toolbar houses the Preset Browser providing simple access to the many included assets. The main 3D workspace is split into three views; a Preview window that provides an extremely fast near final preview of the rendered scene, and camera view providing direct control over the positioning of the camera and a perspective view allowing users to adjust light and camera elements from a third person perspective. To render in this view, press 'F9' or select the 'Render' command from the 'Render' options of the menu bar.

Legacy Layouts

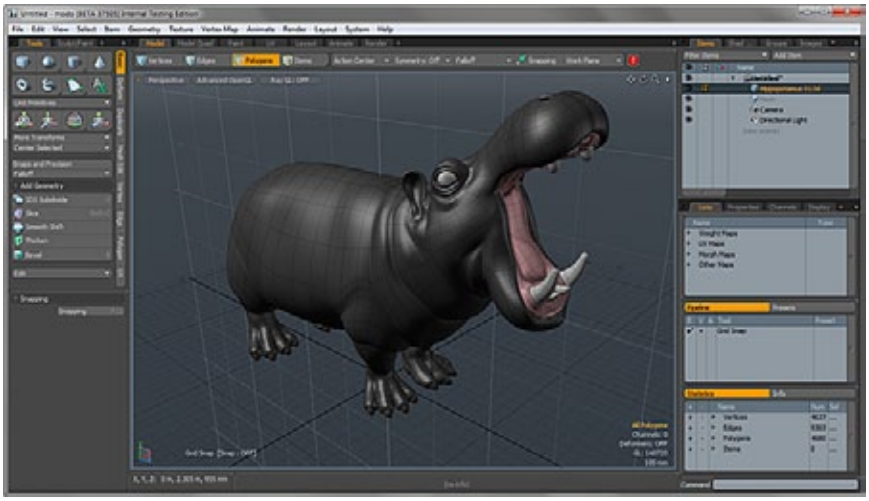
With each new version of modo released, we always try fine-tune the workflow of the default interface, as well as make changes necessary to incorporate new features. We're aware some users prefer these legacy layouts from previous version and provide them as a convenience. They also work well when following older tutorial material, aiding newer users who may not be familiar enough with modo to find functions used in the training.

The various layout provided are listed below with brief explanations of their utility. These can be loaded from the menu bar using the "Layout > Layouts" commands and selecting the layout from the menu. Users can easily restore the default layout by selecting the '601 Default Layout' option form the menu at any time. In cases where the factory default was mistakenly overwritten, users may reset them using the menu bar command "Layout > Restore" and then select the layout to recover.

201 Default Layout

modo 201 introduced rendering to the already powerful modeling package, necessitating the addition of the Shader Tree tab to the toolbox, housed on the left panel of the interface. This allowed users access to both the Item List and the Shader Tree simultaneously. Rendering and UV editing tabs were also added to the main GL viewport panel in the center facilitating a simpler workflow for those tasks.

301 Default Layout

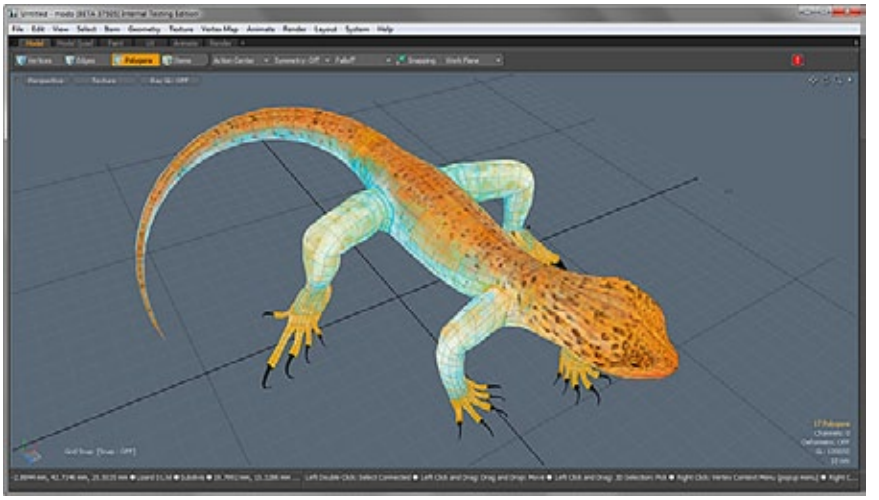


Further refining the powerful package that was 201, animation and sculpting functionality are added and the interface further streamlined, keeping the workflow simple an intuitive with an ever increasing tool set. Toolboxes are standardized on the left panel and item lists on the right.

401 Default Layout

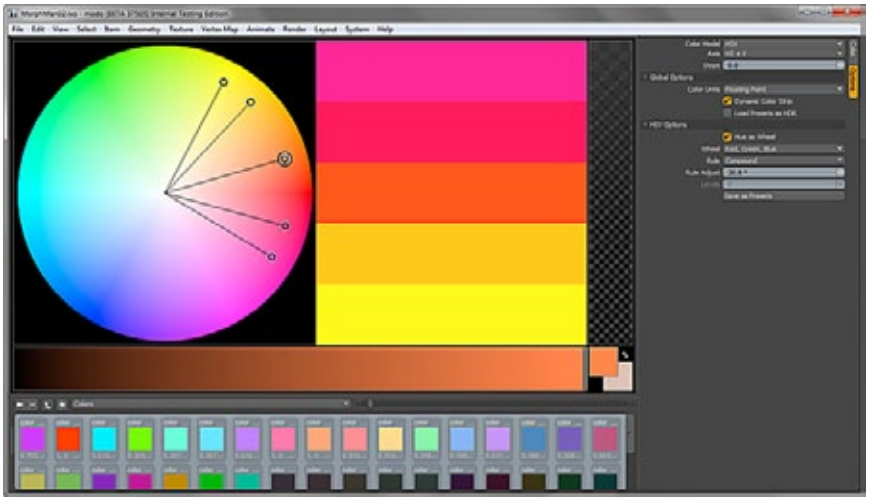
modo 401 introduced much greater refinement to the animation capabilities, as well as a variety of rendering related functions, including replicators and fur. Tabbed workspaces become the norm, further focusing the toolset for each function. Toolboxes remain on the left panel with the items listed on the right hand side. All tabbed layouts have further refinement, including the paint and sculpt and animation layouts receiving their own toolboxes. The 'Layout' tab adds preset functionality aiding in the quick setup of scenes.

3D Sparse



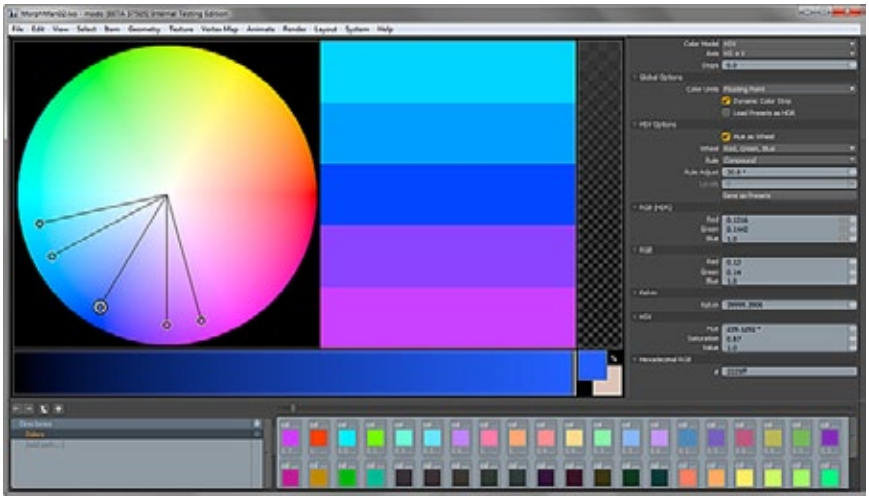
The 3D Sparse layout is focused toward maximizing the openGL viewport view, where most of the modeling and painting action occurs. This maximized view trades the convenience of toolboxes and items lists for screen real-estate. Users fluent in pie-menus can find the tools they need quickly using keyboard shortcuts.

Color Picker



Color being such an integral part of texturing, lighting and rendering, modo has an advanced color picker available. For designing color schemes for a scene, it may be helpful to utilize the Color Picker layout that maximizes the standard color picker to full-screen size. From within this view, users can select color models, set scheme rules and save sets of colors in a preset for application elsewhere in modo.

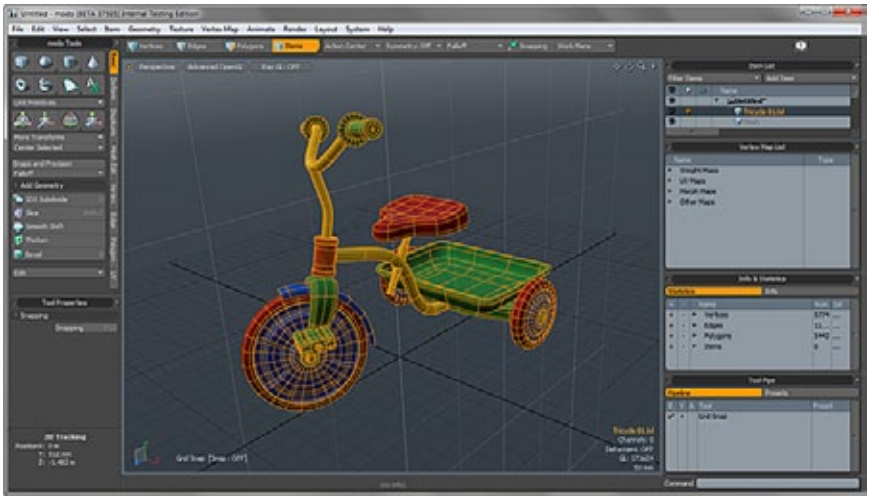
Theme Editor



The Theme Editor is essentially the color picker with all the options visible within a

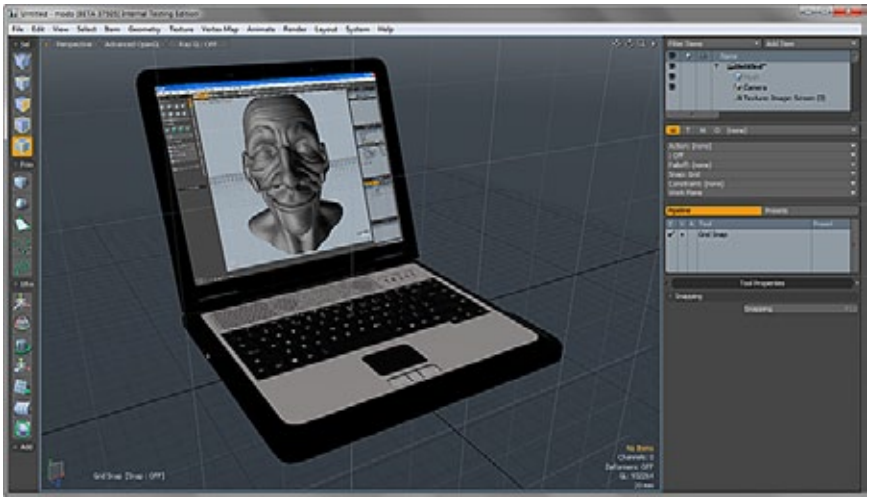
single viewport, and additional options visible within the preset viewport included at the bottom of the interface.

Model



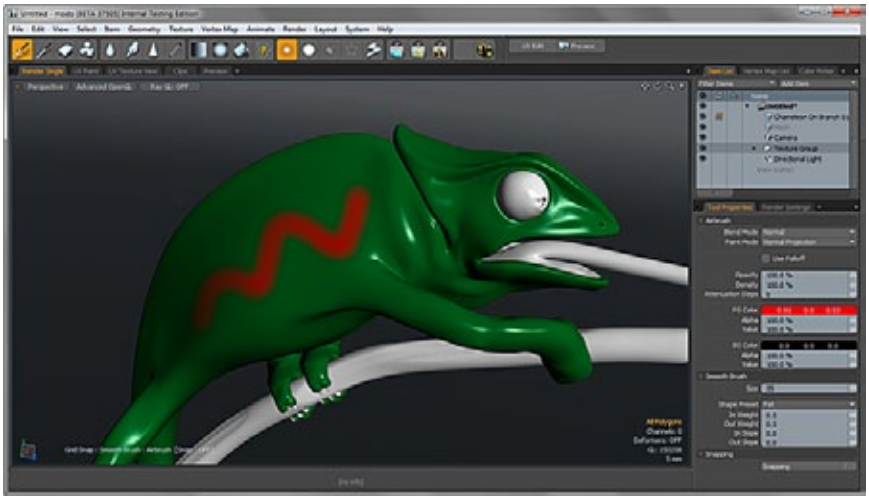
The Model Layout is nearly identical to the Default Layout of modo 101, 102 and 103. This is a highly focused modeling environment. As such, it is a good choice for work that does not require any painting or rendering tools for it pushes these out of the way of the artist so that the focus is wholly on modeling.

Simplified



The 'Simplified' layout consists of a Menu bar, the single Perspective 3D Model View, a buttons only version of the modo Tools tool bar, a headless Mesh List, Vertex Map List and Tool Pipe, a Tool Qualifiers form, and a Tool Properties form. Similar to the 3D Sparse layout where tabs and other conveniences are traded for screen real-estate, the simplified layout reduces the tool menus to icons, maximizing the viewports overall size. For users not fans of using a pie-menu workflows, the Simplified layout provides and easier path to using the most common tools, also providing good utility to users working on smaller screens often associated with laptops.

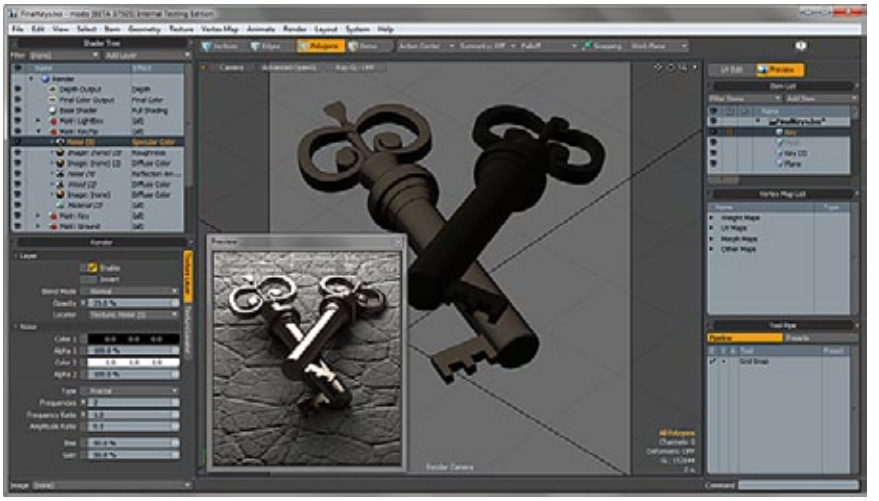
Paint



The Paint Layout looks and feels like a dedicated painting application. This layout pushes modeling and rendering tools to the background so you can focus on texturing your masterpiece. The layout consists of the painting tools across the top of the interface in icon form and a large workspace underneath. The main workspace is a tabbed viewport providing quick access to Render Single, UV Paint, UV Texture View, Clips and Preview. By default you are in a Camera view.

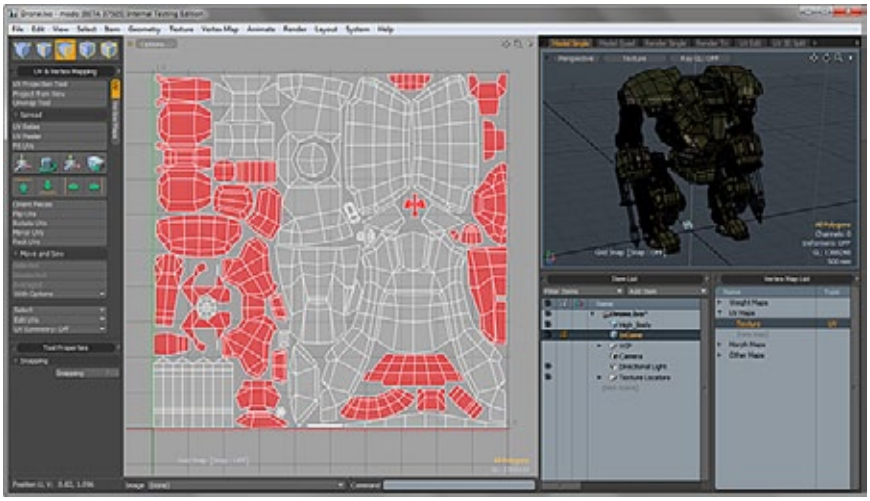
Paint tools are presented as icons along the upper left and a rapid way to access other useful views is provided to the upper right. On the right hand side of the layout is a viewport group with the Item list, Vertex Map list and a color picker for quick access. Beneath that viewport is another tabbed viewport group containing a Tools Properties section and Render Settings view. The first provides access to the current paint tool's properties so that you can control the tool with precision. You can also press the "K" key with your cursor over the 3D viewport to bring up a popover of the tool's properties directly under your mouse. The Render Settings tab contains the Shader Tree and a Render properties form.

Render



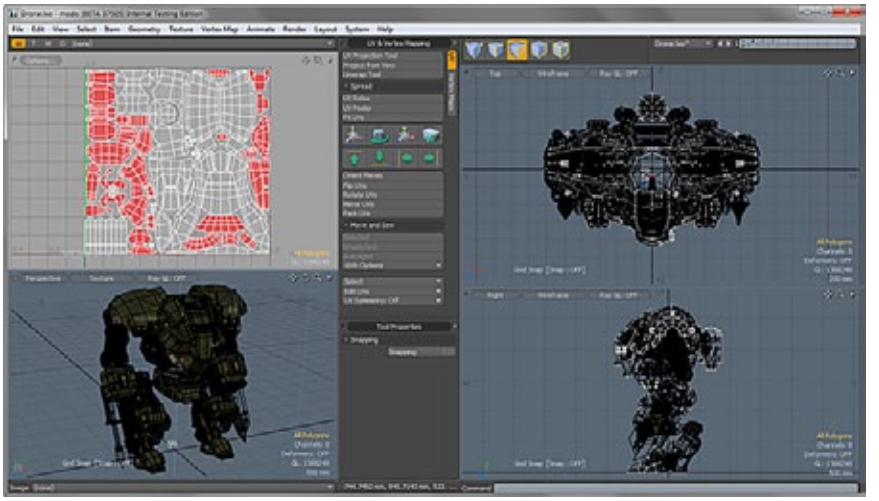
The Render Layout provides streamlined access to the main options you will need when focusing on the task of rendering your project. On the left hand side is the Shader Tree and Render Properties form so you can create and edit the various material and texture items and settings. The main 3D workspace provides you with the direct control over scene layout using a Camera view. On the right hand side the Item List allows you to create, select and edit items such as meshes, cameras and lights. Just above the Item List are three "Quick Access" buttons which open Palettes containing modeling tools, vertex mapping tools and a preview window, in that order. To render in this view, hit F9 or select the 'Render' command from the Render options of the menu bar. Above the addition of the 'Render Preview' window was added by pressing the 'F8' key.

UV Edit



The UV Edit Layout is provided for intensive UV editing sessions. The layout consists of three major areas and two smaller sections that can be expanded or contracted as needed. On the left hand side you will find the selection modes (in icon form or under a Quick Access button), the UV & Vertex Mapping tool bar and also a Tool Properties section. Just to the right of that section is the main UV edit viewport and to the right of that is a tabbed 3D viewport. These two views provide ample space for viewing your UV map and your textured model in the upper right as you edit.

Just underneath the UV and 3D viewports are additional viewports that collapse and hide nicely when the UV or 3D viewport is expanded. Under the UV view is an Image list and Command History and under the 3D viewport sits an Item List and Vertex Map List. By dragging the border between the UV or 3D view and the views beneath them you can expand and collapse the viewports as needed.

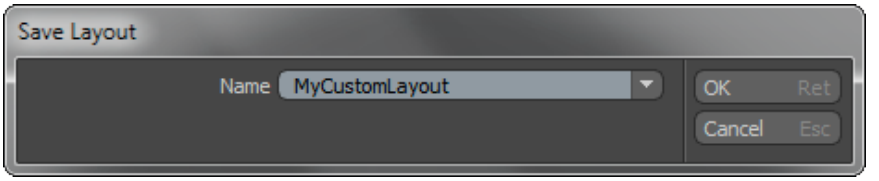


The 'UV Edit 101' layout consists of a Menu bar and four viewports that flank a central set of tool bars. On the left hand side are UV Texture and Perspective views. On the right hand side are a Top and Right view. In the middle of the display is a central UV and Vertex Mapping toolbar, plus a collapsed Vertex Maps toolset and a Tool Properties section. Rounding out this layout are a Mesh List, and a set of Selection mode icons accessible under a Quick Access button.

Customizing your Layout

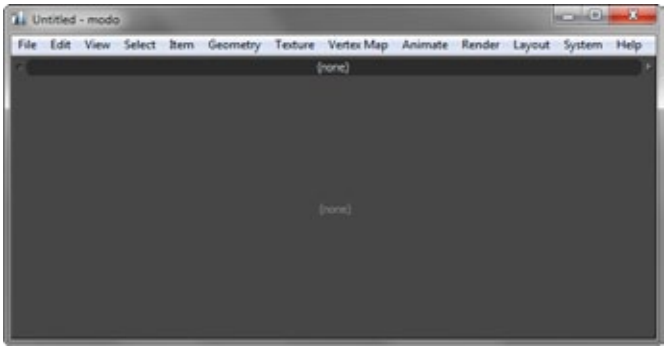
modo provides one of the most, if not the most flexible UI of any 3D application, providing a variety of controls allowing users to customize the program to their liking. The controls to do so are simple and allow users a great degree of flexibility. When creating a completely custom layout, the first thing to do is define a new layout with which to modify, this ensure that the system defaults are not mistakenly changed.

To do this, go to the layout controls in the menu bar "Layout > Save As..." and in the popup requestor enter a unique layout name.



Press 'OK'. Next, load the interface to make it active by selecting the "Layout > Layouts > [MyCustomLayout]", it will appear at the bottom of the list. Selecting it makes it

active, although nothing will have changed because the new interface is a saved version of the previous. Now go back and select "Layout > Clear" and you'll have nothing but a gray window with a menu bar, like this...

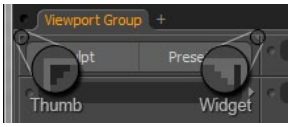


The controls to customize the layout are all that are left within the viewport. In the upper left corner, below the menu bar, you'll notice a tiny shaded dot next to the title bar, this is called the 'Thumb'; viewport *frame* specific controls all reside here. On the opposite side of the title bar rests a small right facing arrow called the 'Widget', this is where users control the contents of a frame. Popup menus from both are accessible from a RMB+click (right mouse button click). Every window has these controls making it easy for users to modify any layout to their liking.

To finish our little exercise, we'll continue recreating a single tab of the default interface. To do that, the interface will need to be divided. For the first step, select the Widget arrow, and from the menu choose 'Viewport Group', this nests two viewports inside one another. Now you'll notice two widgets on the right, for the upper widget, select the 'Tabbed Viewport' option from the menu. This is the makings of the tabbed interface, where each tab holds a different focused layout. In the nested viewport below, RMB+click the Thumb, and from there choose the option 'Split Vertically'. This divides the window into two equal sized viewports stacked on top of one another. The viewports can be resized by hovering the mouse pointer directly over the border between the frames, LMB+click and drag to position the edge in a new location. Dragging the border toward the top makes the first viewport. To finish off dividing the windows, follow this diagram, splitting viewports horizontally or vertically, as necessary.

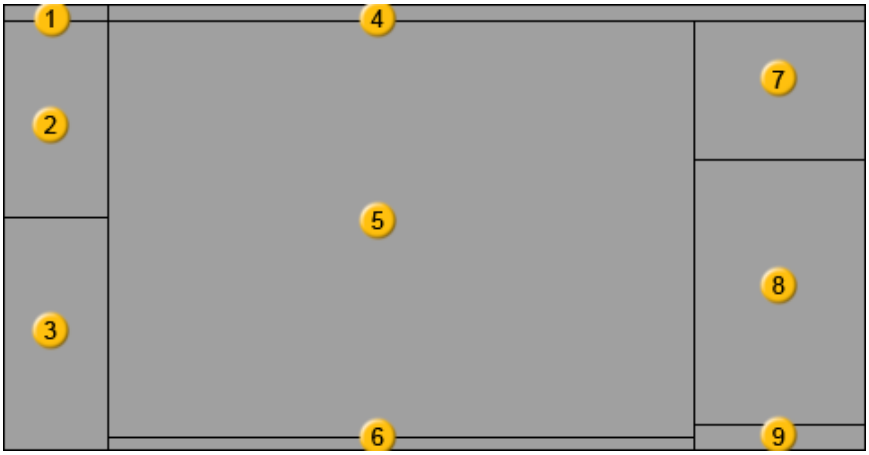
The top most bar was split horizontally once, and the lower window was split horizontally twice, resulting in three columns. Each of those columns was split vertically once, with an additional split in the right column. Position the viewport frames so the layout loosely mimics the above template. If you accidentally split a viewport, don't worry, you can easily delete viewport panes from the Thumb menu command 'Delete'.

Next we will populate the windows with content. The most common viewports are easily accessible form the Widget arrow on the right side of the title bar of each viewport. RMB+click on the them and then select the viewport type from the popup menu. We'll travel in a top to bottom, and left to right fashion filling in the viewports. In the first viewport, select "Form View" first from the top of the Widget menu, this changes the window to a generic form but it still looks empty, so again from the Widget, choose the menu option "Viewport Settings > Sculpt and Presets Quick Access". This adds the quick access buttons to the viewport, but the pesky title bar gets in the way. This is removed from the 'Thumb', RMB+click and select the option 'Min Header' to minimize that. Don't worry, the 'Thumb' and 'Widget' haven't disappeared, they are just very tiny triangles in opposing corners of the window illustrated below.



A little harder to see now, but there non-the-less. That window is finished, so in the next below it, RMB+click the widget and select "Toolbars > modo Tools" and again

minimize the header. Below that window in the next frame, select "Properties > Tool Properties" and again minimize the header.



For the remaining six windows, the menus are as follows--

4.) From the Widget -> "Form View" and then again "Viewport Settings > modo Modes".

5.) From the Widget -> "3D Edit > 3D Model View"

6.) Widget -> "Status Feedback"

7.) Here we'll add some more Tabs, so select from the Widget "Tabbed Viewport" and them RMB+Click on the tab itself and choose "Data Lists > Item List". Click the "+" icon to the right of the tab and then choose "Data Lists > Shader Tree", that adds an additional tab.

8.) Viewport 8 is kind of a special case, as it serves multiple purposes, with nested viewports, similar to the main window. So from the Widget, select "Application > Viewport Group", and like before, set the upper widget to "Tabbed Viewport" and split the lower window twice for 3 even sized viewports. In the top viewport, set "Data Lists > Vertex Map List", in the next window down set "Data Lists > Tool Pipe" and in the third, set "Info > Info and Statistics". Next, across the top of the group, add an additional tab like before and select "Properties > Item Properties". Minimize the Header in all the viewports you just created (as before with the Thumb), forgoing the rest of the tabs for now.

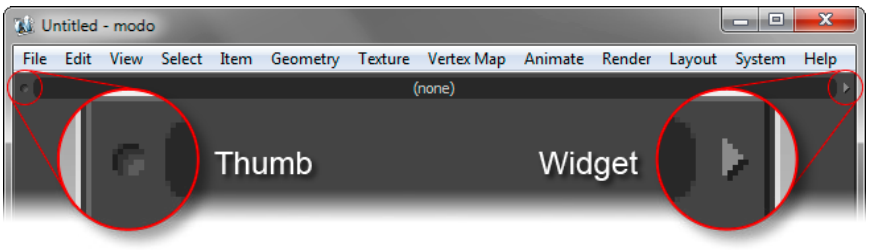
9.) From the Widget -> "Utility > Command History".

That pretty much re-creates the modeling view of the default interface (less a few tabs for brevity), from an essentially blank slate. Make sure to save it again using the 'Save As..." command and select the same name you established at the beginning of the tutorial. Closing modo will force if to be written to the config file ensuring it is saved for future sessions.

Between the commands of the 'Thumb' and the 'Widget', a great deal of customization power can be wielded. Any viewport from modo can be recalled from the Widget arrow, although some are available only within the 'Viewport Settings' submenu choices. Another option users should be aware of is that many viewports will change when scaled, providing slightly different ways of interacting with the info contained therein. These options can be useful in setting up interfaces that mimic other popular applications. A thorough understanding of the common viewports will help in achieving your intended result.

Layout Controls

The interface of modo is made up from a flexible number of windows called 'viewports', that can each have their own contents. Outside of the controls of the 'Layout' option of the menu bar, a couple of interface elements exist to easily control these viewports. They are called the 'Thumb' (also known as the *Dimple*) and the 'Widget'. They appear in the upper corners of each viewport.



RMB+clicking on either icon opens a context menu with a variety of options for controlling viewports. The 'Thumb' is used for manipulating the viewport frame, meaning the viewport itself can be split, deleted and otherwise customized using its commands, while the 'Widget' controls the contents of the viewport, where users can choose from the entire array of possible options. Users can open either menu by way of a RMB (right mouse button) click on the icon. Any of the various menu options van then be chosen with a LMB+click.

Thumb

Controlling the viewport

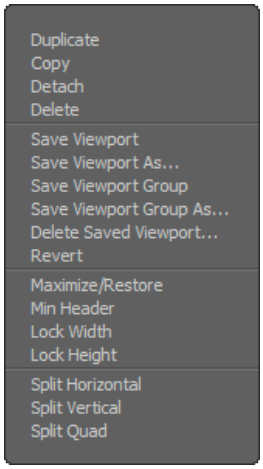
LMB+click-dragging on 'Thumb' icon allows users to detach the viewport into a floating 'Window'.

Shift+LMB+click-dragging on 'Thumb' icon will clone the current viewport into a floating 'Window'.

Ctrl+LMB+click-dragging on 'Thumb' icon will split the viewport in the direction of the drag (either horizontally or vertical).

LMB+click-dragging on a viewport divider will scale/resize the associated viewports.

Thumb menu--



Duplicate: Creates a copy of the selected viewport and deposits it as an embedded viewport on the right hand side of the layout, leaving the original in place.

Copy: Creates a detached copy of the selected viewport leaving the original in place.

Detach: Removes the embedded viewport and floats it in its own Window.

Delete: Removes the current viewport from the interface.

Save Viewport: Saves any changes made to the current viewport preset.

Save Viewport As: Saves a new viewport Preset allowing the user to set the name and category.

Save Viewport Group: Saves any changes made to the current viewport as well as any child viewports to the current preset. Child viewports are defined as those that are nested within an overall viewport frame, such as the Mod-Quad layout.

Save Viewport Group As: Saves a new viewport group Preset allowing users to set the name and category.

Delete Saved Viewport: Provides users a means for users to remove viewport custom-made viewport presets.

Revert: Restores the viewport to its default preset settings.

Maximize/Restore: Fills the viewport group frame with the current viewport, this option is highly dependant on the viewports layout. This is mostly useful in the case of maximizing a 3D viewport inside a collection of 3D views, as in the case of the Mod_ Quad preset.

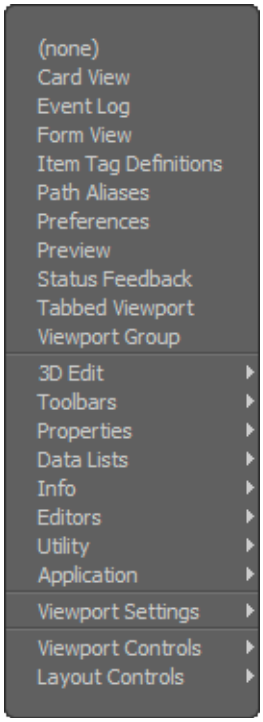
Min Header: Removes the viewport header, leaving mini versions of the Thumb and Widget in their respective corners, preserving screen real-estate when necessary. This is convenient when using viewports for simplified data displays, such as the coordinates viewport in the bottom left hand side of the default modo layout.

Lock Height: Fixes the viewport height at the current setting. Subsequent resizing of the interface won't affect any particular window with their size locked, allowing for fixed size viewports.

Lock Width: Fixes the viewport width at the current setting. Subsequent resizing of the interface won't affect any particular window with their size locked, allowing for fixed size viewports.

Split Horizontal: Evenly splits the current viewport into two viewports of the same type, next to each other.

Split Vertical: Evenly splits the current viewport into two viewports of the same type, one on top of the other.



Widget

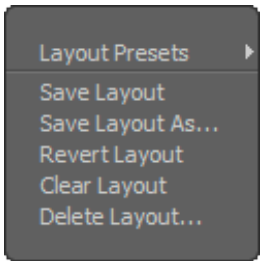
Widget Menu--

The Widget menu is content sensitive, changing the menu options based on the category of the currently loaded viewport preset. The sections are separated by thin shaded lines. The upper most section will change to the most common types of viewports for any given category. The next section down provides all the defined categories and their related viewport types.

Viewport Settings: The 'Viewport Settings' menu contains all the viewport specific options. For 'Form View' viewports, this will be a long list of the various forms available, such as the mode options buttons. For the Shader Tree, this will be 'Filter' and 'Preview Icon' options. The settings of many of the viewport options are covered in their respective documentation pages under the 'modo Viewports' section of the interface page of the documentation.

Viewport Controls: The 'Viewport Controls' menu duplicates the functionality (and therefore the menu) of the Thumb context menu as a convenience.

Layout Controls: The 'Layout Controls' sub-menu provides the same functionality of the options available in modo's menu bar under the '"Layout" menu as a convenience. The options available are as follows--



Layout Presets: Users can choose from the various preset layouts, and can load custom saved layout from this menu as well. These layouts (referred to as Legacy Layouts) are covered on the Legacy Layouts page of the Documentation.

Save Layout: The 'Save Layout' command will let users save any interface customization into the currently loaded preset (which by default is the 'modo 601 Default'). This action will overwrite the original layout replacing it fully.

Save Layout As: The 'Save Layout As' command will let users save any interface customization into a new modo Default Layout that will appear at the bottom of the 'Layout Presets' list. When invoked, a menu will open requesting a name for the new preset.

Revert Layout: The layout Revert options provides users a means to reset any layouts to 'Factory Original' should one be mistakenly modified.

Clear Layout: Removes all viewports from a layout, leaving only a single empty "(none)" viewport form. A tutorial on building a viewport from an empty frame can be found on the 'Customizing the Layout' page of the documentation.

Delete Layout: The 'Delete Layout' command opens a popup window allowing users to remove any custom created interface layouts. It is not possible to remove the default layouts.

Additional controls for the layout are available in the 'Layout' menu of the modo menu bar that are not part of the 'Layout Controls'.

Lock Thumb Dragging: Users can interactively modify viewports using several combination actions when dragging the mouse pointer over the 'Thumb' icon. Users may find from time to time they mistakenly modify the layout inadvertently and may wish to disable the 'Thumb Dragging' functionality by toggling this option.

Lock Divider Dragging: Users can interactively resize viewports by LMB+click and dragging the mouse pointer when over the frame dividers. Users may find from time to time they mistakenly modify the layout inadvertently and may wish to disable the 'Divider Dragging' functionality by toggling this option.

Windows/Palettes: While all the default layouts all have viewports that are embedded, it is also possible to have floating viewports. When viewports are floating (or detached) they can be 'Window' or 'Palette' style. The only difference between them is that 'Palettes' have a thinner header with a single control for closing the Palette, whereas Windows provide full OS specific controls for minimizing/maximizing and closing the viewport. The menu bar command "Layout > Windows/Palettes" provides additional options to open custom pre-made viewports or varying types.

Toggle Palettes: Users can toggle the visibility of floating Windows and Palettes using this command or the keyboard shortcut for the ' ` ' (accent grave key, same as the ~ tilde).

System and Files

File Commands

modo provides all the standard controls necessary for opening, closing and otherwise managing files to and from disk. Most options perform from operating system specific dialog boxes. File managementis largely accomplished through the commands in the menu bar under the 'File' menu. User need only to LMB+click on a menu command to invoke the operation.

New: The 'New' command will create a new *empty* (blank) scene file and add it to the 'Item List'. By default, the scene will be named "untitled". If other currently opened scenes are also named "untitled", modo will append a number to the name, such as "Untitled (2)", "Untitled (3)" and so on. Use the 'Save As' option to name the scene.

Open: The 'Open' command allows users to load any previously saved modo scene files, as well as variety of other 3D object and image file formats. Scene files are added to the 'Item List', as are other 3D object formats. Images are added to the 'Images' list of the currently active scene file.

Open Recent: The 'Open Recent' command allows users directly open a file from a list of files previously opened/saved files, bypassing the 'Load Scene' window. To use Open Recent, simply go to the menu bar under "File > Open Recent" for a list ,and choose the appropriate file name from the list. If you happened to move the file to a different location and then try to use 'Open Recent', you will get an error message letting you know that file is no longer at that location. Should this occur, use the 'Open' command and navigate to the files new location. The list of Recent files are stored in the modo.cfg (aka the config file). If you delete this file, your list of recent files will also disappear.

Import: The 'Import' command will append a saved file into the current scene. Items will be added to the bottom of a scene in the 'Item List' into its own subgroup with its name corresponding to the imported scene or object.

Close Window: Dismisses the current floating window viewport.

Close Scene: The 'Close Scene' command will close the currently active (selected) scene. If the file was not recently saved, modo will also ask the user to save it. If the closed scene was the only one currently in the 'Items' list, modo will automatically create a new blank scene.

Close All: This command will close all currently open items in modo, including images and scene files. modo will default to a new empty scene. If any scene or image file has changed since the last save, a popup dialog will as if the user wishes to save recent changes.

Save: The 'Save' command will allow users to save the currently active scene as a file to disk. An operating system specific dialog box will popup if the file has not been previously saved requesting a file name as well as a location. Users may also specify a format. LXO is the native format for modo and should be used for files requiring further refinement in modo at a later time. For more information on other formats, see Import/Export.

Save As: The 'Save As" command will open a file requester allowing users to specify a file name for the currently active scene file as well as location. Users may also specify a format. LXO is the native format for modo and should be used for files requiring further refinement in modo at a later time. For more information on other formats, see Import/Export.

Save All: The 'Save All' command will save all currently opened scene and image files in one step. For scenes not previously saved, i.e. no filename or path specified, an appropriate dialog box will popup.

Save Incremental: The 'Save Incremental' command will automatically save version of the file with a numeric value appended to the end of the filename allowing users to save iterations of a scenes progress. Each time 'Save Incremental' is invoked, a new file is created with the numerical value incremented by '1'. Once saved, the file become the active file and all future 'Save' commands will save to this increment.

Export As: modo will let users save their work in several different formats by using the 'Export As' command found under the 'File' menu. Users can save as a LightWave LWO, X3D, Collada DAE, DXF, OBJ or FBX file format. This is different than using 'Save As' and choosing the format because Export will not change the current setting of the file as it is inside modo. Which is to say, when you save a model in a particular format modo will remember this and each subsequent save will be in the same format. If you save the model as an 'OBJ' and then use "Export As" to save an 'FBX' file, modo still considers the file in memory to be an 'OBJ'. Export simply allows users to quickly export in different formats without changing the internal file.

Revert: The 'Revert' command will remove the currently active scene from memory and load in the last saved version abandoning all subsequent changes made since the last time the file was saved. A popup requester will ask users to confirm this procedure because *it is not undoable*. Selecting 'Yes' will load in the previous version, and selecting 'No' will cancel the procedure.

Consolidate Scene: The 'Consolidate Scene' command will save a *copy* of the current scene, in its current location, and copy all associated linked files (such as imported images and MDD files) to a subfolder of the save location. The duplicate file will have the text '_bundled' appended to its filename. Consolidate Scene will also adjust the

internal file link locations to *relative paths*. This command will aide users in transferring files between workstations, ensuring that all the associated files are transferred with the base scene.

Edit Commands

Undo/Redo

The user can easily reverse an action by using the 'Undo' command. Undo will step back one step at a time through the sequence of undoable operations. All commands which change model state can be undone, except for special exceptions which make changes too large to be undone. Model commands which cannot be undone display a warning which must be dismissed by the user. Commands which change only the user interface state cannot be undone. They can be mixed with model commands, but they do not undo or redo as the user steps backwards and forwards through prior actions. Users may view a list of undoable commands in the 'Command History' viewport.

Undo: (Ctrl/⌘+Z) This option reverses the previous step up to the maximum number specified in the Preferences.

Redo: (Ctrl/⌘+Shift+Z) This option restores any steps that have been undone.

Clipboard

modo provides a special memory buffer for temporarily holding data referred to as the *clipboard*. A clipboard is extremely useful (nay *essential*) function when moving data around from layer to layer and scene to scene. modo adheres to the standard 'Cut', 'Copy' and 'Paste' methodologies universal to nearly all Windows and Macintosh applications. Users can use the commands to manipulate text or numerical values when working in data input fields, or with geometry when working in the GL viewports.

Elements in the clipboard cannot be saved directly, and are cleared from memory when modo quits, and are therefore not persistent across modo sessions. Text and numerical data is not application specific and can travel in and out of modo. Any geometric data stored in the clipboard is relevant only to modo. This is because each data type has its own buffer. Alphanumeric characters are copied to the system clipboard, while geometric data uses a modo specific clipboard.

The modo geometry clipboard will only store vertex and polygon data. For instance, when cut/copy/pasting edges, the result will only be the associated vertex data. When selecting elements in 'Items' mode, cut/copy/paste act as if in 'Polygons' mode and only copies the geometry. The clipboard support all associated vertex maps and tags as well, so cut/copy/paste will retain selection sets, material/part tags and when pasted

into a different scene, related material surfacing will come along for the ride as well.

To use, user need only to make a selection of some type and then invoke the command form the menu bar under "Edit", or by using the associated keyboard shortcut.

Cut: (Ctrl/⌘+X) This option removes the selected element and stores it in the *clipboard*.

Copy: (Ctrl/⌘+C) This option copies the selected element, leaving the original in place and stores it in the *clipboard*.

Paste: (Ctrl/⌘+V) This option inserts the contents of the *clipboard* into the target layer or field.

TIP: When managing large scene files, or files with multiple layers, it's best to use the 'Import' option instead.

Delete

The Delete command applies to any type of selected data. For geometric data this is exactly like the *cut* command removing the data except that it is not *moved* to the clipboard. Deletion of vertices removes the vertices, but will leave any remaining vertices/polygons (if applicable). If all the vertices of a polygon are deleted, then the polygon itself will be removed. Deletion of polygons also deletes the associated vertices if they are not needed for any other polygons that stays behind. Deleting an edge removes any adjoining polygons leaving a hole. When in Items mode, deleting an item removes the layer, including all geometry contained within from the Item List. When text is selected in an input field, delete will remove the selected section.

Delete: (Delete key) Deletes the selected element from the scene entirely.

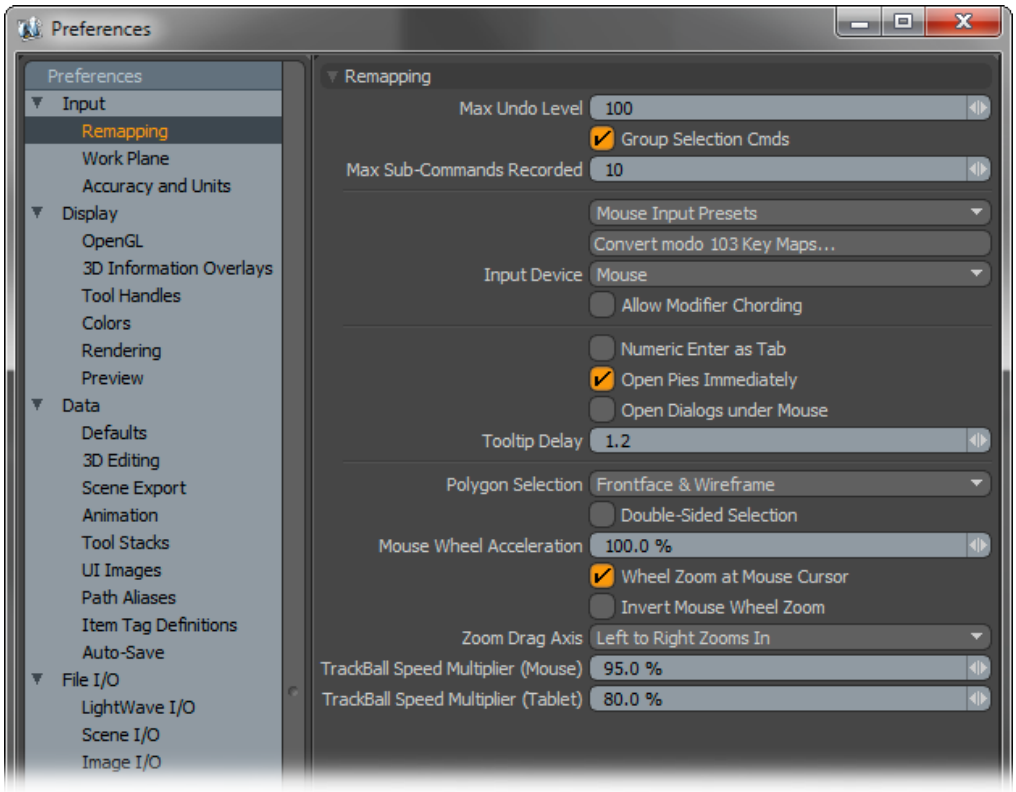
Remove

Remove is effectively the same function as 'Delete', however the one notable difference is that when working with edges, removing them eliminates the edge itself, but leaves the neighboring polygons in place essentially resulting in a polygon *merge*, joining the neighboring polygons.

Remove: (Backspace key) Removes the selected element from the scene.

Preferences - Input

Accessible from the menu bar under "System > Preferences...", the modo *preferences* define the default behavior for a variety of options and functions in modo, further customizing the application to suit users particular tastes and needs. The Preferences are categorized into four main functions- Input deals with ways users interact with the application, 'Display' controls the various way viewports display geometry and information. The 'Data' section controls default behaviors for data created in modo and finally 'File I/O' controls how scenes are imported and exported when dealing with external applications.



Remapping--

Max Undo Levels: The undo history is viewable within the 'Command History' viewport. The 'Max Undo Levels' controls the maximum number of steps backwards users can go in the history. The default value of '100' is generally very good at allowing users leeway to step backwards through the history. Increasing the value will allow users to go further back in time, so to speak, but will also increase memory usage. For memory intensive scenes, it may be beneficial to reduce this value to reduce memory overhead.

Group Selection Cmds: When this option is *enabled*, modo will group multiple consecutive selection actions into a single undoable action. The grouping size is determined by the 'Max Sub-Commands Recorded' Value.

Max Sub-Commands Recorded: This option determines the maximum number of selection commands that can be grouped into a single undoable action. 'Group Selection Commands'

Mouse Input Presets: Users coming to modo who are familiar with another 3D application might find themselves unable to switch into modo mode. The 'Mouse Input Presets' are meant to adjust the mouse navigation controls to mimic those of the chosen application reducing users navigational stumbling, making it easier for switching back and forth between the other program. A variety of options are available representing a number of popular 3D programs.

Convert modo 103 Key Maps: This legacy option converts modo 103 keymaps to the current version. User would need to load v103 keymap configs using the 'Config Import' function found in the file menu prior to running this option.

Input Device: This option determines how modo interprets input commands, users can choose either a 'Mouse' or 'Tablet' as the main form of input. Tablets typically use an absolute positioning based on a stylus' position on the tablet surface, whereas a Mouse uses a relative position based only on the initial position of the cursor. Users using a Tablet with relative positioning should set this option to 'Mouse'.

Allow Modifier Chording: When enabled, modifier keys can trigger 'Chording', when disabled, only mouse buttons can trigger chording. Chording is the use of multiple mouse buttons simultaneously. It should be noted that chording has been widely recognized as a key contributor to carpal tunnel syndrome and other repeated stress injuries. Luxology recommends that users avoid chording to prevent unnecessary muscle strain.

Numeric Enter as Tab: When enabled, pressing the 'Enter' key on the numeric keypad will function the same way as pressing the Tab key does, by advancing the cursor to the next logical data field.

Open Pies Immediately: This option determines weather there is a slight delay to the opening of Pie Menus when disabled, or if the menus open immediately when enabled. For users not accustomed to Pie Menus, the delay may be helpful in eliminating menus from popping open unexpectedly on errant key presses.

Open Dialogues Under Mouse: This option forces new dialog boxes to always open at the current mouse position (such as the 'Save As' and 'Open' file dialogues). When disabled, dialog boxes will open in the same position as the last open dialog box.

Tooltip Delay: Defined in seconds, this option determines the length of time before the tooltip info display opens when hovering the mouse pointer over a button or function of modo. The default value is 1.2 seconds.

Polygon Selection: This option determines the behavior of lasso or paint selection modes when selecting components in the OpenGL viewports.

Frontface & Wireframe- Default behavior is only forward facing polygons are selected, except when in Wireframe mode where both the front and back facing polygons are selected.

Frontface & Backface- Default behavior is front and back facers are always selected regardless of viewport style.

Frontface only- Default behavior is to select only front facing geometry regardless of viewport style.

Always Raycast- Probably means something important

Double Sided Selection: When this option is enabled, any polygons tagged with a material that is defined as 'Double Sided', users can then paint select these polygons from their back side. When disabled, users are unable to paint select polygons from the back.

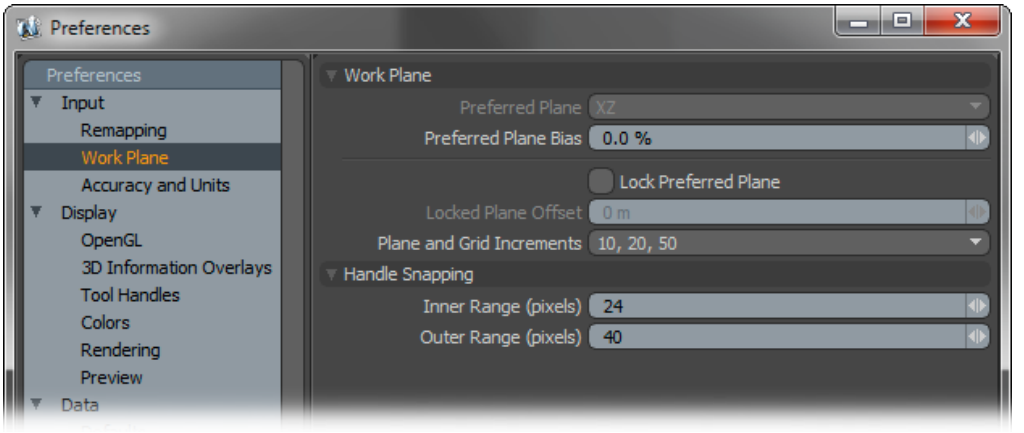
Mouse Wheel Acceleration: This percentage value determines the speed at which modo zooms in or out when using a mouse's scroll wheel. Higher values will zoom faster, while lower values will zoom more slowly.

Wheel Zoom at Mouse Cursor: When this option is enabled, zooming in any OpenGL viewport with the mouse scroll wheel will center the zoom based on the mouse's position over the viewport, when disabled, the mouse's position is disregarded and the viewport will zoom based on its center.

Invert Mouse Wheel Zoom: This option inverts the zooming behavior of the mouse wheel. Pushing forward on the wheel will push out instead of in.

Zoom Drag Axis: This option determines the directional behavior when hauling in a 3D viewport when zooming. The options are 'Left to Right', 'Right to Left', 'Bottom to Top' and 'Top to Bottom'.

Trackball Speed Multiplier: This option determines the rotation speed for using Trackball rotation when navigate in an OpenGL viewport. There are separate settings for both 'Mouse' and 'Tablets'.



Workplane--

Users can favor certain planes when using the Work Plane, especially in instances where users don't want to lock the plane, but it would be helpful to have it face a certain direction, such as modeling a rough city, users could rotate the views but keep the Work Plane mostly facing upwards (like it was the ground) when creating new geometry.

Preferred Plane: When the 'Preferred Plane Bias option is set above 0%, uses can choose a preferred Work Plane. Based on the bias amount, modo will attempt to keep the workplane positioned at the preferred plane determined by its two major axes.

Preferred Plane Bias: When working with the Work Plane, modo tries to keep the plane perpendicular to the viewport window when navigating and rotating the view. Increasing the Plane Bias option will increase modo's favor of the Preferred Plane. Setting a very high value means modo would almost always favor the preferred plane.

Lock Preferred Plane: Users may also prefer to lock the Work Plane to a specific plane facing direction by enabling this option. Once fixed, rotating the viewport will have no effect on the position or angle of the workplane.

Locked Plane Offset: When the Work Plane is locked, users can additionally set an offset (from the origin), providing accurate control over the position and placement of the workplane when used as a construction plane.

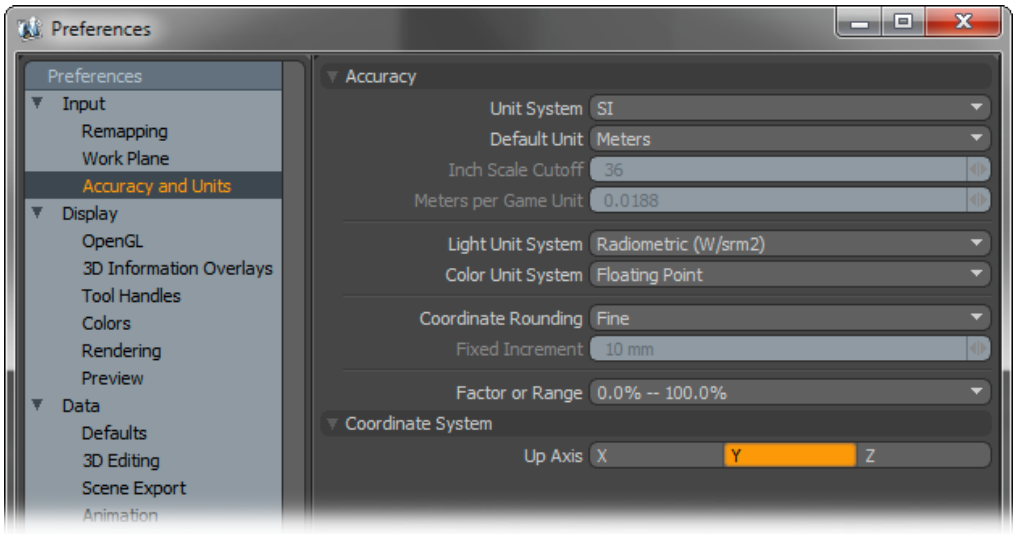
Plane and Grid Increments: As users zoom in and out of a scene, the Work Plane grid adjusts the density of the grid divisions adding or removing subdivisions based on the Plane Increments. Users can modify this behavior using this option.

Handle Snapping--

These are the default values, in pixels, for the distance from a snapping target where the cursor first gets snapped.

Inner Range: Determines the actual snapping range.

Outer Range: Determines the range to highlighting elements for snapping to.



Accuracy--

Unit System: Users can choose their preferred measurement units system based on several options. Once set, this is the measurement input method modo will use for all numerical input values-

SI- The International System of Measurement (abbreviated 'SI') is the modern form of Metric, complies to universal base units.

Metric- A universal system of measurement based on powers of '10' -millimeter, centimeter, meter, kilometer.

English- A historical measurement system based on Imperial Units -mils, inch, foot, yard and mile

Game Units- An arbitrary unit of measurement (defined by the 'Meters per Game Unit' setting).

Unitless- An arbitrary decimalized unit of measurement based on cubic meters, essentially 1 unit = 1m.

Default Unit: This option is dependant on the 'Unit System' and determines the default unit used when no unit is specified.

Inch Scale Cutoff: When the 'English' Unit System is selected, the 'Inch Scale Cutoff' determines the cutoff where modo will display strictly inches as feet and inches. For example, when set to the default 36, 27 inches will display as 27", but 42 inches will display as 3' 6".

Meters per Game Unit: When utilizing the arbitrary 'Game Units' systems, users can use this value to determine the scale of a single unit. This allows users to work with even whole numbers. So if a 1m equivalent is really 1.375 real world meters in modo, users an simply enter 1.375 here and then specify units normally in the numerical input fields.

Light Unit System: This option determines the default Light Unit modo uses when specifying a lights brightness-

Radiometric- Measurement of the electromagnetic spectrum (which included visible light); defined by power of radiation.

Photometric- Measurement of brightness of light as perceived by human eye; defined by luminous intensity.

Color Unit System: Determines the color unit system used for specifying colors in modo-

Floating Point- Determines colors based on decimals, uses 0-1 range (and beyond for HDR colors).

Percentage- Determines colors based on percentages, uses 0%-100% range.

Integer- Determines colors based on 8bit value scale, uses 0-255 range (2 to 8th power = 256 values)

Hexadecimal- A computer numerical system similar to binary represented by 16 characters. 2bit Hex produces 256 values, similar to Integer color units. Widely used in specifying color on the web, uses 00.00.00-ff.ff.ff range.

Coordinate Rounding: This option controls how mouse input is converted, via the Work Plane, into 3D coordinates.

None-- This means that no coordinate rounding is done. Every mouse move gives unclamped coordinates (typically with lots of decimals). This option is basically the raw 2D -> 3D transform. Useful for working freehand.

Normal-- This option attempts to give clean, round coordinates based on the users view transform; so as the mouse moves, users can see values that are nice in your

current unit system display in the information tab. The step size will get smaller or larger as the user zooms in or out. Users may need to move the mouse cursor 2-3 pixels to see values update.

Fine-- This option is similar to 'Normal' but optimizes for closer to one step of coordinate rounding from one pixel of mouse movement. Gives the user finer grained input, but can be difficult to hit exact values.

Fixed-- This option uses the 'Fixed Increment' preference to put a lower limit on both coordinate rounding and the grid. If the user sets the fixed increment to 10mm that means the grid will never get finer than 10mm, and all input will be rounded to the nearest 10mm even at high zoom. When zoomed out, however, the grid will show larger values but the step size will always be a multiple of the fixed increment.

Forced Fixed-- This option is similar to 'Fixed', but forces the size of the grid and the input step to match the increment exactly no matter the zoom level.

Fixed Increment: When the Coordinate Rounding is set to either 'Fixed' options, this value determines the Fixed coordinate rounding grid.

Factor or Range: This option determines surfacing and material value units, users can specify values as a percentage 0%-100% or as a float value 0.0-1.0

Coordinate System--

Up Axis: This option determines the major axis that is considered the default up direction for modo. 3D programs typically use 'Y' as up, where CAD applications typically use 'Z' as the up direction. When importing geometry, if it is always 90° off, changing the up axis to match the originating application may resolve the issue. This can also be set on a per scene basis in the 'Scene Item'.

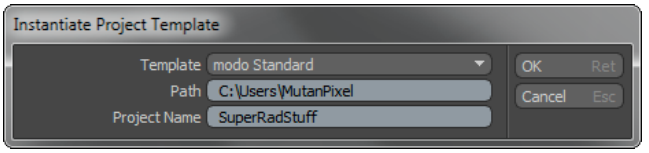
Project Directories

modo supports the use of project directories, a hierarchical arrangement of folders for related files, meant to ease the management and transfer of complex projects. By using the Project Directory commands in modo, users who adhere strictly the generated directory structure will be able to move projects freely from system to system, and platform to platform without fear of missing files or lost link associations. The setup and use of Project directories has been designed to be as transparent and intuitive as possible within modo. The basic methodology for Project Directories, is to first define the project and its folder structure (via the 'New Project' command) prior to saving a scene. Then when saving the scene, modo will default to the 'Scenes' folder within the project, images will save to the 'Images' folder, and so forth. To load a previously saved project, user need only to specify the project directory using the

'Choose Project' command, and then open (with the normal 'Open' command) any files requiring further editing or rendering; modo bases the default paths off those specified by the project file (saved into the project folder when created).

New Project

To work with a project directory, user first need to generate the directory structure, this is done with the "New Project' command found in the menu bar under "File > New Project...". Invoking this command opens an operating system specific file browsing window. Within the window, users can browse to any directory available to the system. LMB+click 'OK" once the appropriate directory is selected and modo opens a modo specific dialog box.



Within this window, users can modify the 'Path' if necessary (this is the path as specified in the previous window) and set the 'Project Name'. The project name becomes the root project directory folder. LMB+clicking 'OK' generates the default folders within the specified location, and sets modo to use the new created structure. Once a Project is specified as such, modo's 'Save As' dialogues will all default to the appropriate project directory folder.

Choose Project

In order to open or utilize a previous or transferred modo project, users need only to place the files into the appropriate location on their hard drive. In modo, users can use the menu bar command "File > Choose Project". This opens an operating system specific file browser, where users can navigate to the saved project, and select the root folder (the folder based on the originally specified 'Project Name'). LMB+click 'OK' to set the project, now when opening any project files, modo will adhere to the paths specified in the luxproject config file automatically saved into the project directory.

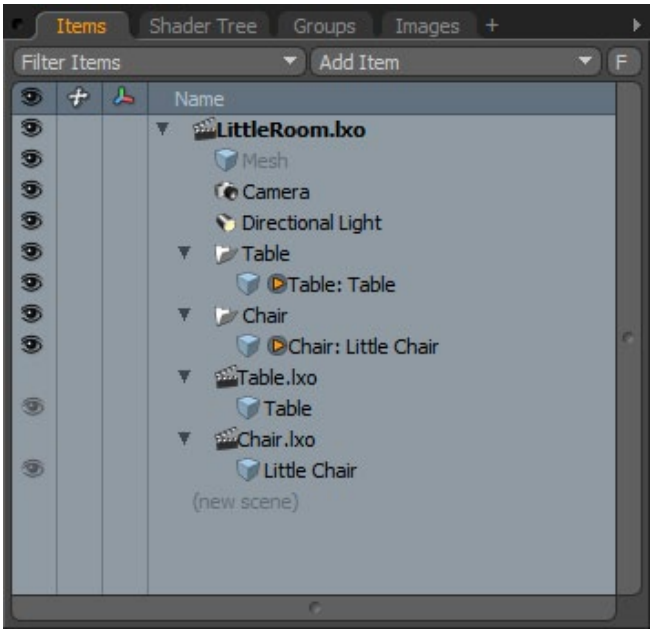
Import Reference

The Import Reference option from the menu bar under "File >Import Reference..." allows users to add items to a scene that is referenced from a separate file saved on disk. The advantage to reference files is that this allows the inclusion of data sets into a project that will be updated individually. When the master file is loaded again the updated reference file will automatically be included. Consider this simple example.

You wish to construct a scene that contains two key elements, a table and chair for instance. By building each element and saving them separately as Table.LXO and Chair.LXO you can then open a new file and import each as references. This provides extended flexibility in the master scene as well as for using those elements in future projects.

For maximum flexibility, files imported as references will contain all items from the original file, including mesh, camera, and texturing items. While the most obvious use for Reference files is to import mesh components, users can use Reference importing to bring dedicated lighting setups or entire project files with their meshes, lights, cameras and textures. In the table and chair example we only want each reference to contain the mesh and texture items. For this reason we would save mesh only versions of the files so that the master file does not contain lighting and cameras from each of the reference files.

When files are imported into a project as references the reference item can not be edited directly at the component level. The reference can have item level transforms as well as local material and texture overrides when in the master file. If users wish to perform component level edits on the geometry, they must be edited from the original file, which can be quickly selected from the Item List.



The master project file is "LittleRoom.lxo" at the top of the Item List. Inside this project file you can see the two referenced files, 'Table' and 'Chair'. Upon import modo creates Group items to hold and organize the item(s) from the reference file. The reference

items can also be easily identified by the reference icon that appears just to the left of the item name.

At the bottom of the Item list users can see the original 'Chair.lxo' and 'Table.lxo'. To perform component level edits simply LMB+click on the file there and modo will *isolate* that portion of the master file for specific edits. Any edits done on the file will propagate into the reference version inside the master as well. Of course, users can also, at any time load the original reference file directly into modo for editing. It is safe to delete items from a reference item group such as lights, cameras and meshes. This is treated as a *local override* and will not effect the actual file(s) on disc, rather the master scene file will only load the required data upon next launch.

Import/Export

One of the original axioms of modo has always been "works well with others". Users rarely work in a single application, so it is important to be a good citizen in the 3D world by supporting standards. modo supports a wide range of file formats for transferring data around. Users can either read the various formats by either opening them directly "File > Open" and selecting the proper file, or by importing "File > Import...". The difference being that opening files makes them their own 'Scene', and importing merges the saved file into the current scene. Users can save out the various formats using the 'Save As' command "File > Save As" and specifying the intended format, or by exporting "File > Export", the difference between them being that saving the file as a specific format replaces whatever is in memory, and subsequent saves will continue to overwrite the saved file in the same format (as one would expect). Exporting keeps the same scene in memory, only writing the selected format to disk.

The supported formats include--

LightWave Object (.LWO)

modo reads and writes LWO2 format for geometry, supporting a full array of features including UV, Weight, Morph and Color vertex maps. Equivalent surfacing data is translated as well, when applicable. Unsupported information is encapsulated and retained, as long as users *remain* in the LWO format, saving to any other format discard the extra data. The loading and saving of LightWave's *scene* files is not currently supported.

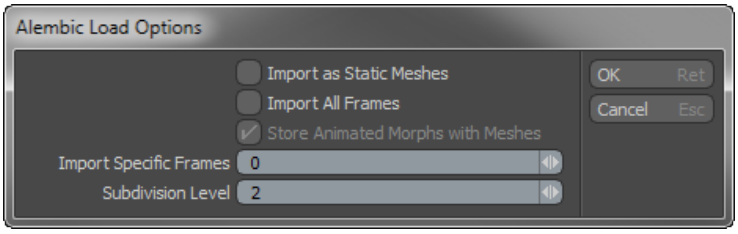
Wavefront Object (.OBJ)

OBJ is a file format developed by Wavefront Technologies for its Advanced Visualizer software package. The file format being open has been adopted by many other 3D

applications because of its simplicity to implement, and for the most part has become a universal exchange format. OBJ supports geometry, layers and simple surfacing. Vertex normal maps are also supported. The OBJ format provides for only a single UV map per object, and all vertices must be mapped (no partial UVs are allowed and will be ignored on export). modo will both load and save OBJ format.

Alembic (.ABC)

Established by Sony Imageworks and ILM, Alembic is an open computer graphics interchange framework. Alembic distills complex, animated scenes into a non-procedural, application-independent set of baked geometric results. The format currently supports the import and export of both static (non-moving) and animated meshes (including subdivision surfaces) and camera data. While the format does not directly support material information, part and material tags are exchanged with the mesh data. When importing files, users are presented with the following dialogue-



Import as Static Meshes: Converts the loaded scene items to the more efficient, though un-editable 'Static Mesh' item type. This does not apply to items with deformations.

Import all Frames: When loading an animated scene, load the transformations and deformations for the entire sequence. If off, load the scene for the frame specified in "Import Specific Frames"

Store Animated Morphs with Meshes: If *enabled*, will load all the mesh deformations into the scene file directly. If *disabled*, the deformations will be referenced from the original Alembic file, and loaded as necessary.

Import Specific Frames: Loads the scene with the transformations & deformations for the defined frame(s) only.

Subdivision Level: When loading subdivision meshes, this value defines their current subdivisions level amount.

McNeel Rhino (.3DM)

Rhino is a NURBS based 3D modeling application. Rhino has the ability to internally convert its curve based NURBS models into a triangulated polygon model. modo will

read the triangulated polygon data from any Rhino 3DM format file (modo cannot read NURBS data directly). modo will both Read and Write to the 3DM file format.

COLLADA (.DAE)

COLLADA is a modern open source 3D interchange format managed by the Khronos Group. The modo COLLADA exporter can read and write geometry with normals and multiple UV texture coordinate sets, materials, lights, cameras, effects, and transform animations, all using a *common* technique that is compatible with a wide variety of third-party tools. COLLADA is the most robust of the exchange formats, and if supported by the target application, it is the recommended format.

Autodesk fbx (.FBX)

FBX is originally a file format developed by Kaydara for their FilmBox application, now owned by Autodesk. It has been adopted as a universal exchange format between digital content creation applications because of it's robust support of object and animation data. modo reads and writes the libraries supplied by Autodesk, that support interchange of geometry, lights, cameras, basic material and surfacing, UVs and transform animations.

Autodesk 3DStudio (.3DS)

3DS is the original native file format used for the DOS version of 3D Studio (precursor to MAX). modo can read geometry data stored in this format.

Autodesk dxf (.DXF)

DXF is a simplistic geometry format developed for the interchange of data between CAD applications. DXF does not support , surfacing, material, vertex maps or animation. modo both reads and writes this format.

Web3D Standard (.X3D)

X3D is an XML based 3D standard and is the successor to VRML on the internet, it support geometry and basic surfacing. modo is only able to write out to this format.

VideoScape (.GEO)

Aegis VideoScape was a (very) early 3D rendering and modeling program on the Amiga (precursor to LightWave). Users can both read and write geometry into this legacy format, though it is most useful for reading as very few modern programs support the format.

Encapsulated PostScript (.EPS)

EPS files are a 2D vector format developed by Adobe for exchange of vector drawing information created in their Illustrator application. Though generally used for print applications, EPS files are imported into modo as *Curves*, where users may freeze them into faces for further manipulation, or render the curves directly, or use them as a basis for patch modeling. The only way to get EPS file *out* of modo is by way of the Export UVs to EPS command in the menu bar "Texture > Export UVs to EPS".

HPGL Plotter File (.PLT)

PLT is a 2D vector-based plotter format originally devised for use with AutoCAD. modo only exports to this format.

Bitmap Image Formats

Bitmap images are an essential component of texturing, painting and image based sculpting. They are also useful as backdrops for modeling, providing reference for shape and scale when modeling. They can also be used to literally light the scene, where inofrmation captured in the image can illuminate rendered images. With so many different uses, it is necessary to support a great number of image formats.

There are several ways users can load images into modo. One is from the Images viewport, typically located near the 'Items List' and 'Shader Tree' viewports of the default interface. Using the 'Add Clips' button in the 'Images' viewport, users can open a OS specific dialog box to locate and select an image for loading. Once loaded, images in the 'Clips' list will appear throughout the application in applicable menus. Users can also load images by simply dragging and dropping them from the operating system onto modo. When users drag and drop images, if dropping them onto geometry they will appear as a texture in a material. If dragged into a 3D viewport, they will appear as a Backdrop Item. When loading a bitmap, that is not applied to geometry or a Backdrop Item, drag it into a non-3D viewport. Items can be dragged from the 3D views directly onto groups (where they become members of the group) or into the Schematic viewport. Users can also load image directly from any menu for selecting images, using the 'Add Image' command present in the menu. Once an image is loaded, users can reference the specific documentation for a particular function for additional information on applying image.

Read Formats

This is a listing of the image map formats modo can read and use for texturing, backdrops, brushes, and variety of other functions.

Format Name	Extension	Bit Depth per channel	Compressed	Type
Flexible Precision Image Format	.FLX	HDR	Yes	Lossless
Targa	.TGA	8 bit + Alpha	No	
Windows BMP	.BMP	8 bit + Alpha	No	
Graphic Interchange Format	.GIF	256 colors	Yes	Palette
Radiance High Dynamic Range	.HDR	HDR	Yes	Lossless
JPEG 2000	.JP2	8 bit	Yes	Lossy
JPEG 2000 16-bit	.JP2	16 bit	Yes	Lossy
JPEG 2000 16-bit Lossless	.JP2	16 bit	Yes	Lossless
JPEG	.JPG	8 bit	Yes	Lossy
Portable Network Graphic	.PNG	8 bit + Alpha	Yes	Lossless
Portable Network Graphic 16 bit	.PNG	16 bit + Alpha	Yes	Lossless
PSD (Photoshop Document)	.PSD	8 bit	No	
SGI RGB	.SGI	8 bit	No	
Tagged Image File Format	.TIF	8 bit + Alpha	Yes	Lossless
16-Bit TIF	.TIF	16 bit + Alpha	Yes	Lossless
16-Bit Uncompressed TIF	.TIF	16 bit + Alpha	No	
Open EXR	.EXR	HDR	Yes/No	
Zoomify (large image support)*	.XML	8 bit	Yes	Lossless

*Zoomify is a new format for supporting large images based on smaller tiles of JPEG images. modo only loads the tiles necessary to render or display, meaning users can apply massive textures, with little affect on system performance. Users can export Zoomify formatted images using the "FIle > Export > Zoomify" function in Photoshop (CS4/5 only) or users can download 'Zoomify Express' for free at Zoomify.com.

Write Formats

This is a listing of formats modo writes out when creating new images.

Format Name	Extension	Bit Depth per channel	Compressed	Type
Flexible Precision Image Format	.FLX	HDR	Yes	Lossless
Targa	.TGA	8 bit + Alpha	No	
Windows BMP	.BMP	8 bit + Alpha	No	
Radiance High Dynamic Range	.HDR	HDR	Yes	Lossless
JPEG 2000	.JP2	8 bit	Yes	Lossy
JPEG 2000 16-bit	.JP2	16 bit	Yes	Lossy*
JPEG 2000 16-bit Lossless	.JP2	16 bit	Yes	Lossless
JPEG	.JPG	8 bit	Yes	Lossy*
Portable Network Graphic	.PNG	8 bit + Alpha	Yes	Lossless*
Portable Network Graphic 16 bit	.PNG	16 bit + Alpha	Yes	Lossless*
PSD (Photoshop Document)	.PSD	8 bit	No	
SGI RGB	.SGI	8 bit	No	
Tagged Image File Format	.TIF	8 bit + Alpha	Yes	Lossless
16-Bit TIF	.TIF	16 bit + Alpha	Yes	Lossless
16-Bit Uncompressed TIF	.TIF	16 bit + Alpha	No	
Open EXR (+layers)	.EXR	HDR	Preference	Lossless*

*A modo preference option provides for users to determine the amount of compression -- go to "System > Preferences" then "File I/O > Image I/O".

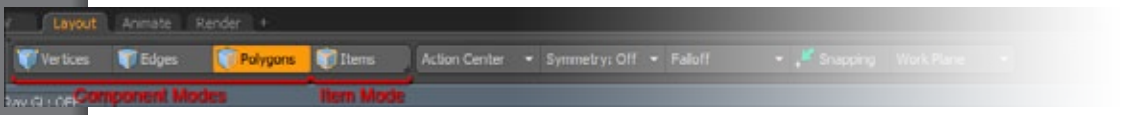
The modo Toolbox

The sheer number of modeling tools and options available may seem overwhelming at first, but to aide in users understanding they are logically grouped by function. Once users grasp the overall utility of each group, its easier to appreciate the subtleties of the individual tools. 'Modeling Primer' provides and overview of working with modo's toolset, while the following reference sections detail the operation and application of the tools and modeling functions individually.

Modeling with a modo Mindset

modo is a polygon modeling tool

For all intents and purposes modeling wise, modo is a polygon modeling tool. Yes, other options are provided for generating forms, such as spline and bezier curve patching, or working with different types of subdivision surfaces, but the ultimate goal of those operations is to eventually create polygons for rendering or for export to their ultimate destination. modo provides a variety of ways to manipulate and shape those polygons into the desired shape. Some tools manipulate polygons directly, such as the transform and deform tools, others provide means to manipulate polygons indirectly, such as the sculpting of subdivision surface models with a displacement map. There are as many methodologies to reaching a finished result as there are users, everyone has their own unique approach, but of the various styles, there are some basics to understand for modo's way of working that will ease the overall learning curve allowing each user to reach their goal and find their own individual style.



Components versus Items

One of the areas that is often confusing to new users of modo are the various selection modes. When modeling, users work in the three component modes -'Vertices', 'Edges' and 'Polygons', this is where the actual geometry is created and shaped. 'Items' mode is useful for positioning modeled object in a scene for final render and is a necessity of Animation. When in any individual selection *mode*, only selections of that type can be made. The important distinction is that when editing an *Item*, users are editing the layer itself (think of it like a container), while editing a component is editing the contents of the layer (what's inside the container). Modifications made in the Vertex, Edge and Polygon modes happen *around the center point*, while transforms applied while in Item mode *move the entire item* including its center point.

Interacting with modo

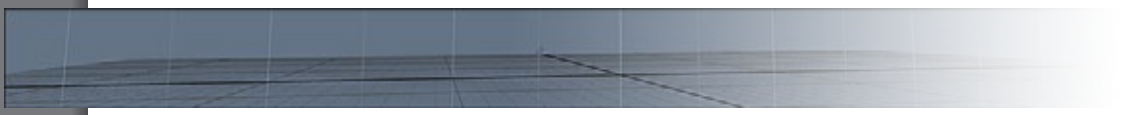
Interacting with modo itself is done mostly through the manipulation of the mouse, taking advantage of all three buttons and the scroll wheel (some may prefer a *trackball* or *tablet*, both typically have functions that can be remapped to mimic the options available on a mouse). Mouse button activity is supplemented by keyboard commands, which are at times used in unison with the mouse. This can take some getting used to. (see Mousing Standards for more information) Given that modo is a 3D application that users interact with in a 2D fashion, this may at times feel awkward, but produces the most predictable results given today's technology (Hey, where's my 3D monitor and force feedback gloves?).

Users are encouraged to read through the interface section of the documentation; familiarizing oneself with the many viewports, tabs and panels will help in working in modo. Nothing kills the creative impulse more quickly than fruitlessly searching for some often used tool or function. The organization of modo is designed to reduce this as much as possible. Main functions are organized as Tabs across the top, such as Modeling, Sculpting or Rendering. When in any of these areas, only the tools necessary for the task at hand are provided, streamlining the process. This way of working also makes it easy to jump in and out of tasks easily, providing for a comfortable non-linear workflow. Users can model, render, animate and then model some more with very little worry that one process will have a negative affect on the others. In fact, a non-linear workflow is encouraged in modo. Users *should be* rendering and tweaking along the way toward the final result. This is meant to eliminate a lot of the guess work and surprises traditionally associated with creating in '3D'.

Using Tools

To perform an operation, users must left mouse button (LMB) click on the tool icon or menu option. For one shot commands, such as 'Delete', 'Fit' or 'Hide', that is all that is necessary to produce a result. For interactive tools, like 'Move', 'Rotate' and 'Bevel', this makes the tool active and draws the appropriate tool handle within the 3D viewport, displaying the necessary Property options. To produce a result users will need to either 'Haul' somewhere in the viewport (hauling is the act of LMB+click dragging the mouse pointer left/right or up/down within the viewport with an active tool and can only be done while a tools is active) or hover the mouse pointer over the tools handles and then LMB+click and drag when a single handle turns a yellow color,

constraining the action to a specific axis. Additionally, users can interact by changing values in the properties form, by default under the toolbox. Modifying values won't affect what's displayed in the viewport unless the user LMB+clicks somewhere in the 3D view to activate interactive mode for the tool, then modification to attribute values will update in real time in the viewport providing interactive feedback. When the desired result for a tool is achieved, users can simply drop the tool by pressing the 'Esc' or 'Q' keys to finalize the application of it. Once a tool is dropped, interactive editing of tool attributes is no longer possible.



Work Plane

Users should understand that actions don't arbitrarily happen in 3D space within a viewport. For example, when creating points, the individual vertices are created at the intersection of the 'Work Plane' and where the user clicks in the viewport, providing predictable and repeatable results. Understanding the workplane and its functions is important to understanding how modo works. The Work Plane is the light gray grid that is visible in any viewport. In a perspective view, users may notice that the grid will change position; this is the Work Plane orienting itself toward the user, trying to stay as close as possible to parallel with the viewport window, while remaining aligned to the nearest axis. Users can also modify the Work Plane to a fixed arbitrary position, based on a component selection. This temporarily orients the entire modo universe to align with that selection, so all functions now work in relation to the Work Plane making it easier to perform task that would be very difficult otherwise. For more information regarding using the Work Plane, please reference that page of the documentation.

Action Centers

When performing any action on a selection, the position of the center of origin and the axis orientation of the tool handle itself has a dramatic effect over the result when applied. Yes, users can orient the Work Plane to suit many situations, but constantly modifying the Work Plane would get tedious, so modo has what are called 'Action Centers', automatic combinations of center positions and axis directions for

tools. There are a variety of options, based on different criteria; some are oriented to selections, some to a viewports orientation, while others are based on fixed positions, like the items center (0,0,0). Action Centers often packaged with tools to provide specific functions. Take for instance the Flex tool, this combines a rotation tool, with an action center that orients itself automatically to the selection border making it easy to pose characters. Understanding Action centers are necessary to a fluid experience with modo, it is recommended users reference the Action Center page for more information on working with them.

Falloffs

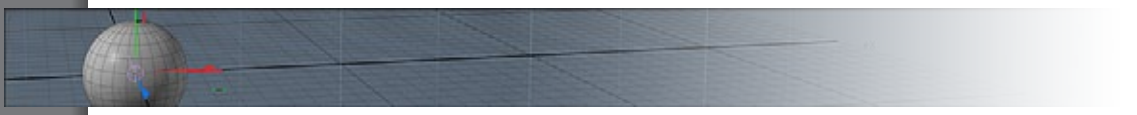
Ordinarily, when performing an action on a selection, the application is uniform across the selection area. Falloffs provide a means to vary the strength of the action across a given area. Once activated, all of modo's tools respect the falloffs and apply themselves appropriately, opening up a variety of option that would be incredibly difficulty to reproduce otherwise. When enabled, a Falloff will *snap* to a selection when a tool is active, scaling to fit within the selections bounding box making them incredibly easy to apply and use. Falloffs can also be packaged along with tools, and often are. They are the basis of the many deformation tools - Rotate combined with a linear falloff produces the *Twist* tool, Move combined with the same falloff produces the *Shear* effect. Falloffs may appear simple, but wield an incredible amount of power when working in modo. For more information on using and applying Falloffs, please reference that section of the documentation.

History and modo

When users apply a function that produces undesirable results (or they simply chang their mind), previous commands can be un-done by way of the command history list, a listing of all the functions applied in a scene. Pressing the Ctrl/⌘+Z steps users back through the command history , up to the maximum number of steps defined by the preference setting (100 by default). The modo Command History is held in a buffer while modo is open, but is not saved with the specific scene. Once a scene is closed,

the Command History is cleared, freeing the memory for the next operation. For more on using the Command History, please reference that page of the documentation.

Mousing Standards



modo employs a great number of keyboard based shortcut keys, despite this fact, users will still largely interact with the application by way of the mouse (or pen and tablet if that's what is preferred). When working in the various 3D viewports, modo has two main styles of mouse input, 'hauling' and 'handle manipulation'. Hauling is when users press the left mouse button (referred in these docs as LMB) and click holding down the button, and then drag anywhere within a 3D viewport while a tool is active (meaning *selected*, activated tools will be highlighted in the toolbox). Handle manipulation is is when users hover over a tools manipulator handle, which is present in the viewport when a tool is active, this action highlights the handle yellow wherein user may then LMB+click and drag the handle, constraining the action to the particular function the handle represents.

To make users workflow more fluid, modo employs a set of standards for all mouse behaviors depending on the type of tool. These standards allow us to leverage all three mouse buttons and qualifier key combinations to produce variations on tool behaviors. There are several basic types of tools - geometry creation tools, geometry editing (transform) tools and curve creation tools. The standards for mousing are slightly different between these categories so that we have mapped the most commonly desired behaviors to the primary mouse inputs for each style of tool. When you activate a tool in modo you can always see the mousing behaviors listed at the bottom of the modo interface in the Tool Tips viewport.

The Status Bar appears at the bottom of the default interface providing feedback on the current tool.

The basic rules of mousing are quite simple. Understanding these rules will allow you to take full advantage of the flexibility and speed of the tools. The following tables quickly illustrate the various modes, followed by detailed descriptions and examples of each operation type.

Transform Tools

LMB	MMB	RMB	Shift+LMB	Ctrl+LMB
Reset tool and haul parameters	Clone and haul parameters	None (reserved for falloff modifiers)	Continue hauling parameters w/o reset	Constrain parameters

Reset tool and haul parameters.
Definition: At the mouse click the tool properties are reset. As the mouse is dragged the values are changed.

Example: When using the Axis Rotate tool with the Automatic Action Center, clicking in the 3D viewport sets the pivot position for the rotation. LMB click and drag will reset the tool properties to 0° and set the pivot for the rotation under the mouse. Dragging will then change the amount of rotation.

Clone and haul parameters
Definition: Re-applies the current/last properties of the tool and allows continued editing from those values.

Example: The Axis Rotate tool has been used to rotate the selection by 45 degrees. Using clone and haul parameters will reapply that rotation so the selection will then be at 90 degrees from its' original position. The tool widget will still be at 45 degrees since the value was cloned. You are now able to continue hauling to edit the value.

None
Definition: The right mouse button when transforming geometry has no direct action. This option is reserved for setting the action center and defining the range of an active falloff.

Continue hauling parameters without reset.
Definition: The tool properties are edited from the current value without resetting any of the tools parameters or action center info.

Example: When using Axis Rotate, this mode will leave the current rotation amount in place as well as the rotate tools pivot. Dragging the mouse in this mode will simply increase or decrease the rotation value.

Constrain parameters
Definition: Uses initial mouse drag to determine which properties to edit and constrains the mouse to only edit those values.

Example: When hauling the Move tool, two properties are edited simultaneously

depending on the current Work Plane. If you are editing on the XY plane, hauling will edit both the X and Y move value. Using Constrain parameters, the first direction the mouse is moved determines the editable axis. In this case, holding CTRL and dragging to the right will edit the X move value and the Y value will remain unchanged.

Geometry Creation Tools

LMB	MMB	RMB	Shift+LMB	Ctrl+LMB
Haul parameters	Clone and haul parameters	Haul alternate parameters	Reset tool and haul parameters	Constrain parameters

Haul parameters
Definition: The tool properties are edited from the current value without resetting any of the tools parameters or action center info.

Example: Using the Polygon Bevel tool, the first click activates and allows you to haul both the inset and shift values freely. Subsequent clicks will simply add to the existing edit values. This enables you to click and drag repeatedly on a single bevel to edit the values.

Clone and haul parameters
Definition: Re-applies the current properties of the tool and allows continued editing from those values.

Example: The polygon bevel tool can be applied repeatedly with the same inset and shift properties by using Clone and haul parameters without additional dragging.

Haul alternate parameters
Definition: Some creation tools provide hauling edits for secondary properties via the RMB.

Example: The Cube primitive tool allows the segment value to be edited by dragging the RMB while the tool is active. The axis is determined by the orientation of the work plane.

Reset Tool and Haul Parameters
Definition: Any current tool action is applied, the tool properties are reset and a new application of the tool begins.

Example: When using the polygon bevel tool this behavior allows you to begin a new bevel and immediately haul the inset and shift values. With this method you can quickly apply a series of bevels, each with its own inset and shift values.

Constrain parameters
Definition: Constrains multiple parameters of the geometry creation tool simultaneously.

Example: When creating a sphere, holding down the 'Ctrl' key constrains all three axes to the same value creating an equally sized sphere.

Curve tools

LMB	MMB	RMB	Shift+LMB	Ctrl+LMB
Create new curve node	Delete Node	None	Drag Node+Child Node	Move Constrained

Create New Curve Node
Definition: LMB clicking with a curve-type tool (Curve, Bezier, Tube, etc) will create a new curve point after the currently selected point. To add a point between two existing points, first select the point that you intend to be "behind" the new point. LMB click/dragging on a curve point will edit the location of that point.

Example: The Tube tool allows you to click repeatedly in the 3D viewport to define new tube control points. Clicking on an existing point will select it so that you can add new points in the middle of the tube.

Delete Node
Definition: MMB clicking directly on a curve handle will delete it immediately.

Example: When working with the Curve tool you may add more nodes that you need. Simply MMB clicking on a node will delete it from the curve.

None
Definition: Right mouse button has no direct action wehn editing curves.

Drag Node and Child Nodes
Definition: Holding down the Shift key while using the LMB to move a curve node will also move any nodes that are "children" of that node.

Example: When using Solid Sketch you may want to move an entire "branch". To do this, simply hold the Shift key while LMB dragging the node.

Move Constrained
Definition: Ctrl-LMB dragging a curve control point will constrain your movement along the first axis you drag toward.

Example: When using the Tube tool, hold the CTRL key down and click on a control point in the tube. Dragging along an axis will constrain the move to that axis so that you can not adjust the position of the point on the other axis.

modo Tool Reference

Creating Geometry

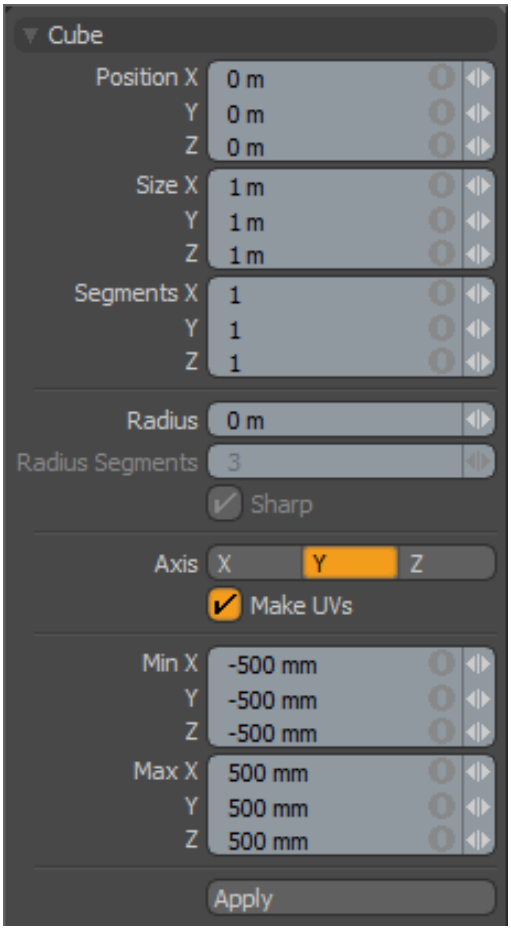
modo offers a variety of different tools for quickly generating geometry. Primitive items such as Spheres and Cubes are a good way to build basic forms which can then be edited, added to, modified and otherwise molded into the form the users wants using any of the deform or mesh editing tools. Drawn curves are useful for creating wireframe skeletons of a form where patches are then used to define the final surface. Functions like Solid Sketch allow users to explore forms more readily without the need to manipulate specific geometry. For information on using a specific tool, click on any of the link options below to be directed to the proper page.

Primitives		Drawing	Other
Cube	Torus	Pen	Solid Sketch
Sphere	Ellipsoid	Curve	Text
Cone	Tube	Bezier	Vertex Tool
Cylinder	Teapot	Sketch	Patch Tool
Capsule		Arc	

Cube

The Box primitive provides the user with a simple method for creating boxes or cubes. There are many parameters available to the user to create variations on the default 1 meter cube. By dragging in the 3D viewport with the Box tool active modo the user can create the initial plane for the box. Once the initial plane is created, clicking away from one of the handles will snap that handle to the mouse position so that the user can add dimension to the shape, or simply change the current values.

There are also "Alternate" commands assigned to all of the Primitive Tool icons. The alternate commands allow you to CTRL click on the icon to quickly create a unit primitive without interacting with the tool or SHIFT click the icon to create a unit primitive inside a new Mesh Item layer.



Cube--

Center X,Y,Z: These three values establish the 3D location for the center of the cube.

Size X,Y,Z: These three values are used to establish the dimensions of the cube. Set to 1m, 1m, 1m by default, provides us with a 1 meter cube. If you want to create a 6 foot plank that is 2 inches by 4 inches, you can type into the X,Y, and Z numeric fields respectively: 6' 2" 4" and modo will perform the measurement translation to the current unit mode (Metric, SI, or English).

Segments X, Y, Z: By default the primitive box uses a single segment on each axis. By increasing this value you can increase the number of "cuts" in each side of the box. This is useful if you plan to deform the cube as these segments act as "hinges" during deformation operations.

Radius: This distance value sets a curved edge amount. When radius is set to a positive non-zero value, the edge of the box is rounded off via extra segments added to the geometry. The number of segments is determined by two additional settings; Radius Segments and Sharp.

Radius Segments: When the Radius value is set to a positive, non-zero value, additional segments are added to round the cube. The radius segments field allows the user to set how many edges are added to smooth out the corners.

Sharp: This will add an extra set of polygons to the edge of the rounded edge so that the adjacent faces are not affected by surface smoothing. This allows the surface material to use smoothing to further round the edges without losing the flat appearance of the faces of the cube.

Axis: Defines the major axis for the cubes orientation. On the cube this will only affect the direction of radius segments.

Make UVs: When this button is active a UV map will be automatically generated for the

geometry created with the tool. This is a very useful option if you plan to UV map the model you are creating from the primitive as it provides a baseline UV map that you can massage later in the modeling process. In many cases this can reduce the amount of work required to map the model.

Min XYZ/Max XYZ: Users may also define a cube based on specific X, Y and Z bounding box locations in 3D space which can be specified here. This makes it easy to place a cube on the ground plane for instance by making the 'Min Y' value '0' and the 'Max Y' value '1m'.

TIP: The Box primitive supports symmetrical creation. When Symmetry is activated, creating the primitive shape will create an identical version across the specified axis.

Sphere

The sphere primitive provides the user with a simple method for creating balls or spheres. There are many paramaters available to the user to create variations on the default 1 meter sphere. By dragging in the 3D viewport with the Sphere tool active you can create the initial plane for the sphere. Once the initial plane is created, clicking away from one of the handles will snap that handle to the mouse position so that you can add dimension to the shape, or simply change the current values. If you wish to drag out a perfect sphere, hold the CTRL key while dragging in the 3D view.

There are also "Alternate" commands assigned to all of the Primitive Tool icons. The alternate commands allow you to CTRL click on the icon to quickly create a unit primitive without interacting with the tool or SHIFT click the icon to create a unit primitive inside a new Mesh Item layer.

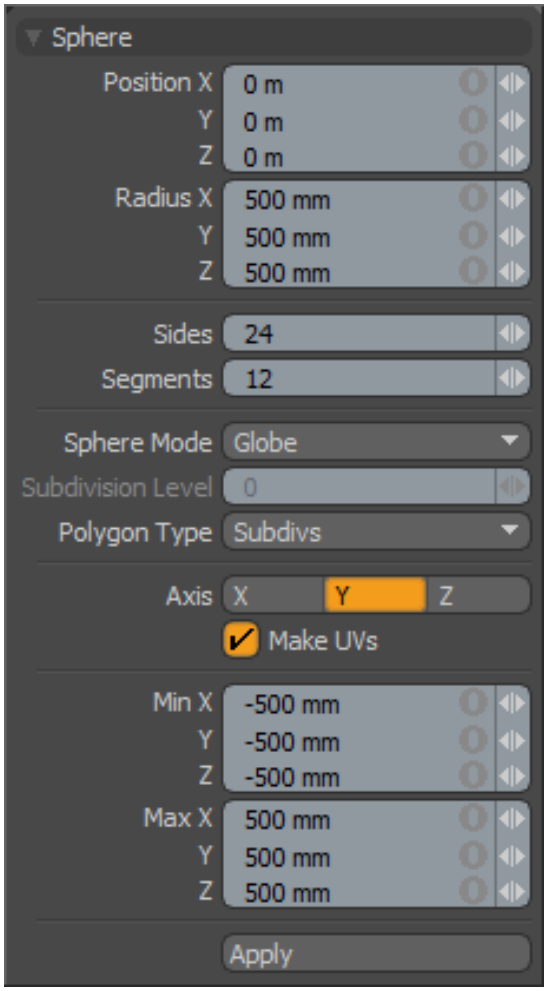
Sphere--

Position X,Y,Z: These three values establish the 3D location for the center of the ball.

Radius X,Y,Z: These three values are used to establish the dimensions of the sphere. Set to 1m, 1m, 1m by default, provides us with a 2 meter ball.

Sides: The sides are the latitudinal edges of the sphere running from top to bottom. By default the primitive ball uses 24 sides. By increasing this value you can increase the number of "cuts" in the ball. This is useful if you plan to deform the ball as these segments act as "hinges" during deformation operations. This also increases the smoothness of the sphere by adding extra facets. If you plan to apply subdivision

surfaces to the mesh, you can use a lower number of segments as the subdivision will smooth out the shape.



Segments: The segments are the longitudinal edges of the sphere running from right to left. By default the primitive ball uses 12 segments. By increasing this value you can increase the number of "cuts" in the ball. This is useful if you plan to deform the ball as these segments act as "hinges" during deformation operations.It also increases the smoothness of the sphere by adding extra facets. If you plan to apply subdivision surfaces to the mesh, you can use a lower number of segments as the subdivision will smooth out the shape.

Sphere Mode: Options for various Sphere types, as follows--

Globe-- The default setting creates a ball with the controls described above.

Quadball-- This option creates a SDS spehere made entirely of polygons containing only four vertices (quads). When this option is active the Sides and Segments settings are disabled and the Subdivision Level control activates. Increasing this value will add complexity to the quadball primitive surface.

Tesselated-- By default the ball primitive is composed of mostly quad polygons with some triangles at the poles. Activating the Tesselated flag will create the sphere from all triangles. When this is active the Tesselation Level sets the resolution of the mesh created. Using a tesselated sphere with a high tesselation level will create a more perfect sphere than the default sides and segments, but with many more polygons. Tesselated spheres are not recommended if you plan to use subdivision surfaces.

Subdivision Level: For the 'Quadball' and 'Tessleated' sphere mode options, the

Subdivision level sets the number of subdivisions, which determines the number of polygons used to define the overall shape. Larger number produce more polygons that are rounder with a smoother surface.

Polygon Type: Users can use this option to determine the type of polygons created-

Face- Creates standard unsmoothed polygon faces.

Subdivs- Creates Subdivision Surface smoothed polygons.

Catmull-Clark- Creates Pixar Catmull-Clark smoothed polygons.

Axis: Defines the major axis direction, most useful for 'Globe' type sphere as it controls the direction of the segments along the radius.

Make UVs: When this button is active a UV map will be automatically generated for the geometry created with the tool. This is a very useful option if you plan to UV map the model you are creating from the primitive as it provides a baseline UV map that you can massage later in the modeling process. In many cases this can reduce the amount of work required to map the model.

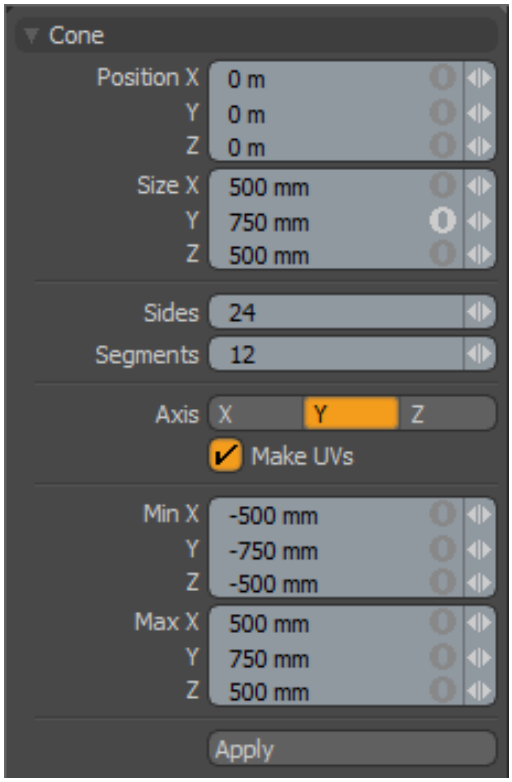
Min XYZ/Max XYZ: Users may also define a sphere based on specific X, Y and Z bounding box locations in 3D space which can be specified here. This makes it easy to place a sphere on the ground plane for instance by making the 'Min Y' value '0' and the 'Max Y' value '1m'.

TIP: The Sphere primitive supports symmetrical creation. When Symmetry is activated, creating the primitive shape will create an identical version across the specified axis.

Cone

The Cone primitive provides the user with a simple method for creating conical shapes. There are many paramaters available to the user to create variations on the default 1 meter cone. By dragging in the 3D viewport with the Cone tool active you can create the initial plane for the cone. Once the initial plane is created, clicking away from one of the handles will snap that handle to the mouse position so that you can add dimension to the shape, or simply change the current values. If you wish to drag out a 3D cone, hold the CTRL key while dragging in the 3D view.

There are also "Alternate" commands assigned to all of the Primitive Tool icons. The alternate commands allow you to CTRL click on the icon to quickly create a unit primitive without interacting with the tool or SHIFT click the icon to create a unit primitive inside a new Mesh Item.



The parameters for the cone tool include:

Position X,Y,Z: These three values establish the 3D location for the center of the cone.

Size X,Y,Z: These three values are used to establish the dimensions of the cone. Set to 50 cm, 75 cm, 50 cm by default. As a radius they produce a cone that is .75 meter tall and 1 meter across at the base.

Sides: The sides are the latitudinal edges of the cone running from top to bottom. By default the primitive cone uses 24 sides. By increasing this value you can increase the number of "cuts" in the cone.This also increases the smoothness of the cone by adding extra facets.

Segments: The segments are longitudinal edges of the sphere running from right to left. By default the primitive cone uses 12 segments. By increasing this value you can increase the number of "cuts" along the cone. This is useful if you plan to deform the cone as these segments act as "hinges" during deformation operations.

Axis: This X,Y,Z choice allows you to quickly change the orientation of the cone.

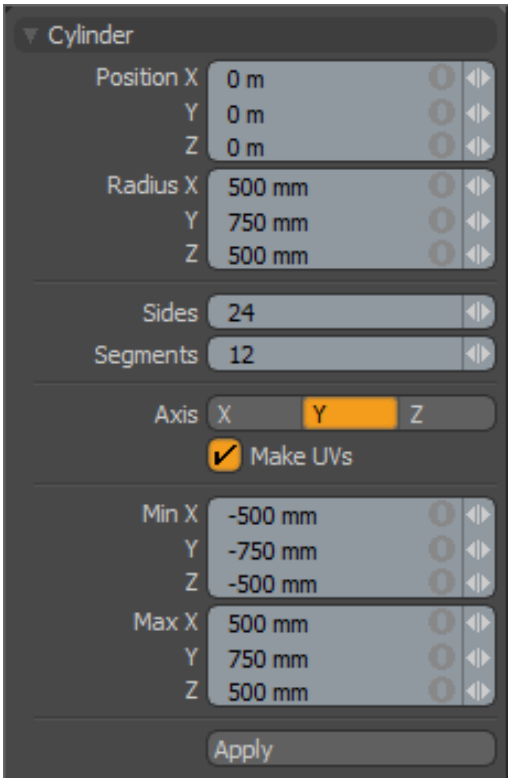
Make UVs: When this button is active a UV map will be automatically generated for the geometry created with the tool. This is a very useful option if you plan to UV map the model you are creating from the primitive as it provides a baseline UV map that you can massage later in the modeling process. In many cases this can reduce the amount of work required to map the model.

Min XYZ/Max XYZ: Users may also define the shape based on specific X, Y and Z bounding box locations in 3D space which can be specified here. This makes it easy to place a cone on the ground plane for instance by making the 'Min Y' value '0' and the 'Max Y' value '1m'.

TIP: The Cone primitive supports symmetrical creation. When Symmetry is activated, creating the primitive shape will create an identical version across the specified axis.

Cylinder

The Cylinder primitive provides the user with a simple method for creating cylinders or tubes. There are many paramaters available to the user to create variations on the default 1 meter cylinder. By dragging in the 3D viewport with the Cylinder tool active modo the user can create the initial end cap for the cylinder. Once the initial cap is created, clicking away from one of the handles will snap that handle to the mouse position so that the user can add dimension to the shape, or simply change the current values.



There are also "Alternate" commands assigned to all of the Primitive Tool icons. The alternate commands allow you to CTRL click on the icon to quickly create a unit primitive without interacting with the tool or SHIFT click the icon to create a unit primitive inside a new Mesh Item layer.

The parameters for the cylinder tool include:

Position X,Y,Z: These three values establish the location for the center of the cylinder.

Radius X,Y,Z: These three values are used to establish the dimensions of the cylinder based on the radius from the center position.

Sides: This value determines how many edges are used to define the cylinder's circumference. The larger the number of sides, the smoother the cylinder will appear. If you plan to use subdivision surfaces on the cylinder, use a low number.

Segments: By default the primitive cylinder uses 12 segments. By increasing this value you can increase the number of "cuts" along the cylinder. This is useful if you plan to deform the cylinder as these segments act as hinges during deformation operations.

Axis: This X,Y,Z choice allows you to quickly change the orientation of the primitive.

Make UVs: When this button is active a UV map will be automatically generated for the geometry created with the tool. This is a very useful option if you plan to UV map the

model you are creating from the primitive as it provides a baseline UV map that you can massage later in the modeling process. In many cases this can reduce the amount of work required to map the model.

Min XYZ/Max XYZ: Users may also define a cylinder based on specific X, Y and Z bounding box locations in 3D space which can be specified here. This makes it easy to place the cylinderon the ground plane for instance by making the 'Min Y' value '0' and the 'Max Y' value '1m'.

TIP: The Cylinder primitive supports symmetrical creation. When Symmetry is activated, creating the primitive shape will create an identical version across the specified axis.

Capsule

The Capsule primitive provides the user with a simple method for creating capsules or rounded edge tubes. There are many parameters available to the user to create variations on the default 1 meter capsule. By dragging in the 3D viewport with the tool active you can create the initial plane for the primitive. Once the initial plane is created, clicking away from one of the handles will snap that handle to the mouse position so that you can add dimension to the shape, or simply change the current values. If you wish to drag out a 3D capsule, hold the CTRL key while dragging in the 3D view.

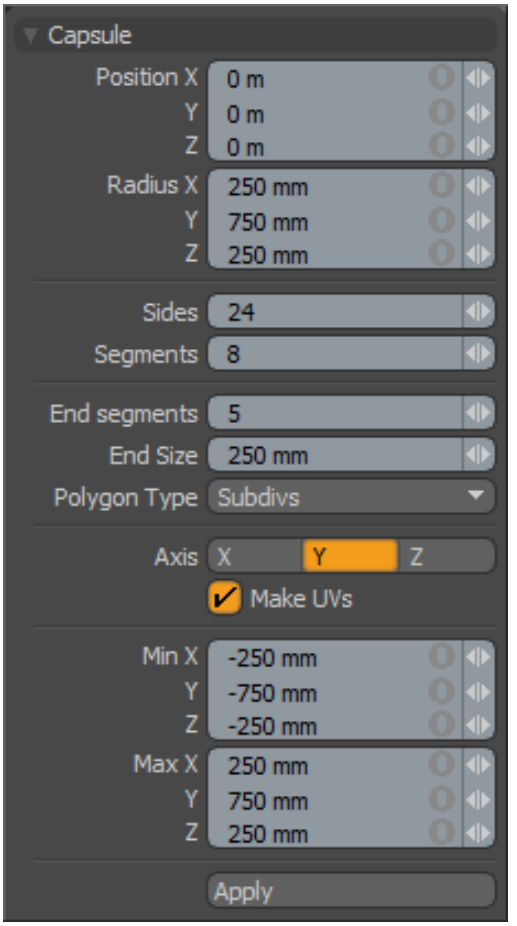
There are also "Alternate" commands assigned to all of the Primitive Tool icons. The alternate commands allow you to CTRL click on the icon to quickly create a unit primitive without interacting with the tool or SHIFT click the icon to create a unit primitive inside a new Mesh Item.

Capsule--

Position X,Y,Z: These three values establish the 3D location for the center of the primitive.

Radius X,Y,Z: These three values are used to establish the overall dimensions of the capsule. Set to 25cm, 75cm, 25cm by default, being radius value, they produce a capsule that is .5 meters wide and 1.5m long.

Sides: The sides are the latitudinal edges of the capsule running from top to bottom. By default the primitive capsule uses 24 sides. By increasing this value you can increase the number of "cuts" in the shape. This is useful if you plan to deform the capsule as these segments act as "hinges" during deformation operations. This also increases the smoothness of the capsule by adding extra facets.



Segments: The segments are the longitudinal edges of the capsule running from right to left. By default the primitive uses 8 segments. By increasing this value you can increase the number of "cuts" in the capsule. This is useful if you plan to deform the shape as these segments act as "hinges" during deformation operations. It also increases the smoothness of the shape by adding extra facets.

End Segments: This value is used for creating the rounded end caps on the capsule. This value sets the number of longitudinal edges on the end cap. Increasing this number produces a more round cap. The default value is 5.

End Size: The end size determines the height of the cap. The default value is 50cm.

Polygon Type: Users can use this option to determine the type of polygons created-

Face- Creates standard unsmoothed polygon faces.

Subdivs- Creates Subdivision Surface smoothed polygons.

Catmull-Clark- Creates Pixar Catmull-Clark smoothed polygons.

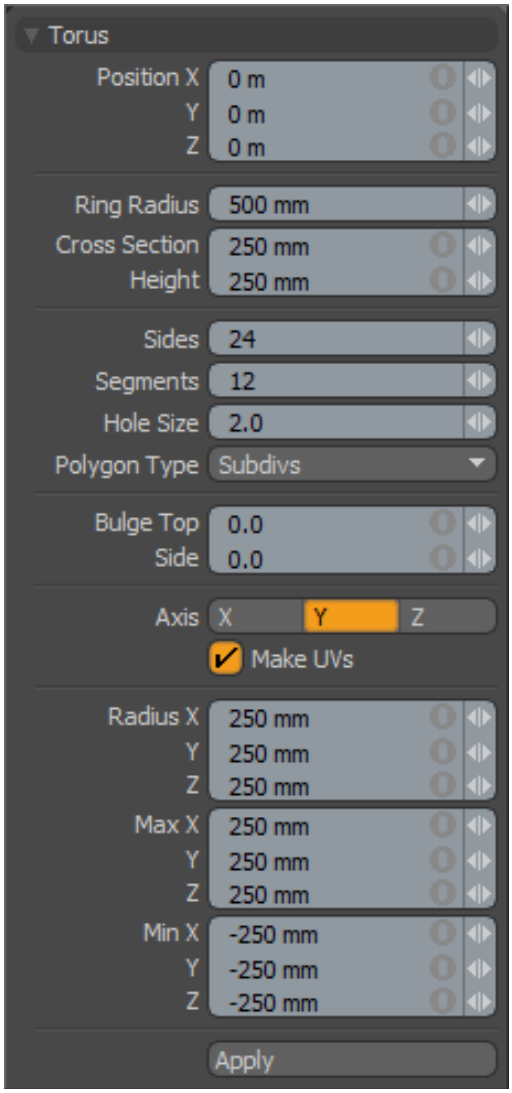
Axis: This X,Y,Z choice allows you to quickly change the orientation of the primitive.

Make UVs: When this button is active a UV map will be automatically generated for the geometry created with the tool. This is a very useful option if you plan to UV map the model you are creating from the primitive as it provides a baseline UV map that you can massage later in the modeling process. In many cases this can reduce the amount of work required to map the model.

Min XYZ/Max XYZ: Users may also define the shape based on specific X, Y and Z bounding box locations in 3D space which can be specified here. This makes it easy to place a ellipsoid on the ground plane for instance by making the 'Min Y' value '0' and the 'Max Y' value '1m'.

TIP: The Capsule primitive supports symmetrical creation. When Symmetry is activated, creating the primitive shape will create an identical version across the specified axis.

Torus



The Toroid primitive provides the user with a simple method for creating toroids or "doughnuts". There are several parameters available to the user to create variations on the default torus. By dragging in the 3D viewport with the tool active you can create the initial plane for the primitive. Once the initial plane is created, clicking away from one of the handles will snap that handle to the mouse position so that you can add dimension to the shape, or simply change the current values. If you wish to drag out a 3D torus, hold the CTRL key while dragging in the 3D view.

There are also "Alternate" commands assigned to all of the Primitive Tool icons. The alternate commands allow you to CTRL click on the icon to quickly create a unit primitive without interacting with the tool or SHIFT click the icon to create a unit primitive inside a new Mesh Item.

Torus--

Position X,Y,Z: These three values establish the 3D location for the center of the primitive.

Ring Radius: Determines the overall outside width of the Torus.

Cross Section/Height: A Cross Section Width and Height setting determining the sizing

of the ring cross section of the Torus.

Radius X,Y,Z: These three values are used to establish the dimensions of the torus. The tool is set to 50cm, 50cm, 50cm by default.

Sides: The sides are the latitudinal edges of the torus running from top to bottom. By default the primitive uses 24 sides. By increasing this value you can increase the number of "cuts" in the shape. This is useful if you plan to deform the mesh as these segments act as "hinges" during deformation operations. This also increases the smoothness of the capsule by adding extra facets. If you plan to apply subdivision surfaces to this shape you can use a lower number of sides.

Segments: The segments are the longitudinal edges of the torus running from right to left. By default the primitive uses 12 segments. By increasing this value you can increase the number of "cuts" in the shape. This is useful if you plan to deform the shape as these segments act as "hinges" during deformation operations. It also increases the smoothness of the shape by adding extra facets. If you plan to apply subdivision surfaces to this shape you can use a lower number of segments.

Hole Size: This value sets the radius of the hole in the middle of the torus.

Polygon Type: Users can use this option to determine the type of polygons created-

Face- Creates standard unsmoothed polygon faces.

Subdivs- Creates Subdivision Surface smoothed polygons.

Catmull-Clark- Creates Pixar Catmull-Clark smoothed polygons.

Bulge Top/Side: Rasing this value above zero (0) attenuates the shape toward a more cubic or box like form.

Axis: This X,Y,Z choice allows you to quickly change the orientation of the primitive.

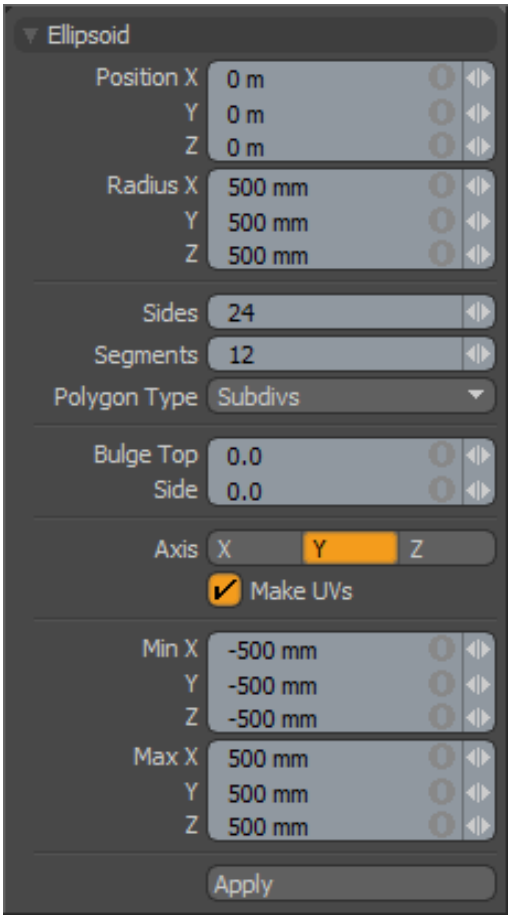
Make UVs: When this button is active a UV map will be automatically generated for the geometry created with the tool. This is a very useful option if you plan to UV map the model you are creating from the primitive as it provides a baseline UV map that you can massage later in the modeling process. In many cases this can reduce the amount of work required to map the model.

Radius: Determines the size from the center of the Torus to the outer bounding box edge, the radius of the Torus.

Min XYZ/Max XYZ: Users may also define the shape based on specific X, Y and Z bounding box locations in 3D space which can be specified here. This makes it easy to place the torus on the ground plane for instance by making the 'Min Y' value '0' and the 'Max Y' value '1m'.

TIP: The Box primitive supports symmetrical creation. When Symmetry is activated, creating the primitive shape will create an identical version across the specified axis.

Ellipsoid



The Ellipsoid primitive provides the user with a simple method for creating ellipses or "pills". There are many parameters available to the user to create variations on the default ellipse. By dragging in the 3D viewport with the tool active you can create the initial plane for the primitive. Once the initial plane is created, clicking away from one of the handles will snap that handle to the mouse position so that you can add dimension to the shape, or simply change the current values. If you wish to drag out a 3D ellipse, hold the CTRL key while dragging in the 3D view.

Ellipsoid--

Position X,Y,Z: These three values establish the 3D location for the center of the primitive.

Size X,Y,Z: These three values are used to establish the dimensions of the ellipsoid. The default values are 50cm, 50cm, 50cm

Sides: The sides are the latitudinal edges of the ellipsoid running from top to bottom. By default the primitive ellipsoid uses 24 sides. By increasing this value you can increase the number of "cuts" in the shape. This is useful if you plan to deform the ellipsoid as these segments act as "hinges" during deformation operations. This also increases the smoothness of the shape by adding extra facets.

Segments: The segments are the longitudinal edges of the ellipsoid running from

right to left. By default the primitive uses 12 segments. By increasing this value you can increase the number of "cuts" in the primitive. This is useful if you plan to deform the shape as these segments act as "hinges" during deformation operations. It also increases the smoothness of the shape by adding extra facets.

Polygon Type: Users can use this option to determine the type of polygons created-

Face- Creates standard unsmoothed polygon faces.

Subdivs- Creates Subdivision Surface smoothed polygons.

Catmull-Clark- Creates Pixar Catmull-Clark smoothed polygons.

Bulge Top: Increasing this value will change the shape from an elongated sphere to a rounded capped cylinder.

Side: Increasing this value will push the edges to the corners of the shape converting it into a flat sided ellipse.

Axis: This X,Y,Z choice allows you to quickly change the orientation of the primitive.

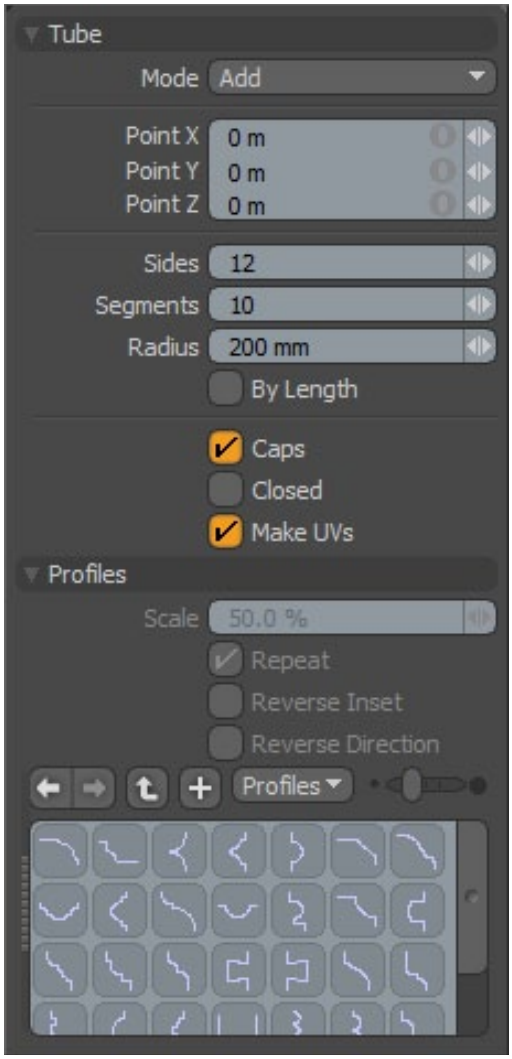
Make UVs: When this button is active a UV map will be automatically generated for the geometry created with the tool. This is a very useful option if you plan to UV map the model you are creating from the primitive as it provides a baseline UV map that you can massage later in the modeling process. In many cases this can reduce the amount of work required to map the model.

Min XYZ/Max XYZ: Users may also define the shape based on specific X, Y and Z bounding box locations in 3D space which can be specified here. This makes it easy to place a ellipsoid on the ground plane for instance by making the 'Min Y' value '0' and the 'Max Y' value '1m'.

TIP: The Ellipsoid primitive supports symmetrical creation. When Symmetry is activated, creating the primitive shape will create an identical version across the specified axis.

Tube

Using the Tube primitive you can quickly click down a number of control points and modo will create a tube based on the a spline curve interpolated through them. The Tube tool has several options.



Tube--

Mode: In 'Add' mode (default) each click on the 3D viewport creates a new control point in the tube. Clicking on an existing control point allows you to move it freely in space. To add a new control point between two existing points, select the first point of the desired pair and then click between them. 'Edit' mode disables the ability to create new control points so that existing control points can be edited quickly without the concern of accidentally adding a new control point by simply missing one of the points when clicking to edit it. The ' Delete' will remove control points when directly clicked on.

Point X/Y/Z: This value triplet is set interactively as you drag a control point in the 3D viewport. If you wish to set a control point to a very specific location you can also use the tool property sheet to enter the X,Y and Z coordinates directly.

Sides: This value determines how many polygons (sides) will be used to make the ring of the tube. If you use a low number such as 3 you would create a triangular tube, whereas a higher number such as the default 12 would create a very round tube. If you plan to use subdivision surfaces on the tube once it is created, you can lower the number of sides to 4 or 5 and the resulting mesh will still be very smooth once subdivision surfaces are active.

Segments: This value determines the number of polygons that are used to span the gap between each control point. If you set this value to one, you will see that the tube looks very linear as there are not enough connecting polygons to interpolate a smooth curve. Lower numbers such as one or two can work if you plan to use subdivision surfaces after creating the tube. Using larger numbers will create a more smooth flowing tube at the expense of additional geometry.

Radius: This value sets the thickness of the tube. It is important to keep in mind that if you plan to use subdivision surfaces, the resulting mesh will be somewhat smaller than the original cage polygons. In this case, you may wish to increase the radius prior to completing the tube so that when you activate subdivision surfaces, the resulting mesh maintains your intended thickness.

By Length: When *enabled*, tube sections will be recomputed as equal lengths across the curve rather than by the control point locations.

Caps: This boolean button option determines whether or not the ends of the tube are closed. By default this setting is turned on so that the tube is closed on both ends. You can remove the caps by turning this button off.

Closed: Activating the Closed flag will create a curve loop rather than a curve line so that the resulting tube has its first and last points connected together via another set of segments.

Make UVs: When this button is active a UV map will be automatically generated for the geometry created with the tool. This is a very useful option if you plan to UV map the model you are creating from the primitive as it provides a baseline UV map that you can massage later in the modeling process. In many cases this can reduce the amount of work required to map the model.

Profiles--

Mini Preset browser for viewing various profiles. Works the same as the standard preset browser. Profiles, when selected, apply themselves along the length of the tube.

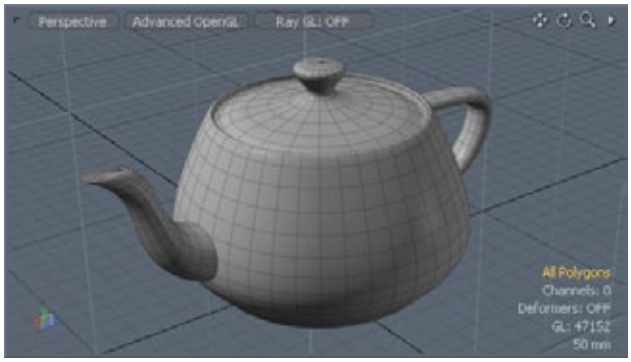
Scale: With this setting, users can adjust the size of the profile applied to the tube primitive.

Repeat: When this option is toggled, it enables/disables the repeating of the profile between each curve point.

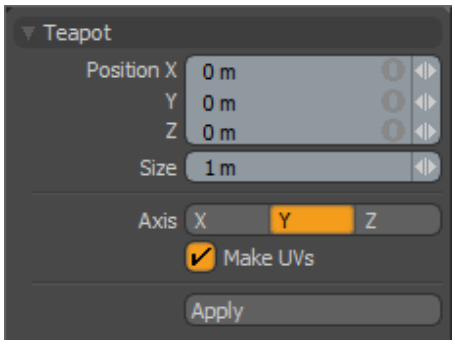
Reverse Inset: This option toggles the inset direction of the profile, reversing it horizontally. Areas that we initially pushed in when activated would push out.

Reverse Direction: This option toggles direction of the profile, reversing it vertically, providing a flipped orientation for the profile.

Teapot



The teapot primitive in included with modo due to its historical significance. In the early years of 3D modeling and rendering, Martin Newel created a computer generated three dimensional teapot. The original "Utah teapot" was actually taller than the famed teapot we see so many places today. Legend has it that during a demo using a system with non-square pixels Jim Blinn scaled the teapot rather than scaling the image. As this dataset became more widely distributed than the original, his edit changed the aspect of teapots in 3D software for years to come. If you are interested in following teapot lore, simply type "Utah Teapot" into your favorite search engine.



There are also "Alternate" commands assigned to all of the Primitive Tool icons. The alternate commands allow you to CTRL click on the icon to quickly create a unit primitive without interacting with the tool or SHIFT click the icon to create a unit primitive inside a new Mesh Item.

Teapot--

Position XYZ: Defines the center point for the teapots bounding box.

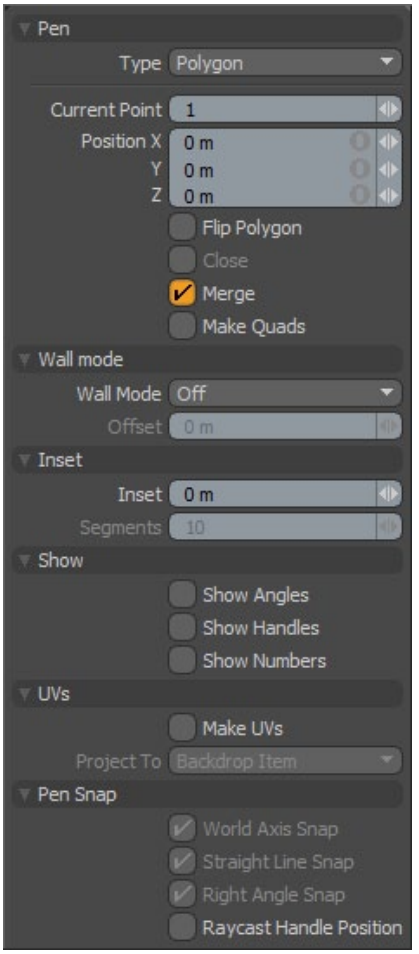
Size: Defines the overall size of the teapots longest bounding edge.

Axis: This X,Y,Z choice allows you to quickly change the orientation of the primitive.

Make UVs: When this button is active a UV map will be automatically generated for the geometry created with the tool. This is a very useful option if you plan to UV map the model you are creating from the primitive as it provides a baseline UV map that you can

massage later in the modeling process. In many cases this can reduce the amount of work required to map the model.

Pen



The Pen tool allows users to create geometry vertex by vertex by clicking in the 3D viewport. One can use the Pen tool to create Polygons (default), lines, vertices, spline patches or subdivision surface polygons. When the tool is active, users can also click directly on any of the vertices it has created in the current session and drag them in 3D space for direct editing, they'll turn yellow when the mouse is directly over them. Clicking away from a vertex will create a new vertex in that location. To start a new line without dropping the tool, press shift LMB-click. Clicking on previous vertices made within the same tool session will weld the new vertex/line section together. To insert a vertex, highlight a previously created one, and click away from it and the new vertex will be inserted after the selected vertex. When creating Quad Strips with the 'Make Quads' option enabled, an occasional triangle can be created by pressing the 'Ctrl' key.

Pen--

Type: This pop-up button allows you to select between different modes-

Polygons-- In Polygon mode, each click of the mouse creates a new vertex in the polygon currently being created by the tool.

Lines-- The Line type will create two vertex polygon line segments.

Vertices-- The Vertices mode will simply create new vertices with each click leaving you with a vertex cloud.

Spline patches-- Spline patch mode acts exactly like Polygon mode but creates polygons of type Spline Patch.

Subdivision Surfaces-- When using 'Subdiv' mode the pen tool simply creates polygons tagged as SDS.

Current Point: You can change this setting to change the current vertex to be edited by the numeric values listed as Position X, Y, Z. Since the tool may have created several vertices, it is useful to be able to choose one numerically. The vertices are numbered in sequence as they are generated by the tool.

Position X/Y/Z: This numeric entry field allows you to precisely edit each vertex created by the tool. Simply set the Current Point value to the vertex you wish to edit and enter the numeric values for the X, Y and Z positions.

Flip Polygon: This option will simply invert the surface normal of the polygon(s) created.

Close: The 'Close' toggle will automatically draw a line between the first and last vertex created making a closed loop.

Merge: The 'Merge' function will automatically 'Merge' or combine vertexes within close proximity to each other when created.

Make Quads: This option will automatically place vertices to create quads with each new click after the first two vertices have been set. This mode makes it very simple to create polygon strips. Pressing the 'Ctrl' key will create a single triangle. Option is only compatible with the 'Type' > 'Polygons' setting.

Wall Mode--

Wall Mode: With mode type set to Polygons, Wall mode will allows the user to draw out polygon strips that look like a building floor plan. effectively creating the foundation for rooms. There are 4 settings, 'Off', disables wall mode, 'Inner' will draw the wall segment on the inside of the polyline, 'Outer' will draw the wall segments on the outside of the polyline, 'Both' will extend the wall segments on both sides of the polyline.

Offset: The Offset value represents the thickness of the wall, when in wall mode. Note then when wall mode is set to both, that offset will be doubled. This value can also be set interactively with the RMB-click in the viewport, much like adjusting brush size for painting.

Inset--

Inset: When 'Wall Mode' is enabled, this option allows each new vertex to be beveled into multiple segments for quickly flattening or rounding corners.

Segments: Only available when Inset is non-zero this value determines how many vertices to create in order to facilitate the inset value. Increasing this value makes more complex and rounded corners. Using a value of one results in flattened corners.

Show--

Show Angles: Toggles the display of corner angles between opposing edges in degrees, drawn in ghosted form over active geometry.

Show Handles: Toggles the display of tool handles offering users the ability to adjust the previously created vertex by axis.

Show Numbers: Toggles display of the vertex point order values in the viewport.

UVs--

Make UVs: When enabled this setting will automatically make UVs for the created geometry further defined by the 'Project To' setting.

Project To: Determines the way that the UV map is created

Action Axis-- With this setting, the Pen tool determines the UV map projection axis based on the current Work Plane for the created polygon. UVs are created in space relative to where they are drawn in 3D space, so only points drawn with the 0m to 1m world space will appear within the 0-1 UV space.

Backdrop Item-- Useful for drawing polygons over a Backdrop item, the UVs are projected onto the backdrop item to match position. So, for instance, if you created a polygon over the tire of car backdrop image, the new polygon would inherit the UV position of the tires location in the image relative to UV space.

U/V Direction-- When in 'Wall Mode', this setting will automatically generate a row of regular quad UV coordinates for the polygons along either the U direction (horizontal) or V direction (vertical) as specified.

Pan Snap--

World Axis Snap: When this is active the pen tool handles will snap along the world XYZ axis once you have created more than one vertex.

Straight Line Snap: Will try to snap the handle to a position that will create a straight line between the next and previous vertices in the polygon being created.

Right Angle Snap: Will effect a snap when the angle created by the current, previous and next vertices create a 90 degree angle.

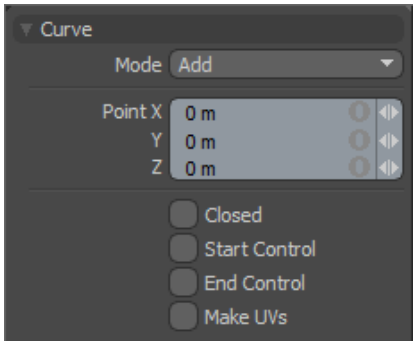
Raycast Handle Position: With this option enabled, vertices behind the background mesh will be unselectable. What this means is that vertices hidden behind backwards facing polygons will be ignored for snapping and editing.

Curve

The Curve tool allows you to create a spline curve by clicking down a series of control points through which the curve is interpolated. This is a hermite curve type and can be very useful for creating modeling guides, quickly roughing out a shape, or setting up an outline for lofting with the extrude, lathe, or other multiply tools. Once you have clicked to create several control points, you can also click directly on an existing control point to drag it around, allowing you to change the shape of the curve.

Adding new control points to the curve is simple. Clicking anywhere on the screen, so long as you are not selecting an existing control point, will create a new control point in the curve. It is important to note that the new control point will be added after the currently selected point. Therefore, if you wish to add a point in a specific location on the curve, be certain to first select the point that would be on the curve before where the new point will be located.

The Curve tool has several parameters including: Mode, Point Position (XYZ), Closed toggle, Start Control, End Control, and Make UVs.



Curve--

Mode: This popup choice allows you to change the behavior of mousing. The options are Add, Edit and Delete. The default Add setting behaves as described above. When set to Edit the ability to add control knots is removed and the focus of the tool is entirely on editing the existing control points. The delete mode provides a simple method for deleting existing control points. When the delete mode is active clicking on a control point will remove it from the curve.

Point Position: The X,Y,Z values of the currently active point can be set precisely by typing in the desired 3D coordinates.

Closed: Activating the Closed toggle causes the spline curve to complete the loop by connecting the first and last points in the curve. Rather than create spline curves, this sets the tool to generate spline loops.

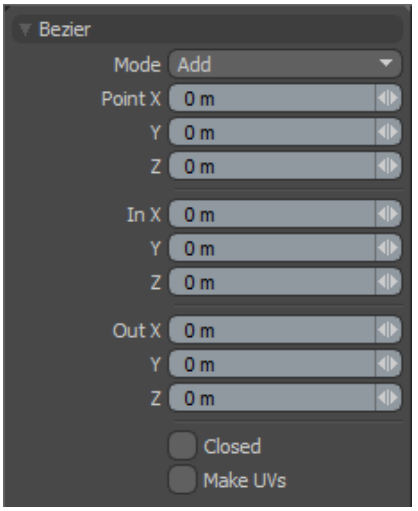
Start Control: This toggle converts the first point in the curve to a control point that does not contribute to the actual curve length, but acts as a handle so that the curvature at the new "first" point can be modified. When activated the curve length from the control point to the next point is drawn in the handle color to indicate that it is no longer part of the geometry, but simply a handle.

End Control: This toggle converts the last point in the curve to a control point that does not contribute to the actual curve length, but acts as a handle so that the curvature at the new "end" point can be modified. When activated the curve length from the control point to the next point is drawn in the handle color to indicate that it is no longer part of the geometry, but simply a handle.

Make UVs: This toggle activates auto-generation of UV texture coordinates along the curve.

Bezier

A Bezier Curve is a type of curve defined by control vertices. Bezier Curves were originally developed for computer modeling in automotive design, but are popular in many Vector drawing applications, providing a more familiar means of drawing curves. In modo, the Bezier tool is found in the 'Basic' subtab of the modeling toolbox. Each vertex is controlled by two other points that control the endpoint tangent vectors. Bezier curves have control handles that are used to adjust curvature.



Bezier Curves support 3 modes: 'Add', 'Edit' and 'Delete'. You can use these modes to create additional points, edit them, or remove them from the curve, during the drawing operation. Once the tool is dropped, interactive handle editing within the tool itself is lost, users may re-activate the editing ability by first selecting the Curve itself in 'Polygons' mode before activating the tool again, LMB+clicking in the viewport will again display the editing handles. Points on the Bezier Curve can be positioned precisely by using the Point XYZ fields of the tools properties form.

Bezier--

Mode: The various bezier curve mode options available are--

Add-- The default mode, when users click in the viewport, additional points are added to the curve. At each control point position two handles extend out that allow users to adjust the curvature of the line between vertices. While drawing users can hover over any point or handle where it will turn yellow, users can then LMB-click and drag the control point or handle to further edit the curve. Control points may be added mid-curve by selecting the preceding control, highlighting it yellow, and clicking the position where the new control point is desired.

Edit-- In edit mode users can freely LMB+click drag and of the control points or

handles along the curve to change the look to the desired shape. While pressing the 'Ctrl' key, users may break the tangency of the control handles, allowing for sharp corners, holding 'Shift' while clicking a handle will restore tangency.

Delete-- In delete mode, users can LMB+click on any control point along the curve to remove it from the bezier.

Point XYZ: Users wishing finer point control can assign specific XYZ values in these input fields for the currently selected (highlighted) control point.

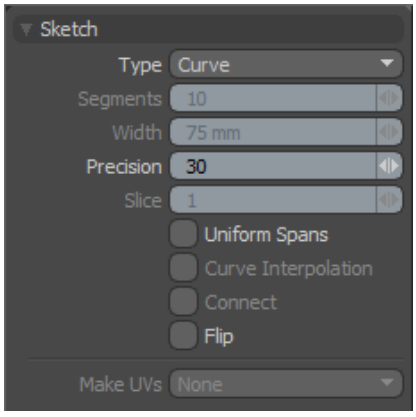
In XYZ/Out XYZ: Users wishing finer handle control can assign specific XYZ values in these 'In' and 'Out' input fields for the currently selected (highlighted) control point.

Closed: The 'Closed' toggle option adds an automatic curve segment between the first and last control point positions producing a closed curve.

Make UVs: This toggle activates auto-generation of UV texture coordinates along the curve.

Sketch

The Sketch tool allows you to quickly create a curve by drawing in the 3D view. To use the tool, click and drag in any 3D viewport. Use a mouse or tablet to draw the curve and modo will convert your sketch into the curve type you've chosen. Strokes, while active have a couple little widgets associated with them, a plus icon at the root or base and a small dot at the tip. Strokes can be manipulated by clicking on the plus icon to move the entire curve and clicking on the dot at the tip will allow the user to rotate and scale the tip around the base. With 'Constrain to Background' enabled, the sketch tool is a quick and easy way to add polygons, retopologizing any background geometry.



Sketch--

Type: This pop-up allows you to choose between the various sketch tool types-

Curve-- When set to Curve (*default*), the Sketch tool creates a spline curve based on the sketch path you create.

Face-- When set to Face, the tool creates a polygon by drawing a straight line from the end of your sketch back to the beginning so as to close the loop.

Lines-- When set to Line, the tool will generate a series of 2 vertex poly-lines that follow the contour of your sketched curve.

Quads-- When set to quads, the tool will create a strip of quad (4 sided) polygons centered along the contour of the sketched curve.

Segment: When set to quads, the segments value specifies the number of polygons in a segment strip. Somewhat like uniform spans, in that subsequent strokes if longer or shorter may contain more or less segments, however the spans will roughly equal those in the initial segment value until changed.

Width: Sets the width of the polygon strip. Setting only affect type-quads. One can also RMB-click in the viewport to interactively set the width value.

Precision: This value sets how finely modo interpolates the drawn stroke, lower values produce strokes with greater numbers of points, higher values produce strokes with lower numbers of points. Think of it roughly like the distance in pixel the pointer travels before creating a new point.

Slice: When set to quads, this value specifies the number of slices in a poly strip, so a 16 polygon strip with a slice value of 2 would produce 32 polygons > 16 long by 2 wide.

Uniform Spans: For curves and polylines, this setting creates a uniform distance between generated vertexes regardless of stroke speed or shape. When set to 'Quads', Uniform Spans creates straight line quad strips between the start and end point of the drawn stroke.

Curve Interpolation: Continuously evaluates the stroke, when set to Type-Quads, with uniform spans, the polygon strips will follow along the drawn path

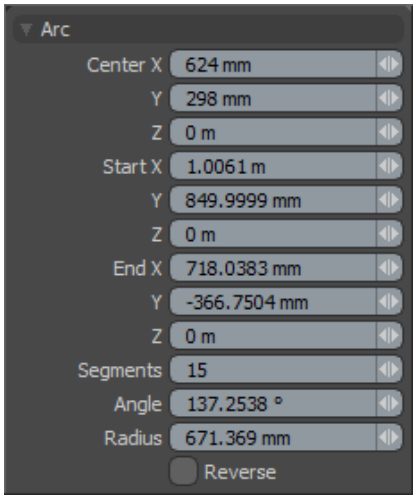
Connect: When drawing out quad strip polygons, with 'Connect' enabled, adjoining strips in close enough proximity to each other, will join together.

Flip: Flip will reverse the normal facing direction of polygons. If you're drawing out polygons and you don't see anything, they are probably facing away from you, use the flip command to reverse their facing direction.

Make UVs: When creating Quad polygon strips, Make UVs will automatically make appropriate UVs. Choose witch direction you wish the lengths to align with U (horizontal) or V (vertical).

Arc Tool

Found in the menu bur, under "Geometry > Primitive Tools > Arc Tool", the Arc Tool provides users a means to make closed segments of a circular shape, think of a pie with wedges of it missing.



Arc--

Center: X,Y and Z coordinate values that determine the center position of the arc. Users can interactively adjust this value by LMB+click and dragging the handle in the 3D viewport.

Start X/Y/Z: X,Y and Z coordinate values that determine the origin angle of the arc. Users can interactively adjust this value by LMB+click and dragging the handle in the 3D viewport.

End X/Y/Z: X,Y and Z coordinate values that determine the termination angle of the arc. Users can interactively adjust this value by LMB+click and dragging the handle in the 3D viewport.

Segments: Determines the number of edge segments that define the curve. Higher values produce a smoother, more round looking arc.

Angle: Determines the total length of the arc as a number of degrees from 'Start' to 'End'. A full circle has 360°.

Radius: Determines the length or size of the arc from its 'Center' point to the edge of the arc.

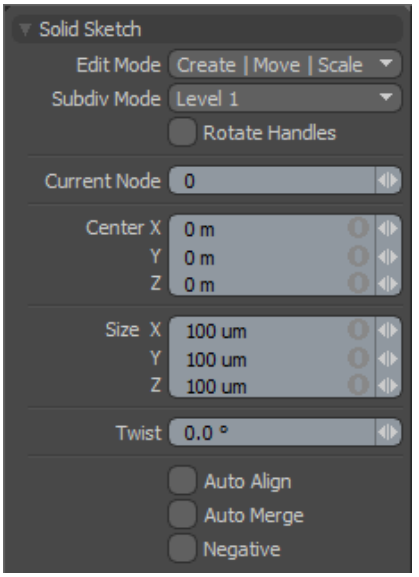
Reverse: Inverts the Arc value creating the reverse shape.

Solid Sketch

The Solid Sketch tool provides a quick method for "roughing" out organic shapes. By clicking down nodes Solid Sketch connects the dots with a smooth flowing mesh. Each node can spawn new branches to create complex shapes quickly and easily. For instance, you might create a hand by starting with a single node that branches to five new nodes each having a series of nodes that follow them to create fingers.

When a mesh is created with Solid Sketch, the tool also places two point lines inside the mesh so that when you reactivate the tool it can recover the state of the tool from these two point lines. It should be noted that any additional geometry in the mesh item at that time would be deleted as it clears all polygons prior to activating the tool.

When you click to create a node you can immediately drag to the right and left to scale the node. Once you have released the mouse button you can use the various handles to move, scale, rotate and twist the node. Additionally, holding the Shift key and dragging on the center handle will scale on all three axes.



Solid Sketch--

Move: Center handle that is always visible even when the node is not selected. This is also the handle you click on to select the node for editing. Clicking and dragging this handle will move the node in the two axes defined by the current workplane. As mentioned earlier, Shift-dragging this handle will change the behavior to scale the node on all three axes uniformly.

Rotate: The rotate handles are the Red, Green and Blue rings around the selected node. These represent rotation around the Z, Y and X axes respectively. To rotate the node simply click and drag on the desired ring handle. The affect of using the rotation handles is to actually move the children nodes of the current node as if the selected node were being reoriented.

Stretch: The red, green and blue squares just outside the rotation rings are used for scaling on the three axes individually. To scale the node on all three axes hold the Shift key while dragging the center most handle.

Twist: The yellow handle allows a simple twisting effect to the node. Twisting the node will not affect the position of the nodes connected to it.

All of these values can be controlled via the tools property form. Further there are additional controls available via the tool properties.

Edit Mode: The default value of "Create|Move|Scale" allows you to create and scale the node on initial click-drag and then use the various handles to control the node further. In this mode you can delete a node by MMB clicking directly on the node. You can also set the Edit Mode to Destroy if you want to quickly click on several nodes to destroy them.

Subdiv Mode: This setting determines how many tessellation levels to apply to the mesh using the SDS algorithm. Setting this value to 0 reveals the polygonal cage. Increasing the value to two will create additional edge loops at each node. This can be particularly useful when creating complex branching from a single node.

Auto Align: With this active each node reorients the geometry it creates to more smoothly blend with the geometry segments around it.

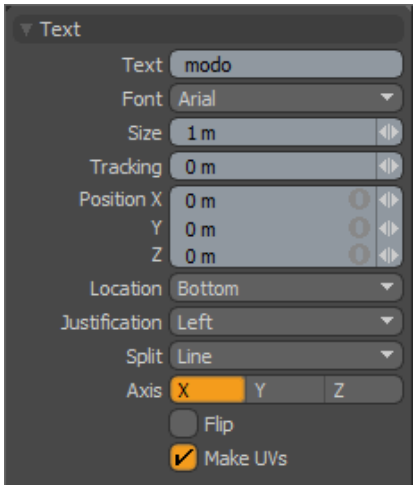
Auto Merge: Since each node typically generates a cube of polygons around it,

bunching up nodes close together would cause the mesh to pinch. The Auto Merge setting will reduce a nodes created geometry by merging it into other nodes once they are too close together.

Negative: Off by default, this option causes a node to repel neighboring geometry. This can be useful to create divots in the mesh.

Text

The Text Tool simplifies the process of creating 3D text using system fonts. To use the tool simply choose the desired font, type the text into the "Text" field and click in the 3D viewport to set the start location for the text. The Text tool creates a special polygon of "text" type. These special "text polygons" contain metadata in the form of polygon tags that store all the information about the font used, size, kerning amount, etc. As long as the text remains as this text polygon the text tool can be reactivated with the text polygon selected and any of these attributes can be edited live. When the text polygon is beveled or extruded modo will convert them to "face" polygons and the text metadata is purged.



Properties for the tool include:

Text: Enter the desired text into this field.

Font: This popup choice give access to all fonts installed on the system. The font used for the text string can be changed at anytime using this popup as long as the text remains in the native text polygon type.

Size: Use this value to set a scale for the text.

Tracking: This control allows fine tuning of the space between each letter.

Position XYZ: This XYZ triplet value determines the starting position for the text string in 3D space.

Location: The Location popup choice shifts the position of the font up or down relative to the start location. The options include Bottom, Middle and Top. The default value of Bottom sets the bottom, left most point on the font at the start location. Changing the value to Middle will position the font at it's middle (in height) and the Top setting pushes the font down such that the top-left most point on the font rests at the start location.

Justification: This value adjusts the text position to the right and left. The options

include Left, Center and Right. The default value of Left puts the left most edge of the text at the start location. Changing this value to Center will position the text such that the center of the text string rests at the start location. Using the Right option will push the text string to the left so that the right-most edge of the text rests at the start location.

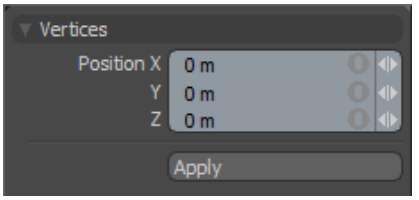
Split: The Split setting determines how many text polygons to create. By default the entire text string is a single text polygon, via the Line setting. Changing this setting to Word will break the string into a single text polygon for each discreet word as determined via the use of spaces. For instance "Hello World" would be two text polygons, one for "Hello" and one for "World". Changing the vale to Character will create a text polygon for each letter in the text string. The Word and Character settings are very useful when combining the text tool with various generators such as Curve or Scatter as they provide better granularity for shaping the text.

Axis: This control determines the direction the text will face.

Flip: This toggle determines the polygon normal direction for the text polygon faces.

Make UVs: When this value is enabled (*default*) modo will create a UV map for all polygons created.

Vertex Tool



Found in the 'Vertex' subtab of the modeling toolbox in modo, the Vertex tool can be used to quickly plot points in 3D space. To use the tool simply click the left mouse button on a 3D viewport to create a point under the mouse. One should notice that the point won't just draw anywhere in the viewport, but will draw where the mouse pointers position intersects the Work Plane. In a perspective viewport, changing the point of view can modify the Work Plane which in turn changes where points will be created. To change the position of the most recent point, either LMB+click directly on the current vertex and drag or hold down the 'Shift' key and drag anywhere in the 3D viewport. To numerically edit the most recently created point, you can change the X,Y, or Z values in the tools properties panel by adjusting the 'Position; attribute.

On their own, single point vertices won't render, but they can be useful in creating object positions when used as a point source for a 'Replicator' item. There are also some commands that complement the Vertex tool for making geometry.

Make Polygons--

The 'Make Poly' command will use the currently selected vertices to create a polygon. The order in which the vertices were created (or selected) will determine the direction of the polygons normal. When run with an edge loop selection, a new polygon will be created from those edges and selection order is not important. This command can be triggered by pressing the 'P' key on the keyboard, or on the menu bar under under 'Geometry > Polygon > Make...' which offers several additional options --

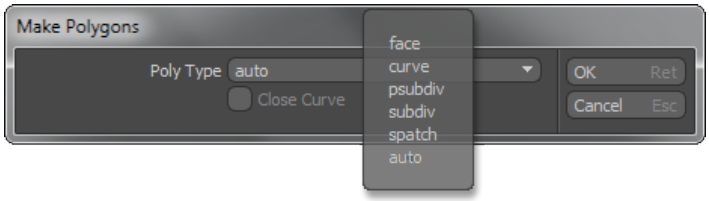
Face-- Produces a regular polygon based on the drawn vertices.

Curve-- Generates a spline curve from the points with the additional option to 'Close Curve' joining the in and out vertices into a loop.

Psubdiv/Subdiv-- Generates a Pixar 'Catmull-Clark' Subdivision Surfaces polygon eliminating the need to toggle the newly created geometry manually.

Spatch-- Produces an spline patch curve for further patching, requires 3 or 4 vertices.

Auto-- The 'Auto' option matches the polygon type from a neighbor polygon connected to the selected vertex. If the selected vertices are floating, modo chooses the 'Face' polygon type.



Make Open Curve--

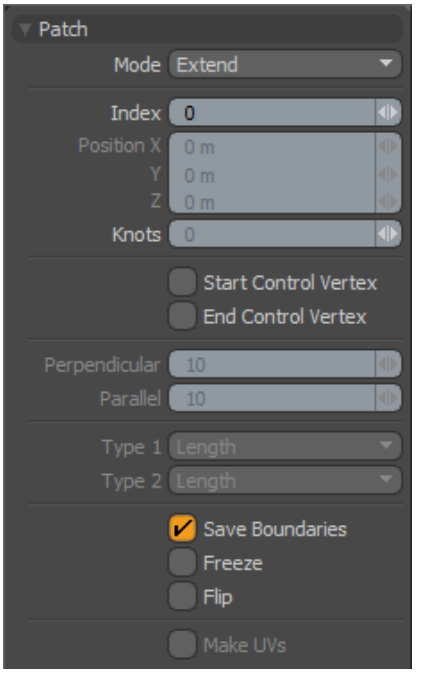
The 'Make Open Curve' command will use the currently selected vertices to generate an open spline curve. When using the 'Vertex Tool', the first drawn vertex determines the base of the curve. If created from a selection of vertices, the first vertex selected would then become the base. Determining the base of a curve is important for 'Fur Guides'. The command may be invoked by pressing the "Shift+O" keyboard shortcut, or from the Curve Palette available from the menu bar 'Geometry > Curve Palette'.

Make Closed Curve--

The 'Make Closed Curve' command will use the currently selected vertices to generate a closed spline curve joining the first and last vertices into a continuous loop. The command may be invoked by pressing the "Ctrl+P" keyboard shortcut, or from the Curve Palette available from the menu bar 'Geometry > Curve Palette'.

TIP: Sometimes vertices may not be visible in the viewport until selected, this can be easily remedied by changing the visibility option. Hover the mouse pointer over the window you wish to modify visibility for and press the 'o' key, this opens the drawing and visibility options panel, toggle the 'Show Vertices' option under the 'Visibility' subtab to make them visible. This a per *viewport* setting.

Patch Curves



Patching is a means of generating a smooth surface from 3 or 4 curves that define the contour edges of the surfaces. Multiple patches can be combined to create a closed volume (like a contoured bottle) The tool can be found under the 'Duplicate' tab of the toolbox, or within the Curves palette. To make a Spline Patch do the following:

1) Use the Curve drawing tool to draw curves that cross correctly (i.e. share endpoints).

2) On the default interface, under the 'Model' or 'Model Quad' layouts, users can find the modeling tools toolbar, click the 'Duplicate' sub tab.

3) Select the 'Patch Curves' tool, it can be found between the 'Bridge' and 'Curve Extrude' buttons.

4) In the viewport click the curve edges, by clicking on a sequence of edges, the user can define a patch.

5) Edges can also be clicked and dragged extending patch edges if necessary.

You can interactively adjust the control points on the curve while the patches are remade to follow the new contours. You can pick edges and set the number of segments for that edge as the patching updates to match. Patching will generate UVs that follow the grid of quads generated for each patch.

Mode: Users can interactively select edges to define the patch surface.

Extend- Grap and drag edge handles to extend patch edges.

Move Knot- Interactively adjust specific knots in the patch.

Move Edge- Move selected edge.

Edit Edge- Select edges and modify the number of knots.

Knots: Defines the number of divisions along the length of a curve for editing.

Start/End Control Vertex: The flag for side curves to extend allowing control over curvature for end segments.

Perpendicular: The number of divisions along a perpendicular side.

Parallel: The number of divisions along a parallel side.

Save Boundaries: Retains the boundary curves.

Freeze: When the tool is dropped, the 'patch' surface becomes editable polygons.

Flip: Reverses the normal direction of the patched surface.

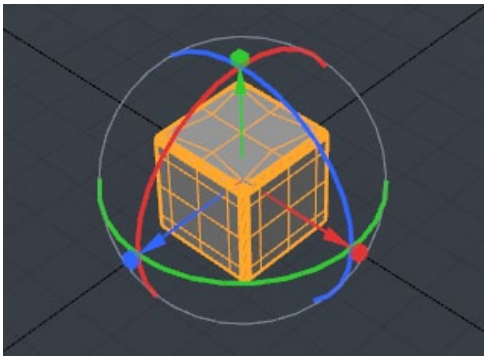
Make UVs: Generates an automatic UV maps for the patched surface for texturing.

Transforming Geometry

The transform tools in modo offers users the ability to quickly, easily, and most importantly, accurately apply transformations to items in modo, whether *hauling* or applying values numerically. All the transform tools are universal in how they work, respecting the settings for Falloffs, Action Centers and the Work Plane, and can be applied in any of the various modes- component modes 'Vertices', 'Edges' and 'Polygons', 'Items', 'Center', 'Pivot' and even 'Materials'. For information on using a specific tool, click on any of the link options below to be directed to the proper page.

Transform			
Transform	Rotate	Scale	Tack
Move	Axis Rotate	Uniform Scale	Center Commands

Transform



The Transform tool combines Move, Rotate and Scale transformations into a single tool. By using the various handles the user can quickly Position, Rotate and Scale from one convenient tool. Holding the 'Ctrl' key will constrain the position transform to a single axis when hauling, and the rotation transform to 15° increments. After applying a transform amount, users may press the MMB to re-apply the previous offset amount, making it easy to apply an edit multiple times. The command can be activated in the 'Basic' subtab of the Modeling viewports, or by pressing the 'Y' keyboard shortcut.

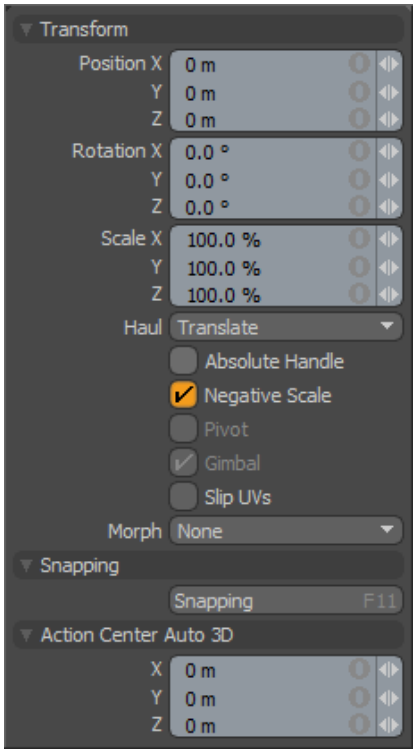
In the component modes, 'Vertices', 'Edges', and 'Polygons', the tool affects the selected geometry directly; when nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected (*highlighted*) in the 'Item List'. In 'Items' mode the tool affects all the currently selected items in the Item List, if nothing is selected, then the tools will appear to do nothing. In material mode, it works the same way as in any of the component modes. Lastly, in the 'Center and 'Pivot' modes it modifies the Center or Pivots respectively. Note that users will need to physically *select* either the Center or the Pivot itself, depending on the mode, prior to activating the tool, as in Items mode, with no selection the tool will appear to do nothing.

When users define any of the various Falloffs modo provides, application of the tool can be attenuated across a user defined area. For more on using Falloffs, please reference the Selection Falloffs page of the documentation.

Once the tool becomes active, modo draws the relevant tool handles within the viewport- arrows for *moving*, boxes for *scaling* and circles for *rotating*. At this point, users may simply click and haul anywhere inside the desired viewport to interactively transform the geometry. Additionally, users may prefer to utilize the various tool handles, constraining the application of the function to one or two axes. The various tool handles are color coded to match the coloring of the different axes, red for X,

green for Y and blue for Z. Hover the mouse over a specific handle until it changes to a yellow color and then LMB-click and drag to modify. To interactively scale the size of the handles themselves, users may use the ' - ' and ' = ' keys on the keyboard to adjust them; minus makes them smaller, while equals makes them larger.

The Transform tool also respect the settings for 'Action Center', and will originate the appropriate action from the defined position. For more information on understanding and using Actions Centers, please reference the Action Center page of the documentation



Transform--

Position: Users may use the X,Y and Z input fields to apply specific offset distance values (calculated from position the object was in when the tool was activated). When adjusting object interactively in the viewport, the 'Position' input fields will display the current offset distance applied, providing useful user feedback.

Rotation: Users may use the X,Y and Z input fields to apply specific rotation values. When adjusting object interactively in the viewport, the 'Rotation' input fields will display the current offset distance applied, providing useful user feedback.

Scale: Users may used the X,Y and Z input fields to apply specific scaling values. When adjusting object interactively in the viewport, the 'Scale' input fields will display the current scale factor applied, providing useful user feedback.

Haul: Users have three options available when hauling in the viewport (i.e. dragging the mouse when not using the tool handles), 'Translate' will allow the users to move the selection, 'Rotate' allows the user to rotate, and 'Scale' would scale the selection.

Absolute Handle: The 'Absolute Handle' toggle applies an additional function to the tool handles allowing users to scale the handle itself without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.

Negative Scale: The 'Negative Scale' toggle enabled/disabled modo's ability to interactively apply negative scale values when manipulating the tool handles in the

viewport. When disabled, users can always input negative values in any of the input fields directly.

Pivot: The 'Pivot' toggle allows users to easily modify an items pivot position without the need to go into Pivots mode. The 'Gimbal' option must be disabled in order to adjust the 'Pivot' option.

Gimbal: With the 'Gimbal' toggle *enabled*, users can assign rotation values numerically, but may arrive at a situation that produces gimbal lock, the loss of one rotation axis when another's value is too close. Users can avoid this by *disabling* this setting, but are then unable to apply specific values within the input fields.

Slip UVs: (*Only in Component Modes*) UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can enable the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes*) The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

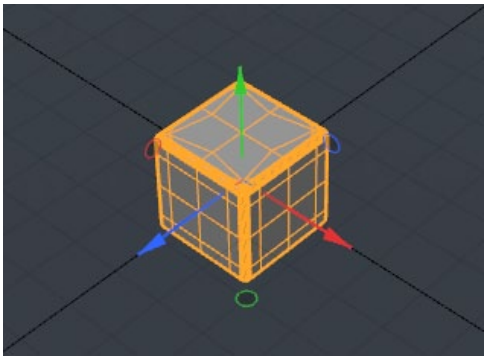
None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Child Compensate: (*Only in Items Mode*) The 'Child Compensate' toggle when *disabled* will pass along the transform value to any child items of the current selection, when *enabled*, the child items will ignore any transform values.

Move



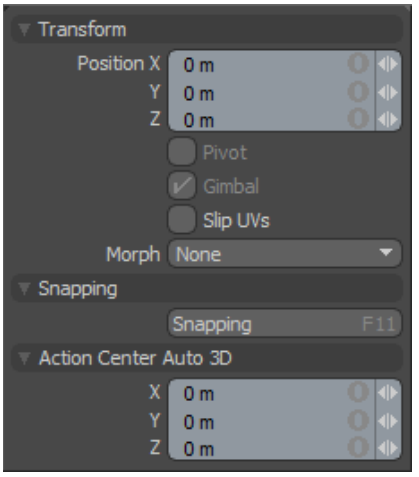
The Move tool obviously allows users to move, or *translate* a selection across a specific distance. By using the various handles the user can easily constrain the movement to one or two axes. Holding the 'Ctrl' key will constrain to a single axis when hauling. After applying a transform amount, users may press the MMB to re-apply the previous offset amount, making it easy to apply an edit multiple times. The command can be activated in the 'Basic' subtab of the Modeling viewports, in the menu bar under 'Edit >Transform > Move' or by pressing the 'W' keyboard shortcut.

In the component modes, 'Vertices', 'Edges', and 'Polygons', the tool affects the selected geometry directly; when nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected (*highlighted*) in the 'Item List'. In 'Items' mode the tool affects all the currently selected items in the Item List, if nothing is selected, then the tools will appear to do nothing. In material mode, it works the same way as in any of the component modes. Lastly, in the 'Center and 'Pivot' modes it modifies the Center or Pivots respectively. Note that users will need to physically *select* either the Center or the Pivot itself, depending on the mode, prior to activating the tool, as in Items mode, with no selection the tool will appear to do nothing.

When users define any of the various Falloffs modo provides, application of the tool can be attenuated across a user defined area. For more on using Falloffs, please reference the Selection Falloffs page of the documentation.

Once the tool becomes active, modo draws the relevant tool handles within the viewport. At this point, users may simply click and haul anywhere inside the desired viewport to interactively position the geometry. Additionally, users may prefer to utilize the various tool handles, constraining the application of the tool, the arrows to a single axis, and the circles to to two axes. The various tool handles are color coded to match the coloring of the different axes, red for X, green for Y and blue for Z. Hover the mouse over a specific handle until it changes to a yellow color and then LMB-click and drag to modify. To interactively scale the size of the handles themselves, users may use the ' - ' and ' = ' keys on the keyboard to adjust them; minus makes them smaller, while equals makes them larger.

The Transform tool also respect the settings for 'Action Center', and will originate the appropriate action from the defined position. For more information on understanding and using Actions Centers, please reference the Action Center page of the documentation



Transform

Position: Users may use the X,Y and Z input fields to apply specific offset distance values (calculated from position the object was in when the tool was activated). When adjusting object interactively in the viewport, the 'Position' input fields will display the current offset distance applied, providing useful user feedback.

Pivot: The 'Pivot' toggle allows users to easily modify an items pivot position without the need to go into Pivots mode.

Gimbal: The Gimbal setting is grayed out in *move* mode as it only applies to rotations.

Slip UVs: (*Only in Component Modes*) UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can enable the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes*) The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Child Compensate: (*Only in Items Mode*) The 'Child Compensate' toggle when *disabled* will pass along the transform value to any child items of the current selection, when *enabled*, the child items will ignore any transform values.

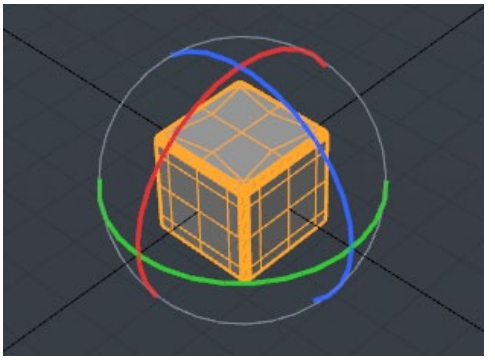
Snapping--

Please reference the 'Snapping' Section of the documentation.

Action Center Auto 3D--

Please reference the 'Action Center' section of the documentation.

Rotate



The Rotate tool obviously allows users to rotate a selection by a specific number of degrees. By using the various handles the user can easily constrain the movement to a single axis. Holding the 'Ctrl' key will constrain the the rotation to fixed 15° intervals. After applying a transform amount, users may press the MMB to re-apply the previous offset amount, making it easy to apply an edit multiple times. The command can be activated in the 'Basic' subtab of the Modeling viewports, in the menu bar under 'Edit >Transform > Rotate' or by pressing the 'E' keyboard shortcut.

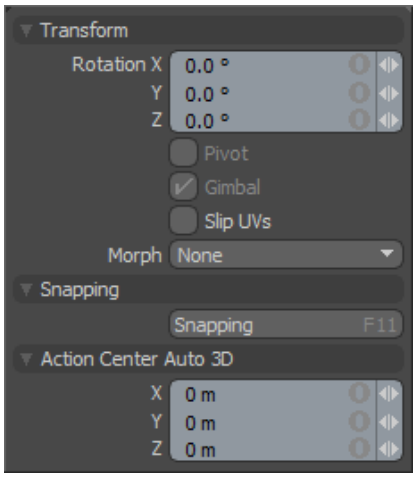
In the component modes, 'Vetices', 'Edges', and 'Polygons', the tool affects the selected geometry directly; when nothing is selected, it affects all active forground geometry. Active foreground items are those that are visible and selected (*highlighted*) in the 'Item List'. In 'Items' mode the tool affects all the currently selected items in the Item List, if nothing is selected, then the tools will apppear to do nothing. In material mode, it works the same way as in any of the component modes. Lastly, in the 'Center and 'Pivot' modes it modifies the Center or Pivots respectively. Note that users will need to physically *select* either the Center or the Pivot itself, depending on the mode, prior to activating the tool, as in Items mode, with no selection the tool will appear to do nothing.

When users define any of the various Falloffs modo provides, application of the tool can be attenuated across a user defined area. For more on using Falloffs, please reference the Selection Falloffs page of the doumentation.

Once the tool becomes active, modo draws the relevant tool handles within the viewport. At this point, users may simply click and haul anywhere inside the desired viewport to interactively rotate the geometry. Additionally, users may prefer to utilize the various tool handles, constraining the application of the tool to a single axis. The tool handles are color coded to match the coloring of the different axes, red for X, green for Y and blue for Z. In some action center modes, there is an additional gray *screen space*

handle around the outside of the tool handles, which rotates the selection based on the viewport cameras postion. Hover the mouse over a specific handle until it changes to a yellow color and then LMB-click and drag to modify. To interactively scale the size of the handles themselves, users may use the ' - ' and ' = ' keys on the keyboard to adjust them; minus makes them smaller, while equals makes them larger.

The Rotate tool also respect the settings for 'Action Center', and will originate theaction from the defined position. For more information on understanding and using Actions Centers, please reference the Action Center page of the documentation.



Transform

Rotation: Users may use the X,Y and Z input fields to apply specific rotation values. When adjusting object interactively in the viewport, the 'Rotation' input fields will display the current offset distance applied, providing useful user feedback.

Pivot: The 'Pivot' toggle allows users to easily modify an items pivot position without the need to go into Pivots mode.

Gimbal: With the 'Gimbal' toggle enabled, users can assign rotation values numerically, but may arrive at a situation that produces gimbal lock, the loss of one rotation axis when another's value is too close. Users can avoid this by disabling this setting, but are then unable to apply specific values within the input fields.

Slip UVs: (Only in Component Modes) UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can *enable* the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes*) The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Child Compensate: (*Only in Items Mode*) The 'Child Compensate' toggle when *disabled* will pass along the transform value to any child items of the current selection, when enabled, the child items will ignore any transform values.

Axis Rotate

The Axis Rotate tool obviously allows users to rotate a selection by a specific number of degrees. The Axis Rotation tool is different from the standard Rotate tool in that it is limited to editing a single axis at a time, determined by the Work Plane position. It also has a different on-screen control display than the other transform tools providing a greater degree of control over the amount of rotation depending on how long or short the user drags out the handle length. Holding the 'Ctrl' key will constrain the the rotation to fixed 15° intervals. After applying a transform amount, users may press the MMB to re-apply the previous offset amount, making it easy to apply an edit multiple times. The command can be activated in the 'Basic' subtab of the Modeling viewports, users may need to press the 'Shift' key to show alternate tool. User may also activate the tool from the menu bar under 'Edit >Transform > Axis Rotate' or by pressing the 'Alt+Shift+E' keyboard shortcut.

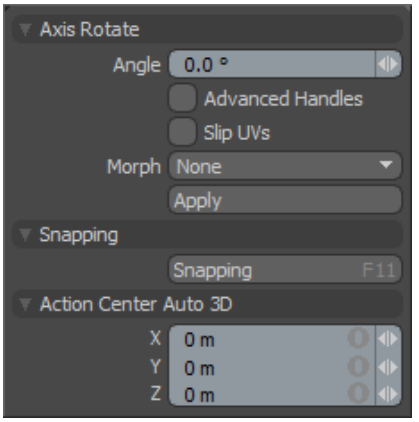
In the component modes, 'Vertices', 'Edges', and 'Polygons', the tool affects the selected geometry directly; when nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected (*highlighted*) in the 'Item List'. In 'Items' mode the tool affects all the currently selected items in the Item List, if nothing is selected, then the tools will appear to do nothing. In material mode, it works the same way as in any of the component modes. Lastly, in the 'Center and 'Pivot' modes it modifies the Center or Pivots respectively. Note that users will need to physically *select* either the Center or the Pivot itself, depending on the mode, prior to activating the tool, as in Items mode, with no selection the tool will appear to do nothing.

When users define any of the various Falloffs modo provides, application of the tool can be attenuated across a user defined area. For more on using Falloffs, please reference the Selection Falloffs page of the documentation.

Once the tool becomes active, modo draws the relevant tool handle within the viewport.

At this point, users may simply click and haul anywhere inside the desired viewport to interactively rotate the geometry, depending on the action center, this may also set the rotation center based on the mouse click position intersecting the Work Plane. Additionally, users may prefer to utilize the on screen display, either clicking on the wheel to rotate to a specific amount, or by dragging the handles on the value bar across the top of the viewport (when 'Advanced Handles' is enabled). When clicking on the wheel, a handle is drawn displaying the number of degrees. When using this handle the rotation sensitivity is adjusted based on the distance the mouse is from the handles as you drag. This can be thought of as extending a lever. The longer the lever, the more mouse movement is required to rotate the element and thus the granularity of control is increased. To interactively scale the size of the handles themselves, users may use the ' - ' and ' = ' keys on the keyboard to adjust them; minus makes them smaller, while equals makes them larger.

The Rotate tool also respect the settings for 'Action Center', and will originate the action from the defined position. For more information on understanding and using Actions Centers, please reference the Action Center page of the documentation.



Axis Rotate--

Angle: Users may use the 'Angle' input field to apply specific rotational value, calculated in degrees. When adjusting object interactively in the viewport, the 'Angle' input fields will display the current rotation amount, providing useful user feedback.

Advanced Handles: Determines whether or not modo displays the additional information and controls in the viewport.

Slip UVs: (*Only in Component Modes*) UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can *enable* the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes*) The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

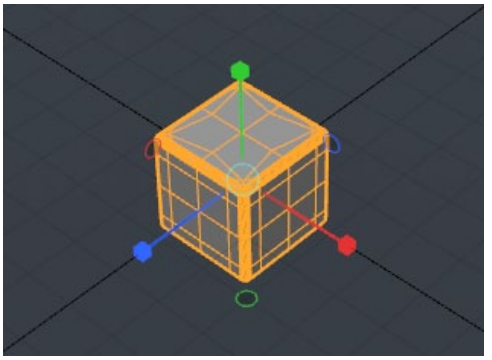
None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Child Compensate: (*Only in Items Mode*) The 'Child Compensate' toggle when *disabled* will pass along the transform value to any child items of the current selection, when *enabled*, the child items will ignore any transform values.

Scale



The Scale tool allows users to make any selection larger or smaller by changing the scale percentage factor. By using the various handles the user can quickly scale in a single axis dimension. After applying a transform amount, users may press the MMB to re-apply the previous offset amount, making it easy to apply an edit multiple times. The command can be activated in the 'Basic' subtab of the Modeling viewports, in the menu bar under 'Edit >Transform > Scale' or by pressing the 'R' keyboard shortcut.

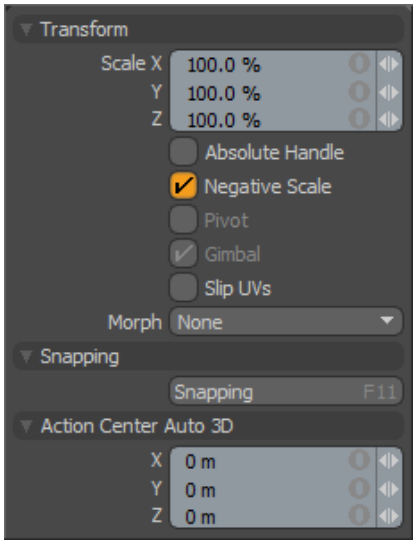
In the component modes, 'Vertices', 'Edges', and 'Polygons', the tool affects the selected geometry directly; when nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected (*highlighted*) in the 'Item List'. In 'Items' mode the tool affects all the currently selected items in the Item List, if nothing is selected, then the tools will appear to do nothing. In material mode, it works the same way as in any of the component modes. Lastly, in the 'Center and 'Pivot' modes it modifies the Center or Pivots respectively. Note that users will need to physically *select* either the Center or the Pivot itself, depending on the mode, prior to activating the tool, as in Items mode, with no selection the tool will appear to do nothing.

When users define any of the various Falloffs modo provides, application of the tool can be attenuated across a user defined area. For more on using Falloffs, please reference the Selection Falloffs page of the documentation.

Once the tool becomes active, modo draws the relevant tool handles within the viewport. At this point, users may simply click and haul anywhere inside the desired viewport to interactively transform the geometry. Additionally, users may prefer to utilize the various tool handles, constraining the application of the function to one or all three axes. The various tool handles are color coded to match the coloring of the different axes, red for X, green for Y and blue for Z. Hover the mouse over a specific handle until it changes to a yellow color and then LMB-click and drag to modify. To

interactively scale the size of the handles themselves, users may use the ' - ' and ' = ' keys on the keyboard to adjust them; minus makes them smaller, while equals makes them larger. Additionally for the Scale tool, there is a small circle at the intersection of all three handles that users can use to uniformly scale the selection interactively.

The Transform tool also respect the settings for 'Action Center', and will originate the action from the defined position. For more information on understanding and using Actions Centers, please reference the Action Center page of the documentation.



Transform

Scale: Users may used the X,Y and Z input fields to apply specific scaling values. When adjusting object interactively in the viewport, the 'Scale' input fields will display the current scale factor applied, providing useful user feedback.

Absolute Handle: The 'Absolute Handle' toggle applies an additional function to the tool handles allowing users to scale the handle itself without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.

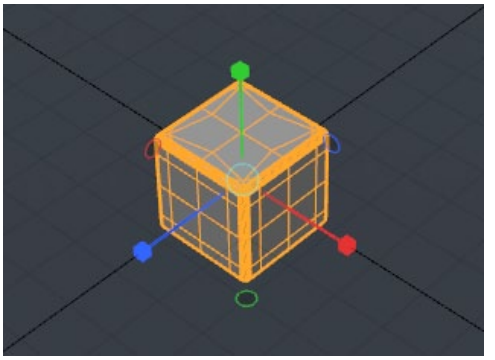
Negative Scale: The 'Negative Scale' toggle enabled/disabled modo's ability to interactively apply negative scale values when manipulating the tool handles in the viewport. When disabled, users can always input negative values in any of the input fields directly.

Pivot: The 'Pivot' toggle allows users to easily modify an items pivot position without the need to go into Pivots mode.

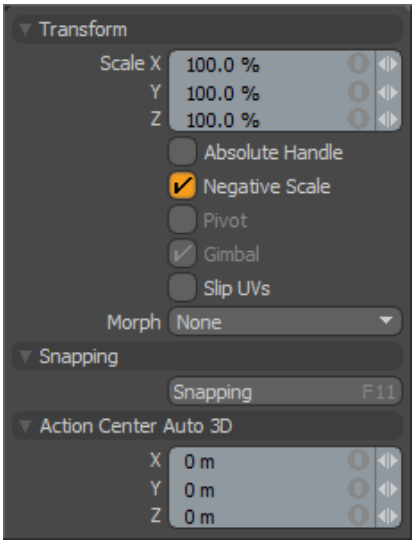
Gimbal: The Gimbal setting is grayed out in *scale* mode.

Child Compensate: The 'Child Compensate' toggle when disabled will pass along the transform value to any child items of the current selection, when enabled, the child items will ignore any transform values.

Uniform Scale



The Uniform Scale tool is identical to the normal Scale tool (above), except that users may haul anywhere in the viewport and all three axes are constrained forcing all scaling operating to be uniform. This ensures the mesh will not become distorted when scaling.



Transform

Scale: Users may used the X,Y and Z input fields to apply specific scaling values. When adjusting object interactively in the viewport, the 'Scale' input fields will display the current scale factor applied, providing useful user feedback.

Absolute Handle: The 'Absolute Handle' toggle applies an additional function to the tool handles allowing users to scale the handle itself without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.

Negative Scale: The 'Negative Scale' toggle enabled/disabled modo's ability to interactively apply negative scale values when manipulating the tool handles in the viewport. When disabled, users can always input negative values in any of the input fields directly.

Pivot: The 'Pivot' toggle allows users to easily modify an items pivot position without the need to go into Pivots mode.

Gimbal: The Gimbal setting is grayed out in scale mode.

Slip UVs: (*Only in Component Modes*) UV values are generally fixed to specific vertices,

subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can *enable* the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes*) The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Child Compensate: (*Only in Items Mode*) The 'Child Compensate' toggle when *disabled* will pass along the transform value to any child items of the current selection, when *enabled*, the child items will ignore any transform values.

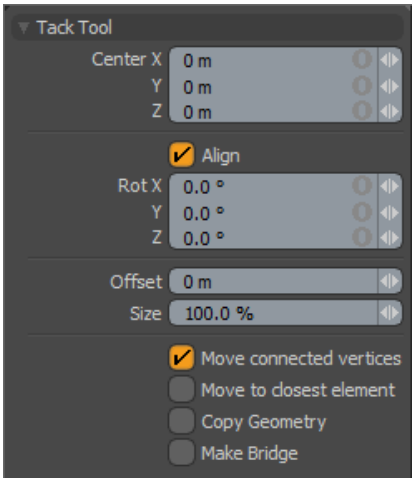
Tack

The Tack Tool allows you to quickly move and rotate a polygon and/or all connected vertices to any other polygon in the model. To use the tool simply choose a single polygon in the mesh you wish to move and re-align. With the tool active, click on another polygon in your project. The initial mesh will move and rotate so that the selected polygon is aligned to the polygon that you clicked on.

Options for the tool are:

Center: This triplet is automatically set with each click of the tool. The value will represent the 3D coordinates of the element clicked on. After the value has been set automatically it is possible to edit the values directly to move the mesh on any of the

three axes.



Align: With this option active the mesh is moved and rotated to align to the surface normal of the target element. Disabling this option will result in the mesh only moving to the target element.

Rotate: This triplet is automatically set with each click of the tool but only when the Align option is active. After the value has been set automatically it can be manually edited to offset the rotation from the element normal.

Offset: Users may use this value to determine an offset distance perpendicular to the surface geometry is tacked on to.

Size: Users may use this value to scale the tacked geometry.

Move connected vertices: By default this option is active so that the selected polygon and all connected vertices will be moved and aligned to the target element. When this option is disabled only the selected polygon will be transformed.

Move to Closest Element: Snaps the geometry centered onto the closest component element -polygon, edge or vertex.

Copy Geometry: Duplicates the geometry when tacking instead of simply moving the original. Simultaneously holding down 'Shift' when clicking will make a copy per mouse click.

Make Bridge: When this setting is enabled, modo will attempt a Bridge operation between the selected polygon and the polygon the user clicked to tack the object, making a single connected mesh from the two surfaces.

Center Commands

modo offers a number of centering commands that allow users to quickly Center a geometry selection at the world origin (0,0,0) position. The commands can be found in the modeling tabs under the basic subtab - 'Center Selection'. Clicking the menu button provides a popup with options that constrain to specific axes, or 'All'. The commands can also be invoked from the menu bar 'Edit > Center Selected'. There is no dialog box or user specified setting for these commands.

The behavior of Center depends on the selected data and the command arguments

used. If a single vertex is selected, Center will move the vertex to position 0 for the defined axis (or axes). When an edge is selected the center point of the edge will be placed at the centered axis (or axes). With a polygon selected the polygon will be moved so that the center of the polygon rests at the center of the defined axis (or axes). When there are more than one vertex, edge or polygon selected, the center of the bounding volume of that selection will be placed at the center of the defined axis (or axes).

The Center commands are meant to work on component selections. Should users wish to center an item while in item mode, they should use the 'Reset' function available in the items properties, returning that item layers center to the world origin. Users should note that this will only center the item itself, and will not modify the geometry within the item layer.

IMPORTANT: If users apply the Center commands to a *mesh item* that already has a transformation applied, it is recommended the user also apply the freeze option in the items properties panel, returning its center to the world origin. If they don't, strange behavior will likely occur.

Deforming Geometry

modo's various deform tools offer users the means to produce soft organic deformations such as with Soft Move and Sculpt or even hard mechanical forms with Shear and Bend. Many of modo's deforming tools are hybrid tools combining functions of Action Centers, Falloffs and Transforms into a single tool. These tools come complements of the Toolpipe, a function of modo that allows users to make their own tools from a combination of functions. Any way you look at it though, these *tools* offer a wide array of options to contort and otherwise shape the geometry into any form the user wishes.

Deform Tools		Rotate Deform	Scale Deform
Element Move	Quantize	Twist	Push
Flex	Symmetry Tool	Bend	Sculpt
Soft Move	Soft Sel. Move	Vortex	Taper
Soft Drag	Soft Sel. Rotate	Swirl	Bulge
Shear	Soft Sel. Scale		Flare
Smooth	Soft Sel. Transform		
Jitter			

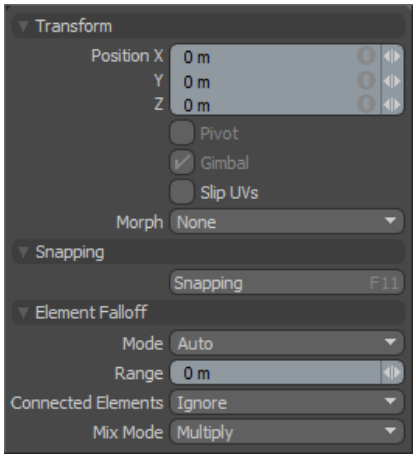
Element Move

The Element Move tool provides users with a very fast method for editing individual component elements a mesh (either vertices, edges and polygons). The tool uses the first mouse click to determine the element directly under the mouse. Pre-highlighting of the element will prompt the user which element will actually be transformed when the mouse button is clicked. When over the proper element, users can simply LMB+click drag the mouse to move the element in 3D space, or you can release the mouse button and use the handles to move the element along that axis.

There are two modes for the tool, 'Manual' and 'Automatic'. By default the tool works in Automatic mode which means the element type to be edited is determined at mouse down. If you click on an edge you will edit the edge and if you click on a polygon you will edit the polygon. This allows you to edit components of the mesh independently of the

current selection state. The Manual mode forces the tool to act only on the component type defined by the current global selection mask. If selection is set to Vertices and the tool is in manual mode, you will only be able to edit vertices.

The Element Move tool also has a Falloff value which will create a spherical area of influence around the element under the mouse. This value can be set via the properties panel for the tool or by RMB dragging in the 3D viewport. The falloff is represented in the 3D view by a yellow wireframe sphere so you can interactively adjust the region.



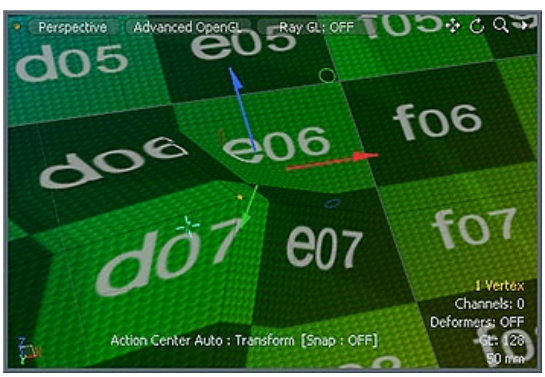
Transform--

Position: When adjusting object interactively in the viewport, the 'Position' input fields will display the current offset distance applied, providing useful user feedback. Users may use the X,Y and Z input fields to apply specific offset distance values (calculated from position the object was in when the tool was activated).

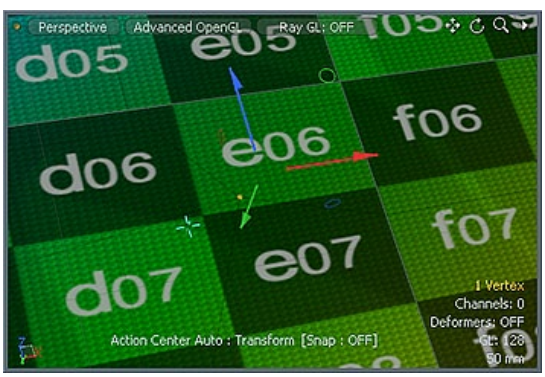
Pivot: Disabled for the Element Move tool.

Gimbal: Disabled for the Element Move tool.

Slip UVs: (Only in Component Modes) UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can enable the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.



'Slip UVs' function disabled, note texture warping.



'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes*) The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Snapping--

Please reference the 'Snapping' Section of the documentation.

Element Falloff--

Mode: Here users can define the automatic center that the transform offsets from when applied, as well as the element type selected when hovering. 'Auto' is the default and will select whatever element the mouse is over, positioning the center where the mouse pointer intersects the surface it is modifying. Other options allow users finer control.

Range: With the 'Range' setting, users may define a soft falloff region that will attenuated the transform over the ranges distance, defined as a distance.

Connected Elements: Defines what connected geometry will do when an element is transformed.

Ignore-- Ignores all connected elements and only moves the element specified.

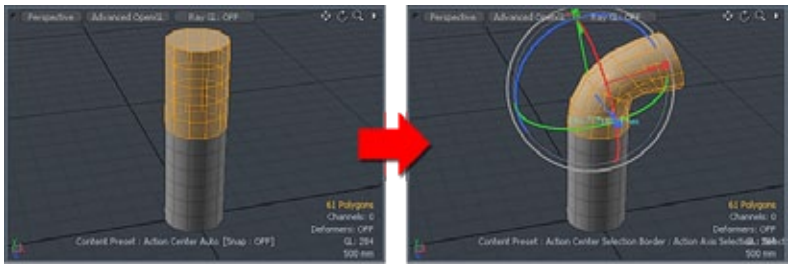
Use Connectivity-- Only affects single surface/connected elements, unconnected elements within range will be ignored.

Rigid Connections-- Moves all connected elements equally the specified distance.

Edge Loops-- Moves the connected loop, defined by all the connected quad polygons/edges/vertices in a single row.

Mix Mode: In instances where there are multiple falloffs applied to a transform, the mix mode defines how each interacts with the other.

Flex



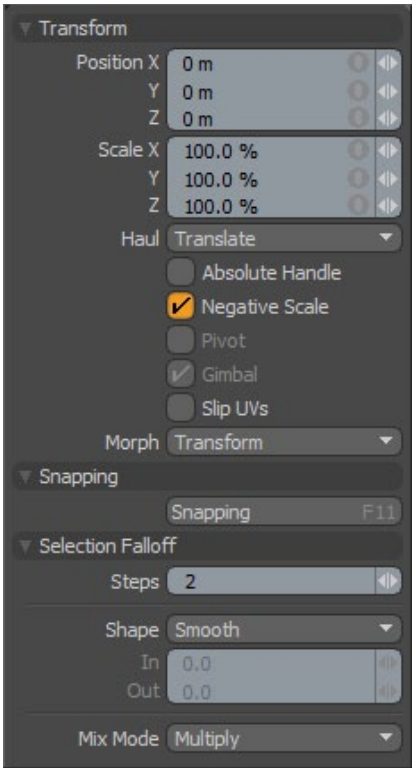
Found in the 'Deform' subtab of the modeling toolbox, the Flex tool leverages the 'Selection Border' action center and 'Selection' based falloff combined with the standard Transform tool to create a tool that provides a more natural, organic bending type deformation. Used in combination with 'Select Closed Loop' command, it provides the ability to pose meshes very simply. The tool has the standard transform widget which allows for direct manipulation of move, rotate and scale for the currently selected elements. The amount of transform applied to the selection is based on the Steps parameter. The default value of two results in a smooth falloff starting at the boundary of the selection. The tools control palette includes direct numeric input of all transform values as well as a Haul pop-up to determine what users wish to 'Haul', Absolute Handle option, Negative Scale choice, and direct control over the Selection falloff options.

Transform--

Position: Users may use the X,Y and Z input fields to apply specific offset distance values (calculated from position the object was in when the tool was activated). When adjusting object interactively in the viewport, the 'Position' input fields will display the current offset distance applied, providing useful user feedback.

Scale: Users may used the X,Y and Z input fields to apply specific scaling values. When adjusting object interactively in the viewport, the 'Scale' input fields will display the current scale factor applied, providing useful user feedback.

Haul: Users have three options available when hauling in the viewport (i.e. dragging the mouse when not using the tool handles), 'Rotate' allows the user to rotate, and 'Scale' would re-size the selection.



Absolute Handle: The 'Absolute Handle' toggle applies an additional function to the tool handles allowing users to scale the handle itself without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.

Negative Scale: The 'Negative Scale' toggle enabled/disabled modo's ability to interactively apply negative scale values when manipulating the tool handles in the viewport. When disabled, users can always input negative values in any of the input fields directly.

Position Only: (*Only in Items Mode*) When enabled, only changes the position of the selected items, rotation is left unchanged.

Pivot: Disabled for the 'Flex' tool.

Gimbal: Disabled for the 'Flex' tool.

Child Compensate: (*Only in Items Mode*) The 'Child Compensate' toggle when disabled will pass along the transform value to any child items of the current selection, when enabled, the child items will ignore any transform values.

Slip UVs: (*Only in Component Modes*) UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can enable the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes*) The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Snapping--

Please reference the 'Snapping' Section of the documentation.

Selection Falloff--

Steps: This value determines the extent of the Falloff range, where each 'Step' represents a loop of polygons or a ring of edge sections. The greater the 'Steps;' value, the further the Falloff attenuates.

Shape: The strength of the Falloffs influence can be controlled along the extent using a 'Shape Preset'-

Linear- Attenuation of Falloff occurs evenly across its range.

Ease-In- Strength of Falloff is greater toward the 'Start' position.

Ease-Out- Strength of Falloff is greater toward the 'End' position.

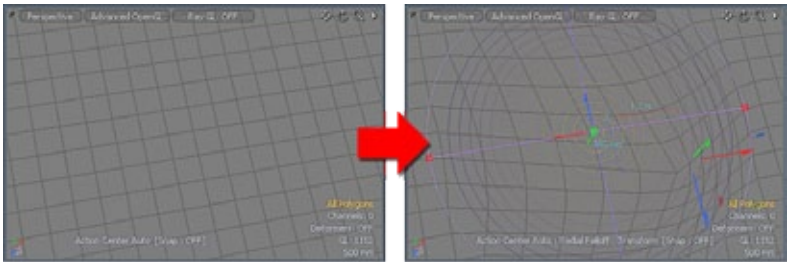
Smooth- Strength of Falloff is greater toward the center of the Falloff.

Custom- Users can use the 'In'/'Out' options to fine tune strength of Falloff.

In/Out: The 'In' value determines the strength of the Falloff nearer to the 'Start' position, where the 'Out' value determines the strength on nearer the 'End' side of the Falloff.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.

Soft Move

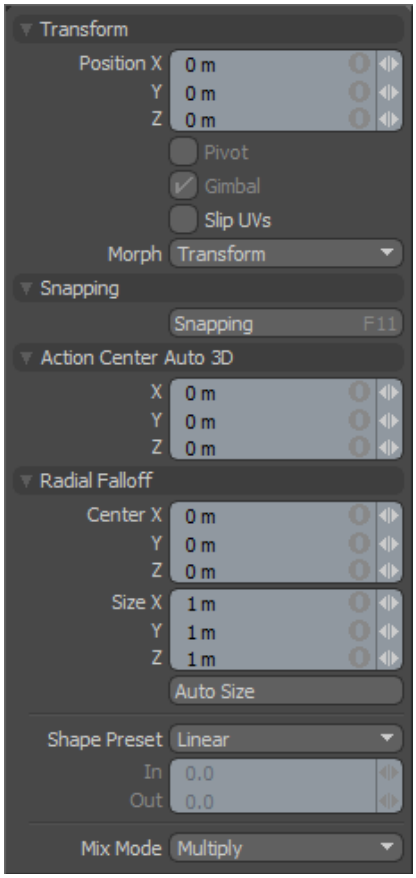


Found in the 'Deform' subtab of the modeling toolbox, the Soft Move is a tool preset that consists of the Transform tool with Radial Falloff. This lets users select vertices, edges or polygons and move them, attenuating the amount of movement to neighboring items based on the size of the radial falloff producing a soft, organic deformation. Use the right-mouse button to define the radial falloff interactively.

Transform--

Position: Users may use the X,Y and Z input fields to apply specific offset distance values (calculated from position the object was in when the tool was activated). When adjusting object interactively in the viewport, the 'Position' input fields will display the current offset distance applied, providing useful user feedback.

Pivot: Disabled for the 'Soft Move' tool.



Gimbal: Disabled for the 'Soft Move' tool.

Child Compensate: (*Only in Items Mode*) The 'Child Compensate' toggle when disabled will pass along the transform value to any child items of the current selection, when enabled, the child items will ignore any transform values.

Slip UVs: (*Only in Component Modes)* UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can *enable* the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes)* The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Snapping--

Please reference the 'Snapping' Section of the documentation.

Radial Falloff--

Center X/Y/Z: Defines the Center of influence, where the strength of the Falloff will be greatest (100%). The strength of the falloff will attenuate toward the outer bounds of the Spherical volume, the area outside the volume receives no tool influence.

Size X/Y/Z: Defines the radius of a perfect circle from the 'Center' and determines the outer area of the Falloff where there is no affect.

Auto Size: Users can select this option to automatically size the Falloff's 'Center' and 'Size' values to match the bounding box of the current selection.

Shape Preset: The strength of the Falloffs influence can be controlled along the extent using a 'Shape Preset'-

Linear- Attenuation of Falloff occurs evenly across its range.

Ease-In- Strength of Falloff is greater toward the 'Start' position.

Ease-Out- Strength of Falloff is greater toward the 'End' position.

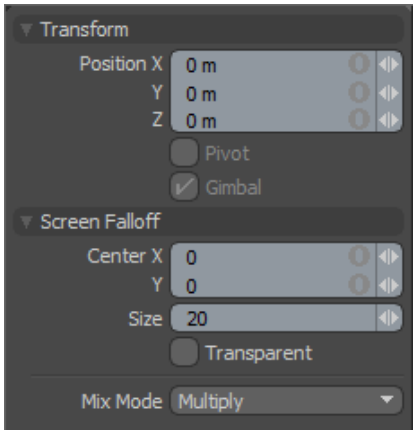
Smooth- Strength of Falloff is greater toward the center of the Falloff.

Custom- Users can use the 'In'/'Out' options to fine tune strength of Falloff.

In/Out: The 'In' value determines the strength of the Falloff nearer to the 'Start' position, where the 'Out' value determines the strength on nearer the 'End' side of the Falloff.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.

Soft Drag

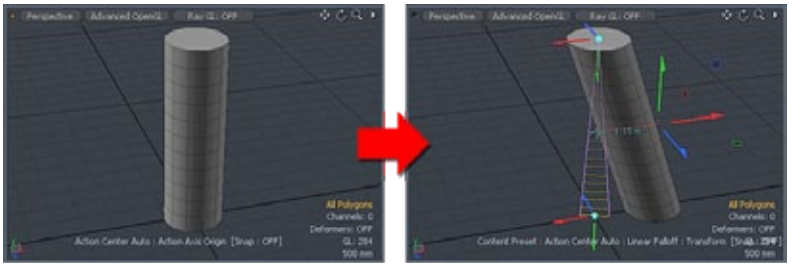


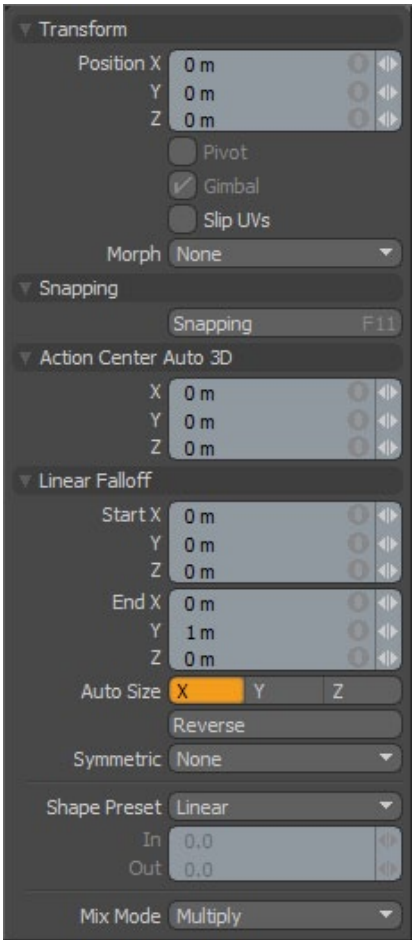
Soft Drag is a tool preset that consists of the Move tool combined with Screen Falloff. This lets you select vertices, edges or polygons and drag them, affecting their neighbors to a greater or lesser degree based on screen falloff. Use the right-mouse button to define the screen falloff circle.

You can access Soft Drag from the Deform tab of the modo Tools toolbar, or from the menubar by choosing Edit > Deform > Soft Drag.

Turn the Transparent button on in the Screen Falloff section to allow the Soft Drag selection to happen through the mesh. With this off, only geometry facing the view will be affected.

Shear





Found in the 'Deform' subtab of the modeling toolbox, the Shear tool combines the Transform tool with a Linear falloff. At first click the linear falloff is set perpendicularly to the mouse position and automatically scaled to match the currently selected geometry. Dragging the mouse at this time will deform the object with an attenuated shearing effect.

To precisely place the falloff users can either use the handles to manipulate the falloff after it has been autosized or by RMB dragging away from the handles to reset the falloff and drag it out interactively.

Transform--

Position: Users may use the X,Y and Z input fields to apply specific offset distance values (calculated from position the object was in when the tool was activated). When adjusting object interactively in the viewport, the 'Position' input fields will display the current offset distance applied, providing useful user feedback.

Pivot: Disabled for the 'Soft Move' tool.

Gimbal: Disabled for the 'Soft Move' tool.

Child Compensate: (*Only in Items Mode*) The 'Child Compensate' toggle when disabled will pass along the transform value to any child items of the current selection, when *enabled*, the child items will ignore any transform values.

Slip UVs: (*Only in Component Modes*) UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can *enable* the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes*) The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Snapping--

Please reference the 'Snapping' Section of the documentation.

Linear Falloff--

Start X/Y/Z: Defines the starting position of the Falloff as a specific XYZ coordinate value. This end of the line and everything beyond it receives the maximum amount of influence attenuating toward the 'End' position.

End X/Y/Z: Defines the end position of the Falloff as a specific XYZ coordinate value. This end of the line and everything beyond it receives no influence attenuating toward the 'Start' position.

Auto Size X/Y/Z: Users can select either of these three options to automatically scale the 'Start' and 'End' points of the Falloff to match the bounding box size of the selected elements along one of the three axes.

Reverse: Users can select this option to invert the 'Start' and 'End' points reversing the influence of the Falloff.

Symmetric: Users can choose from several options to automatically mirror the influence area of the Falloff-

None- Symmetric function is disabled

Start- Influence of Falloff is symmetrically mirrored across the 'Start' position.

End- Influence of Falloff is symmetrically mirrored across the 'End' position.

Shape Preset: The strength of the Falloffs influence can be controlled along the extent using a 'Shape Preset'-

Linear- Attenuation of Falloff occurs evenly across its range.

Ease-In- Strength of Falloff is greater toward the 'Start' position.

Ease-Out- Strength of Falloff is greater toward the 'End' position.

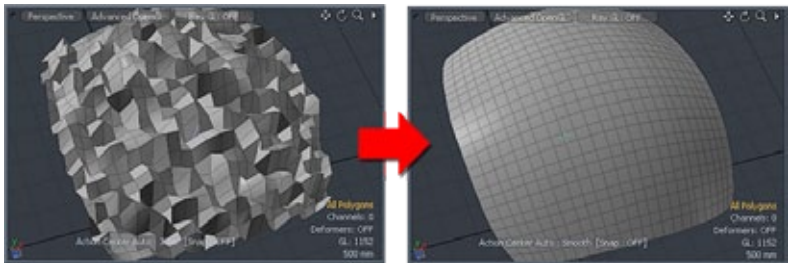
Smooth- Strength of Falloff is greater toward the center of the Falloff.

Custom- Users can use the 'In'/'Out' options to fine tune strength of Falloff.

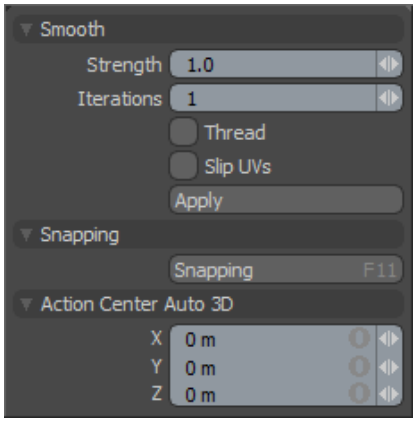
In/Out: The 'In' value determines the strength of the Falloff nearer to the 'Start' position, where the 'Out' value determines the strength on nearer the 'End' side of the Falloff.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.

Smooth



Found in the 'Deform' subtab of the modeling toolbox (sometimes as a sub-option of the 'Jitter' or 'Quantize' tools), the Smooth tool attempts to refine any jagged rough surface (also occasionally called jittered polygons). The settings to achieve this vary depending on the target geometry. The Smooth tool will operate in the 3D and UV viewports making it an excellent tool for smoothing overall topology or specific areas of a UV map. It also employs any active Falloffs (including a Background Constraint) allowing users to target specific areas for the smoothing application.



Smooth--

Strength: Controls the amount of smoothing applied to the model.

Iterations: Determines the number of times the Smoothing is applied. Combinations of both settings will control the overall smoothing of the model.

Thread: On complex models it may be useful to enabled the 'Thread' option providing faster feedback on the smoothing results, but reduces overall system interactivity as the smoothing calculations are processor intensive.

Slip UVs: UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can enable the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

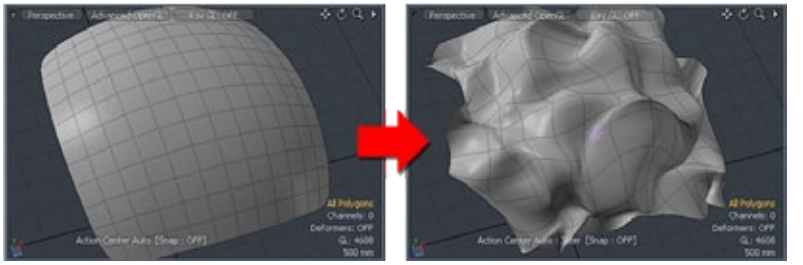
Snapping--

Please reference the 'Snapping' Section of the documentation.

Action Center Auto 3D--

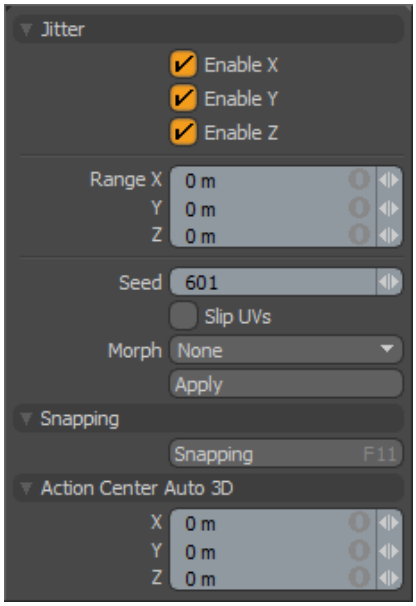
Please reference the 'Action Center' section of the documentation.

Jitter



Found in the 'Deform' subtab of the modeling toolbox (sometimes as a sub-option of the 'Smooth' or 'Quantize' tools), the Jitter tool procedurally repositions selected Items or Components based on a seed number and a range value for X, Y, and Z. Basically it is applying a random offset to each vertex or item position for a more random, organic appearance. Users can interactively apply the Jitter by dragging in the viewport, or numerically entering the values in the Tool Properties Range fields, then click the Apply button. Changing the Seed value will change the random numbers producing

a variation on the results. It also employs any active Falloffs allowing users to target specific areas for the jitter application.



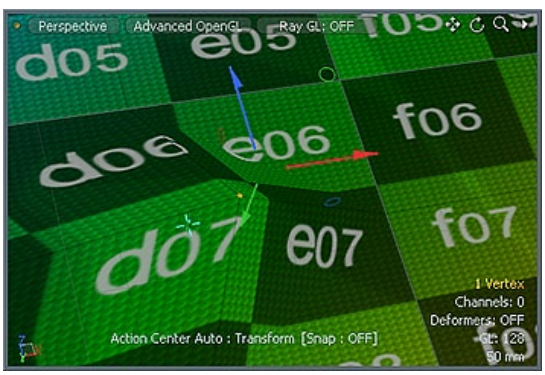
Jitter--

Enable X/Y/Z: Users can enable the random jittering movement on a per axis basis by toggling the option for each one axial direction.

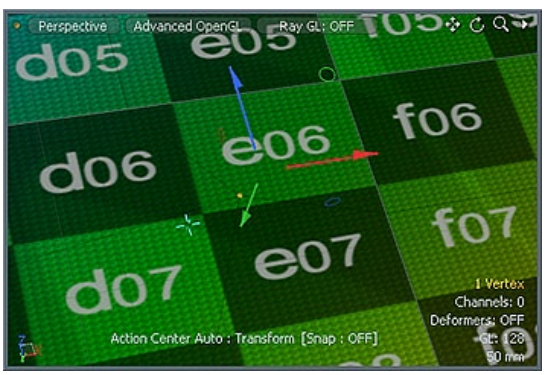
Range X/Y/Z: Controls the maximum random offset distance, this value is like adjusting the strength of the jittering effect.

Seed: The 'Seed' value is the initial number used when generating the random procedural values. Each Seed will produce different random results.

Slip Us: UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can *enable* the 'Slip Us' function so as to not disturb any existing UV mapping applied to the geometry.



'Slip UVs' function disabled, note texture warping.



'Slip UVs' function enabled, texture remains even.

'Slip Us' function disabled, note texture warping.

'Slip Us' function enabled, texture remains even.

Morph: The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of

modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

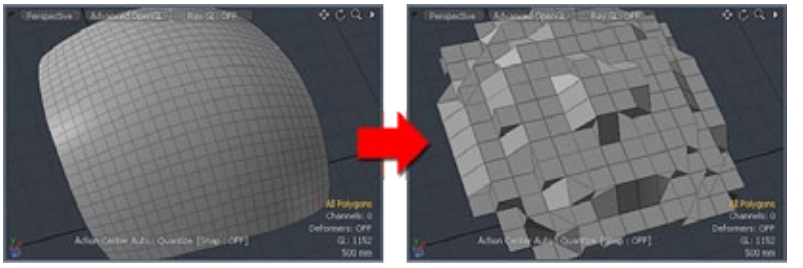
Snapping--

Please reference the 'Snapping' Section of the documentation.

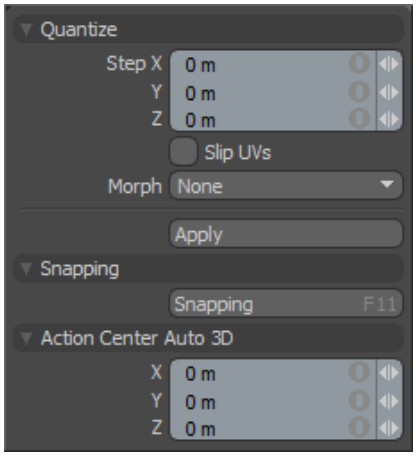
Action Center Auto 3D--

Please reference the 'Action Center' section of the documentation.

Quantize



Found in the 'Deform' subtab of the modeling toolbox (sometimes as a sub-option of the 'Smooth' or 'Jitter' tools), the Quantize tool is a deformation tool that evaluates the contour of the selected geometry and shuffles the vertices to align to a three dimensional grid. As the Step value for each axis is increased "voxel" size (the size of each block in the 3D grid) increases. Each vertex is moved to the closest matching point in the voxel.



Quantize--

Step X/Y/Z: Defines the Grid spacing for each axis individually.

Slip Us: UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can enable the 'Slip Us' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip Us' function disabled, note texture warping.

'Slip Us' function enabled, texture remains even.

Morph: The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Snapping--

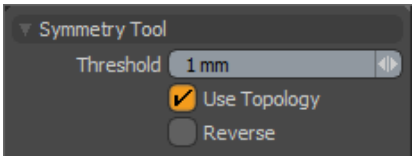
Please reference the 'Snapping' Section of the documentation.

Action Center Auto 3D--

Please reference the 'Action Center' section of the documentation.

Symmetry Tool

The Symmetry tool allows interactive correction of minor symmetry errors. The tool uses a distance tolerance to decide if vertices have a corresponding vertex on the opposite side of the symmetry axis (*Symmetry must be active for the tool to perform*). If the tool locates a vertex within the tolerance amount it will move that vertex to match its partner on the other side of the mesh.

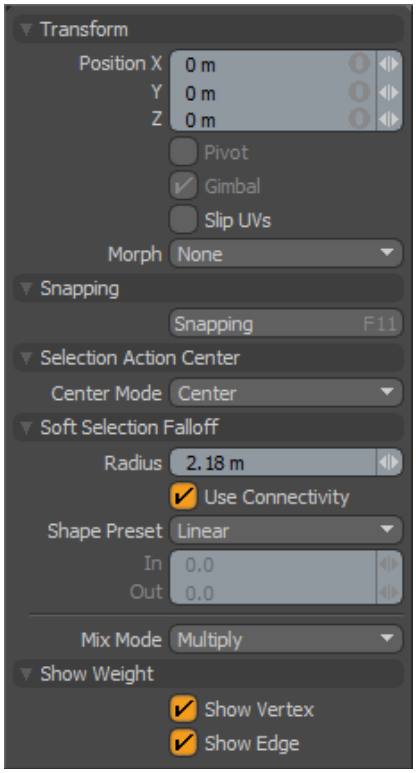


Use Topology Symmetry mode: The Symmetry Tool has been enhanced to include a Use Topology option. Instead of looking directly across an axis to find a corresponding vertex it now works recursively from a point that it knows to be symmetrical, and assumes that similarly connected neighbors should also be symmetric. This allows for a much more robust solution as there is less chance of a vertex looking at the wrong corresponding vertex. To run the tool turn symmetry on, run the Symmetry Tool under the Geometry tab, click in the viewport and drag to increase the threshold distance. Non-symmetrical vertices will be highlighted and will disappear as the threshold is increased. If all the highlighted verts do not disappear then you can continue applying the threshold value by clicking the middle mouse button. The Use Topology option only works if some part of the model is symmetrical to start with. If this isn't the case then Use Topology should be turned off.

To use the tool, activate symmetry on the desired axis then select the tool and click in the 3D viewport. You can interactively adjust the Threshold value by dragging in the 3D viewport. Or you can set the threshold value precisely by entering a numeric value on the properties form. When a vertex is adjusted it becomes highlighted with a large yellow square. If there are remaining vertices out of symmetry you can try to use

the MMB click to re-invoke the tool with the same settings. This will often clean up "stragglers".

Soft Select Move



Found in the 'Deform' subtab of the modeling toolbox (sometimes as a subtool of one of the other Soft Selection Tools), the Soft Select Move tool is a deformation tool that moves or translates the selected geometry rigidly, while gradually attenuating the translation outside of the selected area defined by the Soft Selection 'Radius'. As the Radius increases, so does the softness of the transformation.

Transform--

Position: Users may use the X,Y and Z input fields to apply specific offset distance values (calculated from position the object was in when the tool was activated). When adjusting object interactively in the viewport, the 'Position' input fields will display the current offset distance applied, providing useful user feedback.

Pivot: Disabled for the 'Soft Select Move' tool.

Gimbal: Disabled for the 'Soft Select Move' tool.

Slip UVs: UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can *enable* the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes*) The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Snapping--

Please reference the 'Snapping' Section of the documentation.

Selection Action Center--

Please reference the 'Action Centers' page of the documentation.

Soft Selection Falloff--

Radius: Determines the range outside of the selected edge to attenuate the falloff.

Use Connectivity: When enabled only affects single surface/connected elements, unconnected elements within range will be ignored.

Shape: The strength of the Falloffs influence can be controlled along the extent using a 'Shape Preset'-

Linear- Attenuation of Falloff occurs evenly across its range.

Ease-In- Strength of Falloff is greater toward the 'Start' position.

Ease-Out- Strength of Falloff is greater toward the 'End' position.

Smooth- Strength of Falloff is greater toward the center of the Falloff.

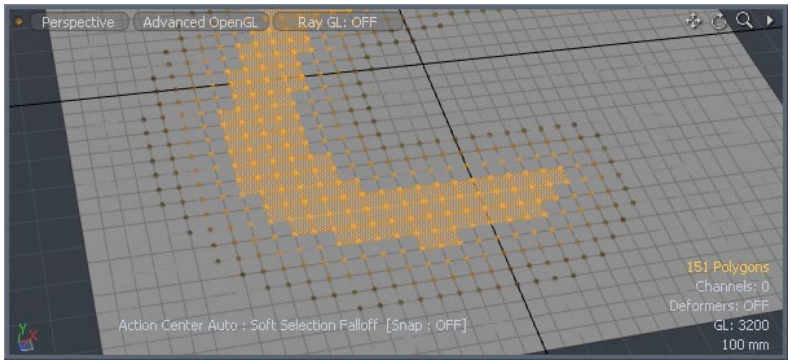
Custom- Users can use the 'In'/'Out' options to fine tune strength of Falloff.

In/Out: The 'In' value determines the strength of the Falloff nearer to the 'Start' position, where the 'Out' value determines the strength on nearer the 'End' side of the Falloff.

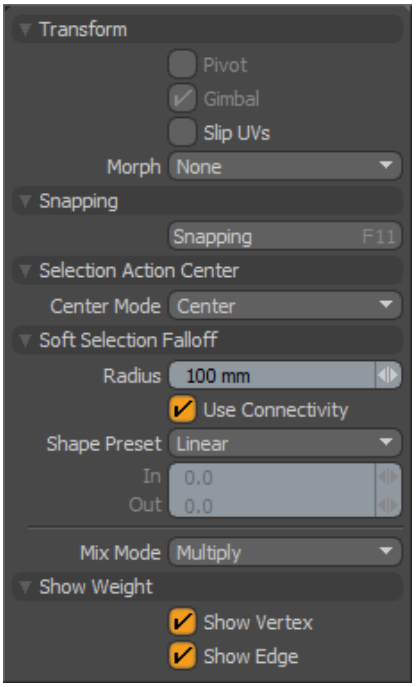
Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.

Show Weights--

Show Vertex/Show Edge: Users can use these toggles to enable/disable the display of geometry highlights for both vertices and edges, giving a visual reference to the application of the Soft Selection falloff (demonstration of the vertices display mode below).



Soft Select Rotate



Found in the 'Deform' subtab of the modeling toolbox (sometimes as a subtool of one of the other Soft Selection Tools), the Soft Select Rotate tool is a deformation tool that rotates the selected geometry rigidly, while gradually attenuating the rotation amount outside of the selected area defined by the Soft Selection 'Radius'. As the Radius increases, so does the softness of the transformation.

Transform--

Pivot: Disabled for the 'Soft Select Rotate' tool.

Gimbal: Disabled for the 'Soft Select Rotate' tool.

Slip UVs: UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can enable the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes*) The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Snapping--

Please reference the 'Snapping' Section of the documentation.

Selection Action Center--

Please reference the 'Action Centers' page of the documentation.

Soft Selection Falloff--

Radius: Determines the range outside of the selected edge to attenuate the falloff.

Use Connectivity: When *enabled* only affects single surface/connected elements, unconnected elements within range will be ignored.

Shape: The strength of the Falloffs influence can be controlled along the extent using a 'Shape Preset'-

Linear- Attenuation of Falloff occurs evenly across its range.

Ease-In- Strength of Falloff is greater toward the 'Start' position.

Ease-Out- Strength of Falloff is greater toward the 'End' position.

Smooth- Strength of Falloff is greater toward the center of the Falloff.

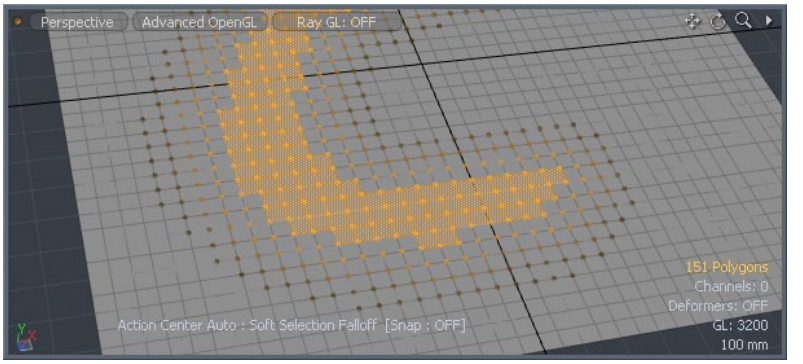
Custom- Users can use the 'In'/'Out' options to fine tune strength of Falloff.

In/Out: The 'In' value determines the strength of the Falloff nearer to the 'Start' position, where the 'Out' value determines the strength on nearer the 'End' side of the Falloff.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.

Show Weights--

Show Vertex/Show Edge: Users can use these toggles to *enable/disable* the display of geometry highlights for both vertices and edges, giving a visual reference to the application of the Soft Selection falloff (demonstration of the vertices display mode below).



Soft Select Scale

Found in the 'Deform' subtab of the modeling toolbox (sometimes as a subtool of one of the other Soft Selection Tools), the Soft Select Scale tool is a deformation tool that scales or re-sizes the selected geometry rigidly, while gradually attenuating the scale amount outside of the selected area defined by the Soft Selection 'Radius'. As the Radius increases, so does the softness of the transformation.

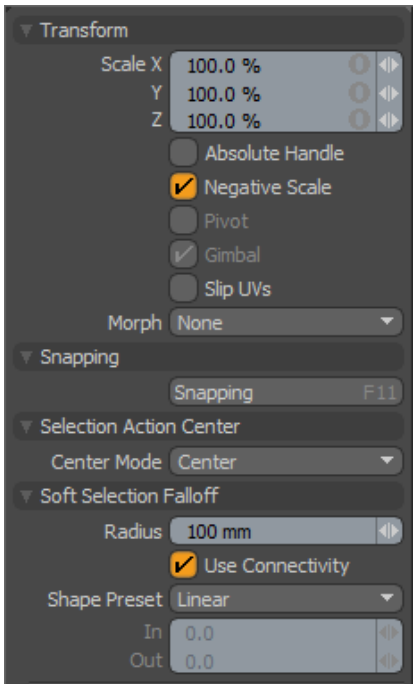
Transform--

Scale: Users may used the X,Y and Z input fields to apply specific scaling values. When adjusting object interactively in the viewport, the 'Scale' input fields will display the current scale factor applied, providing useful user feedback.

Absolute Handle: The 'Absolute Handle' toggle applies an additional function to the tool handles allowing users to scale the handle itself without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.

Negative Scale: The 'Negative Scale' toggle enabled/disabled modo's ability to

interactively apply negative scale values when manipulating the tool handles in the viewport. When disabled, users can always input negative values in any of the input fields directly.

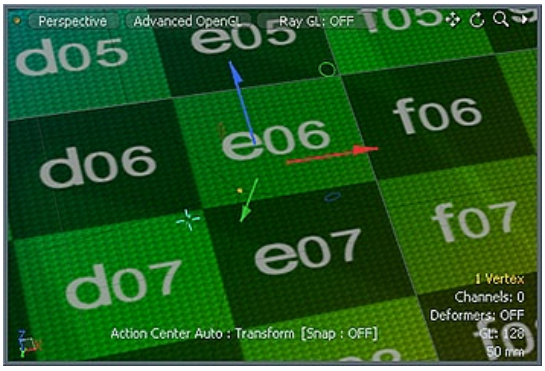


Pivot: Disabled for the 'Soft Select Scale' tool.

Gimbal: Disabled for the 'Soft Select Scale' tool.

Slip UVs: UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can *enable* the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.



'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes*) The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Snapping--

Please reference the 'Snapping' Section of the documentation.

Selection Action Center--

Please reference the 'Action Centers' page of the documentation.

Soft Selection Falloff--

Radius: Determines the range outside of the selected edge to attenuate the falloff.

Use Connectivity: When enabled only affects single surface/connected elements, unconnected elements within range will be ignored.

Shape: The strength of the Falloffs influence can be controlled along the extent using a 'Shape Preset'-

Linear- Attenuation of Falloff occurs evenly across its range.

Ease-In- Strength of Falloff is greater toward the 'Start' position.

Ease-Out- Strength of Falloff is greater toward the 'End' position.

Smooth- Strength of Falloff is greater toward the center of the Falloff.

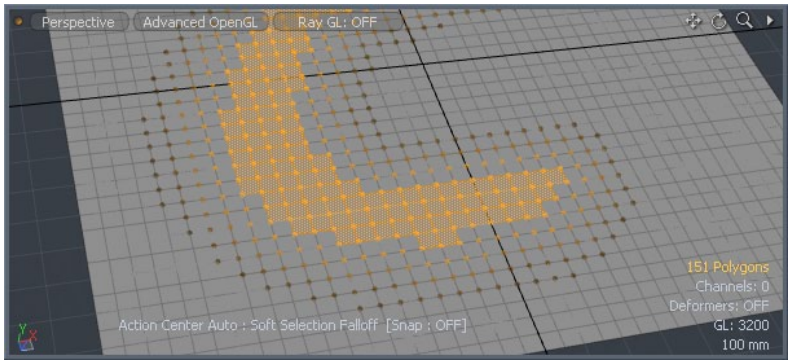
Custom- Users can use the 'In'/'Out' options to fine tune strength of Falloff.

In/Out: The 'In' value determines the strength of the Falloff nearer to the 'Start' position, where the 'Out' value determines the strength on nearer the 'End' side of the Falloff.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.

Show Weights--

Show Vertex/Show Edge: Users can use these toggles to *enable/disable* the display of geometry highlights for both vertices and edges, giving a visual reference to the application of the Soft Selection falloff (demonstration of the vertices display mode below).



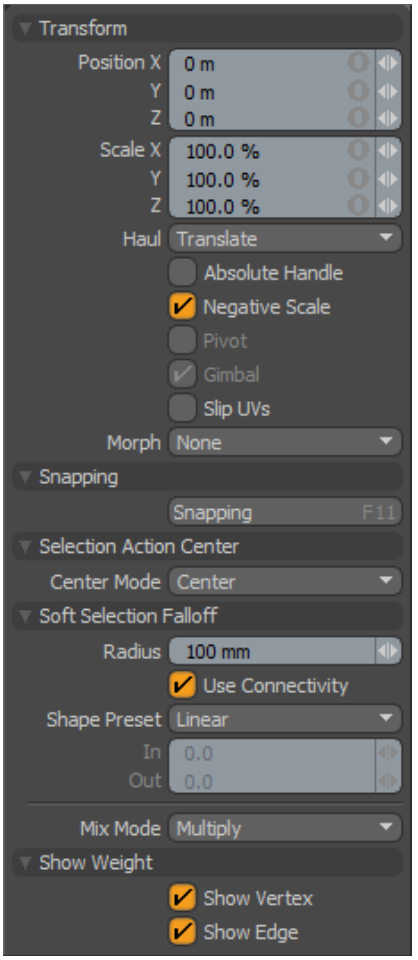
Soft Select Transform

Found in the 'Deform' subtab of the modeling toolbox (sometimes as a subtool of one of the other Soft Selection Tools), the Soft Select Transform tool is a deformation tool that allows users to simultaneously move, rotate and scale the selected geometry rigidly, while gradually attenuating the translation outside of the selected area defined by the Soft Selection 'Radius'. As the Radius increases, so does the softness of the transformation.

Transform--

Position: Users may use the X,Y and Z input fields to apply specific offset distance values (calculated from position the object was in when the tool was activated). When adjusting object interactively in the viewport, the 'Position' input fields will display the current offset distance applied, providing useful user feedback.

Scale: Users may used the X,Y and Z input fields to apply specific scaling values. When adjusting object interactively in the viewport, the 'Scale' input fields will display the current scale factor applied, providing useful user feedback.



Haul: Users have three options available when hauling in the viewport (i.e. dragging the mouse when not using the tool handles), 'Translate' will allow the users to move the selection, 'Rotate' allows the user to rotate, and 'Scale' would scale the selection.

Absolute Handle: The 'Absolute Handle' toggle applies an additional function to the tool handles allowing users to scale the handle itself without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.

Negative Scale: The 'Negative Scale' toggle enabled/disabled modo's ability to interactively apply negative scale values when manipulating the tool handles in the viewport. When disabled, users can always input negative values in any of the input fields directly.

Pivot: Disabled for the 'Soft Select Transform' tool.

Gimbal: Disabled for the 'Soft Select Transform' tool.

Slip UVs: (*Only in Component Modes*) UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can *enable* the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes*) The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Snapping--

Please reference the 'Snapping' Section of the documentation.

Selection Action Center--

Please reference the 'Action Centers' page of the documentation.

Soft Selection Falloff--

Radius: Determines the range outside of the selected edge to attenuate the falloff.

Use Connectivity: When *enabled* only affects single surface/connected elements, unconnected elements within range will be ignored.

Shape: The strength of the Falloffs influence can be controlled along the extent using a 'Shape Preset'-

Linear- Attenuation of Falloff occurs evenly across its range.

Ease-In- Strength of Falloff is greater toward the 'Start' position.

Ease-Out- Strength of Falloff is greater toward the 'End' position.

Smooth- Strength of Falloff is greater toward the center of the Falloff.

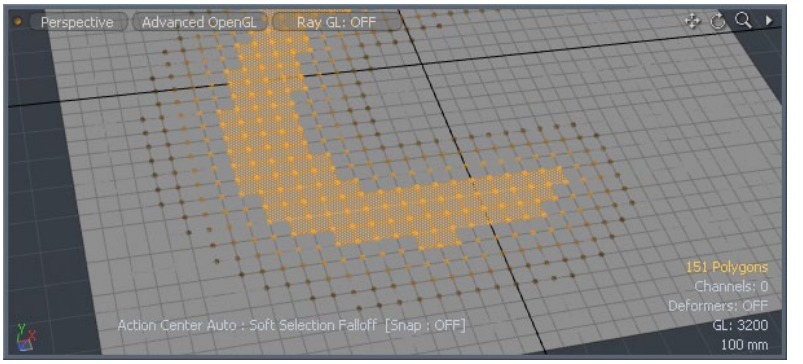
Custom- Users can use the 'In'/'Out' options to fine tune strength of Falloff.

In/Out: The 'In' value determines the strength of the Falloff nearer to the 'Start' position, where the 'Out' value determines the strength on nearer the 'End' side of the Falloff.

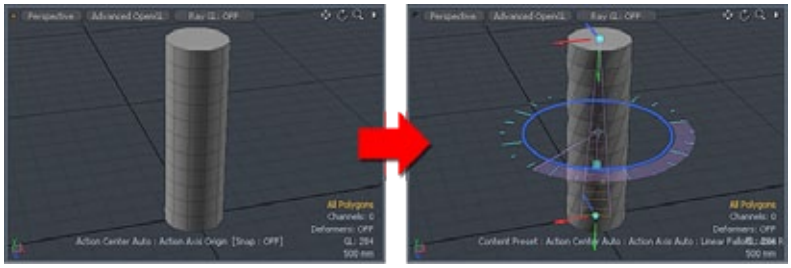
Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.

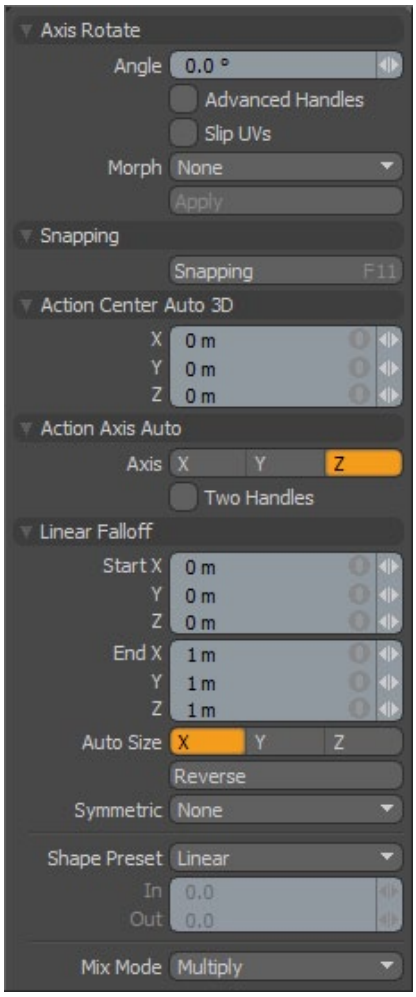
Show Weights--

Show Vertex/Show Edge: Users can use these toggles to *enable/disable* the display of geometry highlights for both vertices and edges, giving a visual reference to the application of the Soft Selection falloff (demonstration of the vertices display mode below).



Twist





Found in the 'Deform' subtab of the modeling toolbox, the Twist tool combines the Axis Rotate tool with a Linear Falloff. At first click, the linear falloff is set perpendicular to the mouse and autofit to the current geometry. Dragging at this time will create a twisting effect.

Users can more precisely set the effect of the tool by adjusting the falloff by RMB dragging out the linear falloff to a specific position and then using the LMB drag to effect the rotation which will create a twisting effect down the length of the falloff so long as the rotation axis is perpendicular to the falloff widget.

Axis Rotate--

Angle: Users may use the 'Angle' input field to apply specific rotational value, calculated in degrees. When adjusting object interactively in the viewport, the 'Angle' input fields will display the current rotation amount, providing useful user feedback.

Advanced Handles: Determines whether or not modo displays the additional information and controls in the viewport.

Slip UVs: (*Only in Component Modes*) UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can

enable the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes*) The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Snapping--

Please reference the 'Snapping' Section of the documentation.

Linear Falloff--

Start X/Y/Z: Defines the starting position of the Falloff as a specific XYZ coordinate

value. This end of the line and everything beyond it receives the maximum amount of influence attenuating toward the 'End' position.

End X/Y/Z: Defines the end position of the Falloff as a specific XYZ coordinate value. This end of the line and everything beyond it receives no influence attenuating toward the 'Start' position.

Auto Size X/Y/Z: Users can select either of these three options to automatically scale the 'Start' and 'End' points of the Falloff to match the bounding box size of the selected elements along one of the three axes.

Reverse: Users can select this option to invert the 'Start' and 'End' points reversing the influence of the Falloff.

Symmetric: Users can choose from several options to automatically mirror the influence area of the Falloff-

None- Symmetric function is disabled

Start- Influence of Falloff is symmetrically mirrored across the 'Start' position.

End- Influence of Falloff is symmetrically mirrored across the 'End' position.

Shape Preset: The strength of the Falloffs influence can be controlled along the extent using a 'Shape Preset'-

Linear- Attenuation of Falloff occurs evenly across its range.

Ease-In- Strength of Falloff is greater toward the 'Start' position.

Ease-Out- Strength of Falloff is greater toward the 'End' position.

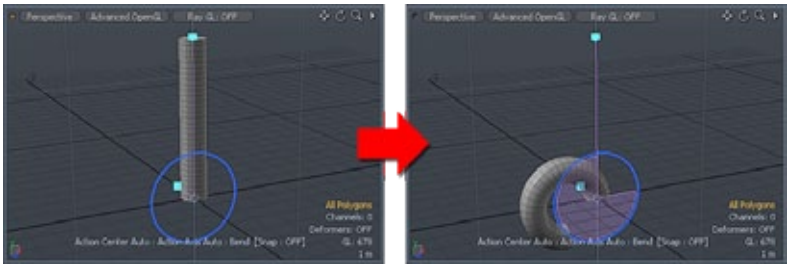
Smooth- Strength of Falloff is greater toward the center of the Falloff.

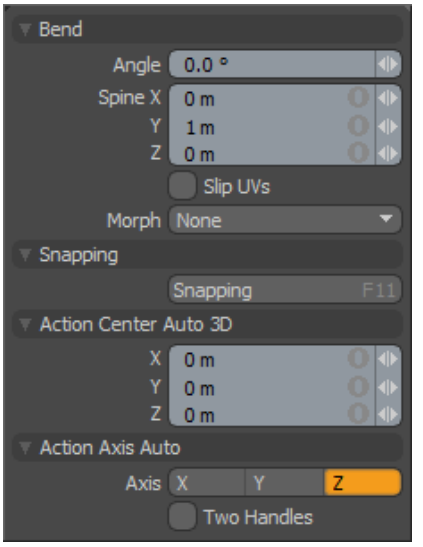
Custom- Users can use the 'In'/'Out' options to fine tune strength of Falloff.

In/Out: The 'In' value determines the strength of the Falloff nearer to the 'Start' position, where the 'Out' value determines the strength on nearer the 'End' side of the Falloff.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.

Bend





Found in the 'Deform' subtab of the modeling toolbox, the Bend tool allows users to quickly deform selected geometry in such a way as to simulate bending the mesh. The simplest example would be to take a straight cylinder or tube and bend it over as if you were bending a flexible pipe. The Bend tool consists of a bend "spine" and the bend amount disc. When activated the Bend tool uses the current action center and axis settings to determine the handle position and orientation. The spine handle position is set based on the selected geometry.

Bend--

Angle: This value represents the amount of bend that is received by the geometry at the end of the spine. The bend tool automatically fades this amount across the geometry selection down towards the base of the bend tool to accomplish its intended bending effect.

Spine X/Y/Z: This triplet value sets the end of the spine handle. The handle can be adjusted interactively in the 3D viewport by LMB+click and dragging or numerically on the property form.

Slip UVs: UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can enable the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

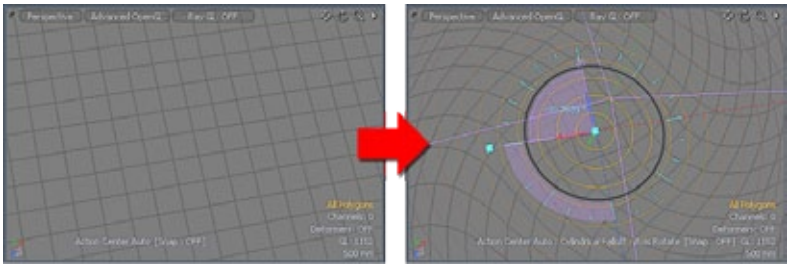
Snapping--

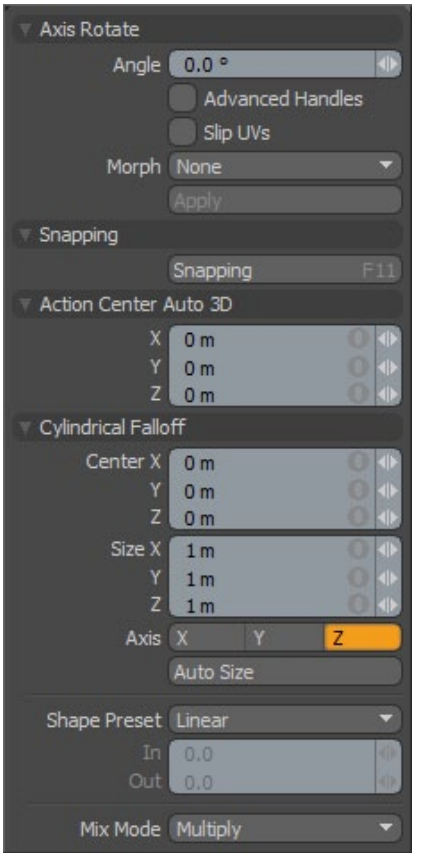
Please reference the 'Snapping' Section of the documentation.

Action Auto Axis--

Please reference the 'Action Center' section of the documentation.

Vortex





Found in the 'Deform' subtab of the modeling toolbox, the Vortex tool combines the Axis Rotate tool with the Cylindrical Falloff. At first LMB click the falloff region is created in a 2D plane centered on the mouse position and automatically scaled to match the current geometry. The Rotate widget is also centered on the mouse position and any dragging at this time will create a vortex effect on the mesh. For more precise control of the effected region, use the RMB to drag out the falloff area first and the LMB drag to affect the rotate.

Axis Rotate--

Angle: Users may use the 'Angle' input field to apply specific rotational value, calculated in degrees. When adjusting object interactively in the viewport, the 'Angle' input fields will display the current rotation amount, providing useful user feedback.

Advanced Handles: Determines whether or not modo displays the additional information and controls in the viewport.

Slip UVs: (*Only in Component Modes*) UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can *enable* the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes*) The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Snapping--

Please reference the 'Snapping' Section of the documentation.

Cylindrical Falloff--

Center X/Y/Z: Defines the Center of influence, where the strength of the Falloff will be greatest (100%). The strength of the falloff will attenuate toward the outer bounds of the Cylinder. The Falloff volume extends upward and downward toward infinity.

Size X/Y/Z: Defines the radius of a perfect circle from the 'Center' and determines the outer area of the Falloff where there is no affect.

Axis: Defines the Projection Axis for the cylinder determining the direction for the Falloff.

Auto Size: Users can select this option to automatically size the Falloff's 'Center' and 'Size' values to match the bounding box of the current selection.

Shape Preset: The strength of the Falloffs influence can be controlled along the extent using a 'Shape Preset'-

Linear- Attenuation of Falloff occurs evenly across its range.

Ease-In- Strength of Falloff is greater toward the 'Start' position.

Ease-Out- Strength of Falloff is greater toward the 'End' position.

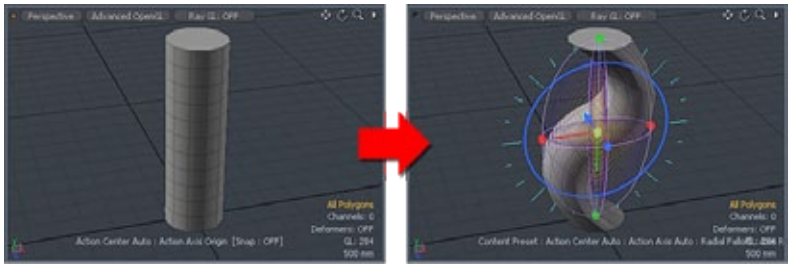
Smooth- Strength of Falloff is greater toward the center of the Falloff.

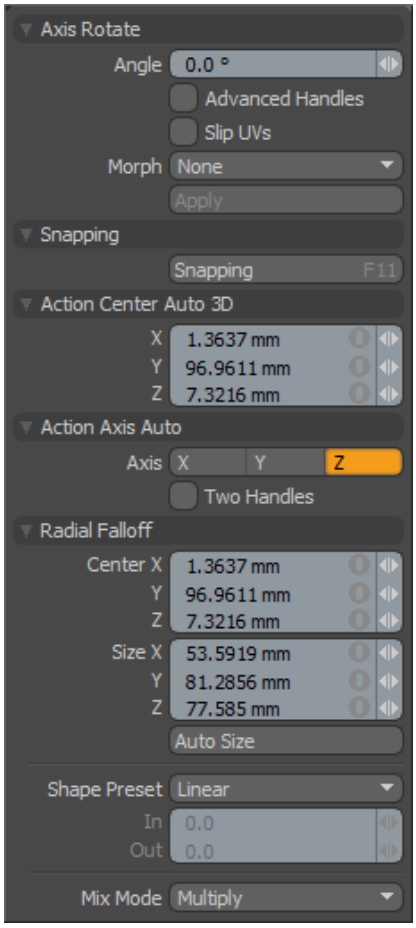
Custom- Users can use the 'In'/'Out' options to fine tune strength of Falloff.

In/Out: The 'In' value determines the strength of the Falloff nearer to the 'Start' position, where the 'Out' value determines the strength on nearer the 'End' side of the Falloff.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.

Swirl





Found in the 'Deform' subtab of the modeling toolbox, the Swirl tool combines the Axis Rotate with a Radial Falloff. The first mouse click will automatically fit the radial falloff to the current geometry. Dragging in the 3D view will apply a rotation modulated by this falloff. For more specific control over the region of influence you can use the right mouse button to quickly define the falloff region, then use the left mouse button to apply the rotation. To precisely place the falloff you can either use the handles to manipulate the falloff after it has been autosized or you can RMB drag away from the handles to reset the falloff and drag it out interactively.

Axis Rotate--

Angle: Users may use the 'Angle' input field to apply specific rotational value, calculated in degrees. When adjusting object interactively in the viewport, the 'Angle' input fields will display the current rotation amount, providing useful user feedback.

Advanced Handles: Determines whether or not modo displays the additional information and controls in the viewport.

Slip UVs: (*Only in Component Modes*) UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can enable the 'Slip UVs' function so

as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes*) The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Snapping--

Please reference the 'Snapping' Section of the documentation.

Radial Falloff--

Center X/Y/Z: Defines the Center of influence, where the strength of the Falloff will be greatest (100%). The strength of the falloff will attenuate toward the outer bounds of

the Spherical volume, the area outside the volume receives no tool influence.

Size X/Y/Z: Defines the radius of a perfect circle from the 'Center' and determines the outer area of the Falloff where there is no affect.

Auto Size: Users can select this option to automatically size the Falloff's 'Center' and 'Size' values to match the bounding box of the current selection.

Shape Preset: The strength of the Falloffs influence can be controlled along the extent using a 'Shape Preset'-

Linear- Attenuation of Falloff occurs evenly across its range.

Ease-In- Strength of Falloff is greater toward the 'Start' position.

Ease-Out- Strength of Falloff is greater toward the 'End' position.

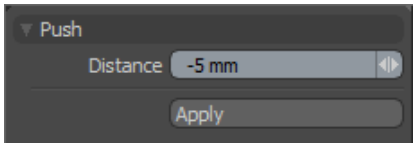
Smooth- Strength of Falloff is greater toward the center of the Falloff.

Custom- Users can use the 'In'/'Out' options to fine tune strength of Falloff.

In/Out: The 'In' value determines the strength of the Falloff nearer to the 'Start' position, where the 'Out' value determines the strength on nearer the 'End' side of the Falloff.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.

Push

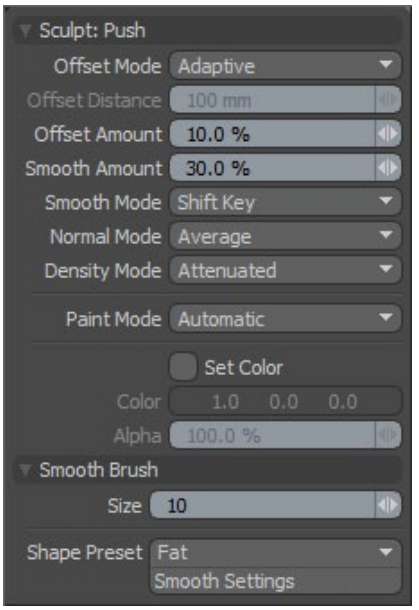


The Push tool is a Deformation tool that simply moves the selected geometry along the selected items normals.

The Push tool can be found on the Deform tab of the modo Tools toolbar, or on the menu bar by choosing Edit > Deform > Push.

The Push tool can be applied interactively by clicking in the viewport to activate the tool, then LMB dragging to increase the push distance. The tool can also be adjusted numerically with the Push Distance in the Tool Properties form. If the tool has not been activated (usually done by clicking in the 3D viewport) you must click "Apply after setting the value numerically (You can also right-click the Push entry in the Tool Pipe, and choose Apply Tool). This is a useful method when working with meshes that are very dense as it avoids interactive adjustments.

Sculpt



Before modo had sculpting there was the 'Sculpt' modeling tool. This was a very simple and direct way to shape polygons using a brush-like paradigm to smoothly deform a surface. When sculpting was introduced to modo, the functionality made the older sculpt tool redundant and unnecessary. For users convenience the 'sculpt' tool remains, using the superior sculpting functionality providing users with a more powerful way modify their mesh, but still retaining the intuitive simplicity of the older tool. To apply, users need only to LMB+click on the icon to enable the tool revealing the options below in the tool properties panel. Users can then LMB+click and drag over their models surface in the 3D viewport to deform it outward along the polygons *normal*. Holding down the 'Ctrl' key will push the surface inward along the normal. Users may adjust the option available to customize the effect of the brush on the surface. For more on sculpting, please see the 'Paint and Sculpting' section of the documentation.

Sculpt Push--

Offset Mode: modo provides two methods for calculating the way the tool moves the surface when brushing.

Adaptive- The adaptive mode will adapt to the surface that is being brushed on based on the setting for 'Offset Amount' and the size of the brush.

Absolute- When in 'Absolute' mode, the 'Offset Distance' setting will become active and users may enter a specific distance value the brush will conform to regardless of brush scale.

Offset Distance: This control is disabled unless the 'Offset Mode' is set to 'Absolute'. When active, the value determines specifically how far to displace the surface.

Offset Amount: Active only when the 'Offset Mode' is set to 'Adaptive', this percentage value modulates the extent of the brushstroke. The total scale of the sculpt deformation will be equal to the brush size times the offset amount percentage. (Note: This is a very important value for controlling sculpted detail and as such it can be adjusted interactively by using the MMB+click and hauling left or right.)

Smooth Amount: The Smooth tool is such a useful method for controlling sculpt detail it is available with almost every sculpting tool in modo and can be accessed by simply holding the Shift key down while sculpting. The Smooth amount attribute allows you to control the amount of smoothing applied when using the Shift qualifier with that specific tool. This is a convenience feature implemented specifically to enhance the sculpting workflow.

Smooth Mode: modo provides alternate methods for applying smoothing while sculpting-

Shift Key-- The smooth mode is enabled when the 'Shift' key is depressed.

Low Pressure-- When using a pressure sensitive tablet, users may apply the 'Low Pressure' option, where light sculpting strokes will have a smoothing affect on the surface. Once the pressure threshold is exceeded, the smoothing effect will attenuate toward the full pressure effect where the brush will work normally as determined by the tool providing a dual function to the brush.

Uniform-- When set to 'Uniform', smoothing is always on while sculpting, thus making softer strokes.

Normal Mode: This option controls the method used for displacing the sculpted vertices. The various options include:

Brush Center-- uses the geometry at the very center of the brush to calculate the direction along which to displace the mesh.

Vertex-- displaces *each vertex* along its own normal.

Average-- calculates an average normal for all the vertices inside the brush.

Screen-- displaces the vertices directly towards the active viewport.

Mouse Down-- uses the normal of the mesh surface directly under the mouse when the LMB is clicked. All displacement is then carried out according to that vector.

X,Y,Z-- allows you to pick a world axis along which to displace the vertices.

Density Mode: This control determines how the sculpt stroke effects areas of the mesh that it has already covered. For instance, brushing back and forth will cause the sculpt tool to run back over itself which can have a number of different effects. This controls the sculpt during a single stroke. Once the stylus comes off the tablet or you release the LMB the density control is reset.

Additive-- continues to add to the displacement each time the brush crosses over its own path with no maximum cap to govern the sculpt.

Attenuate-- continues to add to the displacement as the brush crosses over its own stroke until the displacement reaches the Offset Distance or Amount value.

Max Stroke-- enforces a maximum offset based on the largest displacement value of the current stroke.

Max Map-- caps the sculpt effect at the maximum amount in the current displacement image.

Paint Mode: The 'Paint Mode' determines which method of sculpting is to be used. By default the setting is 'Auto' which will perform mesh based sculpts when there are no displacement images and automatically switch to image based sculpting when there is an image available. There are times when you may want to force modo into mesh based sculpting even if you have an image applied for displacement. This can be very helpful for blocking out shapes and form, as mesh sculpting is only effecting the cage vertices of the base geometry.

Automatic-- chooses between mesh and image sculpting depending on availability of displacement image map texture.

Mesh-- forces the tools to perform mesh sculpting only.

Image Map-- forces the tools to perform image sculpting only.

Set Color: When utilizing Multi-Res sculpting, and enabling the 'Set Color' option, users can simultaneously sculpt the mesh and paint into an Image Map layer using the same settings of the sculpting tool itself.

Color: This value determines the color applied to the image map when sculpting.

Alpha: This value determines the opacity of the stroke applied to the image map.

Smooth Brush--

The Smooth Brush creates a nice feathered airbrush effect over the surface. There are two main options for the Smooth Brush, Size and Shape Preset.

Size: This value sets the radius of the brush in pixels from screen space. This is important to understand as zooming the view will change the result of the brush. With the brush size fixed zooming in on the model will decrease the relative scale the the brush whereas zooming out will increase it. Users can interactively adjust the Tip size value by RMB+click and haul in the viewport, adjusting an on-screen display representing the brushes size.

Shape Preset: The popup option includes several preset brush options for varying effects. This setting drives the profile of the airbrush. Users may also adjust the In and Out Slope and Weight setting to produce further customized brushes.

Linear-- will create a straight ramp from the center to the edge which results in a brush profile that looks like a cone.

Smooth-- will create a brush stroke that is opaque in the center and falls off smoothly to the edge of the brush with an S curve shape.

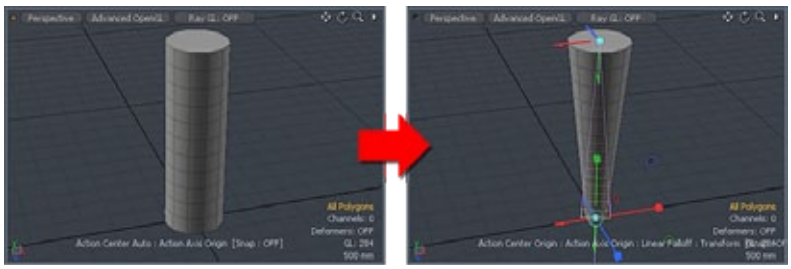
Bulge-- changes the bias of the curve to start out slowly and then ramp more harshly towards the middle, resulting in a brush profile that looks similar to a dome.

Fat-- brush starts out soft and ramps harshly at the outer edge producing a soft 'U' shaped brush that varies the most under light tablet input pressure.

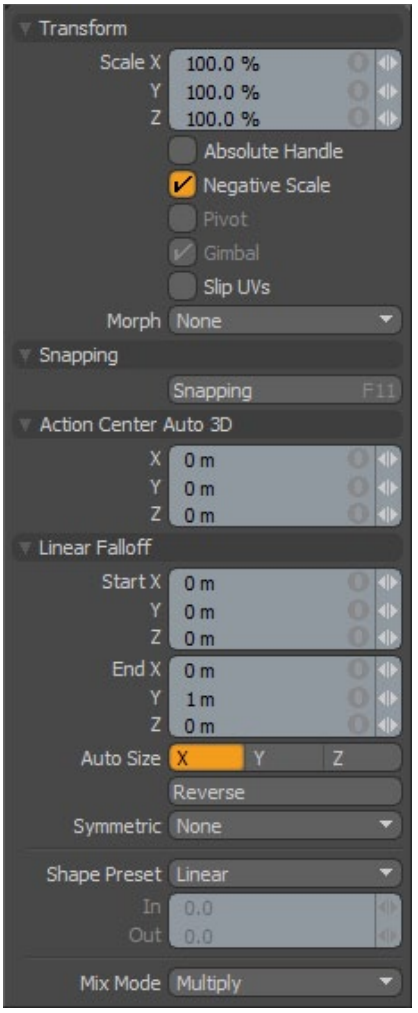
Sharp-- the falloff is very harsh and steep at the center producing a fine stroke.

Very Sharp-- this preset is nearly identical to the 'Sharp' version with a more extreme falloff for even finer strokes.

Taper



Found in the 'Deform' subtab of the modeling toolbox, the Taper tool combines the Transform tool with a Linear falloff. At first click the linear falloff is set perpendicularly to the mouse position and automatically scaled to match the currently selected geometry. Dragging the mouse at this time will deform the object with a tapering effect. To precisely place the falloff you can either use the handles to manipulate the falloff after it has been autosized or you can RMB drag away from the handles to reset the falloff and drag it out interactively.



Transform--

Scale: Users may used the X,Y and Z input fields to apply specific scaling values. When adjusting object interactively in the viewport, the 'Scale' input fields will display the current scale factor applied, providing useful user feedback.

Absolute Handle: The 'Absolute Handle' toggle applies an additional function to the tool handles allowing users to scale the handle itself without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.

Negative Scale: The 'Negative Scale' toggle enabled/disabled modo's ability to interactively apply negative scale values when manipulating the tool handles in the viewport. When disabled, users can always input negative values in any of the input fields directly.

Pivot: Disabled for the 'Taper' tool.

Gimbal: Disabled for the 'Taper' tool.

Child Compensate: (*Only in Items Mode*) The 'Child Compensate' toggle when disabled will pass along the transform value to any child items of the current selection, when *enabled*, the child items will ignore any transform values.

Slip UVs: (*Only in Component Modes*) UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can *enable* the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes*) The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Snapping--

Please reference the 'Snapping' Section of the documentation.

Linear Falloff--

Start X/Y/Z: Defines the starting position of the Falloff as a specific XYZ coordinate value. This end of the line and everything beyond it receives the maximum amount of influence attenuating toward the 'End' position.

End X/Y/Z: Defines the end position of the Falloff as a specific XYZ coordinate value. This end of the line and everything beyond it receives no influence attenuating toward the 'Start' position.

Auto Size X/Y/Z: Users can select either of these three options to automatically scale the 'Start' and 'End' points of the Falloff to match the bounding box size of the selected elements along one of the three axes.

Reverse: Users can select this option to invert the 'Start' and 'End' points reversing the influence of the Falloff.

Symmetric: Users can choose from several options to automatically mirror the influence area of the Falloff-

None- Symmetric function is disabled

Start- Influence of Falloff is symmetrically mirrored across the 'Start' position.

End- Influence of Falloff is symmetrically mirrored across the 'End' position.

Shape Preset: The strength of the Falloffs influence can be controlled along the extent using a 'Shape Preset'-

Linear- Attenuation of Falloff occurs evenly across its range.

Ease-In- Strength of Falloff is greater toward the 'Start' position.

Ease-Out- Strength of Falloff is greater toward the 'End' position.

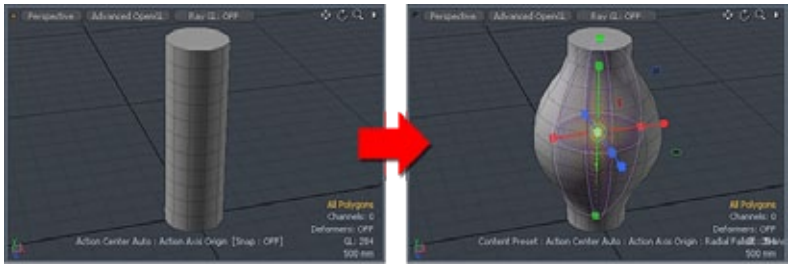
Smooth- Strength of Falloff is greater toward the center of the Falloff.

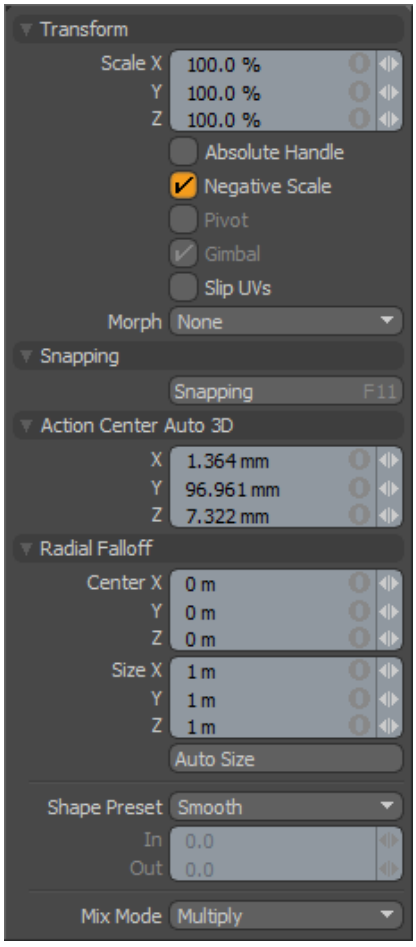
Custom- Users can use the 'In'/'Out' options to fine tune strength of Falloff.

In/Out: The 'In' value determines the strength of the Falloff nearer to the 'Start' position, where the 'Out' value determines the strength on nearer the 'End' side of the Falloff.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.

Bulge





Found in the 'Deform' subtab of the modeling toolbox, the Bulge tool combines the Transform tool with a Radial Falloff. At first activation the radial falloff is auto-fit to the current geometry and the Stretch tool is activated at the center of the mesh. Using the Stretch handles causes an apparent bulging of the object. You can easily define a custom area of the mesh for applying the bulge by using the right mouse button to drag out the radial falloff region. The left mouse button will apply the stretch tool.

Transform--

Scale: Users may used the X,Y and Z input fields to apply specific scaling values. When adjusting object interactively in the viewport, the 'Scale' input fields will display the current scale factor applied, providing useful user feedback.

Absolute Handle: The 'Absolute Handle' toggle applies an additional function to the tool handles allowing users to scale the handle itself without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.

Negative Scale: The 'Negative Scale' toggle enabled/disabled modo's ability to interactively apply negative scale values when manipulating the tool handles in the viewport. When disabled, users can always input negative values in any of the input fields directly.

Pivot: Disabled for the 'Bulge' tool.

Gimbal: Disabled for the 'Bulge' tool.

Child Compensate: (*Only in Items Mode*) The 'Child Compensate' toggle when disabled will pass along the transform value to any child items of the current selection, when *enabled*, the child items will ignore any transform values.

Slip UVs: (*Only in Component Modes*) UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can *enable* the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Morph: (*Only in Component Modes*) The Morph option determines how modo treats stored Morph information when applying transforms to geometry (Move, Rotate, Scale, etc.). In previous versions of modo, in order to transform a Morph along with its base, it needed to be selected in the Vertex Map list. If it wasn't, relative Morph map data when recalled would produce distorted, undesirable results. If users we're unaware of the requirements, it was easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the Morph Map vertex data is dealt with when applying any transforms--

None- Morph data is not affected unless selected, working identical to modo 501 and earlier.

Transform- With this option selected, Morph data is transformed along with base mesh.

Keep Positions- With this option Selected, Morph data is converted into a Absolute Morph Map and all vertices retain their pre-transformed positions.

Snapping--

Please reference the 'Snapping' Section of the documentation.

Radial Falloff--

Center X/Y/Z: Defines the Center of influence, where the strength of the Falloff will be greatest (100%). The strength of the falloff will attenuate toward the outer bounds of the Spherical volume, the area outside the volume receives no tool influence.

Size X/Y/Z: Defines the radius of a perfect circle from the 'Center' and determines the outer area of the Falloff where there is no affect.

Auto Size: Users can select this option to automatically size the Falloff's 'Center' and 'Size' values to match the bounding box of the current selection.

Shape Preset: The strength of the Falloffs influence can be controlled along the extent using a 'Shape Preset'-

Linear- Attenuation of Falloff occurs evenly across its range.

Ease-In- Strength of Falloff is greater toward the 'Start' position.

Ease-Out- Strength of Falloff is greater toward the 'End' position.

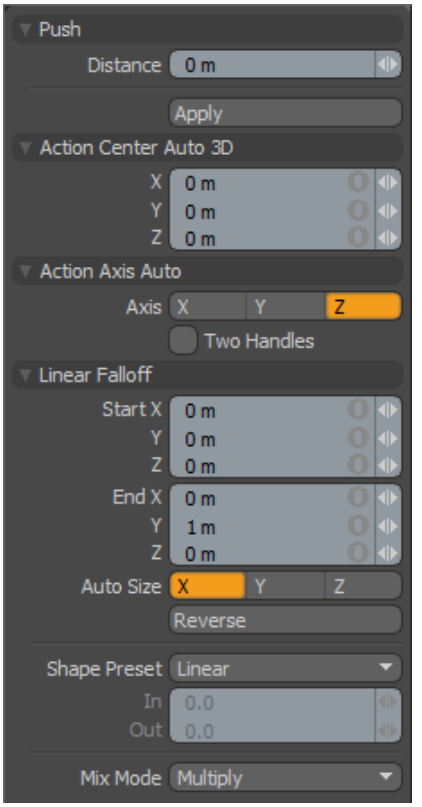
Smooth- Strength of Falloff is greater toward the center of the Falloff.

Custom- Users can use the 'In'/'Out' options to fine tune strength of Falloff.

In/Out: The 'In' value determines the strength of the Falloff nearer to the 'Start' position, where the 'Out' value determines the strength on nearer the 'End' side of the Falloff.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.

Flare



The Flare tool combines the Push tool with a Linear Falloff. The result is the quick ability to "flare" out a specific portion of the mesh using the linear falloff as the modulator.

Distance: The actual amount each vertex is moved out away from its original position. The vertex will be moved directly along its own normal. This value can be positive or negative.

Linear Falloff

Start XYZ: This triplet value sets a 3D position for the beginning of the linear falloff. At this position the falloff value is set to 0. All vertices at this side of the falloff receive no effect from the tool.

End XYZ: This triplet value sets a 3D position for the end of the linear falloff. At this position the falloff value is set to 100. All vertices at this side of the falloff receive 100% effect from the tool.

Auto Size: This option button will snap the linear falloff to the chosen axis and scale it to match the selected geometry.

Reverse: Clicking this option will swap the Start and End values effectively flipping the falloff.

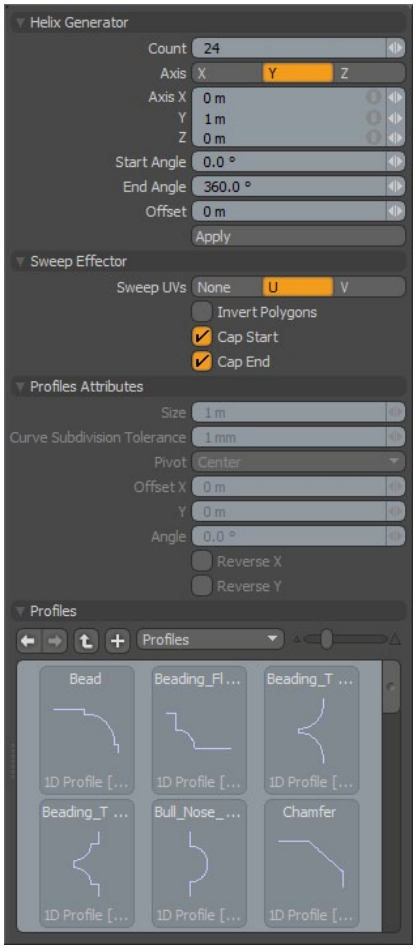
Duplicating Geometry

Often times, it is the details that make for compelling objects; modo's toolbox excels in easing this frequent requirement. Providing a variety of cloning and array tools, producing multitudes of seemingly complex forms from basic geometry. Popular functions like Radial Sweep and Bridge provide simple ways extend and shape a surface very quickly. Extrusions fill out modo's toolbox providing a diverse set of options for generating and extending existing geometry, making it easy to take the simple and make it complex.

Duplicate Geometry

Radial Sweep	Curve Extrude	Curve Clone	Array
Sketch Extrude	Bezier Extrude	Bezier Clone	Radial Array
Mirror	Pen Extrude	Pen Clone	Mesh Paint
Bridge		Scatter Clone	Paste Tool
		Clone	Duplicate

Radial Sweep



The Radial Sweep tool creates geometry by revolving the geometry selection around a central point. The geometry can be a modo Curve profile, or a polygon silhouette shape. This is the common method for creating simple round objects like bottles, plates or goblets. Radial Sweep can also be very useful for creating objects such as springs by using the offset control to not only spin the selected geometry while sweeping it but also move the new elements along the tools axis. Users will not receive satisfactory results when sweeping closed shapes such as cubes and spheres.

Helix Generator--

Count: This value determines the number of new segments generated by the tool.

Axis: Determines along which axis to sweep the geometry.

Axis X/Y/Z: These three values determine the location of the tools Axis control handles. If the values are set to 1 meter on all axes the handles ends will have one end at -1 X, -1 Y, -1 Z and the other at 1 X, 1 Y, 1 Z creating a skewed off axis plane for the array.

By default the Start and End Angles go from 0 to 360 degrees. This creates an array that completes a circle. Changing these two values can create an array that revolves many times or one that leaves one side of the circle open. Setting the end value to greater than 360 degrees is very useful when combined with some non-zero offset value.

Start Angle: This degree setting determines the rotation placement for the first lathed geometry.

End Angle: This degree value determines the rotation placement for the last lathed geometry.

Offset: Moves each successive segment a user defined distance from the previous along the tools axis. This value is very useful when combined with an End Angle greater than 360 degrees so that the array creates a spiral for items such as a spring or telephone cord.

Sweep Effector--

Sweep UVs: Determines along which UV axis to span the newly created geometry.

Invert Polygons: This toggle will simply flip the polygons created by the tool. In some circumstances lathing geometry will turn inside out. In this case simply activate the Invert Polygons control and they will face the correct direction.

Cap Start/End: The Cap controls are only useful when the start and end angles do not result in a complete 360 rotation since in this case the start and end segments are joined together. 'Cap Start' creates a cap on the start of the lathe to avoid a hole in the mesh, 'Cap End' creates a cap on the end of the lathe.

Profile Attributes--

If the user wishes, one can easily sweep a preset profile as readily as selected geometry by LMB clicking any of the profiles in the tool properties Profiles viewport.

Size: The size of 1D or 2D profile to sweep. If the size is 0.0, it uses the real size.

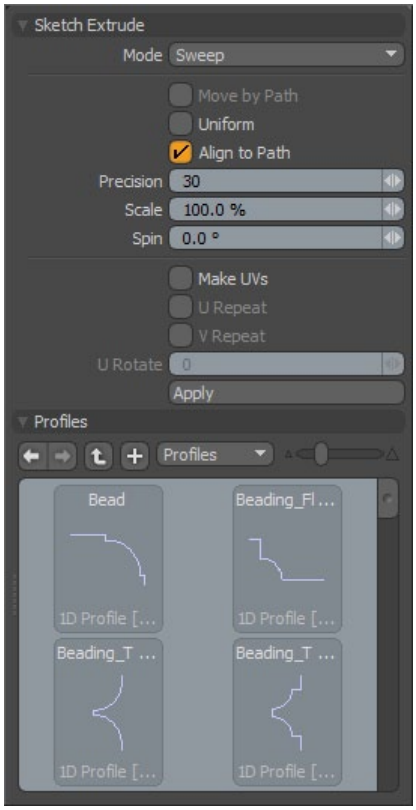
Curve Subdivision Tolerance: This is the minimum height when it subdivides bezier curve to polyline.

Pivot: Specify the pivot position of the profile.

Offset X/Y: These are the offset amount on the profile space.

Reverse X/Y: Reverse the selected profile to horizontal or vertical.

Sketch Extrude



The Sketch Extrude tool allows you to extrude selected polygons by drawing in the viewport.

To use this tool, select a polygon, then choose the Sketch Extrude tool on the Polygon tab of the modo Tools toolbar. Click in the viewport to activate the tool handle, then press and drag on the handle to initiate the sketch. The handle turns to a yellow circle widget. The polygon extrusion will follow the movement of the mouse cursor as you draw in the viewport. The tool has several options available.

Sketch Extrude--

Modes: The modes popup provides a number of useful options for controlling the tool.

Sweep-- This mode allows the user to click and drag on any of the visible handles(cyan dots), in essence sketching out the path for extrusion of the polygons.

Edit Path-- This mode reveals the control points on all user created paths. Clicking and dragging these control points allows further adjustment of the sketched curves. In the

Delete Knot-- In this mode, clicking on a control point will remove it from the curve path. In the 'Delete Path' mode, one can click on the path control handle to remove it.

Uniform Spans- This mode shows handles only at the end of each path. Clicking on any handle will adjust the control knots along the path such that they are distributed uniformly along the path.

Straight- The Straight mode allows you to quickly straighten a path by clicking on the end handle.

Move by Path: This toggle option is only available when using the Edit Path mode. When the option is activated dragging any control point will also move the "children" points that are between the selected control point and the end of the path.

Uniform: This toggle option will constrain all newly generated geometry to be uniformly distributed between the original polygons position and the end of the path.

Align to Path: With this toggle option active the geometry that is created along the path will also be rotated such that it is aligned to the curve. When this option if off all new polygons have the same world rotation as the original polygon.

Precision: The precision value determines how often control points should be created while the path is interactively "sketched". This number represents a screen space pixel count so that if the value is set to 30 (default value) a new control span will be created each time the mouse travels 30 pixels on screen during the sketch.

Scale: The scale percentage will vary the thickness of the generated geometry along the path from the original polygonal location to the control handle at the end of each path.

Spin: This value in degrees will rotate each generated span along the path providing a twisting effect along the sketched curve.

Make UVs: These options provide control over the method of UV generation for the new polygons in the mesh. The default option attempts to create the UVs inside the existing UV space amongst the corresponding geometry. Selecting the U or V Repeat options will overlay each new segment over the last along the selected U or V axis.

Profiles--

Mini Preset browser for viewing various profiles. Works the same as the standard preset browser. Selecting any of the various profiles will modulate the extrusion width with the selected shape over its length. To return to the original polygon selection for the extrusion, ctrl-click the selected profile to deselect it.

Mirror

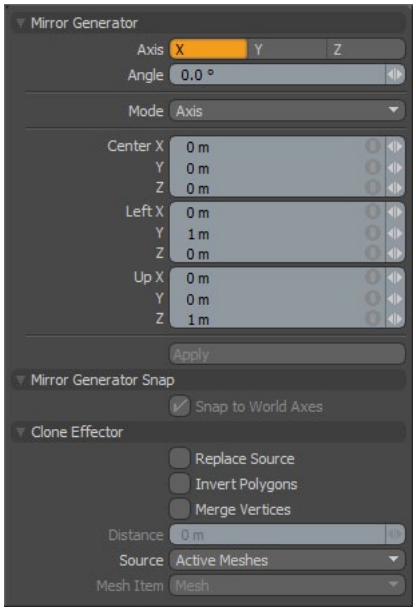
Using the mirror tool you can create a duplicate of the selected geometry in the active layer. When the mirror tool is actived, it creates an axis, across which the geometry is duplicated in a flipped or *mirrored* orientation to the original. With the tool active you can interactively position the mirror plane to change the distance of the mirrored geometry, or change the rotation of the plane to adjust the orientation of the duplicated geometry. The mirror tool has the following options:

Mirror Generator--

Axis (XYZ option): Sets the axis direction for the mirror plane, defining the center position across where the geometry will be cloned.

Angle: This degree value determines rotation of the mirror plane around the Axis. The default value of 180 degrees created a perfect mirror flip. By increasing or decreasing

this value, you can skew the mirror plane to create an acute or obtuse angle between the original and new geometry.



Mode: Determines what tools handles are available to control the mirroring-

Axis- Keep mirror manipulator handle aligned to an upright axis direction, handle allows for single value rotation, adjusting the 'Angle' value.

Free Rotation- Produces a second mirror manipulator handle allowing an addition axis of rotation.

Three Points- Produces a triangular manipulator allowing any degree of freedom in the angle and position of the mirroring plane.

Center (X,Y,Z): The center values set the position for the main mirror plane. This value can be set numerically, or by direct manipulation of the mirror tool handle. To adjust the position, you must drag the center of the handle.

Left (X,Y,Z): This secondary value rotates the mirroring plane. This value can be set numerically, or by direct manipulation of the mirror tool handle.

Up (X,Y,Z): In 'Free Rotation' or 'Three Points' mode, this value provides an additional degree of rotation freedom of the mirroring plane.

Mirror Generator Snap--

Snap to World Axis: When snapping is *enabled* (press the 'X' key to toggle), this option allows users to snap the handles to the axis center positions.

Clone Effector--

Replace Source: Removes the initial geometry, leaving only the mirrored results.

Invert Polygons: Flips polygons normals. Useful when mirroring across a negative plane direction that inverts the polygon face directions.

Merge Vertices: When active this option will attempt to merge any vertices overlapping between the original and new geometry.

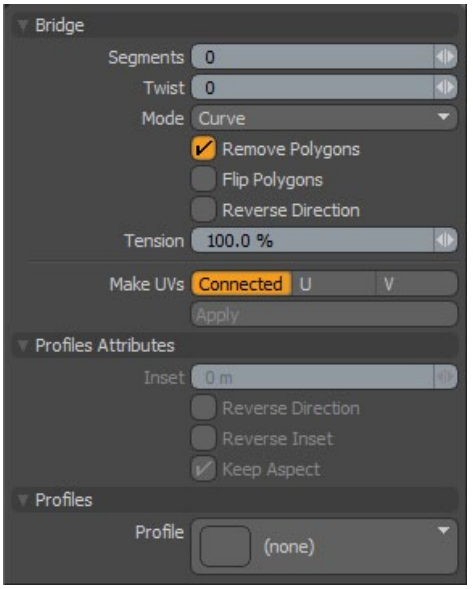
Distance: This function depends on the activation of the 'Merge Vertices' option. The

Distance is a simple measurement to determine the threshold for merging points from the new and original geometry.

Source: Determines which geometry is mirrored. Generally this is left to the 'Active Mesh' option. Other options are useful when cloning multiple items, such as with the 'Scatter Clone' tool.

Bridge

The Bridge tool will create polygons to bridge the area between selected edges or polygons. There are several options for the Bridge tool including the number of segments, a twist amount, interpolation mode (Linear, Curve, Smooth), and options to remove the original polygons and/or flip the new polygons. The resulting polygons can also be modulated by selecting a profile.



Bridge--

Segments: The segments option allows you to set the number of polygons that are created to the bridge. This value can be set interactively by dragging in the 3D viewport while the tool is active, or numerically on the tool properties form.

Twist: The twist value is used when bridging polygon selections. When the value is set to a non-zero number it will turn the bridge polygons between the original connection points. This is very useful in cleaning up bridges between disparate polygon selections where the bridge can become overlapped. In that case using the Twist value will often allow you to "untangle" the bridge manually.

Mode (Linear, Curve, Smooth): This popup provides a simple method for setting the interpolation style of the generated bridge polygons. 'Linear' will create a bridge in a straight line between the original selection. 'Curve' will create an arc between the original selection points. 'Smooth' creates an "ease in/ease out" blend between the original selection sections. The 'Curve' option can be further refined adjusting the 'Tension' setting.

Remove Polygons: This will remove the originally selected polygons used to create the bridge. This is on by default and is the most often preferred behavior.

Flip Polygons: When activated this option will flip the normals of the newly created polygons in the bridge.

Reverse Direction: When enabled, modo will reverse the direction of the bridge.

Tension: This percentage value provides some additional control over the bridge path used when the Curve mode is active. Increasing this value will accentuate the curved effect and decreasing it will mute the effect toward the linear path.

Make UVs: Determines how a UV map is automatically generated for the resulting geometry

Connected- Simply connects the edges between existing UV values as a straight line.

U/V- Fills the entire UV map 1-0 area with the resulting bridged geometry in the specified orientation.

Profile Attributes--

Inset: Specifies the X direction offset amount of 1D profile.

Reverse Direction: If this option is enabled, the 1D profile is evaluated from top to bottom.

Reverse Inset: If this option is enabled, "Inset" will be reversed essentially mirroring the profile.

Keep Aspect: Automatically sets the "Inset" value based on the aspect ratio of the profile.

Profiles--

Mini Preset browser for viewing various profiles. Works the same as the standard preset browser.

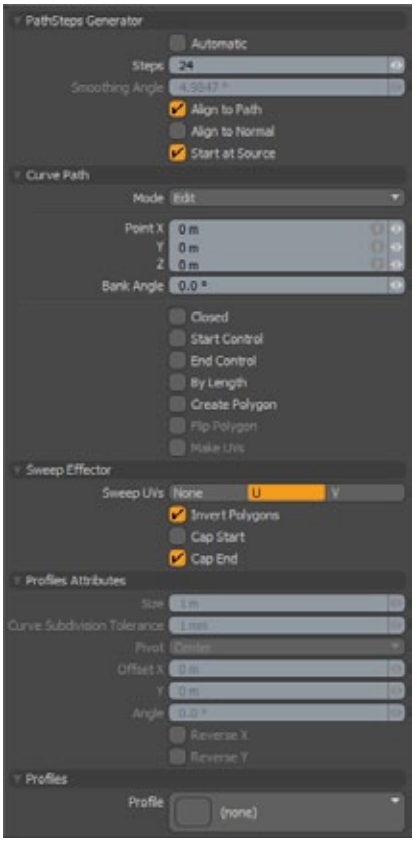
Curve Extrude

The Curve Extrude tool allows users to quickly extrude a geometry selection along the length of a curve. If the curve is pre-existing, select it first, then the polygonal geometry for extrusion. 'Curve Path' will also allow the user to create a path interactively. To use, select the tool and make it interactive by clicking in the viewport. Once activated, a number of properties are available

PathSteps Generator--

Automatic: This toggle control will force a computed number of clones based on the

Smoothing Angle value such that a clone is placed along the path to maintain a level of smoothness at or above the user set level (determined by the 'Smoothing Angle').



Steps: Active only when Automatic is off this is the number of particles generated along the curve path to feed into the Effector.

Smoothing Angle: Determines the angle along the curve where a clone will be created when the 'Automatic' option is *enabled*.

Align to Path: When *enabled* the generated particles will include rotation information so the newly created clone aligns to the path vector direction at the location where it is created.

Align to Normal: When enabled, the generated particles will include rotation information so the newly created clone aligns with any stored normal direction for a vertex, such as for a curve extracted from geometry.

Start at Source: Often a curve used for cloning will have its initial control knot imperfectly aligned with the geometry to be cloned. With Start at Source active the curve is used as a relative guide using the original geometry as the true starting location. Disabling this option may cause the newly created geometry to move away from the intended original location.

Curve Path--

Mode: By default the mode is set to Add. In the 'Add' mode clicking on a curve point and dragging will edit the curves position. Clicking anywhere in the 3D viewport other than the handles will create a new curve control point. To avoid creating new control points when dragging existing points around you can set this mode to 'Edit'. 'Delete' mode allows quick removal of control points by simply clicking on them.

Point X/Y/Z: The XYZ position of the currently selected knot. The user can interactively edit the position by clicking over the knot indicated by the cyan dot, it will turn yellow; the user can now drag its position around in the viewport, or manually enter a position by editing any of the data fields.

Bank Angle: For each path position, a bank angle can be defined that controls the

rotation angle of elements cloned through that position.

Closed: This toggle option will cause the curve to become a closed loop, connecting the end point to the start.

Start Control: Enabling this toggle will cause the first knot in the curve to become a control point but not an actual segment of the curve. This provides a simple method for changing the curvature at the start of the curve.

End Control: Enabling this toggle will cause the last knot in the curve to become a control point but not an actual segment of the curve. This provides a simple method for changing the curvature at the end of the curve.

By Length: This option forces the newly created geometry to be evenly distributed across the entire length of the curve rather than spacing based on the control knots.

Create Polygon: This toggle option will generate a curve type polygon that remains after the tool has been dropped. This is useful if you want to re-use the curve you create with the current tool with additional curve based tools later in your project.

Flip Polygon: This toggle will invert the faces of the generated polygons.

Make UVs: When this toggle is active the tool will populate the current UV map with data for the generated polygons.

Sweep Effector--

Sweep UVs: This popup control allows you to choose between the U or V axis when placing UV data for newly created geometry.

Invert Polygons: Reverses the normal facing direction of the polygons generated by the extrusion. Useful when extruded geometry appears inverted or inside-out.

Cap Start/End: This toggle will force the back end of the extruded geometry to be capped or closed.

Profile Attributes--

Size: The size of 1D or 2D profile to sweep. If the size is 0.0, it uses the real size.

Curve Subdivision Tolerance: This is the minimum height when it subdivides bezier curve to polyline.

Pivot: Specify the pivot position of the profile.

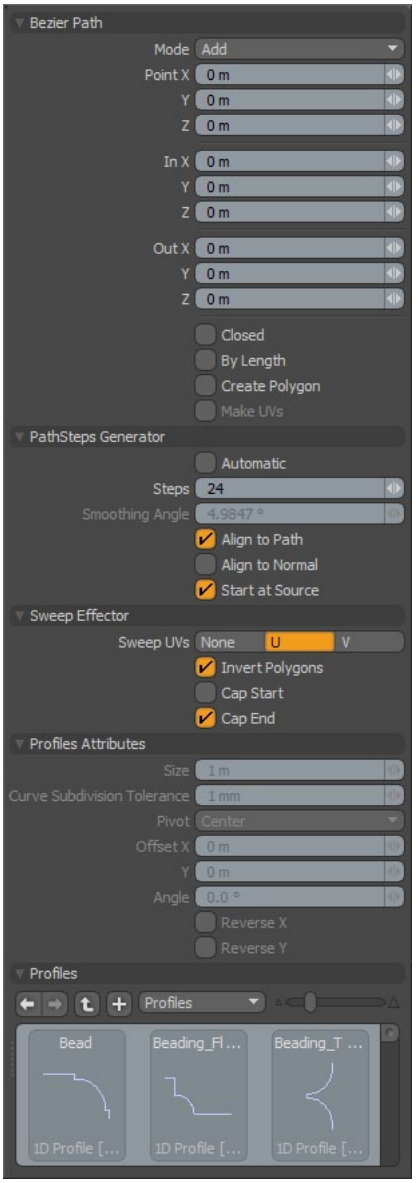
Offset X/Y: These are the offset amount on the profile space.

Reverse X/Y: Reverse the selected profile to horizontal or vertical.

Profiles--

Mini Preset browser for viewing various profiles. Works the same as the standard preset browser. Selecting any of the various profiles will modulate the extrusion width with the selected shape over its length. To return to the original polygon selection for the extrusion, ctrl-click the selected profile to deselect it.

Bezier Extrude



The Bezier Extrude tool allows one to quickly extrude a geometry selection along the length of a curve. If the curve is pre-existing, select it first, then the polygonal geometry for extrusion. 'Bezier Path' will also allow the user to create a path interactively. To use, select the tool and make it interactive by clicking in the viewport. Once activated, a number of properties are available

Bezier Path--

Mode: By default the mode is set to Add. In the 'Add' mode clicking on a curve point and dragging will edit the curves position. Clicking anywhere in the 3D viewport other than the handles will create a new curve control point. To avoid creating new control points when dragging existing points around you can set this mode to 'Edit'. 'Delete' mode allows quick removal of control points by simply clicking on them.

Point X/Y/Z: Defines the XYZ position of the currently selected knot. The user can interactively edit the position by clicking over the knot indicated by the cyan dot, it will turn yellow; the user can now drag its position around in the viewport, or manually enter a position be editing any of the data fields.

In/Out X/Y/Z: Defines the XYZ position of the currently selected in or out bezier handle. The user can interactively edit the position by clicking

over the handle indicated by the cyan rectangle at the end of the purple dotted line, it will turn yellow; the user can now drag its position around in the viewport, or manually enter a position be editing any of the data fields.

Closed: This toggle option will cause the curve to become a closed loop.

By Length: This option forces the newly created geometry to be evenly distributed across the entire length of the curve rather than spacing based on the control knots.

Create Polygon: This toggle option will generate a curve type polygon that remains after the tool has been dropped. This is useful if you want to re-use the curve you create with the current tool with additional curve based tools later in your project.

Make UVs: When enabled, modo will automatically create a UV map for along the length of the extrusion.

PathSteps Generator--

Automatic: This toggle control will force a computed number of segments based on the Smoothing Angle value such that an edge is created along the path to maintain a level of smoothness at or above the user set level.

Steps: Active only when Automatic is off, this is the number of segments generated along the curve path.

Smoothing Angle: Defines a minimum angle for generating a new segment for 'Automatic'.

Align to Path: When enabled the segment edges will align with the curves direction vector producing a consistent thickness.

Align to Normal: Aligns the Extrusion shape to the normal direction of the initial polygon selection.

Start at Source: Often a curve used for extruding will have its initial control knot imperfectly aligned with the geometry to be extruded. WIth 'Start at Source' active the curve is used as a relative guide using the original geometry as the true starting location. Disabling this option may cause the newly created geometry to move away from the intended original location.

Sweep Effector--

Sweep UVs: This popup control allows you to choose between the U or V axis when placing UV data for newly created geometry.

Invert Polygons: Reverses the normal facing direction of the polygons generated by the extrusion.

Cap Start/End: This toggle will force the back end of the extruded geometry to be capped or closed.

Profile Attributes--

Size: The size of 1D or 2D profile to sweep. If the size is 0.0, it uses the real size.

Curve Subdivision Tolerance: This is the minimum height when it subdivides bezier curve to polyline.

Pivot: Specify the pivot position of the profile.

Offset X/Y: These are the offset amount on the profile space.

Reverse X/Y: Reverse the selected profile to horizontal or vertical.

Profiles--

Mini Preset browser for viewing various profiles. Works the same as the standard preset browser. Selecting any of the various profiles will modulate the extrusion width with the selected shape over its length. To return to the original polygon selection for the extrusion, ctrl-click the selected profile to deselect it.

Pen Extrude

The Pen Extrude tool allows one to quickly extrude a geometry selection along the length of a pen path where each knot creates a new segment. To use, select the tool and make it interactive by clicking in the viewport. Once activated, a number of properties are available.

Pen Generator--

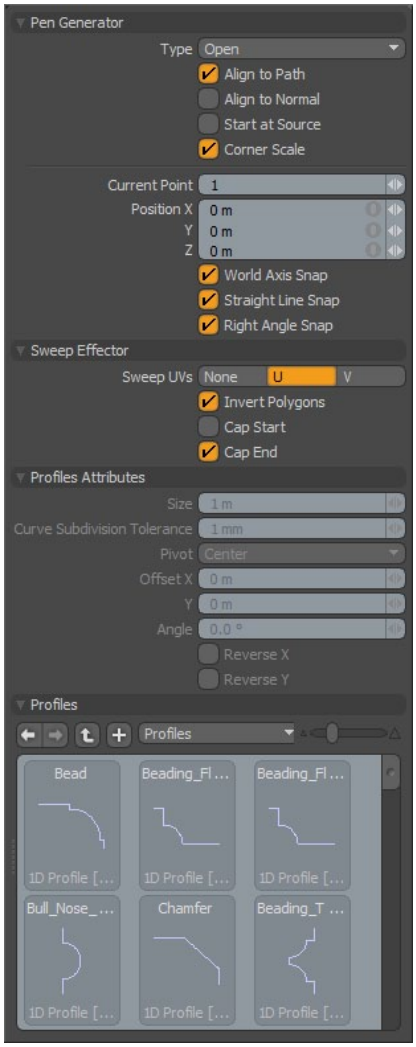
Type: You can choose to generate an 'Open' or 'Closed' path, 'Closed' automatically generates an extra path between the first and last knots, producing a closed loop.

Align to Path: When enabled the segment edges will align with the curves direction vector producing a consistent thickness.

Align to Normal: Aligns the Extrusion shape to the normal direction of the initial polygon selection.

Start at Source: Often times an extrusion will have its initial control knot imperfectly aligned with the geometry to be extruded. WIth 'Start at Source' active the curve is used

as a relative guide using the original geometry as the true starting location. Disabling this option may cause the newly created geometry to move away from the intended original location.



Corner Scale: Automatically scales corner segments as to maintain the proper thickness over the extrusions length.

Current Point: You can change this setting to change the current vertex to be edited by the numeric values listed as Position X, Y, Z. Since the tool may have created several vertices, it is useful to be able to choose one numerically. The vertices are numbered in sequence as they are generated by the tool.

Position X/Y/Z: This numeric entry field allows you to precisely edit each vertex created by the tool. Simply set the Current Point value to the vertex you wish to edit and enter the numeric values for the X, Y and Z positions.

World Axis Snap: When this is active the pen tool handles will snap along the world XYZ axis once you have created more than one vertex.

Straight Line Snap: Will try to snap the handle to a position that will create a straight line between the next and previous vertices in the polygon being created.

Right Angle Snap: Will effect a snap when the angle created by the current, previous and next vertices create a 90 degree angle.

Sweep Effector--

Sweep UVs: This popup control allows you to choose between the U or V axis when placing UV data for newly created geometry.

Invert Polygons: Reverses the normal facing direction of the polygons generated by the extrusion.

Cap Start/End: This toggle will force the back end of the extruded geometry to be capped or closed.

Profile Attributes--

Size: The size of 1D or 2D profile to sweep. If the size is 0.0, it uses the real size.

Curve Subdivision Tolerance: This is the minimum height when it subdivides bezier curve to polyline.

Pivot: Specify the pivot position of the profile.

Offset X/Y: These are the offset amount on the profile space.

Reverse X/Y: Reverse the selected profile to horizontal or vertical.

Profiles--

Mini Preset browser for viewing various profiles. Works the same as the standard preset browser. Selecting any of the various profiles will modulate the extrusion width with the selected shape over its length. To return to the original polygon selection for the extrusion, ctrl-click the selected profile to deselect it.

Curve Clone

The Curve Clone tool allows users to quickly clone geometry by clicking down control points for a curve. The cloned element is determined by the settings of the 'Source' option of the 'Clone Effector'. As the curve is created, the specified geometry will duplicate itself along the curve in real-time (depending on system performance).
One may wish to begin with a pre-created curve, simply put the curve in visible but deselected mesh item prior to activating the tool (also known as a Background item).
Once the second vertex of the curve has been created with the tool, the clone effector will create geometry along it using the various controls.

Path Steps Generator--

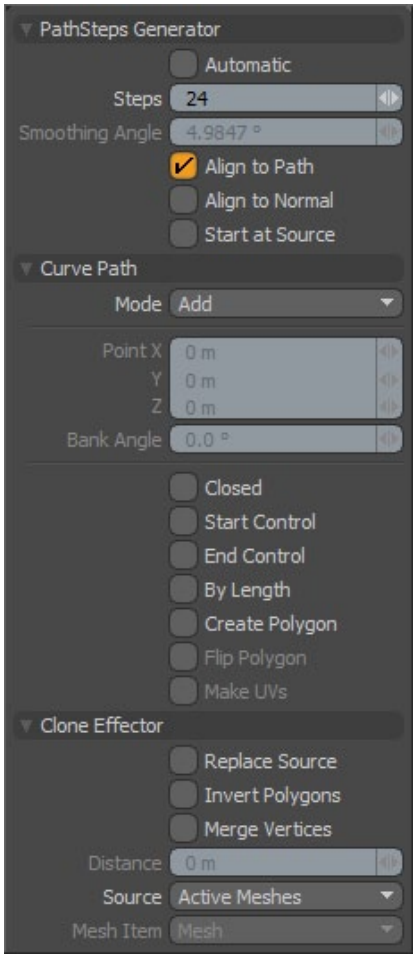
Automatic: This toggle control will force a computed number of clones based on the Smoothing Angle value such that a clone is placed along the path to maintain a level of smoothness at or above the user set level (determined by the 'Smoothing Angle').

Steps: Active only when Automatic is off this is the number of particles generated along the curve path to feed into the Effector.

Smoothing Angle: Determines the angle along the curve where a clone will be created when the 'Automatic' option is *enabled*.

Align to Path: When *enabled* the generated particles will include rotation information so the newly created clone aligns to the path vector direction at the location where it is created.

Align to Normal: When enabled, the generated particles will include rotation information so the newly created clone aligns with any stored normal direction for a vertex, such as for a curve extracted from geometry.



Start at Source: Often a curve used for cloning will have its initial control knot imperfectly aligned with the geometry to be cloned. With Start at Source active the curve is used as a relative guide using the original geometry as the true starting location. Disabling this option may cause the newly created geometry to move away from the intended original location.

Curve Path--

Mode: By default the mode is set to Add. In the 'Add' mode clicking on a curve point and dragging will edit the curves position. Clicking anywhere in the 3D viewport other than the handles will create a new curve control point. To avoid creating new control points when dragging existing points around you can set this mode to 'Edit'. 'Delete' mode allows quick removal of control points by simply clicking on them.

Point X/Y/Z: The XYZ position of the currently selected knot. The user can interactively edit the position by clicking over the knot indicated by the cyan dot, it will turn yellow; the user can now drag its position around in the viewport, or manually enter a position by editing any of the data fields.

Bank Angle: For each path position, a bank angle can be defined that controls the rotation angle of elements cloned through that position.

Closed: This toggle option will cause the curve to become a closed loop, connecting the end point to the start.

Start Control: Enabling this toggle will cause the first knot in the curve to become a control point but not an actual segment of the curve. This provides a simple method for changing the curvature at the start of the curve.

End Control: Enabling this toggle will cause the last knot in the curve to become a

control point but not an actual segment of the curve. This provides a simple method for changing the curvature at the end of the curve.

By Length: This option forces the newly created geometry to be evenly distributed across the entire length of the curve rather than spacing based on the control knots.

Create Polygon: This toggle option will generate a curve type polygon that remains after the tool has been dropped. This is useful if you want to re-use the curve you create with the current tool with additional curve based tools later in your project.

Flip Polygon: This toggle will invert the faces of the generated polygons.

Make UVs: When this toggle is active the tool will populate the current UV map with data for the generated polygons.

Clone Effector--

Replace Source: When this option is enabled, modo will remove the source geometry and position it at the first Point in the path.

Invert Polygons: This option, when *enabled* will invert the normal direction of the cloned geometry.

Merge Vertices: This option, when *enabled* will merge coinciding vertices into a single vertex (based on a specified 'Distance' value).

Distance: Used in conjunction with the 'Merge Vertices' option, the Distance value determines the maximum distance between vertices where merging will occur. Vertices outside this value will remain unchanged. A value of zero will only remove collocated vertices.

Source: Determines the mesh that will be cloned along the Pen tool path-

Active Meshes- This option will use the currently selected geometry as a clone source.

Specific Mesh- This option allows the user to specify a specific Mesh item as the clone source using the 'Mesh Item' option below.

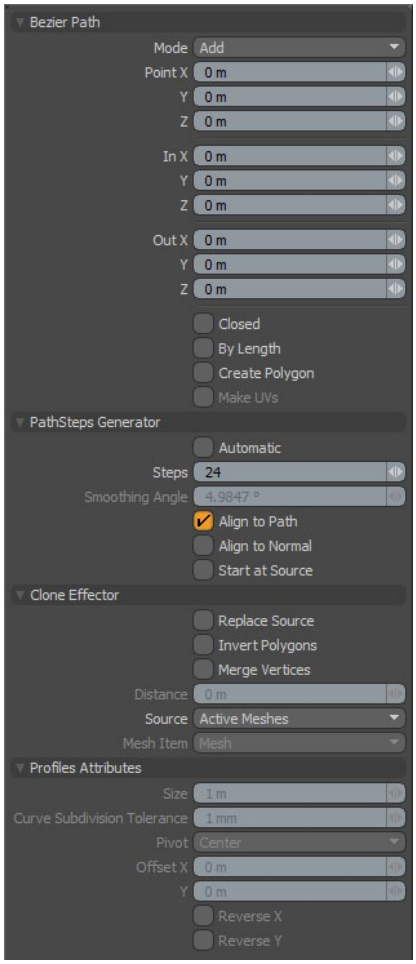
All BG- This option will combine all the currently visible background items cloning them as a single mesh.

Random BG- This option chooses a random background item from all the currently visible background items at each clone position.

Preset Shape- This option allows users to clone Profile shapes from the Preset Browser. Open the Preset Browser 'F6" and select a profile before activating the tool.

Mesh Item: This option works when the Source is set to 'Specific Mesh' and allows users to set the mesh to be cloned.

Bezier Clone



A Bezier Curve is a type of curve defined by control vertices. Each vertex is controlled by two other points that control the endpoint tangent vectors. The Bezier handles are then used to adjust the curvature of the path. Geometry or items may be cloned along this path defined by the settings of the clone effector. Each cloned item will be duplicated as a straight mesh, increasing the polygon count within the scene file. modo also offers users options to clone items as 'Instances' or as 'Replicator' items which do not increase scene geometry. The Bezier Clone tool is available in the menu bar under "Geometry > Duplicate > Bezier Clone". Users may also select the tool directly in the 'Duplicate' subtab of the modeling toolbox found in the 'Model' and 'Model Quad' interface tabs. Users can LMB+click and hold the toolbox button to open a fly out menu providing the alternate tool options.

To use, simply activate the tool by LMB+clicking the tools icon, once active the following properties will appear in the tool properties panel of the default interface. Users can then LMB+click in the 3D viewport to create curve Points at the intersection of the mouse click in the viewport and the Work Plane. The curved path will be created between each path point as subsequent points are added. Bezier Curves support three modes: 'Add', 'Edit' and 'Delete'. You can use these modes to create additional points, edit them, or remove them from the curve, during the drawing operation. Users can interactively edit the curves as well, by selecting a particular point in the path by hovering the mouse over the point and LMB+clicking to change it to yellow selecting it, users can then LMB+click and drag the selected element to modify its position, this also works for editing the bezier handles. Users can also add points between existing positions by selecting a point and then LMB+clicking away from the selected point adding the intermediate point. Points on the Bezier Curve can be positioned precisely by using the Point XYZ fields of the tools properties form. Once

the tool is *dropped* by pressing the 'Q' key, the cloned items will be in effect *set* and no further editing of the path is possible.

Bezier Path--

Mode: The various bezier curve mode options available are--

Add-- The default mode, when users click in the viewport, additional points are added to the curve. At each control point position two handles extend out that allow users to adjust the curvature of the line between vertices. While drawing users can hover over any point or handle where it will turn yellow, users can then LMB-click and drag the control point or handle to further edit the curve. Control points may be added mid-curve by selecting the preceding control, highlighting it yellow, and clicking the position where the new control point is desired.

Edit-- In edit mode users can freely LMB+click drag and of the control points or handles along the curve to change the look to the desired shape. While pressing the 'Ctrl' key, users may break the tangency of the control handles, allowing for sharp corners, holding 'Shift' while clicking a handle will restore tangency.

Delete-- In delete mode, users can LMB+click on any control point along the curve to remove it from the bezier.

Point XYZ: Users wishing finer point control can assign specific XYZ values in these input fields for the currently selected (highlighted) control point.

In XYZ/Out XYZ: Users wishing finer handle control can assign specific XYZ values in these 'In' and 'Out' input fields for the currently selected (highlighted) control point.

Closed: The 'Closed' toggle option adds an automatic curve segment between the first and last control point positions producing a closed curve.

By Length: When the 'By Length' option is enabled, cloned items will be evenly distributed along the path by length rather than by the span, which may produce uneven placement.

Create Polygon: When the 'Create Polygon' option is enabled, dropping the tool will leave the Bezier Curve path instead of removing it from the scene. This is useful should the user wish to use the resulting curve for additional purposes.

Make UVs: Makes UVs for normal bezier curves that are converted to polygons (*disabled for the Bezier Clone tool as it is unnecessary as the curve is not converted to a straight polygon*).

Path Steps Generator--

Automatic: When the 'Automatic' option is enabled, modo will assign the number of clones to the path based on the number of control points. Additional control points will produce further clones on the path.

Steps: Available when 'Automatic' is disabled, users can use this value to determine an exact number of clones along the path.

Smoothing Angle:

Align to Path: This option, when enabled, will rotate the cloned items to align along the paths directional vector, meaning the objects will be rotated to match the direction of the path at the clones position.

Align to Normal: When this option is enabled, modo will try to align the direction of the source objects to a straight up position before cloning the source item(s). If the item is already upright when cloned, this option will likely have no effect.

Start at Source: By default, modo will clone items at the actual world position along the bezier curve. When the 'Start at Source' option is enabled, modo will offset each cloned item the distance between the source items center point and the first curve point,producing an effect as if the curve were initiated precisely at the center of the first source item.

Clone Effector--

Replace Source: When this option is enabled, modo will remove the source geometry and position it at the first Point in the path.

Invert Polygons: This option, when enabled will invert the normal direction of the cloned geometry.

Merge Vertices: This option, when enabled will merge coinciding vertices into a single vertex.

Distance: Used in conjunction with the 'Merge Vertices' option, the Distance value determines the maximum distance between vertices where merging will occur. Vertices outside this value will remain unchanged.

Source: Determines the mesh that will be cloned along the Pen tool path-

Active Meshes- This option will use the currently selected geometry as a clone source.

Specific Mesh- This option allows the user to specify a specific Mesh item as the clone source using the 'Mesh Item' option below.

All BG- This option will combine all the currently visible background items cloning them as a single mesh.

Random BG- This option chooses a random background item from all the currently visible background items at each clone position.

Preset Shape- This option allows users to clone Profile shapes from the Preset Browser. Open the Preset Browser 'F6" and select a profile before activating the tool.

Mesh Item: This option works when the Source is set to 'Specific Mesh' and allows users to set the mesh to be cloned.

Profile Attributes--

When users define the 'Source' as the 'Preset Shape' above, the following Profile attribute become active to determine the size and position of the cloned Preset shape.

Size: Determines the size of the cloned Preset Shape.

Curve Subdivision Tolerance: When the preset shapes curve is converted into a polygon once cloned, this option determines the fidelity of that conversion.

Pivot: This option determines the position of the cloned Preset Shape in relation the Point of the path.

Offset: This option provides finer control over the Preset Shapes position in relation to the path itself. This value is additive to the Pivot position specified above.

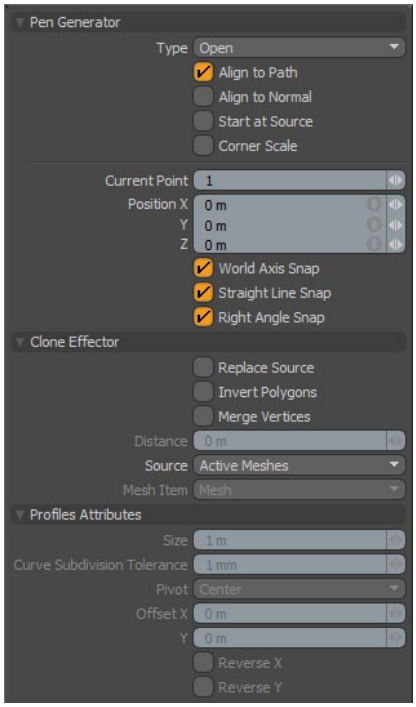
Reverse X/Y: This option flips the Preset Shape in either the 'X' or 'Y' direction.

Pen Clone

The Pen tool is useful in creating linear paths that have perfectly straight line segments between each user defined point. Geometry or items may be cloned along this path defined by the settings of the clone effector. Each cloned item will be duplicated as a straight mesh, increasing the polygon count within the scene file. modo also offers users options to clone items as 'Instances' or as 'Replicator' items which do not increase scene geometry. The Pen Clone tool is available in the menu bar under "Geometry > Duplicate > Pen Clone". Users may also select the tool directly in the 'Duplicate' subtab of the modeling toolbox found in the 'Model' and 'Model Quad' interface tabs. Users can LMB+click and hold the toolbox button to open a fly out menu providing the alternate tool options.

To use, simply activate the tool by LMB+clicking the tools icon, once active the following properties will appear in the tool properties panel of the default interface. Users can

then LMB+click in the 3D viewport to create path Points at the intersection of the mouse click in the viewport and the Work Plane. Lines will be created between each path point as subsequent points are added. Users can edit selected path points using the Position input field. Users can select a particular path by hovering the mouse over the point until it changes to yellow, then LMB+click and drag to modify its position. Users can also add points between existing positions by selecting a point and then LMB+clicking away from the selected point adding the intermediate point. Once the tool has been dropped by pressing the 'Q' key the cloned items will be in effect *set* and no further editing of the path is possible.



Pen Generator--

Type: This option determines the termination of the path, whether is stays 'Open' as the user creates the path, or if it is 'Closed, where a line is drawn between the first and last points making the path a continuous loop.

Align to Path: This option, when *enabled*, will rotate the cloned items to align along the paths directional vector, meaning the objects will be rotated to match the direction of the path at the clones position.

Align to Normal: When this option is *enabled*, modo will try to align the direction of the source objects to a straight up position before cloning the source item(s). If the item is already upright when cloned, this option will likely have no effect.

Start at Source: By default, modo will attempt to place the base of the cloned geometry at each path Point, useful when cloning items in the same layer, the 'Start at Source' option, when *enabled*, will position cloned items based on their Center point.

Corner Scale: The path tool, when used in conjunction with the 'Corner Scale' option for creating geometry will automatically scale *corners* to make a path have a consistent thickness, this same option is also available for cloning items should the user desire their clones match a previously created pen tool path.

Current Point: This option displays/determines the currently selected path 'Point'. Users can select any point along the path by LMB+clicking directly on the 'Points' handle when yellow, or specify a 'Point' manually by adjusting the value in this field.

Position: This trifecta of position values determines the point in 3D space where the current Point is positioned. USers can manually enter a new position with the input fields, or select the Point in the 3D viewport and position it via the mouse.

World Axis Snap: When this option is enabled, modo will attempt to snap Path line segments to 90° angles relative to the World axes.

Straight Line Snap: When this option is enabled, modo will attempt to snap Path line segments to straight lines aligned with the previous two Points in the path.

Right Angle Snap: When this option is enabled, modo will attempt to snap Path line segments to 90° angles relative to the previous path segment.

Clone Effector--

Replace Source: When this option is enabled, modo will remove the source geometry and position it at the first Point in the path.

Invert Polygons: This option, when enabled will invert the normal direction of the cloned geometry.

Merge Vertices: This option, when enabled will merge coinciding vertices into a single vertex.

Distance: Used in conjunction with the 'Merge Vertices' option, the Distance value determines the maximum distance between vertices where merging will occur. Vertices outside this value will remain unchanged.

Source: Determines the mesh that will be cloned along the Pen tool path-

Active Meshes- This option will use the currently selected geometry as a clone source.

Specific Mesh- This option allows the user to specify a specific Mesh item as the clone source using the 'Mesh Item' option below.

All BG- This option will combine all the currently visible background items cloning them as a single mesh.

Random BG- This option chooses a random background item from all the currently visible background items at each clone position.

Preset Shape- This option allows users to clone Profile shapes from the Preset Browser. Open the Preset Browser 'F6" and select a profile before activating the tool.

Mesh Item: This option works when the Source is set to 'Specific Mesh' and allows users to set the mesh to be cloned.

Profile Attributes--

When users define the 'Source' as the 'Preset Shape' above, the following Profile attribute become active to determine the size and position of the cloned Preset shape.

Size: Determines the size of the cloned Preset Shape.

Curve Subdivision Tolerance: When the preset shapes curve is converted into a polygon once cloned, this option determines the fidelity of that conversion.

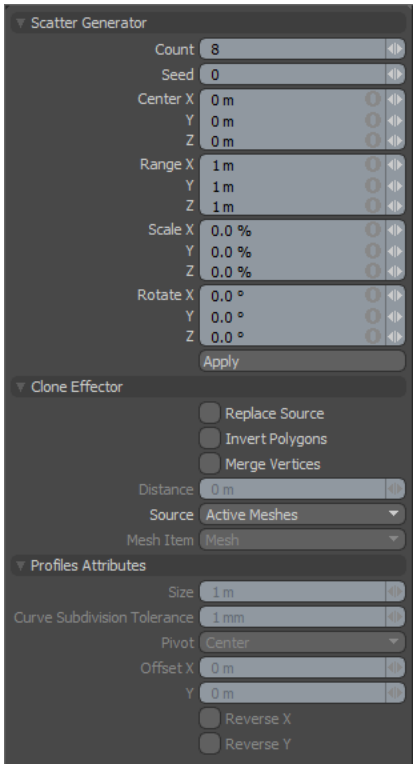
Pivot: This option determines the position of the cloned Preset Shape in relation the Point of the path.

Offset: This option provides finer control over the Preset Shapes position in relation to the path itself. This value is additive to the Pivot position specified above.

Reverse X/Y: This option flips the Preset Shape in either the 'X' or 'Y' direction.

Scatter Clone

The Scatter Clone tool generates a random point cloud onto which clones of the original geometry are placed.



Scatter Clone--

Count: This numeric value determines the total number of clones (including the source).

Center XYZ: This triplet determines the center position of the scatter clone effect.

Range XYZ: This triplet distance value determines the maximum distance on each axis the clones can be from the center of the effect.

Scale XYZ: This triple percentage value determines the maximum amount of randomized fluctuation for any given clone.

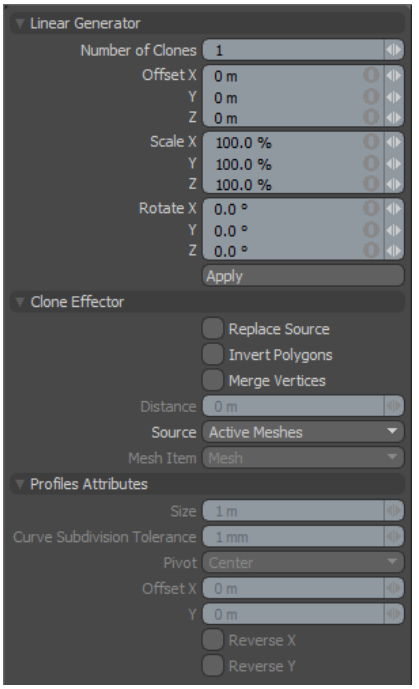
Rotate XYZ: This triple degree value determines the maximum amount of randomized rotation for any given clone.

Replace Source: When activated, this control will remove the original geometry and create a

replacement clone for it at the center of the scatter clone effect.

Clone

The Clone tool will create duplicates of the currently selected geometry in a linear offset determined by user input. Clone can be accessed on the Duplicate tab of the modo Tools toolbar, or by choosing Geometry > Create > Clone. The tool is controlled by the following settings.



Clone--

Number of Clones: This numeric value determines the number of copies of the original geometry to be created.

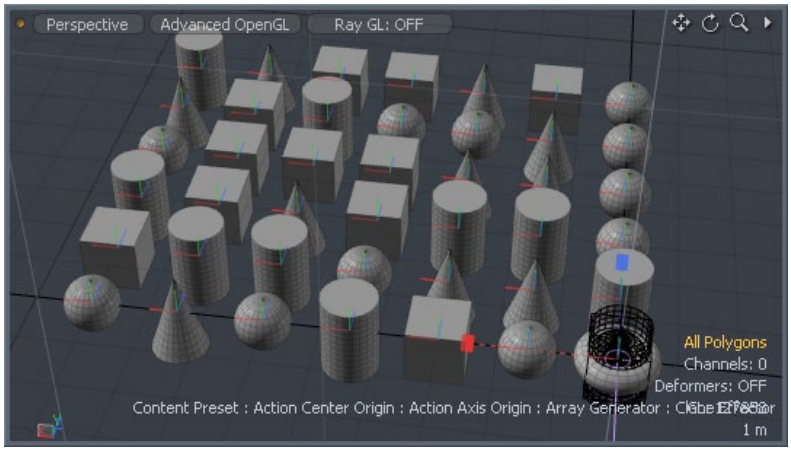
Offset XYZ: These distance values determine how far each clone is from the previous. Setting this value to 1 meter would place each successive cloned object 1 meter away from the previous clone or the original geometry in the case of the first clone.

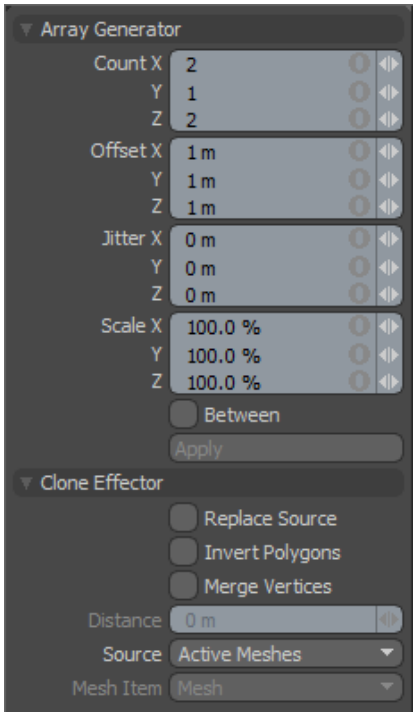
Scale XYZ: These three percentage settings control the relative size of each clone to the one before it. Setting the value to 50% would force each clone to be half the size of the previous.

Rotate XYZ: The rotate control allows you to set a rotational value in degrees for each of the three axes. This rotation amount will be applied to each clone. This can be very useful for creating objects such as spiral staircases etc.

Replace Source: This toggle option will cause the tool to delete and replace the original source geometry. This can be very useful when using scale and rotation values.

Array





The 'Array' tool provides users a means to produce ordered duplicates of a mesh or meshes, cloning a source element into a grid-like pattern. Users can enable this tool by selecting the Array icon from the 'Duplicate' subtab of the modeling toolbox. The 'Array function can also be found in the menu bar under "Geometry >Duplicate > Array". Once the tool has been *enabled*, a series of options become available in the tool properties panel. Users can adjust these settings and press the 'Apply' button to duplicate the geometry into the array, or by LMB+clicking in any 3D OpenGL viewport, users can initiate an *interactive* mode for the tool, where users can manipulate handles in the 3D viewport to adjust the distance setting. By RMB+clicking on the 'Array' button in the toolbox, a pop-up opens showing additional Array tools covered further below.

Array Generator--

Count X/Y/Z: The 'Count' option determines the number of clones in each axial direction. Users should note that the array produces a *volume* of cloned elements that can increase polygon counts very quickly, even with small numbers, such as a 4x4x4 array produces 64 cloned elements. The use of 'Instances' or 'Replicators' (also covered below) can help ease the amount of geometry generated by the Array tool.

Offset X/Y/Z: The 'Offset' option determines the distance away from the source element to produce the cloned elements, calculated as the distance between the source and the first order of clones, subsequent clones will be offset the same amount again. When *enabling* the 'Between' option, this value determines the distance between the first and last clones with all others evenly spaced between those points.

Jitter X/Y/Z: The 'Jitter' controls allow some random variation to be added to any of the three axes (X,Y or Z). Each copied mesh will be positioned with the offset and a random variation using the Jitter distance as the maximum possible offset variation.

Between: The 'Between' option, when enabled will allow users to define a first position and a last position and the rest of the clones will be generated between the existing copies.

Clone Effector--

Replace Source: When this option is *enabled*, modo will remove the source geometry and position it at the first Point in the path.

Invert Polygons: When this option is *enabled* the normal direction on cloned geometry is reversed. Can be useful in instances where negatively cloned elements flip the normals inward producing inside-out gematria.

Merge Vertices: When *enabled*, the 'Merge Vertices' option automatically joins vertices into a single vertex based on the 'Distance' setting.

Distance: If 'Merge Vertices' is *enabled*, the 'Distance' setting determines the maximum distance between neighboring vertices where they will be joined into a single vertex. A setting of '0m' only joins vertices that are collocated.

Source: Determines the source mesh that will be cloned into the array-

Active Meshes- This option will use the currently selected geometry as a clone source.

Specific Mesh- This option allows the user to specify a specific Mesh item as the clone source using the 'Mesh Item' option below.

All BG- This option will combine all the currently visible background items cloning them as a single mesh.

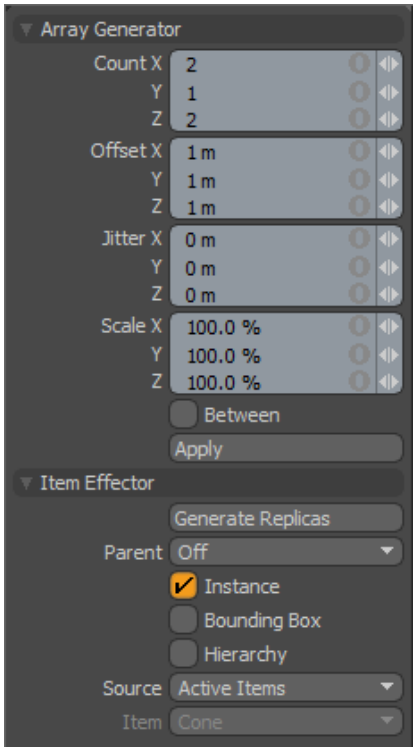
Random BG- This option chooses a random background item from all the currently visible background items at each clone position.

Preset Shape- This option allows users to clone Profile shapes from the Preset Browser. Open the Preset Browser 'F6" and select a profile before activating the tool.

Mesh Item: This option works when the Source is set to 'Specific Mesh' and allows users to set the specific mesh to be cloned.

Instance Array

The 'Instance Array' tool functions in a similar way to the standard Array tool, but produces Instances instead of mesh items or geometry. Instances are useful in that they don't contain any physical geometry, but merely reference a source item, keeping scene files light. Additionally, any edits made to the source geometry are instantly propagated to all the instanced clones making it easy to update the geometry should it be necessary. There is no visual difference between an instance over actual geometry. The 'Instance Array' tool is found under the standard 'Array' toolbox button by RMB+clicking on the array tool to open the pop-up menu and then selecting the 'Instance Array' option.



Array Generator--

Count X/Y/Z: The 'Count' option determines the number of clones in each axial direction. Each clone will be an 'Instance' of the original mesh, containing no geometry itself but referencing a source mesh. Any edits or modifications to the source will be propagates to each instance.

Offset X/Y/Z: The 'Offset' option determines the distance away from the source element to produce the cloned elements, calculated as the distance between the source and the first order of clones, subsequent clones will be offset the same amount again. When *enabling* the 'Between' option, this value determines the distance between the first and last clones with all others evenly spaced between those points.

Jitter X/Y/Z: The 'Jitter' controls allow some random variation to be added to any of the three axes (X,Y or Z). Each copied mesh will be positioned with the offset and a random variation using the Jitter distance as the maximum possible offset variation.

Between: The 'Between' option, when *enabled* will allow users to define a first position and a last position and the rest of the clones will be generated *between* the existing copies.

Item Effector--

Parent: This option determines how Instanced items are treated in the Items List-

Off- Simply clones source item as additional instanced layers in the Items List.

Hierarchy- Creates a cascading array where each subsequent clone will be a child of the previous clone.

Root- Creates instances as child items of the source mesh.

Primary- Creates Instances as children of a Group Locator that is parented to the Source mesh.

Instance: When this option is *enabled* (default) each cloned item will be an Instance of the source mesh. Instances contain no physical geometry, but reference a source layer and are represented in 3D viewports by their pink wireframe representations.

Bounding Box: This option toggles the display mode of instances, when enabled the resulting instances will display as bounding boxes rather than wireframe meshes, providing a small amount of 3D viewport performance increase when many instances are created.

Hierarchy: When this option is *enabled*, any child items of the source geometry will also be cloned, retaining their hierarchy through each subsequent clone.

Source: Determines the source mesh that will be cloned into the array-

Active Items- This option will use the currently selected item(s) as a clone source.

Specific Item- This option allows the user to specify a specific Mesh item as the clone source using the 'Mesh Item' option below.

All BG- This option will combine all the currently visible background items cloning them as a single mesh.

Random BG- This option chooses a random background item from all the currently visible background items at each clone position.

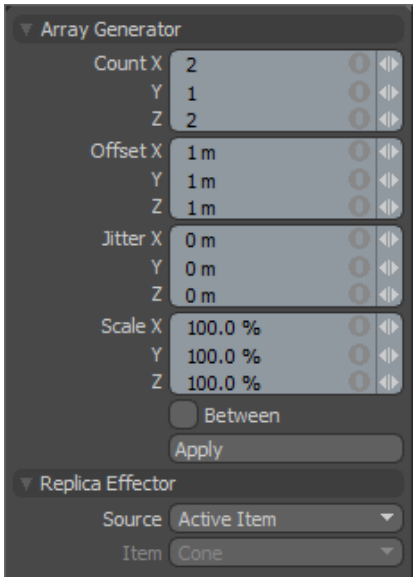
Mesh Layer Preset- This option allows users to clone Mesh Presets from the Preset Browser. Open the Preset Browser 'F6" and select a Mesh preset before activating the tool.

Mesh Item: This option works when the Source is set to 'Specific Item' and allows users to set the target mesh to be cloned.

Replica Array

The 'Replica Array' tool produces and array of cloned items, just like the 'Array' tool,

but generates them using a Replicator instead. Much like Instances, Replicas don't add additional geometry to the scene, all that is created is a point source mesh item layer with vertices representing the item positions and a 'Replicator' item properly defined with a source and prototype. The 'Replica Array' tool is found under the standard 'Array' toolbox button by RMB+clicking on the array tool to open the pop-up menu and then selecting the 'Replica Array' option.



Array Generator--

Count X/Y/Z: The 'Count' option determines the number of clones in each axial direction. Each clone will be an 'Instance' of the original mesh, containing no geometry itself but referencing a source mesh. Any edits or modifications to the source will be propagates to each instance.

Offset X/Y/Z: The 'Offset' option determines the distance away from the source element to produce the cloned elements, calculated as the distance between the source and the first order of clones, subsequent clones will be offset the same amount again. When *enabling* the 'Between' option, this value determines the distance between the first and last clones with all others evenly spaced between those points.

Jitter X/Y/Z: The 'Jitter' controls allow some random variation to be added to any of the three axes (X,Y or Z). Each copied mesh will be positioned with the offset and a random variation using the Jitter distance as the maximum possible offset variation.

Between: The 'Between' option, when *enabled* will allow users to define a first position and a last position and the rest of the clones will be generated between the existing copies.

Replica Effector--

Source: Determines the source mesh that will be cloned into the array-

Active Item- This option will use the currently selected geometry as a clone source.

Specific Item- This option allows the user to specify a specific Mesh item as the clone source using the 'Mesh Item' option below.

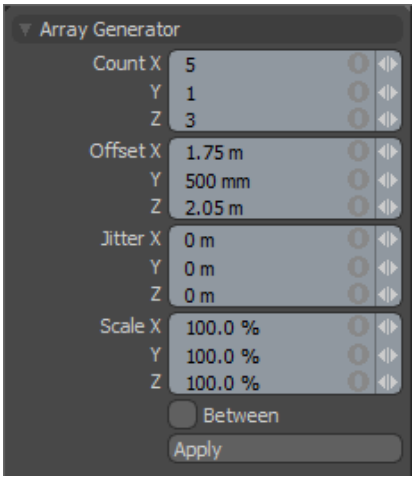
All BG- This option will combine all the currently visible background items cloning them as a single mesh.

Preset Shape- This option allows users to clone Profile shapes from the Preset Browser. Open the Preset Browser 'F6" and select a profile before activating the tool.

Item: This option works when the Source is set to 'Specific Item' and allows users to set the target mesh to be cloned.

Transform Array

The 'Transform Array' tool creates ordered arrays of existing items. In order to use the 'Transform Array tool, users will need to select the target items in which to affect prior to activating the tool. Regardless of their current positions, the items will be moved to the new array positions based on their 'Center' points. The 'Transform Array' tool is found under the standard 'Array' toolbox button by RMB+clicking on the array tool to open the pop-up menu and then selecting the 'Transform Array' option.



Array Generator--

Count X/Y/Z: The 'Count' option determines the number of clones in each axial direction. Each clone will be an 'Instance' of the original mesh, containing no geometry itself but referencing a source mesh. Any edits or modifications to the source will be propagates to each instance.

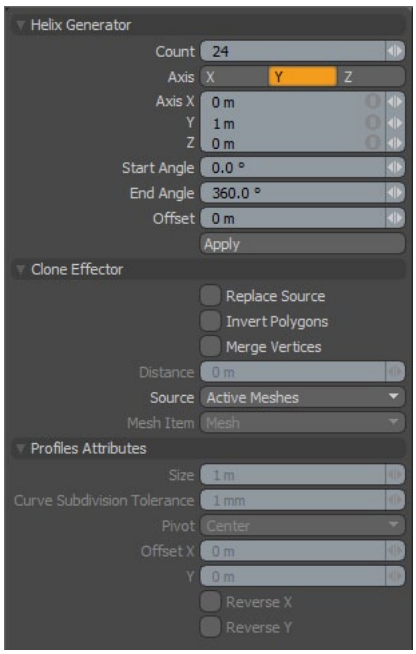
Offset X/Y/Z: The 'Offset' option determines the distance away from the source element to produce the cloned elements, calculated as the distance between the source and the first order of clones, subsequent clones will be offset the same amount again. When *enabling* the 'Between' option, this value determines the distance between the first and last clones with all others evenly spaced between those points.

Jitter X/Y/Z: The 'Jitter' controls allow some random variation to be added to any of the three axes (X,Y or Z). Each copied mesh will be positioned with the offset and a random variation using the Jitter distance as the maximum possible offset variation.

Between: The 'Between' option, when *enabled* will allow users to define a first position and a last position and the rest of the clones will be generated *between* the existing copies.

Radial Array

The Radial Array tool copies the selected geometry into a circular or helix pattern. There are a variety of controls that allow precise control over the count, placement, angles and spiral.



Count: Controls the number of clones created.

Axis: Determines along which axis to create the helix.

Axis XYZ: These three values determine the location of the tools Axis control handles. If the values are set to 1 meter on all axes the handles ends will have one end at -1 X, -1 Y, -1 Z and the other at 1 X, 1 Y, 1 Z creating an skewed off axis plane for the array.

By default the Start and End Angles go from 0 to 360 degrees. This creates an array that completes a circle. Changing these two values can create an array that revolves many times or one that leaves one side of the circle agape. Setting the end value to greater than 360 degrees is very useful when combined with some non-zero offset value.

Start Angle: This degree setting determines the rotation placement for the first cloned geometry.

End Angle: This degree value determines the rotation placement for the last coned geometry.

Offset: Moves each successive clone a user defined distance from the previous along the tools axis. This value is very useful when combined with an End Angle greater than 360 degrees so that the array creates a spiral.

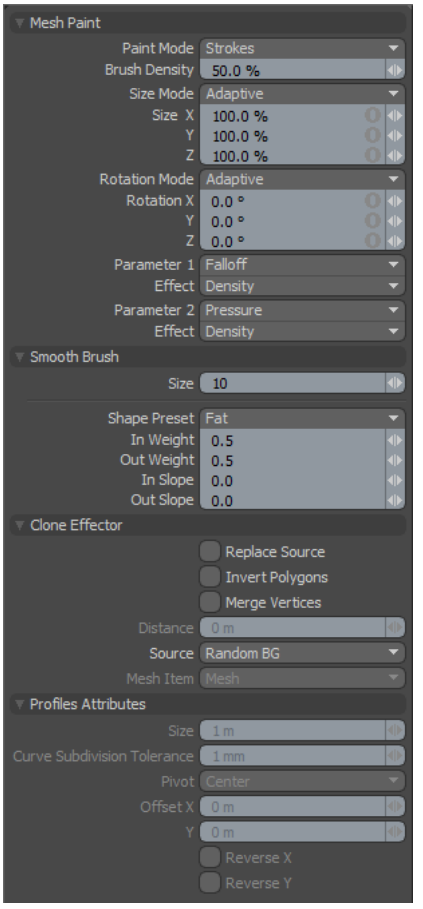
Replace Source: When this toggle is active the original mesh elements cloned by the tool will be replaced with an additional clone.

Mesh Paint

The 'Mesh Paint' tool provides an easy method for directly applying one mesh layer onto the surface of another. For instance, with Mesh Paint you can quickly drag a bolt along the hull of a ship or quickly brush many copies of boulders across a landscape.

Mesh Paint can create actual geometry or mesh *instances* and can be used to accurately place a single copy or apply hundreds of copies with a single stroke.

To use 'Mesh Paint' there must be at least two mesh items in the 'Item List'. The currently selected item is the target of the tool and the background (visible but unselected) item will be the source applied onto the target. With this set up users can simply LMB+click and drag across the selected mesh item and the background mesh will be applied along the brush stroke. In the default mode Mesh Paint will apply a single copy of the background mesh item and slide it along the surface (Slide Mode). There are many controls on the Mesh Paint tool property form that allow users to modify the tool to create brush strokes full of geometry with variable size, scale and rotation based on input from the geometry itself, from a tablet or modulated with a falloff.



The tool can be invoked from the modo standard toolbox, under the 'Duplicate' subtab. It can also be called from the menu bar under "Geometry > Duplicate > Mesh Paint". To create 'Instances' or replicas using 'Replicators', users can access an alternate Mesh Paint tool from the button popup. Hold down a LMB+click over the existing Mesh Paint button to open it. Once the tool is activated, the following attributes appear in the tool properties panel-

Mesh Paint--

Paint Mode: There are a few different modes that control how the BG mesh item(s) are applied with each stroke-

Slide- A single items is dragged over the target surface and positioned where the mouse button is released. Users can also alternately press 'Ctrl' before clicking to interactively set the scale of duplicates.

Strokes- Multiple items are *sprinkled* along the brush stroke providing users a means to clone many items across a surface very easily with randomized placements.

Drop- A single item is placed at the intersection of the mouse click position and the

target surface, holding down the LMB, users drag to interactively scale and rotate the item.

Size Mode: The 'Size' mode options determine the scale of the placed mesh duplicates onto the surface-

Uniform- Retains the original size of the item when placed. Users can adjust the scale with the X,Y and Z values as a multiplier of the original size (Item mode transforms on source are ignored).

Adaptive- Adapts the size of the item to the size brush. Users can adjust the scale with the X,Y and Z values as a multiplier of generated size (Item mode transforms on source are ignored).

Random- Randomly scales each duplicate as determined by the X,Y and Z values. 0% provides no variation, while larger values will increase the amount of variation.

Adaptive+Random- Combined properties of both Size modes allowing users to add some randomness to the adaptive scaling of cloned duplicates.

Rotation Mode: The 'Rotation' mode determines the orientation of the placed mesh duplicates onto the surface-

Uniform- Retains the original orientation of the item when placed. Users can adjust the rotation with the X,Y and Z values as an offset of the original rotation (Item mode transforms are ignored).

Surface Align- Aligns the target surfaces normal facing direction with the up (Y+) direction of the cloned item.

Random- Randomly rotates each duplicate as determined by the X,Y and Z values. 0° provides no variation, while larger values will increase the amount of variation.

Surface Align+Random- First aligns items up direction to the target surfaces normal facing direction and then randomizes it away from that as determined by the X,Y and Z values.

Surface Align+Tablet- First aligns items up direction to the target surfaces normal facing direction, users can then rotate items based on a stylus' orientation to a tablets surface (requires pan tablet with orientation support).

Surface Align+Tablet (no Pitch)- Same as above but ignores 'Pitch' rotation (Z axis).

Screen Align+Tablet- Orientation of view on-screen is first considered, users can then rotate items based on a stylus' orientation to a tablets surface (requires pan tablet with orientation support).

Screen Align+Tablet (no Pitch)- Same as above but ignores 'Pitch' rotation (Z axis).

Parameter 1/2: The 'Parameter' and 'Effect' options provide additional controls over the duplicated geometry using a number of criteria. The 'Parameter' itself is the controlling factor, while the 'Effect' determines what the Parameter controls.

Falloff- Any of the many 'Falloff' options can be used as a controlling Parameter.

Pressure- Strength of the Stylus' pressure against tablet is a controlling Parameter.

Slope- The Slope angle of the target surface is a controlling Parameter.

Altitude- The Altitude (height) of the target surface is a controlling Parameter.

1.0 Slope- Inverse of the Slope Parameter

1.0 Altitude- Inverse of the Alititude Parameter

Effect: Determines the aspect which is controlled by the above 'Parameter'.

None- Nothing is affected, disables 'Parameter' option.

Density- Parameter settings controls 'Density' (number of duplicates).

Size- Parameter settings controls 'Size' (scale of duplicates).

Rotation- Parameter settings controls orientation.

Size+Density- Parameter simultaneously affects both 'Size' and 'Density' attributes.

Smooth Brush--

The 'Smooth Brush' is one of the standard brushes used in modo for painting and sculpting covered on its own page of the Painting and Sculpting section of the documentation.

Size: The size and shape of the Smooth Brush controls the area across where duplicates are sprinkled and their initial density when placing elements with the Strokes mode. Larger brushes (in relation to the surface) would spread duplicates across a larger area. When the 'Size mode is set to 'Adaptive', the size of the brush will also affect the size of the duplicated geometry.

Shape Preset: These are quick options to select from preset brush shapes controlling the initial density of the sprinkled duplicates.

Clone Effector--

The Clone Effector determines what mesh items are grabbed when duplicates are generated. This is general purpose effector used for multiple tools, so some settings may appear out of place and should be disregarded.

Replace Source: When enabled, this option replaces the target target surface with a clone, certainly not useful in regards to Mesh Painting. *Do not enable*.

Invert Polygons: When enabled, this option Inverts the normal facing direction of the duplicated geometry.

Merge Vertices: When enabled, this option merges any overlapping vertices with identical values.

Distance: When 'Merge Vertices' is enabled, users can also define a 'Distance' value that merges any vertices that fall within that range.

Source: The 'Source' option determines the source of the geometry for duplicating onto the target surface. Active meshes are those that are selected and visible in the 'Items List', while background meshes are those that are visible but unselected.

Active Meshes- The Active Meshes duplicated onto the target surface, as duplicates are added, future duplicates will reflect the current active mesh state.

Specific Mesh- Users can specify a specific mesh layer to use as a source using the 'Mesh Item' setting..

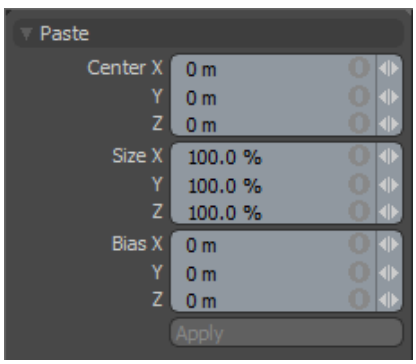
All BG- All BG meshes are considered as a single item and duplicated together.

Random BG- For each duplicate, a random BG layer is used as the source, providing users a means to randomly clone many similar but different items onto a surface producing much greater variety.

Preset Mesh- Users can select a mesh preset item from the preset browser as the source for duplicates.

Mesh Item: When the 'Specific Mesh' option is selected, users can define the particular source mash desired using this option.

Paste Tool



The Paste tool allows you to quickly stamp down copies of any geometry currently in the clipboard buffer. Activating the tool will initially create a "live" copy of the geometry in the clipboard. At this time the copied materials can be edited using standard tool handles for position (Center) and scale (Size). Additionally the Paste Tool offers options for Bias X,Y and Z so that the mesh can be shifted inside the handles.

Options for the Paste tool include Center XYZ, Size XYZ and Bias XYZ.

Center XYZ: The center triplet allows precise numeric placement of the center position of the paste tool handles. You can interactively set the center position by dragging the center handle in the 3D edit viewport, or you can enter the numeric position directly in the property form.

Size XYZ: The size settings allow you to scale the original clipboard mesh interactively with the tool handles or numerically via the tool properties form. The size setting is entered as a percentage value where 100% equals the original size of the geometry in the clipboard.

Bias XYZ: This triplet allows you to shift the mesh inside the tool handle cage. This is very useful when you want to paste a mesh onto existing geometry such as pasting rivets on to a mechanical model. By default the paste mesh is centered on the tool handles. As a result, if you were to paste the mesh with the center handle constrained to existing mesh the pasted mesh would be inset into the existing geometry. Using the Bias control you can easily shift the mesh so that it rests on the center handle rather than being centered on it.

Duplicate Commands

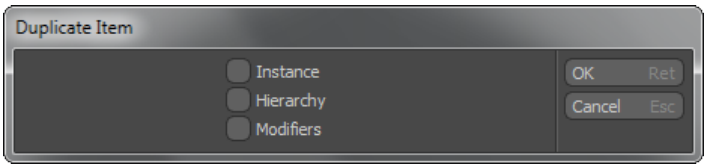
Duplicate--

The 'Duplicate' command is a simple means to create a single copy of any item in modo (or duplicates of multiple items, when more than one item is selected). The command can be found either in the 'Duplicate' subtab of the modo modeling tools panel or in the menu bar under "Item > Duplicate". If the users wishes to duplicate geometry components, such as Points or Polygons, it is better to use the 'Copy' and 'Paste' functionality. There are no dialog boxes or user options, simply invoke the command and an additional item will be created and added to the 'Item List'. modo will automatically append a number to the end of the duplicated item to avoid naming conflicts. The protocol for adding this can be modified in the modo system Preferences. To rename the item, simply RMB+click on the new item layer and choose the 'Rename ' option from the popup context menu.

Duplicate Hierarchy--

The 'Duplicate Hierarchy' command works in identical fashion to 'Duplicate', however, it will also duplicate all descending hierarchy items as well, retaining the parent/child relationships of the originating item. There are no dialog boxes or user options, simply invoke the command and the additional items will be created and added to the 'Item List'.

Duplicate Options...



With the 'Duplicate Options' command found in the menu bar under "Item > Duplicate Options...", users are presented with a dialog box offering several options for duplicating items. The 'Instance' toggle, when enabled, generates an instanced clone of the original item(s). The 'Hierarchy' toggle, when enabled, will duplicate the descending hierarchy chain of the selected item, retaining the parent/child relationships and the 'Modifiers' toggle, when enabled, will properly duplicate any Modifiers associated to the selected item(s).

Instance--

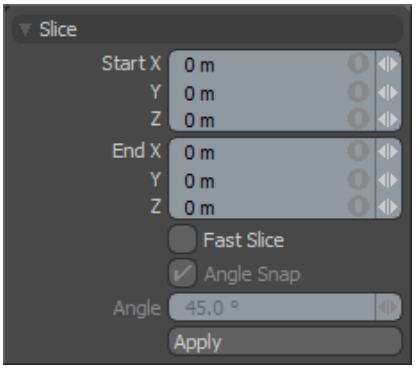
The 'Instance' command found in the menu bar under "Item > Instance" creates a single *instance* copy of the selected item(s). Instances are different than duplicated geometry in that they reference the originating item, so any component modifications made to source item are then automatically transmitted to all descendants. They are also unique in that they do not require additional scene memory and therefore can be duplicated many times over with little to no impact on the performance of the scene. Instanced items are easily recognizable in the 3D viewports by their distinct pink wireframe representations and can only be modified by way of an Item transform. If the user wishes to modify an instance individually, in can be converted to a standard mesh item by RMB+click on the instanced item name in the item list and choose the command "Change Type > Mesh" from the popup context menu.

Edit Geometry

Slicing and dicing isn't only for late night infomercial; modo also offers a variety of different tools to divide geometry in an assortment of ways. Necessary in reshaping existing surfaces, the various slicing functions provide means to add additional geometry to any area of a surface lacking in polygonal density. The Boolean and Drill options provide means to shape forms by way of adding and subtracting simple shapes into more complex ones. The subdivide commands provide a simple option to increase overall density across a mesh for a multitude of reasons, including modo's many sculpting functions.

Slice		Edit	Other
Slice	Loop Slice	Booleans	Cut/Copy/Paste
Curve Slice	Axis Slice	Solid Drill	Delete/Remove
Edge Slice	Julienne	Axis Drill	Mesh Cleanup
Pen Slice	Topology Pen	Subdivide	

Slice



Found in the 'Mesh Edit' subtab of the Modeling toolbox, the ' Slice' tool sets a slicing plane through the currently selected geometry. The slice plane is interactively set by dragging out the tool across a 3D viewport. The first click in the 3D view sets the 'Start' position for the slice. Using a LMB-drag will extend the slicing plane out to the 'End' position which can also be dragged interactively. The orientation of the plane is relative to the 'Work Plane' drawing perpendicular to its current position. The 'Start' and 'End' positions are simply X,Y, and Z positions that are set with the mouse or numerically on the Tool Properties sheet. As long as the tool is active, users can drag the knife by clicking on either of the blue end handles, or LMB+click dragging anywhere along the pink line that connects the start and end points. To create a new slice using the same basic length and axis, but set a new center position, you can MMB-Click at the new location and the slice plane will jump up to that location and create a new slice. To begin a completely new slice, use the Shift-LMB drag. This will leave the

previous slice in place and begin a reset slice action at the mouse down position.

Slice--

Start X/Y/Z: Determines the beginning position that defines the slicing plane. Users can edit the position interactively in the viewport, or accurately define values with the XYZ coordinate inputs.

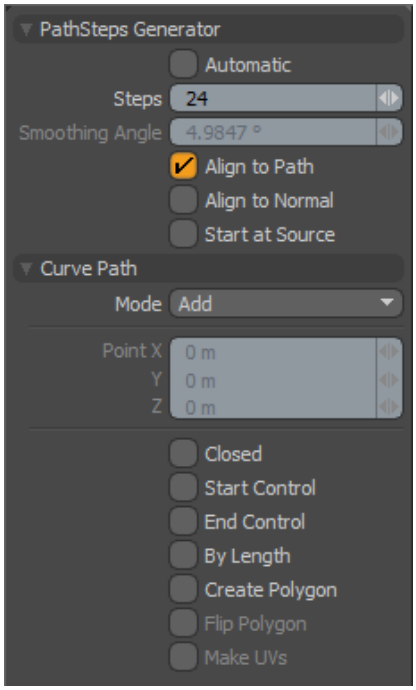
End X/Y/Z: Determines the ending position that defines the slicing plane. Users can edit the position interactively in the viewport, or accurately define values with the XYZ coordinate inputs.

Fast Slice: When this option is *enabled*, the slicing action wont occur as the plane is edited until the mouse button is released. This can help to increase performance when slicing dense meshes.

Angle Snap: When snapping is *enabled* (press the 'X' key to temporarily toggle snapping state) the slicing plane will snap (jump) to the nearest specified angle.

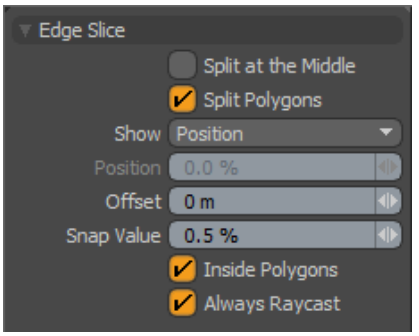
Angle: Determines the snapping angle for 'Angle Snap' in degrees of rotation.

Curve Slice



The 'Curve Slice' tool will cut the mesh with a user defined curve. There are two methods possible, one by creating a curve and placing it in the background (make it visible in the item list, but not selected (highlighted)). When the tool is activated, the background curve will become the cutting source. Second, if no background curve is visible when the tool is enabled, the user can use the mouse cursor to draw a slicing curve interactively in the 3D viewport, editing it just like a normal curve. (See the 'Curve Tool' for more on editing curves) The tool has settings to control the number of steps along the curve which will affect the smoothness of the newly created edges.

Edge Slice



With the Edge Slice tool you can click from edge to edge creating a slice between the edges as you go. After clicking down on an edge, you can drag along the edges length to set a precise cut location. You can also edit previously made edge cuts by clicking directly on the handle point that represents the desired cut point and dragging it to a new location.

Often you may wish to use the Edge Slice to create a closed cut where the last point and the first point are joined by an edge so that you have created a new "edge loop". To do this you can create a final cut on the edge that received the first cut. As you drag the new cut point towards the first cut point on the edge, the color of both points will change when they overlap so you can see that they have been merged. At this point you can deactivate the tool as you have completed the edge loop.

The options for the Edge Slice tool are:

Split at Middle: When active this option will force all cuts to be centered on each edge as it is cut.

Split Polygons: When enabled, the Split Polygons function will create a new edge between the cut points of the previous edges. With this disabled, new points will be added to the polygons, but no edges will connect them.

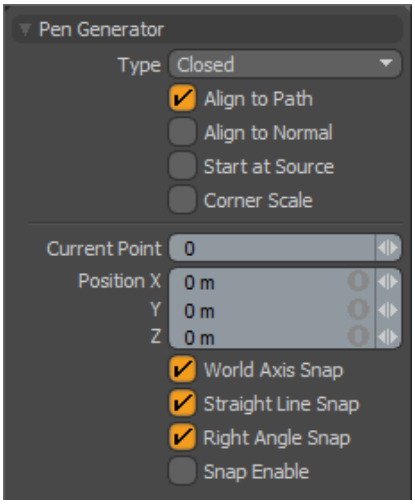
Show (Position/Distance): This pop-up provides the option of seeing the heads up data of the new cut point as a position along the edge as a percentage, or a two distance values indicating the distance to either point in the cut edge.

Count: The Count field shows the number of cuts that have been made in the current tool session. Decreasing this value will remove cut points starting with the most recently made cut.

Snap Value: This percentage value constrains the newly created vertices to specified increments. To adjust the slice vertices without constraint simply set this value to 0%.

Inside Polygons: With this toggle setting active you can click anywhere inside a polygon to create a new edge that goes into the interior of the polygon. This option makes it very convenient to carve in additional details to existing polygons.

Pen Slice



The Pen Slice tool uses the same underpinnings as the Pen tool This allows you to leverage the same structured or organic flow for creating a slice plane as you would when creating a polygon with the standard pen tool.

The tool has options for creating "Open" or "Closed" slices. In the default, Closed mode the tool connects the first and last point in the slice profile to create a closed loop.

Self snapping options include World Axis, Straight Line and Right Angles.

Loop Slice

Loop Slice is a powerful tool that lets you slice additional edge loops into your geometry. Loop Slice is accessed on the Mesh Edit tab of the modo Tools toolbar by clicking the Loop Slice button. To use, simply select two adjoining polygons to define the loop and activate the tool by clicking the button and then clicking in the 3D viewport. When active, a purple bar will appear toward the top of the viewport which the user can manipulate to modify the slicing. The small yellow triangle is the currently active slice when modifying. The far end dots can be dragged to change the size of the input bar.

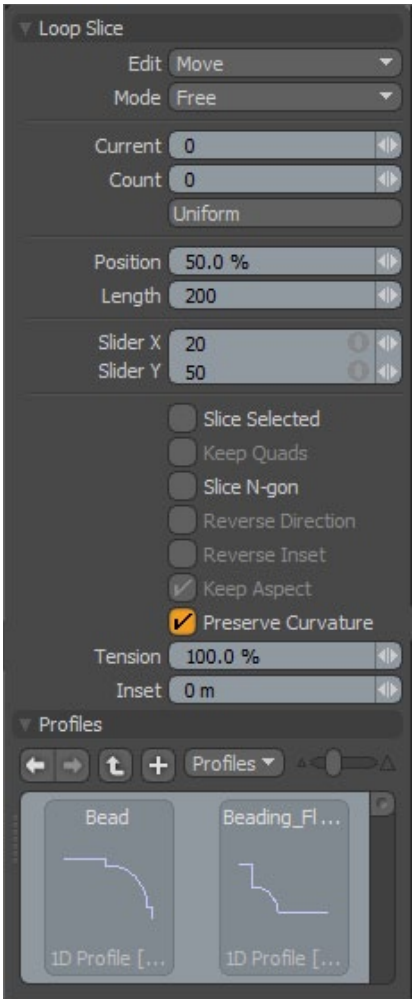
Loop Slice--

Edit: Defines what function the mouse plays, 'Move' will reposition the slice within the loop, by hauling in the viewport or dragging on the slice indicator. 'Add' will create additional slices when clicking on the indicator bar, and 'Remove' will delete them.

Mode: Defines how modo creates and positions slices, 'Free' will allow any position along the indicator bar, 'Uniform' will evenly space the slices within the loop, and 'Symmetry' will mirror the loop slices over the 50% position.

Current: Denotes which slice is the active slice for editing. Slices are number sequentially left to right in the input bar.

Count: The total number of slices added to the loop.



Uniform: Clicking this button will automatically reposition Freely spaced spans evenly across the width of the loop.

Position: Slice position expressed as a spacing percentage in relation to the loop width. 50% is the center.

Length: Controls the length in pixels of the purple slice input bar in the viewport.

Slider X/Y: Controls the position of the input bar in the viewport.

Slice Selected: Restricts the slice only to only the selected portion of the mesh, rather than using the selection to simply determine the loop to cut. This setting will respect corners as well.

Keep Quads: Force all new polys to be quadrangular inside the sliced area.

Slice N-gon: When this option is *enabed*, modo will attempt to continue a loop slice across polygons with more than four sides. For better control over how loop slice slices across the N-gon, select the loop as you wish it to be sliced and enable the 'Slice Selected' option.

Reverse Direction: If this option is enabled, the 1D profile is evaluated from top to bottom.

Reverse Inset: If this option is enabled, "Inset" will be reversed.

Keep Aspect: Automatically sets "Inset" value based on the aspect ratio of the profile.

Preserve Curvature: The new position is computed along a curve through the edges to maintain the overall curvature or surface continuity. Once enabled, the 'Tension' setting becomes available controlling the tightness of the curvature preservation.

Tension: Determines the strength of the curvature preservation. The default value of 100% calculates the curvature similar to a spline generated along the edge's length, placing the new vertex appropriately, where a value of 0% would be like disabling the Preserve Curvature option. Negative values inset the curvature and higher values further offset the vertex out from the curve's position.

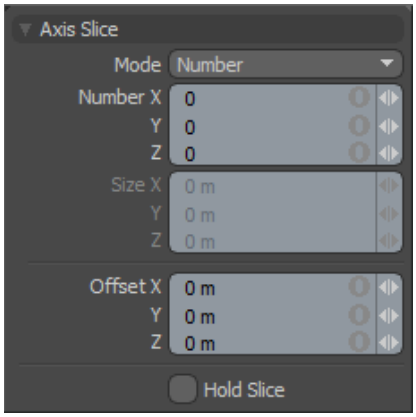
Inset: This specifies the offset amount of 'X' direction of 1D profile.

Profiles--

Mini Preset browser for viewing various profiles. Works the same as the standard 'Preset Browser'. Users can simply LMB+click on a profile to load that particular shape as a 'Loop Slice' cutter. Has an effect of what a router does when cutting in to and engraving a wood panel.

Axis Slice

The Axis Slice tool will create one or more evenly spaced slices aligned on the X, Y or Z axis planes. The number of slices and their positions are determined by the Number, Size and Offset values. The 'Offset' value can be edited interactively by dragging the mouse in the 3D viewport. This tool is useful for quickly *dicing* up a mesh with uniform slices.



Slice Mode: This pop-up option allows users to set the style of slicing. The options are 'Number' and 'Size'. Depending on which option you have selected, the fields below will be highlighted for either Number or Size.

Number X/Y/Z: This triplet field is active only when the tool is in 'Number' mode. When active, this field allows users to set a specific number of slices along within the bounding box area of the selected geometry for each axis.

Size X/Y/Z: This triplet field is active only when the tool is in 'Size' mode. When active this field allows you to set a specific distance between each slice on the three axes.

Offset X/Y/Z: This triplet field sets the center position of the slices. By default the values are 0,0,0. Changing these values will move the slices on their respective axes. Can be adjusted interactively in the viewport by hauling in the viewport.

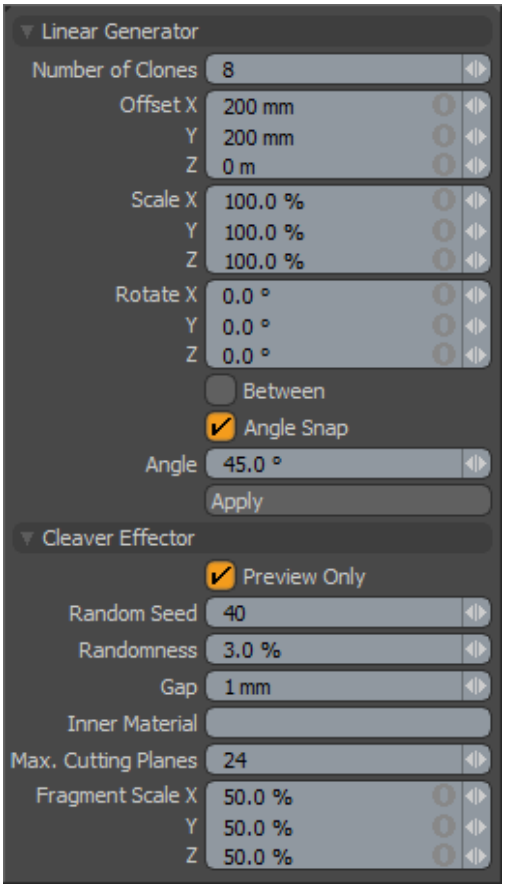
Hold Slice: The 'Hold Slice' option temporarily disabled the slicing operation. Useful when users wish to edit the data input fields without a calculation lag when dicing a dense mesh.

Julienne

The Julienne tool will create multiple evenly spaced slices that can be positioned interactively in the 3D viewport, with resulting geometry that contains inner faces. The number of slices and their positions are determined by the Number and Offset values. The 'Offset' value can be edited interactively by dragging the mouse in the 3D viewport. This tool is useful for quickly dicing up a mesh with uniform slices.

Using the Julienne Tool

The Julienne Tool is found in the modeling toolbox of the 'Model' and 'Model Quad' interface tabs, under the 'Mesh Edit' subtab. The Julienne tool is mainly meant to be used interactively, once the tool icon is clicked, LMB+click and drag in the 3D viewport. The initial click position will set the origin of the tool, this will be relative to the Work Plane, dragging determines the offset. By interactively adjusting the offset location, by hauling in the viewport (off handle dragging), users can adjust the final angle and positions of the slices.



Linear Generator--

Number of Clones: This option determines the total number of possible slices, though depending on the origin and 'Offset' positions, not all the slices will intersect with target mesh.

Offset X/Y/Z: Determines the distance from the initial origin as an 'Offset', this either determines the second slice in the arrangement of slices, or the final slice when the 'Between' option is enabled.

Size X/Y/Z: Determines the scaling of the cutting planes.

Rotate X/Y/Z: Determines the rotations of the cutting planes.

Between: When the 'Between' option is enabled, the slicing planes will be evenly spaced between the origin and the 'Offset' position, when disabled, the cutting planes will extend beyond the 'Offset'.

Angle Snap: When enabled, the tool will gently snap when the angle is close to the value specified.

Angle: Only applies when the 'Angle Snap' option is enabled, specifies the Angle where snapping occurs.

Cleaver Effector--

The 'Cleaver Effector' cuts apart geometry using a series of recursive booleans. When used with an associated generator in the tool pipe, (such as the Linear Generator) the randomness options are ignored.

Preview Only: To speed up the interactivity of the tool, the default 'Preview Only' option won't actually cut the geometry, instead offering a simplified GL preview of the final cuts. Disable this option prior to dropping the tool to actually create the cuts.

Random Seed: Determines the initial value for the random number generator, each producing its own unique results.

Randomness: Determines the strength of the jitter values applied to the cutting planes, when not used with a generator.

Gap: Determines the thickness of the cutting planes. This will be the amount of space between surfaces after the cut is complete.

Inner Material: Users can define a name here and it will be applied as a 'Material' tag to the interior cut areas of the mesh.

Max. Cutting Planes: Maximum number of total cutting planes.

Fragment Scale: These scale factors control the relative separation of the planes in the various dimensions. When not used with a generator, this option can be used to make chunks that are long and thin or flat instead of roughly cubic.

Topology Pen

When creating high resolution assets using functions such as multi-resolution sculpting, it's easy to create geometry that would be difficult to rig and animate. Too many vertices, no attention paid to surface topology. This and other things can lead to odd, unrealistic deformations. The solution for getting a freeform high resolution model into a format where it can be easily animated is to 'retopologize' the model. What this means is to take a high resolution mesh and use that as a base to generate a much lower resolution mesh. The lower resolution mesh can have more thoughtfully placed geometry that is easier to weight and rig. From there a displacement map can be

baked resulting in a final asset that is visually the same as the freeform geometry, but a lot lighter, providing better deformations and smoother interactivity when animating.

In order to facilitate the function of retopologizing models, the Topology pen was created to ease users in creating that lower resolution mesh. This combined with the Pen tool and Topology Sketch, set to a Background Point Constraint should be all that users need to completely retopologize any high resolution geometry. modo also has a dedicated 'Topo' interface tab that is dedicated to the functions of retopologizing models, making all the tools and settings available in a single convenient viewport.

Using the Tool

Retopology works by first having the high resolution mesh as a background element. This means it is visible in the Items List, but is unselected. Next making sure modo is in 'Polygons' component mode, and not 'Items' mode, create a new empty mesh layer by pressing the 'N' key. Select the new Mesh layer in the Items List (renaming it if desired). It is best to begin placing some initial geometry using either the Pen Tool; using the Quads mode is especially helpful in laying down strips of quadrangle polygons, or the Topology Sketch Tool that also lays down a series of quadrangle polygons along a drawn curve that conforms to the background surface. Once the basic contours are mapped out, users can activate the Topology Pen tool and begin to extend edges and adjust the geometry into the final surface. The Topology Pen tool, depending on settings offers a broad range of functions, some even duplicating those of other tools, providing a more convenient and fluid workflow.

By default the Topology Pen tool also activates a Background Constraint type of 'Point', ensuring all vertices created will lie directly on the surface of the background mesh. To use, select the tool by pressing the icon in the 'Topo' interface tab's toolbox. This brings up the properties panel where users can select the proper 'Mode' and options, and manipulate the geometry in the 3D viewport. Users can see the geometric element they'll be editing by the pre-highlighting, drawing a ghosted color over the element as the user moves the mouse. LMB+clicking and dragging will apply the defined action, represented by a small yellow box, the distance this box moves is displayed in the 'Offset X/Y/Z' input fields.

There are also some shortcut modifiers that work while the tool is in 'Move' mode--

LMB = Move

RMB = Move Edge Loop

MMB = Split

Shift+LMB = Duplicate

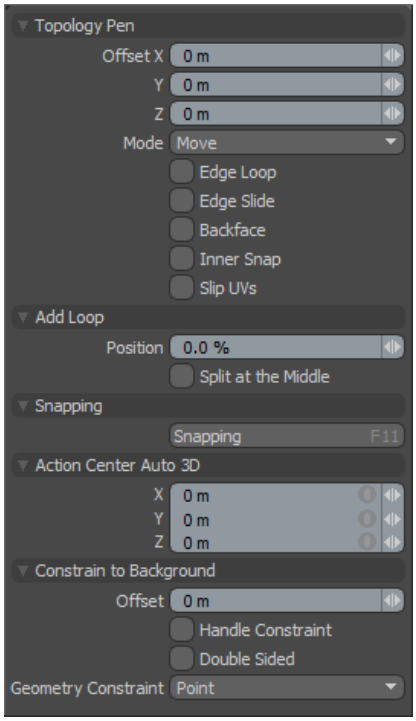
Shift+RMB = Duplicate Loop

Shift MMB = Add Loop

Ctrl+LMB = Slide

Ctrl+MMB = Remove

Smart Snapping



The Topology Pen tool utilizes a custom smart snapping feature that automatically snaps edges together, welding them into a single vertex in the process. The snapping is determined by the proximity of vertices to each other as the user drags the element, snapped edges are indicated by a purple highlight. The welding is not permanent until the mouse button is released meaning users can continue to drag and adjust the position of the element, in turn modifying the snapping, should it incorrectly grab the wrong vertex. Individual vertices can be snapped together as well by simply dragging one vertex onto another.

Topology Pen--

Offset X/Y/Z: As users drag in the 3D viewport with the mouse cursor, a small yellow box denotes the point of reference, the distance it travels is displayed in this input field. Once the location is defined by clicking the mouse, users may also manually adjust these input fields directly to manipulate the target component.

Mode: There are a variety of modes, offering functions directly related to the process of retopologizing geometry, all in a single tool-

Move- When an element is LMB+clicked and dragged the element will move but remain fixed against the background surface as it slides around. Useful in editing, vertices, edges and polygons.

Duplicate- Duplicates an edge as the user drags. LMB duplicates a single edge, With the 'Edge Loop' option *enabled*, or by using the RMB button users can duplicate an entire edge loop simultaneously. Also duplicates individual vertices making triangle polygons.

Remove- Completely removes the element from the geometry, does not modify individual outer edges, but will remove interior edges, vertices and entire polygons. RMB will remove an entire loop.

Split- LMB+clicking and dragging a vertex will create an edge that can then be dragged over and snapped to an adjoining vertex, splitting the geometry with a new edge.

Add Loop- LMB+clicking an edge will create a new edge across the entire span or loop of geometry until another edge or a triangle is encountered. Dragging while the mouse is depressed, will interactively adjust the position of the loop, unless the 'Split at the Middle' option is *enabled*.

Edge Loop: When *enabled*, any adjustments will modify an entire continuous loop of geometry until an opposing edge or triangle is encountered, instead of a single geometric element.

Edge Slide: When *enabled*, the moving of edges and vertices will be restricted to the positions of adjoining edges, in effect sliding along the edges of neighboring geometry. Pressing down 'Ctrl' while in 'Move' mode provides the same functionality.

Backface: Normally, backward facing polygons are ignored for snapping, when 'Backface' is *enabled*, users can also snap geometric components against the back side of a surface.

Innersnap: Interior vertices for a surface (those with no open edges) cannot be snapped to other interior faces, snapping only occurs between perimeter vertices. When 'Innersnap' is enabled, any vertex will snap to any other vertex regardless of its location (in the same layer).

Slip UVs: UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can *enable* the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

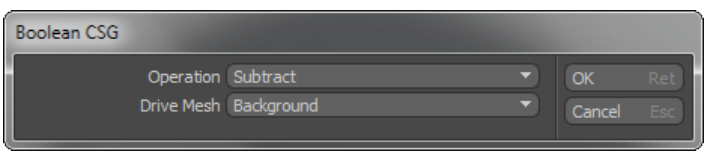
Add Loop--

Position: When the mode is set to 'Add Loop', this value determines the position along an edge where the new line of polygon edges will be added, computed as a percentage along each edge encountered. After adding the loop with a LMB+click in the 3D viewport, this input value can be directly modified further adjusting the resulting position.

Split at the Middle: When this option is *enabled*, new edge loops will be created fixed at the 50% center position.

Booleans

The Boolean command allows the user to combine multiple layers in various modes (Union, Intersect, Subtract and Add). To perform a boolean operation, you must have geometry in a foreground and a background layer. The background layer will be used as the template to act across the foreground layer.



Union: Combines the two layers together as a seamless mesh removing any overlapping geometry.

Intersect: Results in a mesh created only in areas where the two layers are overlapping. All sections of the two layers that are not overlapping are removed.

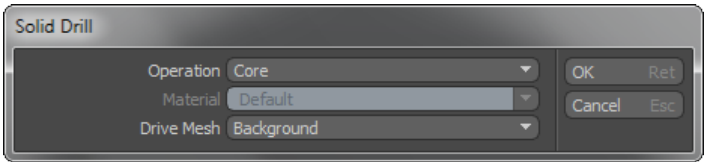
Subtract: Removes the background layer from the foreground layer, cutting away any section of the foreground that overlaps with the background layer. New faces will be created to close the area where the background layer cuts through the foreground.

Add: The geometry from both layers are combined into a single mesh. No geometry is removed, but extra vertices may be created to ensure the meshes are stitched together.

See also Drill and Solid Drill. These commands allow you to perform similarly related boolean operations as well.

Solid Drill

The Solid Drill command allows you to perform Boolean operations onto flat 2D polygons using overlapping 3D geometry. The background layer geometry is used as the Boolean drill operator on the foreground geometry.



Options include:

Core: Cuts away everything outside the 2D outline projected onto the polygons.

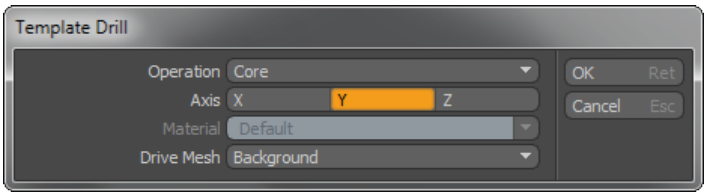
Stencil: Projects the 2D outline onto the polygons without removing either the inside or outside. Applies a default material to the new geometry within the outline.

Tunnel: Cuts a hole using the 2D outline, removes the inside.

Slice: Same as Stencil, does not affect the surface material.

Axis Drill

The Axis Drill tool allows you to perform Boolean like operations using 2D polygons projected down a specified axis. The background layer geometry is used as the Boolean operator onto the foreground geometry.



Options for Drill include:

Core: Cuts away everything outside the 2D outline projected onto the polygons.

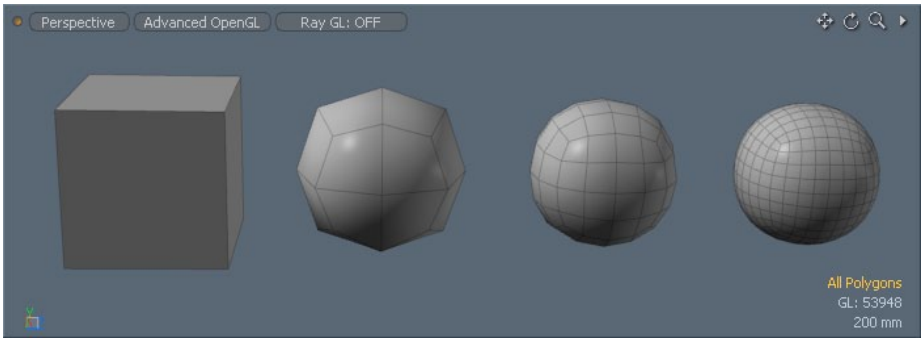
Stencil: Projects the 2D outline onto the polygons without removing either the inside or outside. Applies a default material to the new geometry within the outline.

Tunnel: Cuts a hole using the 2D outline, removes the inside.

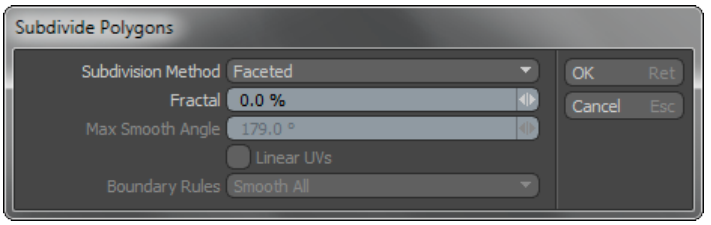
Slice: Same as Stencil, does not affect the surface material.

Axis: Defines across which axis the user wants to project the drill function.

SDS Subdivide



Subdivide Polygons is a technique used to recursively split the currently selected polygons into smaller polygon, resulting in four times the number of polygons than what was started with (each quad polygon being divided into four smaller polygons). Users can activate the command directly in the modeling toolbox under the 'Mesh Edit' sub-tab. Additionally, users can invoke the command by pressing 'D' while in 'Polygons' component mode. For additional options and controls, users can press 'Shift+D" keyboard shortcut, or invoke the menu bar command "Geometry > Polygon > Subdivide" to open the following dialog-



Subdivide Polygons--

Subdivision Method: Users can choose the type of Subdivision that is applied to the geometry-

Faceted- This option will create new polygons whose vertices are perfectly aligned with the plane of the original polygon. Visually there will be no significant change to the shape of the model (unless these polygons are 'Subdivision Surface' polygons), other than added vertices (increased resolution) for further deformation.

Smooth- This option uses the Max Smoothing Angle option to determine where there are *edges* in the model that are beyond the sharpness threshold and should not be *smoothed*. All other polygons that meet with edges below the 'Max Smooth Angle' will not only be broken into 4 additional polygons, but will the resulting vertices will be moved slightly to round out, or "smooth" the surface of the mesh. This can be illustrated quite nicely by creating a sphere with a low number of sides and segments, and then applying Subdivide using the Smooth option.

SDS Subdivide- This option is similar to the Smooth option except that all points in the mesh, including the original points, are moved towards the limit surface that was derived using the original mesh as the control points. This method of subdivision will give you a similar result to simply activating Subdivision Surfaces with a subdivision level of 1 and then freezing the geometry. SDS is a popular choice for adding uniform density to an entire mesh when working with subdivision surfaces.

Catmull-Clark- Catmull-Clark is a Subdivision Surface algorithm similar to modo's SDS Subdivide option, that provides smoothing of the selected limit surface, although with slightly different results. Developed by Pixar's Ed Catmull and Jim Clark, this new subdivision type provides some strong benefits, producing far less distortion of UV map and the ability to apply 'Edge Creasing'.

Fractal: This value allows users to add some randomization to the position of points after they are subdivided. This is sometimes useful when creating organic objects such as rough terrains or rocks.

Max Smoothing Angle: This value is used to determine when the Smooth option should

move the newly created vertices. If the angle between the original vertices is greater than the Max Smoothing angle, the new vertices will fall in the same plane as the originals. It is helpful to think of a cube in this case. Since the cube consists of right angles, if the Max Smoothing Angle is set to be less than 90 degrees, the new vertices will be in the same plane, without any rounding affect.

Linear UVs: When subdividing polygons, the additional vertices created require UV values (so as to not break the existing UVs maps!) which are smoothly interpolated using existing neighboring values. This provides the best, most pleasing results. However, using baked maps (such as a Normal Map, or a Diffuse Color Map for a game character) that require fixed polygon positions, its best to keep the UV's 'Linear', i.e. vertex values are not smoothed before interpolating the new values retaining the initial UV positions.

Boundary Rules: The boundary rules determine how to move the points on the boundaries of a mesh. Creasing will produce a sharper transition between surfaces, smoothing, quite obviously, produces smooth surface transitions. The options are-

Smooth All- All of positions on the boundary are moved by the smooth rule of Pixar Subdivision Surfaces.

Crease All- The points that have two incident edges (valence = 2) are moved by the crease rule, other positions on the boundary are smoothed by two boundary edges.

Crease Edges- The points that have two incident edges (valence = 2) are moved by the smooth rule, other positions on the boundary are smoothed by two boundary edges.

Edit Commands - Cut/Copy/Paste

Cut--

modo supports the ability to 'Cut' geometry selections from the 3D Model View, temporarily holding them in a buffer, where they can be *Pasted* into other layers or other files. To cut geometry from multiple mesh items do the following:

1) In the 'Item List' hold down the 'Ctrl' key and select multiple mesh item layers by LMB+clicking on their layer name.

2) In the 3D Model View, paint or marquee select the target geometry.

3) Use the command "Edit > Cut" or press 'Ctrl-x' on the keyboard.

This moves the geometry from the scene into the internal memory buffer. To *paste* this data into another layer or scene, simply select the target Mesh item layer in the appropriate scene and use the command "Edit > Paste" or press 'Ctrl-v'.

Copy--

modo supports the ability to 'Copy' geometry selections from the 3D Model View, temporarily holding them in a buffer, where they can be Pasted into other layers or other files. To copy geometry from multiple mesh items do the following:

1) In the 'Item List' hold down the 'Ctrl' key and select multiple mesh item layers by LMB+clicking on their layer name.

2) In the 3D Model View, paint or marquee select the target geometry.

3) Use the command "Edit > Copy" or press 'Ctrl-c' on the keyboard.

This clones the geometry from the scene into the internal memory buffer. To *paste* this data into another layer or scene, simply select the target Mesh item layer in the appropriate scene and use the command "Edit > Paste" or press 'Ctrl-v'.

Paste--

modo supports the ability to 'Paste' geometry from within its internal memory buffer into an alternate Mesh layer or scene file. In order to do so, users must first have used the 'Cut' or 'Copy' commands to place the selected components into the memory buffer. Users can use the command "Edit > Paste" of press 'Ctrl-v' placing the geometry from the memory buffer into the target mesh item layer. In the case of multiple layer selections in the Items list, the geometry will be inserted into the highest layer of the list.

Note: Any 'Cut' or 'Copy' operations on edges always 'Paste' as individual vertices (users may need to toggle viewport vertex visibility to see resulting pasted component).

Edit Commands - Delete/Remove

Delete/Remove--

modo has two functions for discarding selected elements from a model or scene, 'Delete' and 'Remove'. Both essentially do the same thing, but depending on the selection type, how they go about it will produce different results.

'Delete' is found in the menu bar under the 'Edit' menu or can be accessed by pressing the 'Del' or 'Delete' key on the keyboard. 'Delete' will universally discard any selected item, be it in the Shader Tree, Item list, Graph Editor or any of the selected components in the 3D viewport. When 'Deleting' a selection, the command eliminates the entire selection, such as with a polygon, it also removes the associated edges and vertices (though doesn't affect the components of surrounding unselected geometry).

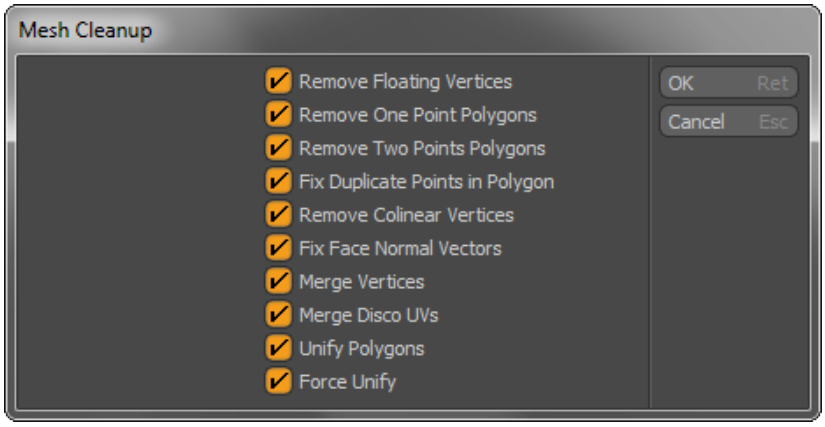
'Remove' is found in the 'Geometry' menu and it intended to discard component selections only. When applied, the 'Remove' options will act the same as 'Delete except in the case of 'Edge' type selections. 'Deleting' an edge will also remove the associated polygons that relied on the edge, where 'Remove' leaves the geometry intact, simply removing the edge itself (example below).

Edge Selection

Delete eliminates surrounding polygons

Remove keeps associated polygon

Mesh Cleanup



The 'Mesh Cleanup' function wraps a bunch of useful functionality into a single command aiding users in repairing problem geometry. Inevitably, even the most conscientious modelers will eventually create some geometry that contains collocated polygons, random floating vertices or other errors that can build up and may eventually cause rendering errors among other problems. The 'Mesh Cleanup' command is a one-stop-shop if you will of error eliminating power. Users can simply invoke the command found in the menu bar under "Geometry > Mesh Cleanup" to open the above dialogue panel. Users can then toggle the different options on or off if applicable and press 'OK' to apply the command to the currently selected geometry. The different options include--

Remove Floating Vertices: Removes single vertices that aren't connected to any polygons.

Remove One Point Polygons: Removes vertices that may be left behind when deleting polygon faces.

Remove Two Point Polygons: Removes edges of polygons where there is a visible line connecting vertices, but no faces.

Fix Duplicate Points in Polygons: Eliminates invalid duplicated vertices within a polygon, merging them into a single vertex.

Remove Colinear Vertices: Removes unnecessary extra vertices along an edge.

Fix Face Normal Vectors: Shifts the vertex order of face polygon in order to compute the correct normal vector. Resolves issues when first, second and last vertex are arranged in a straight line.

Merge Vertices: Combines vertices that have identical coordinates (collocated).

Merge Disco UVs: Each vertex can store multiple UV values, one set for each connected face. In cases where identical values are stored as multiple discontinuous values to a single vertex, this option eliminates them, merging the duplicates, optimizing the UV map.

Unify Polygons: Removes duplicated (collocated) polygons which share identical vertex map values and tags.

Force Unify: Works in conjunction with 'Unify Polygons' but removes all duplicated (collocated) polygons, ignoring vertex map values and tags.

Vertex Component Tools

Some of modo's tools are specific to the various component types - Vertices, Edges and Polygons. For example, there are three Beveling tools depending on the selection mode, each with its own settings; the 'Edge Bevel' has the rounding option where 'Polygon Bevel' does not. Some of the commands are full-on interactive tools, while others are one shot commands that produce a specific result. Some will be used regularly, and others only in rare situations, however, all of their utility could be considered indispensable.

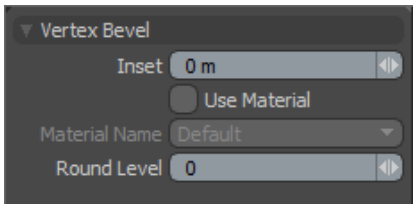
Vertices

Vertex Bevel | New Vertex | Join/Averaged | Split

Vertex Extrude | Set Position | Remove | Merge

Merge Tool | Align | Collapse

Vertex Bevel



Vertex Bevel replaces each selected vertex with 3 vertices which are moved away from the original position. The Roundness option will generate more vertices to create a rounding effect.

This tool is accessed from the Vertex Tab of the modo Tools toolbar. If you select vertices you can also select this tool from the Geometry menu > Bevel option. Unlike Bevel Polygon, only a single handle is available when beveling vertices.

The tools options include:

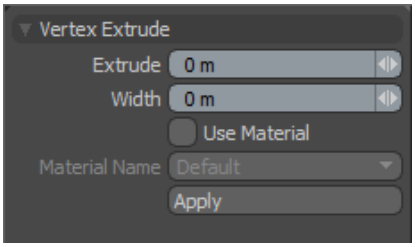
Inset: This value determines the distance of the generated vertices from the original vertex location.

Use Material: With this option on all new polygons created will automatically be assigned a Material polygon tag. The Material name is chosen from the popup just beneath this control.

Material Name: This popup is only active when the Use Material flag is enabled. Use this popup to choose any Material names that already exist in the project and this will be assigned to all polygons created by the bevel operation.

Round Level: Increasing this value will add additional vertices to the newly created edges and apply a rounding effect.

Vertex Extrude



Found on the Vertex tab of the modo Tools toolbar, Extrude Vertex shifts the selected vertex along its normal, away from its original position and creates three additional vertices and edges around the original vertex location.

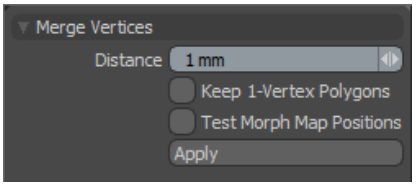
Settings include:

Extrude: This distance value determines how far out the vertex is shifted away from its original location.

Width: This distance value determines how far the newly created edges that surround the original vertex are pushed out away from that location.

Use Material: Determines what material group to assign the newly created geometry. If this is disabled, the new polygons will not have a material group and will use the base material.

Merge Tool



The Vertex Merge tool allows you to interactively adjust a distance threshold value by dragging in the 3D viewport or using the mini-slider. Any two or more vertices whose proximity to each other falls within that range will be merged together.

The options for the tool are:

Distance: This is the value that determines if vertices will be merged together. Any vertices whose distance apart is less than or equal to this value will be merged together.

Keep 1-Vertex Polygons: With this toggle enabled the merging process will leave behind single and two vertex polygons. This value is off by default so that the tool can clean up these error causing polygons. However, there are some cases where a user may want to use the resulting 1 and 2 vertex lines as a starting point for additional geometry.

New Vertex

Found on the Geometry menu under Vertex > Create, the Create New Vertex dialog lets you create vertices one at a time by specifying their location in 3D space.

To create an individual vertex do the following:

1) On the Viewport Shading popup choose Visibility Options > Show Vertices.

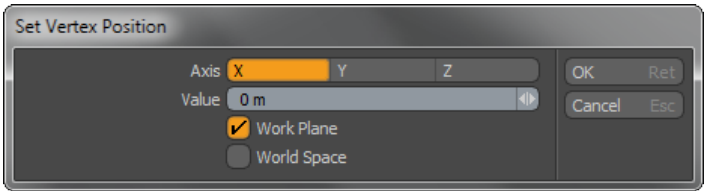
2) On the Geometry menu choose Vertex > Create.

3) On the Create New Vertex floater, specify the X, Y and Z coordinates for the vertex location.

4) Click OK.

The vertex is visible in the 3D model View.

Set Position



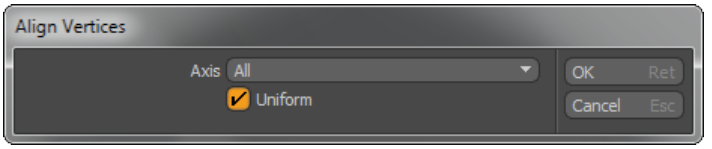
The 'Set Position' command lets users reposition an existing vertex (or many) to a specific position using numeric input, this option makes it easy to for instance center an entire loop of vertices on the zero position for a mirror operation. Set Position can be accessed from the 'Vertex' sub tab of the modo toolbox (under Commands') or users can select the command from the manu bar under "Geometry > Vertex > Set Position". Simply select a number of vertices and invoke the command.

Axis: Determines the plane with which to position the vertices.

Value: Determines the position to assign to the selected vertices.

Work Plane/World Space: Users can assign the position value based off of the 'World Space' position, or users may also select the 'Work Plane' option and position the selected vertices based on the current 'Work Plane' position.

Align

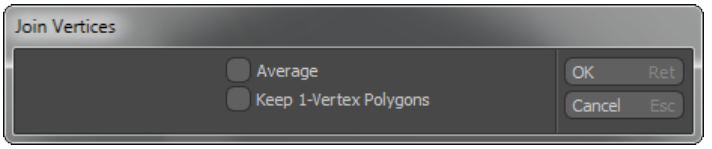


The 'Align' option allows users to align a selection of vertices into a straight row, as a line between the first and last vertex selected.

Axis: Confines the movement of the repositioned vertices to one or 'All' axes, or an axial plane (XY, YZ or ZX).

Uniform: When this option is enabled, the vertices will be evenly (uniformly) spaced along the span.

Join



Join welds the select vertices to the last vertex chosen, and moves them all to that position. Join works on multiple vertices as well.

Available on the Vertex tab of the modo Tools toolbar, the Join command launches the Join Vertices dialog.

Average: When this is turned on, the selected vertices are welded into a single vertex positioned at an averaged distance between the selected points.

Keep 1-Vertex Polygons: When this is turned off 1 vertex polygons are automatically deleted. When this is turned off, if the join operation leaves a single vertex behind, it is allowed to exist.

Join Averaged

Join average welds two vertices together at a central position between the two vertices. Join average can weld multiple vertices as well.

To Join Average multiple vertices do the following:

1) Select the vertices you wish to join together into one vertex.

2) On the modo Tools toolbar, click the Vertex tab.

3) In the Commands section click Join Averaged.

4) In the Join Vertices dialog, you should see a message telling you how many vertices were joined into one. Press OK to close this dialog.

Remove

Geometry Remove deletes the selected vertices, edges or polygons from the 3D Model View.

You can use the Remove command from the Vertex or Edges Tab of the modo Tools toolbar, or choose Geometry > Remove. You can use the Backspace key or Delete to also remove vertices, edges or polygons. Geometry Remove works on individual or selected vertices, edges or vertices.

Collapse

The Collapse command removes the selected element without destroying the integrity of the geometry. Any select polygon, edge or vertex is deleted but no hole is left behind. Instead, the mesh heals, closing any gaps by merging the neighboring elements together.

The Collapse feature can be found as follows:

Vertex: On the Vertex tab of the modo Tools toolbar > Commands > Collapse.

Edge: On the Edge tab of the modo Tools toolbar > Commands > Collapse Edges.

Polygons: On the Polygons tab of the modo Tools toolbar > Commands > Reduce > Collapse.

You can also select whatever geometry you wish to collapse and then choose Geometry > Collapse.

Split

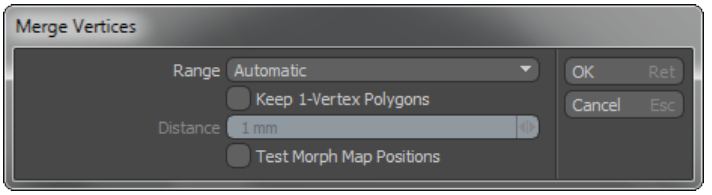
Split Vertices is the opposite of the Join Vertex Command. Instead of combining multiple vertices into a single vertex, it takes a single vertex and creates multiple vertices in the same location, based on the number of polygons that share the point.

Unlike the Join command, Split Vertices creates the multiple vertices in their original

location, it doesn't move them apart. Select any polygon or edge and move that to separate the split vertices.

Split Vertices can be found on the Vertex tab of the modo Tools toolbar, under the Commands section. Or you can select the vertices, then choose Geometry > Vertex > Split.

Merge



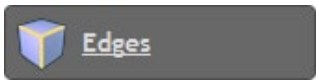
The Merge Vertices command measures the distance between vertices in the currently selected geometry to see if any fall within a specific distance threshold to one another. If any vertices are within that range, they are combined into a single vertex. The resulting vertex is positioned at the site of the most recently selected vertex of the vertices merged.

By default the merge vertices function uses a Range setting of Automatic. This means that only vertices that share the exact location on all three axes will be merged. To increase the distance allowed between two vertices to be merged, use the Fixed option. When the Fixed option is selected, the Distance field allows you to set a distance threshold. For example, if the Distance value is set to one meter, any points that fall within one meter of another point will be merged together.

The "Keep 1-Vertex Polygons" flag is active by default as well. Often when merging vertices within a certain threshold, polygons become flattened into only one vertex. In this case they become 1-vertex polygons. These single vertex polys can cause rendering anomalies, and other issues in the model file when exporting to other systems. You can simply disable this option and all resulting single vertex polys will be removed.

The Merge Vertex command is found on the Vertex tab of the modo Tools toolbar in the Tools section.

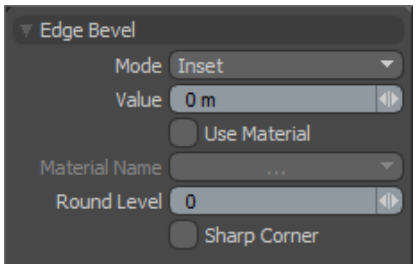
Edge Component Tools



Edges

Edge Bevel	Edge Slide	Spin Edge	Join/Averaged
Edge Extrude	Add Point	Collapse	Split
Edge Extend	Add Loop	Remove	Grow/Fill Quads

Edge Bevel



When applying Edge Bevel, selected edges widen into quad strips of a uniform, specified width. New polygons are centered over the original edge and are oriented with the average normal of the two polygons on either side. Surrounding polygons remain in their same plane, simply shrinking on their affected edges.

Edge Bevel--

Mode: Select either 'Inset' or 'Width'. 'Inset' determines the distance away form the original un-beveled edge. 'Width' determines the actual width of the new bevel (prior to the application of any rounding, if applicable).

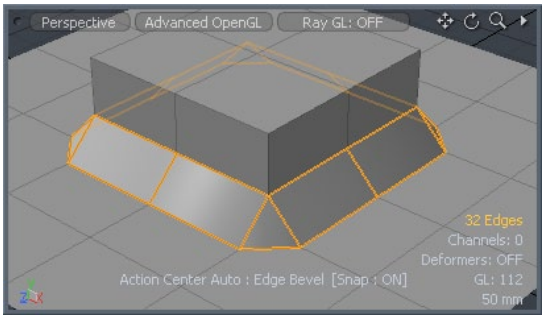
Value: Controls the amount of the Bevel as a distance value, depending on the 'Mode' option setting.

Use Material: When this option is deactivated the tool will use the neighboring material. When activated the user can choose an existing material to be applied to all newly created geometry.

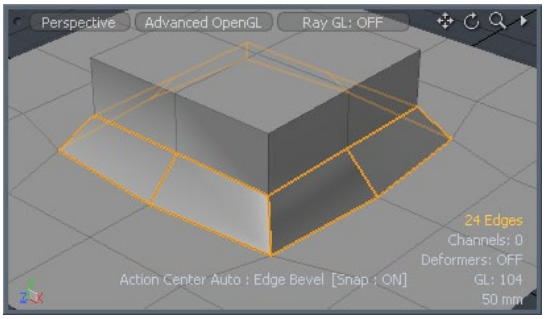
Round Level: This value allows the tool to create additional edges between the original pair generated. Each new edge will be placed in such a way as to create a "rounding" effect between the outer most edges.

Sharp Corner: For edges shared by 3 or more polygons, modo groups the polygons into pairs and creates new faces between each pair, ignoring polygons not part of a pair. At junctions between 3 or more edges, extra polygons are inserted to bridge any gaps

created by the extrusion. Extra polygons are also inserted at the bare ends of edge to keep the shape closed and planar as the edge bevels.

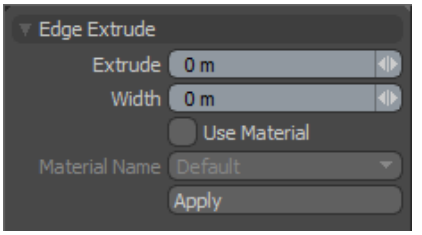


Normal Edge Bevel Corners



Edge Bevel 'Sharp Corners' enabled

Edge Extrude



With the Edge Extrude tool, edges are shifted outward along the average normal of their two neighboring polygons. New faces are added between the edge and their neighbor polygons so that their neighbors can remain in their original plane. Neighbor polygons are shrunk by the same amount and in the same manner as for edge bevel.

Parameters:

Extrude: This distance value determines how far the edge will be shifted out away from its original position.

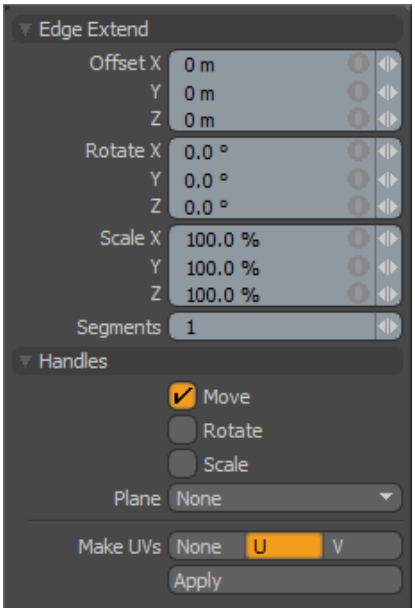
Width: This distance value determines how far the newly generated edges will be shifted away from the original edge location.

Use Material: Turn this on to apply a specific material group to the polygons created with the edge extrusion.

Material Name: Specifies the material group for the Use Material command.

Edge Extend

This tool provides a simple method for creating geometry from selected edges by extending them with options for moving, rotating and scaling the newly created edges.



The edge extend tool includes a customizable tool handle. By default you are presented with move and scale handles but these can be modified in tool properties to include rotation handles and also planar move and scale handles. This allows you to adapt the tool easilyfor a particular modeling operation and alsomeans you can perform more complex edge editing without dropping the tool.

Options for the tool handles include Move, Rotate and Scale widgets which can be displayed or hidden in any combination from the properties form. The Plane option allows you to choose to display planar widgets for Move or Scale to facilitate quick edits of these transforms on 2 axes simultaneously.

To quickly add a series of edge extends you can hold the shift key down and each time you click a then new edge will be created. This is a very quick method for creating profiles by holding shift and click-dragging each new span into place.

Edge Slide

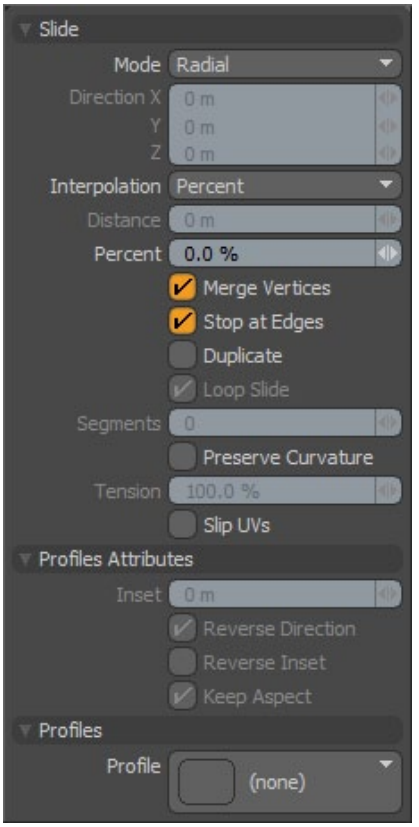
The Slide tool moves selected mesh elements (vertices, edges and polygons) along their connected edges. There are two main modes to the tool (Radial and Linear) which can be used depending on the style of selection and intended result.

Mode: 'Radial Mode' is most useful for sliding entire edge loops such as widening or narrowing a selected edge loop around an eye. When this mode is active the direction fields are disabled as the only relevant input is Distance/Percentage which will indicate how far to slide the selected loop between the next and previous loops.

'Linear Mode' is provides a method to slide elements along either of two intersecting edges and is most useful when sliding vertices or polygon selections. When this mode is selected, the 'Direction X/Y/Z' triplet is activated. When interacting with the tool in a 3D viewport the direction of the handle will determine the edge to slide along.

Interpolation: This pop-up option allows you to choose between Distance and Percent methods for sliding the geometry.

Distance: The Distance option allows the tools numeric entry to be a specific distance based on the current until system so that geometry can be moved a specific number of feet, inches, meters etc.



Percentage: The Percentage option slides the selected geometry based on a percent from it's initial position (0%) to the neighboring edge or edge loop (-100 or 100%).

Merge Vertices: When this option is active, vertices will be merged if the slide tool results in vertices sharing the exact same coordinates. For example, sliding an edge 100% with this option on will result in that edge being merged into the neighbor. This can be particularly convenient when combined with the Stop at Edges option.

Stop at Edges: With this setting active the selected the tool will stop the slide when the selected mesh elements reach the surrounding edges.

Duplicate: When active the Duplicate option will create a new edge rather than simply sliding the selected edge. With this option active it is simple to create new edges or edge loops and place them along the existing curve.

Loop Slide: When sliding an edge selection, 'Loop Slide' constrains the movement along the existing polygonal bounds.

Segments: This specifies the number of segment for profiles. If value is 0 and a 1D profile is selected, the edge slide tool automatically decides the best adaptive segments for the profile. Duplicate must be enabled for Profiles to work.

Preserve Curvature: The new position is computed along a curve through the edges to maintain the overall curvature or surface continuity. Once enabled, the 'Tension' setting becomes available controlling the tightness of the curvature preservation.

Tension: Determines the strength of the curvature preservation. The default value of 100% calculates the curvature similar to a spline generated along the edge's length, placing the new vertex appropriately, where a value of 0% would be like disabling the Preserve Curvature option. Negative values inset the curvature and higher values further offset the vertex out from the curve's position.

Slip UVs: UV values are generally fixed to specific vertices, subsequently further edits to the geometry may warp, deform or otherwise distort the the UV values in

undesirable ways requiring users to adjust the map or redo it altogether. To avoid this undesirable result, users can *enable* the 'Slip UVs' function so as to not disturb any existing UV mapping applied to the geometry.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

'Slip UVs' function disabled, note texture warping.

'Slip UVs' function enabled, texture remains even.

Profile Attributes--

Inset: This specifies the offset amount of X direction of 1D profile.

Reverse Direction: If this option is enabled, the 1D profile is evaluated from top to bottom.

Reverse Inset: If this option is enabled, "Inset" will be reversed.

Keep Aspect: Automatically sets "Inset" value based on the aspect ratio of the profile.

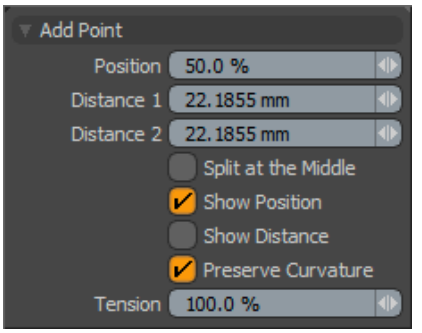
Profiles--

Mini Preset browser for viewing various profiles. Works the same as the standard preset browser.

Add Point

The 'Add Point' tool provides users a means to easily add a vertex point along an edge, or along a curve wherever the user clicks the LMB. Found in the 'Edge' subtab of the modeling toolbox, the following option appear in the tool properties panel when the tool

is activated-



Add Point--

Position: This value determines the position of the new point across the entire edge length, displayed as a percentage, with 50% being the center.

Distance 1/2: The slice position displayed as a numeric distance value from the start (1) to the end (2) of the edge.

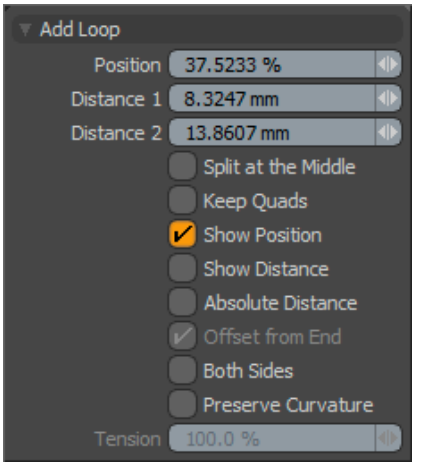
Split at the Middle: When enabled, modo will automatically slice at the middle position regardless of where the edge was clicked.

Show Distance: When enabled, displays the distances between the edge positions and the new position as a numeric value.

Preserve Curvature: When *enabled*, the 'Preserve Curvature' option attempts to maintain the *flow* of surrounding geometry. The new position is computed along a curve through the surrounding edges to maintain the overall curvature of the surface. Once enabled, the 'Tension' setting becomes available.

Tension: Determines the strength of the curvature preservation. The default value of 100% calculates the curvature similar to a spline generated along the edge's length, placing the new vertex appropriately, where a value of 0% would be like disabling the Preserve Curvature option. Negative values inset the curvature and higher values further offset the vertex out from the curve's position.

Add Loop



The add loop tool is the interactive version of Loop Slice tool. Users can easily add an edge loop at the edge where clicked and then see the pre highlighted polygon loop to slice at the mouse position. Found in the 'Edge' subtab of the modeling toolbox, the following options are possible options are available when the tool is activated-

Add Loop--

Position: This value determines the slice position across the entire edge length displayed as a

percentage, with 50% being the center.

Distance 1/2: The slice position displayed as a numeric distance value from the start (1) to the end (2) of the edge.

Split at the Middle: When *enabled*, modo will automatically slice at the middle position regardless of where the edge was clicked.

Keep Quads: Forces the 'Add Loop' tool to produce Quad polygons as possible.

Show Position: When enabled, displays a text overlay of the new position as a percentage.

Show Distance: When enabled, displays the distances between the edge positions and the new position as a numeric value.

Absolute Distance: When enabled, the offset of the new loop is defined by an absolute distance instead of as a percentage.

Offset from End: When enabled, allows the offset of the loop from the opposite end of edge. This option is only available when 'Absolute Distance' is enabled.

Both Sides: When enabled, simultaneously slices loop from both sides of the existing loop providing two edge loops.

Preserve Curvature: When *enabled*, the 'Preserve Curvature' option attempts to maintain the flow of surrounding geometry. The new position is computed along a curve through the surrounding edges to maintain the overall curvature of the surface. Once enabled, the 'Tension' setting becomes available.

Tension: Determines the strength of the curvature preservation. The default value of 100% calculates the curvature similar to a spline generated along the edge's length, placing the new vertex appropriately, where a value of 0% would be like disabling the Preserve Curvature option. Negative values inset the curvature and higher values further offset the vertex out from the curve's position.

Spin Edge

The Spin Edge tool turns the current edge or edges to reorganize the way they separate the two adjoining polygons. For example, if you select an edge, this command will spin it so that it attaches to different vertices while leaving the two bisected polygons in place. It will change the flow of your polygons while maintaining the surrounding mesh.

Collapse

The Collapse command removes the selected element without destroying the integrity of the geometry. Any select polygon, edge or vertex is deleted but no hole is left behind. Instead, the mesh heals, closing any gaps by merging the neighboring elements together.

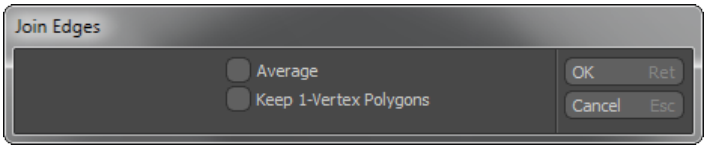
Remove

The Remove Edge command can be used to delete edges from the mesh geometry. This differs slightly from the Geometry > Remove tool in that it doesn't remove attached vertices, it only removes the selected edges. The vertices remain behind.

To remove the edges and vertices, select the edges and then on the menu bar choose Geometry > Remove.

To remove ONLY the edges, select the edges, then on the modo Tools toolbar, click the Edge tab, then Remove Edges on the Commands section. The edges will be removed, the vertices will remain.

Join



Join edge allows you to weld two edges together. Simply select two edges and run the tool. The first selected edge will move to the position of the second selected edge and the vertices will be joined.

Join Averaged

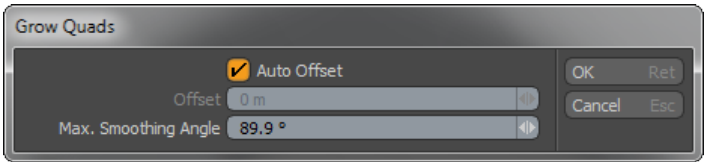
Join Edge Average will weld two edges together and the resulting edge will have a position that is an average of the original edge positions. Simply select two edges and run the tool.

Split

Split edge works like the Split vertex command. Instead of splitting vertices around a selected vertex, it separates mesh polygons along selected edges. In polygon selection mode, the command splits boundary edges of the selected polygons. This is useful when separating polygons which are selected by the close loop selection command. This is also useful for opening up a "hinge" section on a mesh.

Grow Quads

The 'Grow Quads' command will extend a row of quad polygon along a selected edge border.



Auto Offset: Will attempt to match preceding rows size, disable to set a specific offset value.

Offset: Specifies distance of offset.

Max. Smoothing Angle: Specifies maximum smoothing for newly created polygons across angular surfaces.

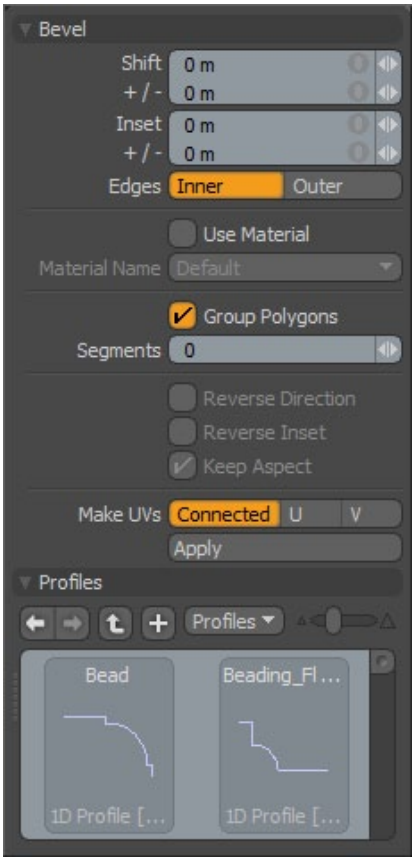
Fill Quads

The 'Fill Quads' command will attempt to fill any open holes within a mesh with quad polygons. Users can control the application of the tool with an edge loop selection.

Polygon Component Tools

Polygons			
Polygon Bevel	Thicken	Triple	Align
Polygon Extrude	Spikey	Split	Merge
Smooth Shift	Inset	Spin Quad	Collapse
Shift	Reduction	Convert to Quads	Remove
		Flip	Unify

Polygon Bevel



The Bevel tool performs a shift and inset on the selected polygons. The shift pushes the polygons away from their starting positions, and the inset scales the polygons. New polygons are created to bridge the gaps that would be left by moving the original polygons from their starting positions.

Shift/Inset: Both inset and shift can be set to positive or negative values. When the values are positive inset and shift work as described above. When the values are negative, shift will move the original polygons back behind their original location and inset will scale the polygons to be larger than their original size.

Each of the Shift and Inset values can be edited by hauling in the 3D viewport, dragging tool handles or by using the HUD sliders (HUD sliders require advanced handle viewing to be active). You can also edit these values numerically on the tool properties sheet. On the properties sheet you will find the following for the bevel tool.

+/- : Under Inset and Shift this field allows you to set a random value range for the Inset or Shift value. If you have a Shift value of 1 meter and a +/1 value of .5 meters, the resulting bevel(s) will be between .5 and 1.5 meters. This can be useful when beveling many polygons simultaneously.

Edges (Inner/Outer): This toggle option determines whether the resulting polygons are created in such a way as to allow them to fall inside the original polygons boundaries, or so that they are outside the original polygon. The default, inner setting is the preferred choice for most bevel scenarios. The outer option is very useful when beveling polygons that are not attached to other polygons. The classic example of using the outer option is for beveling text. In this case you may want the original text polygons to maintain their scale, but would like to bevel an edge "outside" of the original polygons. Activating outer will do just that.

Use Material: This option will also enable a text field. When this option is active, any new polygons created by the bevel tool will have the material name from the text field assigned to them.

Group Polygons: This simple toggle provides a quick method for beveling contiguous polygon selections essentially treating the entire selection as a single polygon. This is incredibly useful for adding fine fillets around selections by creating a series of bevels.

Segments: This specifies the number of edge segment along the bevels shift direction. If it is 0 and a 1D profile is selected, the bevel tool automatically decides the best adaptive segments for the profile.

Reverse Inset: If this option is enabled, "Inset" will be reversed.

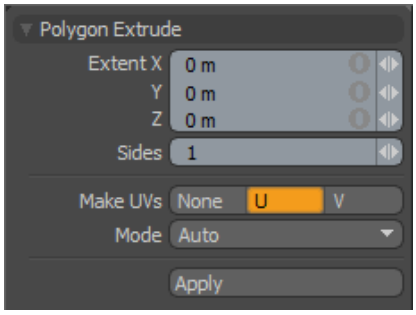
Keep Aspect: When enabled, 'Keep Aspect' automatically sets "Inset" value based on the aspect ratio of the profile.

Make UVs: This pop-up option allows you to choose three UV creation methods. The default "none" option will simply create the UVs for the new faces inside the existing UV coordinates attempting to bevel the UV data along with the mesh. The U and V options will create a completely new set of UV data in the negative UV coordinates with the newly created polygons fanned out in the user selected axis (U or V). These options can be tremendously useful when beveling the geometry results in overlapping UV data as it will create a more manageable starting point for the UV set.

Profiles--

Mini Preset browser for viewing various profiles. Works the same as the standard preset browser.

Polygon Extrude



Polygon Extrusion: Found on the Mesh Edit tab of the modo Tools toolbar, polygon extrusion repositions the selected polygon in space, based on the three tool handles. Additional polygon surfaces are created between the new location of the polygon and the original location of the selected faces.

Use the Sides field to control the number of loops created with the polygon extrusion.

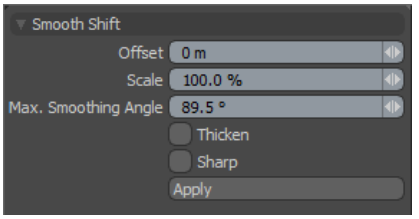
Use the Extents fields to precisely reposition the extruded polygon.

The Make UVs option allows you to control if the tool creates UVs for the new polygons in the existing space or fans them out on the U or V axis.

The Mode setting is set to Auto by default. In this mode the tool will evaluate the

geometry to determine if it should remove the original selected polygon after the extrusion. In that case the tool automatically chose the "Shift" mode. In the case of a mesh where that polygon would be left "inside", the tool will remove it. If the selected polygon(s) are single sided and will not be encased by existing geometry the tool will leave the polygons in question in place. In this case the tool automatically set itself to "Thick". You can override this behavior by simply setting the tool into one of these modes directly.

Smooth Shift

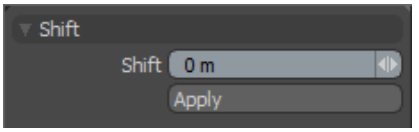


Smooth Shift is a Deformation tool that smooths out sharp edges in polygonal geometry based on a smoothing angle. When the angle between polygons is greater than the assigned smoothing angle, the edge is replaced with two edges which are scaled and offset to round the sharp corners of the mesh. Two handles in the viewport individually control the offset and scale.

Smooth Shift is found on the Polygon tab of the modo Tools toolbar, under the Tools group, or you can access this tool by choosing the Geometry menu > Polygon > Smooth Shift command.

Smooth Shift can be used interactively in the 3D Viewport, or by pressing the Apply button in the Tool Properties form.

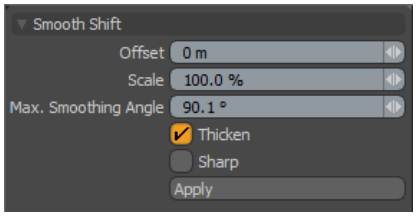
Shift



The polygon Shift tool performs an extrusion on selected polygons along their individual normal axes. This tool is similar to the Extrude tool found on the Mesh Edit tab of the modo Tools toolbar, except that it allows you to extrude multiple polygons in different directions in a single operation.

The polygon Shift tool can be accessed from the Polygon Tab of the modo Tools toolbar under the Tools group > Shift.

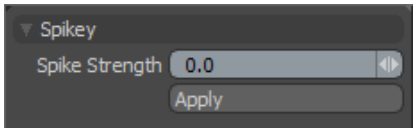
Thicken



Thicken: The Thicken tool allows you to quickly add dimension to flat geometry. This functionality is often referred to as Shell. The Thicken tool can be used to add thickness to the outside of the volume making the object larger or to the inside of the volume creating an inner surface.

Thicken has two controls; Offset and Scale. When you invoke the thicken tool it creates a new set of polygons at the same location as the originals. Setting a positive Offset value moves the selected or original polygons in the direction of their normals. At the same time a set of polygons are created in the exact original location but facing the opposite direction. These two sets of polys are connected with a row of polygons that are roughly perpendicular. Using a negative Offset value will push the original polygons opposite the direction of their original normals and flip them at the same time to avoid the object turning inside out.

Spikey



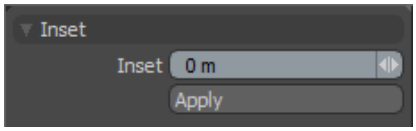
Spikey is a polygon tool that takes selected polygons, adds a vertex to the center of the polygon and then repositions that vertex away from the original polygon based on the spike strength setting, to create a spike. Spikes are created based on polygon face normals for each selected polygon.

Spikey can be accessed by clicking the Polygon tab of the modo Tools toolbar.

Spikey can be fired interactively in the viewport, or by using the Apply button on the Tool Properties form.

The Spikey tool is not available form the menu bar, only from the modo Tools toolbar.

Inset

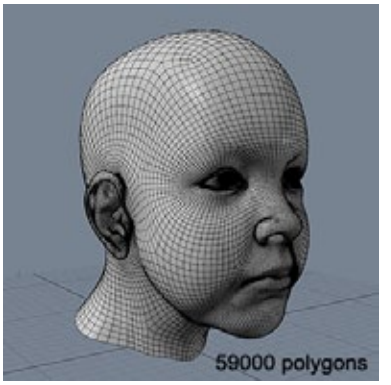


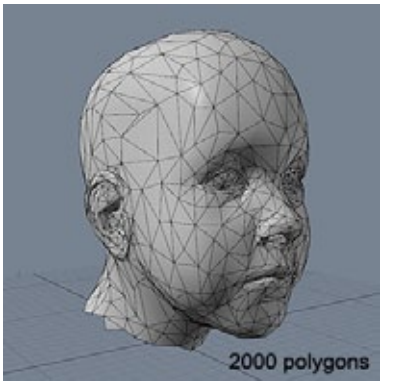
Inset is a polygon creation tool that takes selected polygons and subdivides them to create new geometry within the original selection. The new polygons are scaled duplicates of the original. For example, selecting a single polygon on the side of a cube, and then firing inset, will create a smaller quad polygon framed within the original. Using negative values for inset will force the duplicated polygon to scale larger than the original.

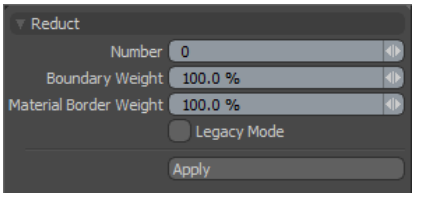
Inset can be found on the Polygon tab of the modo Tools toolbar under the Tools group > Inset.

The Inset value can be specified using the Inset numeric field, then use the Apply button to fire the command with those values.

Polygon Reduce







The Reduction tool works by iteratively removing edges from the mesh while trying to maintain as much of the original contouring as possible. There are three options to the tool: Number, Boundary Weight and Material Border Weight.

Number: When the tool is first activated this field will contain the total number of polygons in the currently visible mesh layer. Clicking and dragging will adjust this number in the tools property and in the mesh. Of course, you can also simply type in the exact number of polygons into the property field.

Boundary Weight: Determines how much emphasis should be given to preserving the details around boundary areas of the mesh.

Material Border Weight: Determines how much emphasis should be given to using polygons to preserve the integrity and detail of the mesh along edges which define a boundary between materials groups.

Triple

The Triple command will subdivide any selected polygon with more than 3 vertices so that it consists only of triangle polygons. This can be very useful for preparing data to be saved in a format that does not support polygons with more than 3 vertices per polygon. It can also be useful for quickly breaking an n-gon down into triangles for similar export issues.

Split

Select two vertices on a polygon to force an edge to be drawn between them. Vertices can also be selected in succession across polygons.

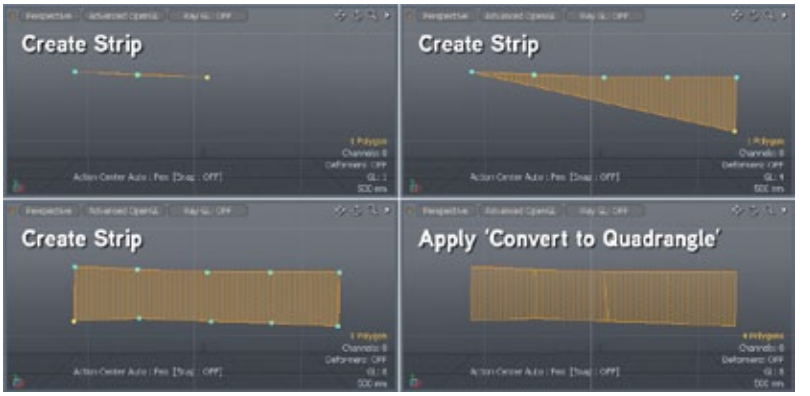
Spin Quads

Spin Quads will change where your edges are attached within the geometry. For example, if you select two adjacent polys, this command will spin them so that they attach to different points while leaving them in place. It will change the flow of your polygons while maintaining the surrounding mesh. Spin Edges does the same thing on an edge level, however you can select one edge and spin it so that it bisects two polygons differently. Spin quads only works with two polygons with the same edge number.

These tools are found on the Edge and Polygon tabs of the modo Tools toolbar, under the Commands groups.

Convert to Quadrangles

Found in the menu bar under "Geometry > Polygon Quadruple" or within the 'Topo' tab toolbox, the 'Convert to Quadrangles' command will take a series of vertices from an N-gon and create a row of regular four sided (quadrangle) polygons. Useful in creating quad strips, especially when retopologizing a model. Using the Pen Tool, lay down two opposing rows of vertices, then invoke the 'Convert to Quadrangles' command and modo will automatically add edges spanning between the opposing vertices making quad polygons.



Flip

The Flip tool will reverse the direction of selected polygons. Polygons are typically single sided, and as such are only visible from one direction. The visible side is determined by the direction of the "normal". The direction of a polygon is initially determined by the order in which it's vertices were created, or selected, to make the polygon. The Flip tool will effectively reorder those vertices so that the polygon faces the opposite direction. For linear polygons, such as curves, the Flip tool reverses the order to force the curve to run the opposite direction.

Align

The Align Polygons tool attempts to automatically make all polygons face the same direction. Use this when you have mesh geometry with polygons that have face normals facing both toward and away from the view. Align Polygons uses the 1st polygon you select as the model, it will attempt to match the face normal direction of that polygon.

Merge

The Merge Polygons option combines selected polygons into a single polygon of n

number of sides. Essentially it removes all interior edges, so the multiple polygons can be treated as a single polygon.

This command can be found on the Polygon tab of the modo Tools toolbar, in the Commands sections on the Reduce pulldown. Expand the Commands section to see Reduce, then press the dropdown arrow and choose Merge.

Or on the menubar choose Geometry > Polygon > Merge.

More complex selections will merge as many polygons together as necessary to remove the target edges. If all the edges that use a vertex are deleted, then the vertex is deleted as well. It may be possible for you to specify selections that cannot all be consistently deleted. In that case the operation does the best that it can without leaving "spikes", which are edges entirely internal to a single polygon.

Collapse

The Collapse command removes the selected element without destroying the integrity of the geometry. Any select polygon, edge or vertex is deleted but no hole is left behind. Instead, the mesh heals, closing any gaps by merging the neighboring elements together.

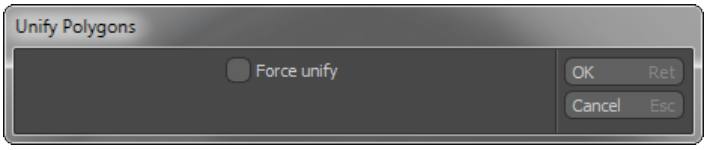
Polygons: On the Polygons tab of the modo Tools toolbar > Commands > Reduce > Collapse.

You can also select whatever geometry you wish to collapse and then choose Geometry > Collapse.

Remove

The Remove Polygon command deletes the selected polygons, completley removing them from the selected mesh item layer. The command is found within the modeling tabs under the 'Polygon' toolbox, or in the menu bar under "Geometry > Remove". Users may also simply press the 'Backspace' key on the keyboard.

Unify



Unify Polygons lets you merge two polygons into a single polygon when you have two

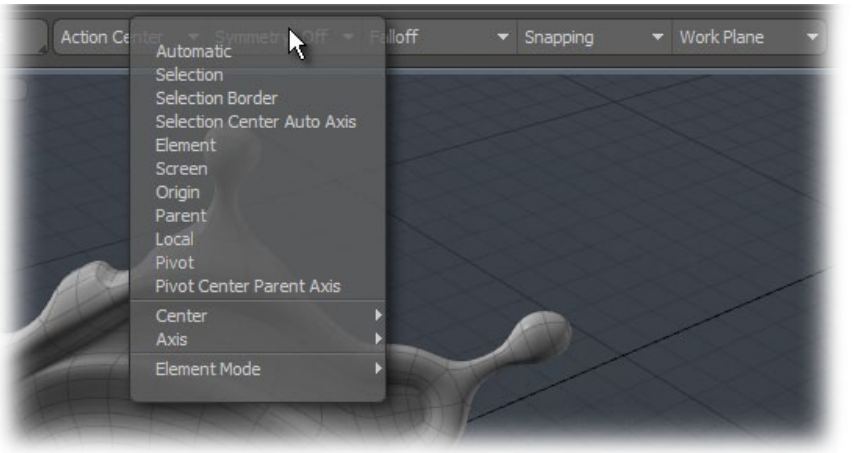
polygons that are in exactly the same space and the polygons share the same vertices.

To access Unify Polygons click the Polygon tab of the modo Tools toolbar, then click the dropdown arrow next to Reduce and choose Unify. The selected polygons in the viewport will be unified.

Tool Modifiers

Action Centers

Action Centers are a definable origin position for modeling and transform actions in modo. While other 3D software programs use the fixed mesh geometry *pivot point* as the center for most actions, modo separates the Action Center into an independent dynamically definable component. This provides users a means to easily change the positions from where the action of any tool originates from. Action Centers can be activated directly in the interface in the Action Center drop down menu or within the menu bar under "Edit > Action Center". If the user wishes, the Action Center can be further broken into its 'Center' and 'Axis' components for even greater control.



Generally user won't see anything happen within the 3D viewport when selecting an Action Center until a tool is made active, as actions centers themselves are represented by the positions of an active tools handle. When users select a new Action Center, the position and orientation of the handle will update to reflect the new axis and center.

Action Center Types

None

When no action center is selected, modo defaults to none, which is essentially the same as Automatic, except there is no ability to explicitly define the axis.

Automatic

When a tool is activated, the 'Automatic' action center positions the tools handle at the center of the current selection (or on all active geometry in the case of no selection) with the axis aligned to the world or Work Plane axes. Users can explicitly select the axis in the properties panel and interactively modify it by adjusting the orientation handle, the thin blue line with the yellow dot. The 'Automatic' Action Center is not fixed like others Action Centers are. Users can define a new center by LMB+clicking away form the tools handle, setting the positions at the intersection of where the pointer clicked in the viewport and the Work Plane.

Selection

'Selection' bases the action center on the average vertex position of the selection in the 3D Model View effectively centering the action center to whatever is selected. Setting the Action Center to Selection will change the tool's handles to have the center and axis aligned to the current selection.

Selection Border

This action center preset combines the Selection Border center with the Selection based axis. This combination centers all tool handles around the border, or boundary, of the selected elements with the orientation of the tool derived from the average normals of all selected elements. The border of selection is defined by the "open edge" area where the selection stops. This is extremely useful for posing character type operations. Selecting a loop of polygons at say, an elbow, using the Select > Close loop command to select the rest of the arm, and activating 'Selection Border' will position a rotate transforms handles aligned to the initial loop selection making it easy to 'bend' the arm.

Selection Center Auto Axis

Selection Center Auto Axis keeps the center of the tool handles centered on the geometry, but realigns the tool handles to the major axes. Selection Center Auto Axis provides an Axis field in the Tool Properties Action Axis Auto section, so you can easily change from one axis to another by clicking the appropriate button.

Element

Setting the Action Center to Element allows you to click on any element in the 3D model to set the action center. If you click on one polygon, modo realigns the tool's handles to its normal. If you click on an edge, or a vertex, modo also aligns the tool handle to the normal of those elements. If you want to Stretch an entire model along the normal of any particular polygon , for example, you simply activate the stretch tool, click on the polygon and now you can scale along that axis.

Element Action Center does not require the element that is selected to be the element that is used for the Action Center. Once you have a polygon selected, you can choose Element Action Center, then click on a different polygon. That polygon will be the reference Action Center for transforming the original selected polygon.

Screen

Screen Action Center allows the action center and axis to be based on the frame of the viewport, or Screen space. When Screen Center and Axis is chosen for any transform, the tool handles will realign themselves to the screen, and the transform becomes in relation to the "picture plane" rather than to any world or local coordinate system.

Origin

The 'Origin' action center uses the world coordinates 0,0,0 for the center of any transform, regardless of item or element position. While less useful with the Move transform, this is handy for Rotate or Stretch if you are building something off the origin to begin with. You can easily rotate around the center of the 3D View using this setting. It also makes it easy to accurately set functions like mirror that require they happen across the origins planar axes.

Parent

The 'Parent' action center uses the pivot position of the parented item (as opposed to the center position). Useful for transforming items in a hierarchy when you wish to keep items in proper relationship to their parents.

Local

The 'Local' action center uses the center of individual element clusters for the operation center, effectively like having a separate axis and center for each selection group. This lets you select multiple elements and have them each rotate around their own local axis. When using the Move tool, the handles are axis-aligned to the last selected polygon. This is particularly useful when working with edges as you can use the stretch tool to scale several edges simultaneously along each ones individual local axes.

Pivot

The 'Pivot' action center uses the pivot position of the item as the center of any transform. This is especially useful for animated items that might not currently near the origin center.

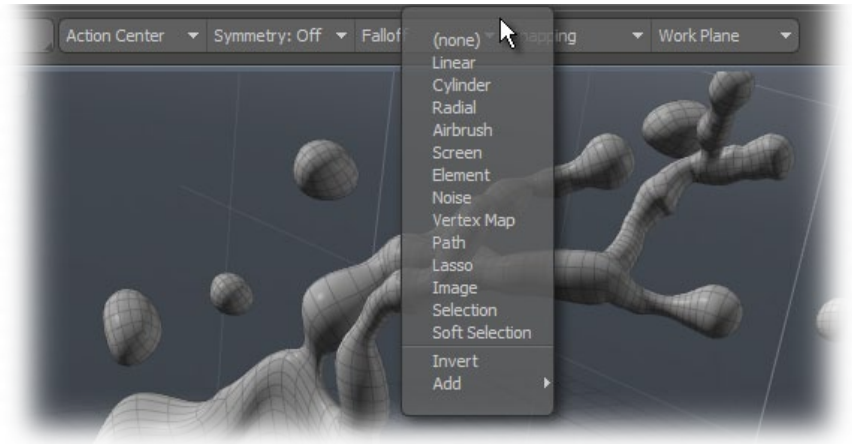
Pivot Center Parent Axis

The 'Pivot Center/Parent Axis' action center uses the current items pivot position as the center, but adjusts the axis to match the parent item.

Falloffs

In modo, the affect any tool has upon a selected region of geometry is rigid, meaning applications of a tool happen at a constant rate across the selected range. modo provides a way to attenuate the affect a tool has over a selection as a *Falloff*, a gradual decrease in intensity across a defined range. Falloffs are universal in application, once activated they are persistent until disabled, and work with all the various tools. As a matter of fact, many tools are simply pre-made combinations of a specific Falloff and a transform; add the Rotate tool to a Linear Falloff and get a Twist tool. This is made possible using the power of the Tool Pipe.

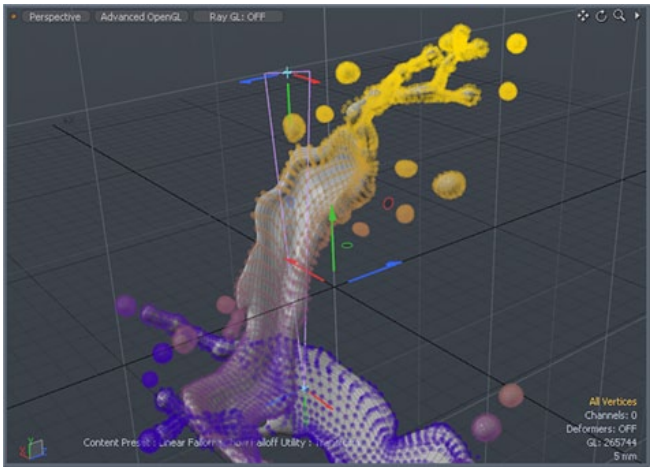
Defining Falloffs is done by selecting the type of Falloff desired form the drop down menu either within the interface, in the 'Falloff' selector menu or in the menu bar under "Edit > Falloff".



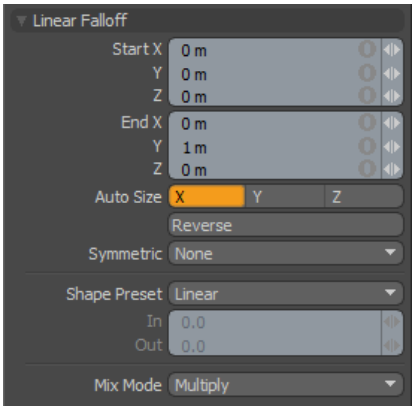
Once selected, users may LMB+ click and drag in the viewport to define the range. Taking its position cue from the Work Plane, the first click defines the *origin* or *center* of the Falloff, and the drag action defines the *size* or *range*. Release the mouse button to

accept the Falloff, and any subsequent tool applications will utilize the Falloff until it is disabled by selecting 'none' or pressing 'Esc'. When a Falloff of is active, its properties are visible within the tools properties panel, as an adjunct to the tools properties itself. Each has its on specific settings. Anytime a tool is in *interactive mode*, users can modify the input values and receive real-time feedback of the results. Additionally, users may adjust the tool directly in the viewport by LMB+click dragging the widgets visible on the Falloff display.

If a tool is active when setting a Falloff, such a the move or rotate tools, the act of simply selecting the Falloff type in the menu will automatically scale the Falloff to the bounding box size of the active selection (which would be all geometry if nothing were selected). This functionality makes it easy to accurately apply Falloffs when modeling. To better visualize the effect a Falloff has over a selection, users can enable the 'Show Falloffs' option in the menu bar under "View > Show Falloffs". This colorize's the vertices depending on the amount of influence a Falloff has over them, dark purple for no influence transitioning toward yellow for full influence as demonstrated by the image below (with a Linear Falloff on the Y axis).



Falloff Types



Linear

Linear Falloff uses a straight line to define Falloff. The line itself determines the direction of the Falloffs influence with the strength of the effect itself reaching out toward infinity. All the vertices at one end of the line receive 100% of the tool's influence. The vertices at the other side receive 0% of the tool's influence. The points between the two lines receive some partial value based on the interpolation between the two end points.

Start X/Y/Z: Defines the starting position of the Falloff as a specific XYZ coordinate value. This end of the line and everything beyond it receives the maximum amount of influence attenuating toward the 'End' position.

End X/Y/Z: Defines the end position of the Falloff as a specific XYZ coordinate value. This end of the line and everything beyond it receives no influence attenuating toward the 'Start' position.

Auto Size X/Y/Z: Users can select either of these three options to automatically scale the 'Start' and 'End' points of the Falloff to match the bounding box size of the selected elements along one of the three axes.

Reverse: Users can select this option to invert the 'Start' and 'End' points reversing the influence of the Falloff.

Symmetric: Users can choose from several options to automatically mirror the influence area of the Falloff-

None- Symmetric function is disabled

Start- Influence of Falloff is symmetrically mirrored across the 'Start' position.

End- Influence of Falloff is symmetrically mirrored across the 'End' position.

Shape Preset: The strength of the Falloffs influence can be controlled along the extent using a 'Shape Preset'-

Linear- Attenuation of Falloff occurs evenly across its range.

Ease-In- Strength of Falloff is greater toward the 'Start' position.

Ease-Out- Strength of Falloff is greater toward the 'End' position.

Smooth- Strength of Falloff is greater toward the center of the Falloff.

Custom- Users can use the 'In'/'Out' options to fine tune strength of Falloff.

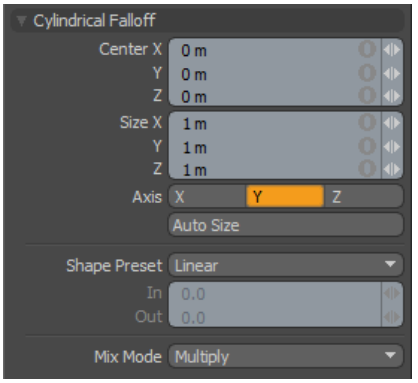
In/Out: The 'In' value determines the strength of the Falloff nearer to the 'Start' position, where the 'Out' value determines the strength on nearer the 'End' side of the Falloff.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.

Cylinder

The Cylindrical Falloff creates a circular Falloff that extends across the length of a virtual cylinder. All vertices at the center of the cylinder receive 100% of the tools influence attenuated toward the outer edge and beyond the cylinder, where vertices receive none of the tools influence.

Center X/Y/Z: Defines the Center of influence, where the strength of the Falloff will be greatest (100%). The strength of the falloff will attenuate toward the outer bounds of the Cylinder. The Falloff volume extends upward and downward toward infinity.



Size X/Y/Z: Defines the radius of a perfect circle from the 'Center' and determines the outer area of the Falloff where there is no affect.

Axis: Defines the Projection Axis for the cylinder determining the direction for the Falloff.

Auto Size: Users can select this option to automatically size the Falloff's 'Center' and 'Size' values to match the bounding box of the current selection.

Shape Preset: The strength of the Falloffs influence can be controlled along the extent using a 'Shape Preset'-

Linear- Attenuation of Falloff occurs evenly across its range.

Ease-In- Strength of Falloff is greater toward the 'Start' position.

Ease-Out- Strength of Falloff is greater toward the 'End' position.

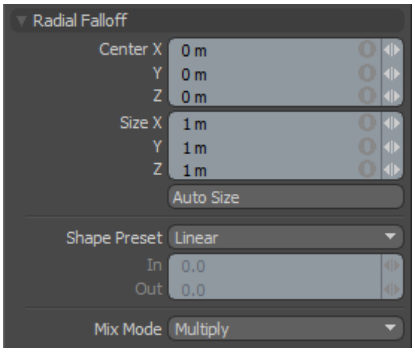
Smooth- Strength of Falloff is greater toward the center of the Falloff.

Custom- Users can use the 'In'/'Out' options to fine tune strength of Falloff.

In/Out: The 'In' value determines the strength of the Falloff nearer to the 'Start'

position, where the 'Out' value determines the strength on nearer the 'End' side of the Falloff.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.



Radial

Radial Falloff uses a sphere where all vertices at the very center of the sphere receive 100% tool influence and any vertices outside the sphere receive 0% of the tool's influence. There is a smooth gradient of tool influence between the center and edge boundary of the sphere that utilizes the Falloff shape preset to determine the interpolation style.

Center X/Y/Z: Defines the Center of influence, where the strength of the Falloff will be greatest (100%). The strength of the falloff will attenuate toward the outer bounds of the Spherical volume, the area outside the volume receives no tool influence.

Size X/Y/Z: Defines the radius of a perfect circle from the 'Center' and determines the outer area of the Falloff where there is no affect.

Auto Size: Users can select this option to automatically size the Falloff's 'Center' and 'Size' values to match the bounding box of the current selection.

Shape Preset: The strength of the Falloffs influence can be controlled along the extent using a 'Shape Preset'-

Linear- Attenuation of Falloff occurs evenly across its range.

Ease-In- Strength of Falloff is greater toward the 'Start' position.

Ease-Out- Strength of Falloff is greater toward the 'End' position.

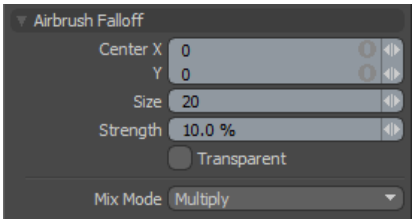
Smooth- Strength of Falloff is greater toward the center of the Falloff.

Custom- Users can use the 'In'/'Out' options to fine tune strength of Falloff.

In/Out: The 'In' value determines the strength of the Falloff nearer to the 'Start' position, where the 'Out' value determines the strength on nearer the 'End' side of the Falloff.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using

the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.



Airbrush

The Airbrush Falloff allows a tools influence to be *painted* onto a mesh. Unlike the other Falloff tools, airbrush continually adds to the tool influence while you drag the brush around the mesh. This tool is similar to the Airbrush tool found in 2D paint programs, except rather than smoothly painting pixels into an image, the brush *paints* the tool influence into the mesh. This can be combined with many tools in modo for a wide variety of effects. Users can RMB+click and drag to define the size of the brush being applied interactively.

To use the 'Airbrush falloff, activate the Falloff, apply the effect, such a 'Move'. At first users will see no results, then LMB+click and drag over the surface of the model to apply the tools action to the geometry.

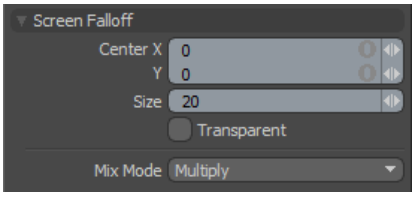
Center X/Y: Determines the current center position in the viewport of the brush tip, users can interactively LMB+click and drag over a models surface to actually apply the Falloff.

Size: Sets the pixel size of the Airbrush tips influence, calculated as a radius from the 'Center' position.

Strength: The 'Strength' value defines the intensity of the Falloffs application when brushing over a surface.

Transparent: When 'Transparent' is disabled, the Falloff only affect polygons facing toward the viewport, when enabled the falloff works through surface also affecting those facing away from the viewports.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.



Screen

Screen Falloff is actually quite similar to Cylinder Falloff, except rather than using one of the major axes to determine the angle of the projection, the screen Falloff projects a cylinder that is perpendicular to the viewport window (i.e. the

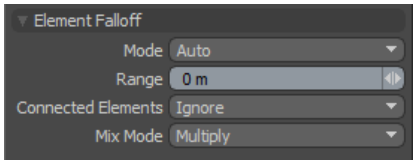
computers *screen*). The Falloff is defined by a disc drawn in screen space extending infinitely into the distance. This disc is used to project a virtual cylinder of influence.

Center X/Y: Defines the Center of influence in *screen space*, where the strength of the Falloff will be greatest (100%). The strength of the falloff will attenuate toward the outer bounds of the Spherical volume, the area outside the volume receives no tool influence.

Size: Sets the pixel size of the Circular area of influence, calculated as a radius from the 'Center' position.

Transparent: When 'Transparent' is *disabled*, the Falloff only affect polygons facing toward the viewport, when *enabled* the falloff works through surface also affecting those facing away from the viewports.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.



Element

Element Falloff limits the tool influence to the element (vertex, edge or polygon) in the mesh that is clicked on when using the current tool. Users can hover the mouse pointer over a component (vertex, edge or polygon) and see a pre-highlight of the target elements. LMB+clicking on the element will apply the tools action to the element, attenuating outward based on the 'Range' value.

Mode: There are several modes available for selecting the element type-

Auto- Users can choose from any component, based on which element is clicked on.

Auto Center- Same as 'Auto', but the 'Range' is calculated from the elements center point.

Vertex- Users are limited to only selecting Vertices.

Edge- Users are limited to only selecting Edges.

Edge Center- Same as Edge', but the 'Range' is calculated from the elements center point.

Polygon- Users are limited to only selecting Polygons.

Polygon Center- Same as 'Polygon', but the 'Range' is calculated from the elements center point.

Range: This value determines the extent of the Falloff range, measured outward from the tool handle. When the 'Center' options are selected, the 'Range' is calculated from the elements center point. At the default 0m value, only the clicked element will be affected.

Connected Elements: Defines what connected geometry will do when an element is transformed.

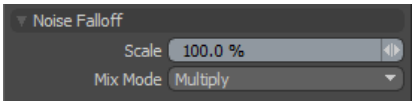
Ignore- Ignores all connected elements and only moves the element specified.

Use Connectivity- Only affects single surface/connected elements, unconnected elements within range will be ignored.

Rigid Connections- Moves all connected elements equally the specified distance.

Edge Loops- Moves the connected loop, defined by all the connected quad polygons/edges/vertices in a single row.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.

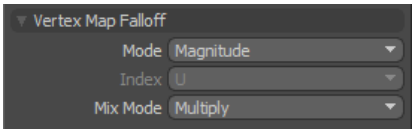


Noise

Noise Falloff uses a three dimensional noise pattern to modulate the influence of the tool. The size of the noise pattern can be adjusted with the scale parameter. Combined with the Move or Stretch tool, this Falloff can be useful for roughing up a surface.

Scale: This value determines the size of the noise in relation to the geometry it is affecting.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.



Vertex Map

The Vertex Map falloff converts values of the currently selected vertex map to values representing falloff strength. While generally a Weight Map is used, being the most intuitive approach, users may also use values of other vertex map types as a falloff as well. The Falloff is determined by the values of an existing vertex map. This allows users the ability to create complex Falloffs that actually can have a inverse negative in instances where negative values can be applied.

If user chooses "Component" for the 'Mode' option, the falloff tool uses the component value specified by the index option. For example, if the current map is a Morph Map and the index is "X", then the falloff sets the X value of morph position to the weight value. "Magnitude" is effective for Morph, RGB, and RGBA maps. If the current map selected is a Morph map, the falloff returns the length of the morph delta position. If it is RGB or RGBA, the falloff returns the luminosity of color vector.

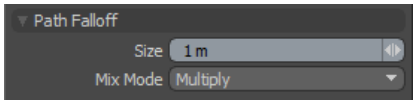
Mode:

Magnitude- Values of the vertex map are converted to strength values and applied to the affected surface(s).

Component- Users can select individual components of a map- Weight Map (1 component- weight value), RGB Map (3 components- R, G or B), RGBA Map (4 components- R, G, B or A), Morph Map (3 components- X, Y or Z), Pick Map (1 component)

Index: When 'Component' mode is selected, users determine the specific component using this option selector.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.

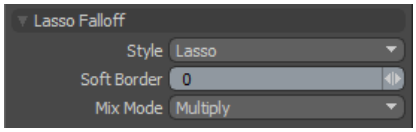


Path

The Path Falloff combines the Curve Generator with the Path Falloff. The curve generator creates a curve and the Falloff reads the curve points to determine where the tool should be active. If the user has the Path Falloff in the Tool Pipe at the same time as a mesh edit or transform tool, they may need to select the Curve Path (in 'Polygons' mode) prior to activating the Falloff. This will allow users to define the curve to be used. Once the basic curve created users can select the tool and begin to edit. Users may also click on the curve control points to edit them while the tool/Falloff combination is active.

Size: This value determines the area of attenuation around the path itself, calculated as a tube around the curves position.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.



Lasso

Several styles of Lasso selections drawn in screen space projecting outward toward infinity,

producing Falloffs that are very specific. Settings allow users to then to define areas of influence around the lasso shape with the 'Soft Border' option. All areas within the Lasso receive 100% of the tools effect; the attenuation of the falloff occurs across the 'Soft Border' distance only outside the Lasso area.

Style: Users can choose from the various Lasso styles-

Lasso- Allows users to draw freeform shapes onto the screen defining the Falloff .

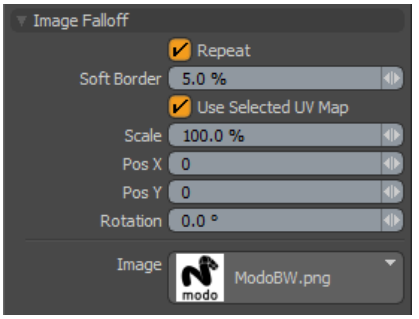
Rectangle- Users can drag out rectangular shapes to define the Falloff area.

Circle- Users can drag out perfectly circular shapes to define the Falloff area. Shape originates from center of the circle.

Ellipse- Users can drag out elliptical shapes to define the Falloff area. Shape originates from the upper or lower corner of the ellipse.

Soft Border: The 'Soft Border' option defines the actual Falloff radius around the Lasso area that attenuates across the distance defined. Calculated in screen-space pixels from the lasso border.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.



Image

Much like the Image Ink option in the painting toolbox, users may apply an image, in screen space, as a Falloff, where the brightness (luminosity) of the image will affect the selected geometry. Additionally, by selecting the 'Use Current UV' option of the Falloff, users can apply the current image as a Falloff in 3D object space opening up a whole variety of options.

To use, user will need to place the masking image into the images folder of the Presets, or add the path to the Preset Browser. Once the image is selected, LMB+clicking in the viewport will activate the image widget, where users can use the controls to position the image within the 3D viewport using the various handles.

Repeat: When this option is enabled, the image is repeated outside of the images borders. Has no effect when 'Use Selected UV Map' is *enabled*.

Soft Border: Attenuates the strength of the images affect over the geometry. At 100% the strength of the falloff is strongest at the center falling off toward the edges of the

image. At 0%, the white areas of the entire image will affect the geometry.

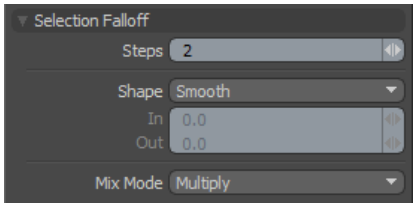
Use Selected UV Map: When this option is enabled, modo will bypass the viewport image, and apply the image's affect to the surface with the currently selected UV map. Users can select UV maps within the Lists viewport under the UV Map section.

Scale: This value determines the size of the image in the 3D viewport, and when 'Repeat' is *disabled*, the size of the Falloffs area of influence as well. USers can LMB+click the right side handle to interactively rotate the image.

Pos X/Y: Positions the center point of the Image within the 3D viewport. Users can also LMB+click the center handle on the widget to interactively position the image.

Rotation: Rotates the image within the3D viewport. Users can LMB+click the top handle to interactively rotate the image.

Image: Users can use the 'Image' button to browse through the preset bitmap textures, selecting the image that will be used for the Falloff.



Selection

Selection Falloff allows you to modulate a geometry edit based on a selection area. The area of influence is at 0% on the selection border and ramps to 100% based on the Steps value. If the Steps are set to 1 there will be only a slight Falloff range at resulting in a "harder" edit. Increasing the number of steps will soften the influence at the selection border and the number of steps can be controlled by RMB (right mouse button), or by editing the text field in tool properties. You might think of these "steps" as rows or loops of selected elements; for each Step the Falloff reduces the selection area and uses the range between the original selection and the reduced selection as the "soft" border for the Falloff area.

Steps: This value determines the extent of the Falloff range, where each 'Step' represents a loop of polygons or a ring of edge sections. The greater the 'Steps;' value, the further the Falloff attenuates.

Shape: The strength of the Falloffs influence can be controlled along the extent using a 'Shape Preset'-

Linear- Attenuation of Falloff occurs evenly across its range.

Ease-In- Strength of Falloff is greater toward the 'Start' position.

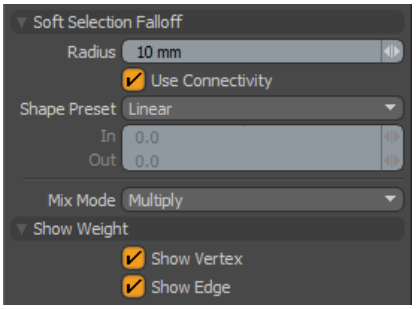
Ease-Out- Strength of Falloff is greater toward the 'End' position.

Smooth- Strength of Falloff is greater toward the center of the Falloff.

Custom- Users can use the 'In'/'Out' options to fine tune strength of Falloff.

In/Out: The 'In' value determines the strength of the Falloff nearer to the 'Start' position, where the 'Out' value determines the strength on nearer the 'End' side of the Falloff.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.



Soft Selection

The Soft Selection Falloff is unique in that it is the only falloff that modulates the geometry outside of the selection.

Radius: Determines the range outside of the selected edge to attenuate the falloff.

Use Connectivity: When *enabled* only affects single surface/connected elements, unconnected elements within range will be ignored.

Shape: The strength of the Falloffs influence can be controlled along the extent using a 'Shape Preset'-

Linear- Attenuation of Falloff occurs evenly across its range.

Ease-In- Strength of Falloff is greater toward the 'Start' position.

Ease-Out- Strength of Falloff is greater toward the 'End' position.

Smooth- Strength of Falloff is greater toward the center of the Falloff.

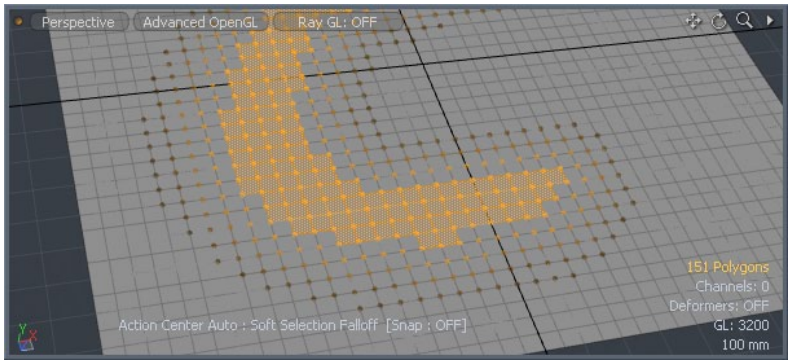
Custom- Users can use the 'In'/'Out' options to fine tune strength of Falloff.

In/Out: The 'In' value determines the strength of the Falloff nearer to the 'Start' position, where the 'Out' value determines the strength on nearer the 'End' side of the Falloff.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.

Show Weights--

Show Vertex/Show Edge: Users can use these toggles to enable/disable the display of geometry highlights for both vertices and edges, giving a visual reference to the application of the Soft Selection falloff (demonstration of the vertices display mode below).



TIP: Commands such as 'Copy', 'Paste' or 'Delete' are typically all or nothing type events and are therefore unaffected by Falloffs.

Selections

Selection Modes and Types

There are seven total selection modes in modo. The first three- 'Vertices, Edges, and Polygons' are known as the *component* selections modes. In these modes users can use the mouse pointer to select the component elements of the geometry for manipulation using modo's many various modeling and transform tools. When in 'Items' mode, users can transform and entire item layer as a whole, move, rotate or scale. If you think of item layers as a container, component edits manipulate the contents of the container, and item edits transform the container itself. Next, the 'Materials' selection mode will allow the user to automatically select a shader tree layer by clicking on a polygonal surface within the 3D viewport (no more guessing which surface belongs to what geometry). The last two modes 'Pivot' and 'Center' allow the user to select the center and pivot elements of an item layer and manipulate them directly modifying their position and orientation to the associated geometry for animation purposes. Users can directly select any of these modes using the keyboard shortcuts-

1 = Vertices

2 = Edges

3 = Polygons

4 = Materials

5 = Items

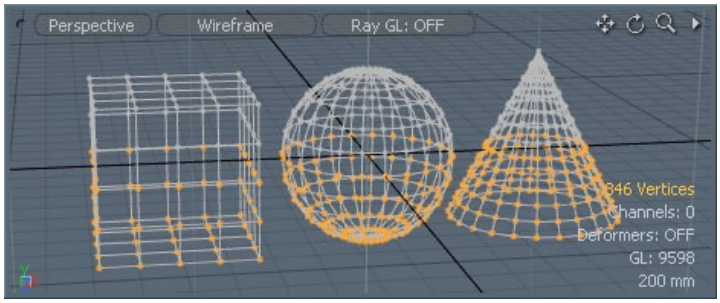
6 = Pivot

7 = Center

Cycle Component Selections

This command cycles through the component selection types of Vertices, Edges and Polygons. In the modo default configuration this is mapped to the 'Spacebar' key. If there is a tool selected the tool is first dropped. If no tool is selected, the command cycles the selection sequentially from vertices to edges to polygon and back to vertices in a continuous loop.

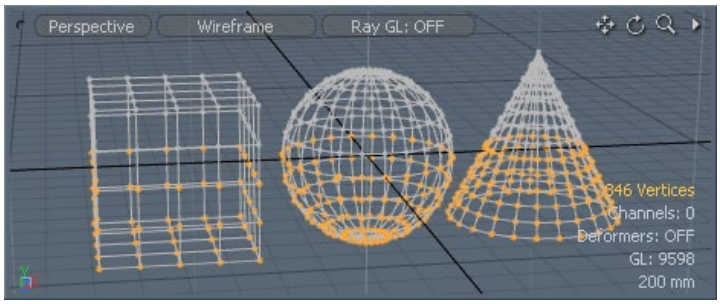
Vertices Selection Mode



Vertices are a single point that represents a position in 3D space. When in Vertices selection mode users can use the basic *paint* or *lasso* selection options to quickly select one or more vertices for precise editing. Selected vertices will be highlighted an orange color to distinguish them from unselected vertices. Using the 'LMB+click and dragging over an object, users can *paint* a selection of one of more vertices or use the RMB+click and drag, drawing a lasso loop around the selection area selecting multiple vertices at once. There are also a number of selection options that can be used with vertices such as Loop Select, Select More/Less and many more (found on the 'Selection Modifiers' page of the documentation). Users can change the Lasso Style by going to the menu bar and choosing "Select > Lasso Style" and making the choice there.

Vertices selection mode can be activated LMB+clicking the 'Vertices' interface button in the modes toolbar. Alternatively, users can use the menu bar command "Select > Vertices", "1" keyboard shortcut, or by RMB+click away from any geometry in the 3D viewport to open a context menu and selecting the 'Vertices' option. And if that isn't enough, users can also switch to Vertices selection mode by simply MMB+clicking on any vertex in the viewport.

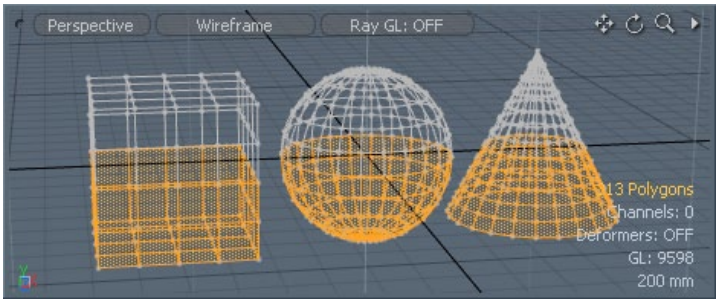
Edge Selection Mode



Edges are defined by two vertices, the edge itself is the resulting line between them. When in 'Edges' selection mode users can use the basic *paint* or *lasso* selection options to quickly select one or more edges of their geometry for precise editing. Selected edges will be highlighted an orange color to distinguish them from unselected edges. Using the 'LMB+click and drag' over an object to *paint* a selection of one or more edges or use the RMB+click and drag, drawing a lasso loop around the selection area selecting multiple edges at once. There are also a number of selection options that can be used with edges such as Loop Select, Ring Select and many more (found on the 'Selection Modifiers' page of the documentation). Users can change the Lasso Style by going to the menu bar and choosing "Select > Lasso Style" and making the choice there.

Edges selection mode can be activated LMB+clicking the 'Edges' interface button in the modes toolbar. Alternatively, users can use the menu bar command "Select > Edges", "2" keyboard shortcut, or by RMB+click away from any geometry in the 3D viewport to open a context menu and selecting the 'Edges' option. And if that isn't enough, users can also switch to Edges selection mode by simply MMB+clicking on any edge in the viewport.

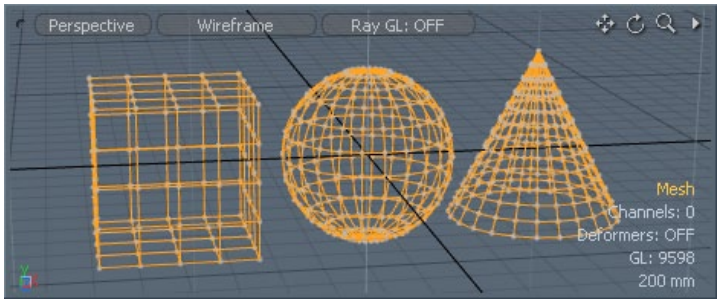
Polygon Selection Mode



Polygons are the closed plane that is created from three or more edges that produces a renderable surface. When in 'Polygons' selection mode users can use the basic *paint* or *lasso* selection options to quickly select one or more polygons of their geometry for precise editing. Selected polygons will be highlighted an orange color to distinguish them from unselected polygons. Using the 'LMB+click and drag' over an object to *paint* a selection of one or more polygons or use the RMB+click and drag, drawing a lasso loop around the selection area selecting multiple polygons at once. There are also a number of selection options that can be used with polygons such as Loop Select, Select Connected and many more (found on the 'Selection Modifiers' page of the documentation). Users can change the Lasso Style by going to the menu bar and choosing "Select > Lasso Style" and making the choice there.

Polygons selection mode can be activated LMB+clicking the 'Polygons' interface button in the modes toolbar. Alternatively, users can use the menu bar command "Select > Polygons", "3" keyboard shortcut, or by RMB+click away from any geometry in the 3D viewport to open a context menu and selecting the 'Polygons' option. And if that isn't enough, users can also switch to Polygons selection mode by simply MMB+clicking on any polygon in the 3D viewport (except in Wireframe draw mode).

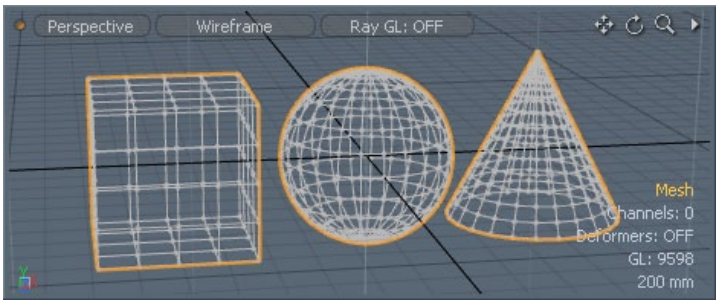
Items



Items represent individual object layers in a scene, whether its a 'Camera', 'Light', 'Locator', 'Mesh' or other type of object, each one is individually an item. When modeling, users are working in one of the component modes, Vertex, Edge, or Polygon. In these modes the users are modifying the *contents* of the layer items, but while in *Items* mode users can select and transform entire layers. Only mesh layers can be modified in the component modes, all other elements are transformed in Items mode. For more information the subtleties of the modes, please see the Items vs Components page of the documentation. All keyframes for animation are set while in Items mode. Component animation is only possible through the use of mesh Deformers.

Items selection mode can be activated by LMB+clicking the 'Items' interface button in the modes toolbar. Alternatively, users can use the menu bar command "Select > Items", "5" keyboard shortcut, or by RMB+click away from any geometry in the 3D viewport to open a context menu and selecting the 'Items' option.

Material Selection Mode

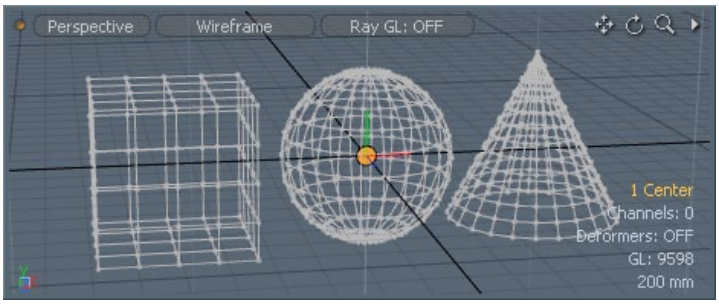


Materials are defined when users make specific polygon selections and assign a 'Material' tag. These tags then produce a material 'Group' item in the Shader Tree that represents the tagged polygons for masking. Any surfacing is applied to the geometry using this mask. Users can then utilize the 'Material' selection mode to select geometry based on the tag definitions. Material selections are displayed as orange outlines surrounding the tags defined area. Users can either use the selection to edit the geometry, or to simply adjust surfacing within the Shader Tree. When a specific material tag is selected in the 3D viewport, the corresponding Group item is also selected in the Shader Tree making it easier to locate a specific Shader Tree layer in a complex scene.

Material selection mode is somewhat of a misnomer, as it is possible to select from several polygon tag types that modo offers. Using the menu bar options found in "Select > Polygon Tag Type", users can set the "Materials" selection mode to select based on Material, Part or Smoothing Group polygon tags.

Materials selection mode can be activated LMB+clicking the 'Materials' interface button in the modes toolbar (found in the flyout menu under the 'Items' button. LMB+click and hold to open). Alternatively, users can use the menu bar command "Select > Materials", "4" keyboard shortcut, or by RMB+click away from any geometry in the 3D viewport to open a context menu and selecting the 'Materials' option.

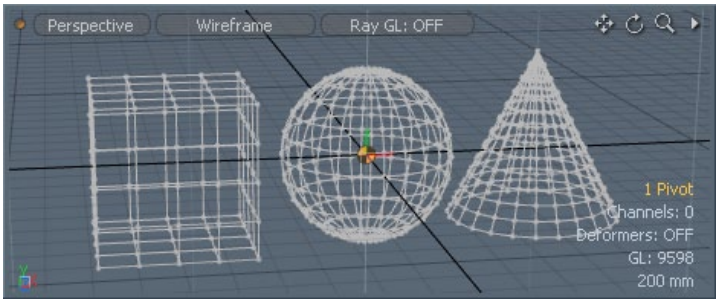
Center Selection Mode



The 'Center' element represents the center location of the associated item layer. Each layer in the 'Items' list has its own corresponding Center position. This location defines where all Position, Scale and Rotate transforms originate from for that particular layer. While users can use 'Action Centers' to dynamically define transform origins for modeling operations, Centers define the origin when animating in modo. By default, the Center is located at the world position origin of 0,0,0 for any item layer. Users can enter 'Centers' mode, selecting the Center element by LMB+clicking on the center icon widget. Users can then use the 'Move' and 'Rotate' transform tools to reposition the center point in relation the the geometry without affecting the geometry position itself.

Center selection mode can be activated by LMB+clicking the 'Center' interface button in the modes toolbar (found in the flyout menu under the 'Items' button. LMB+click and hold to open). Alternatively, users can use the menu bar command "Select > Center", "7" keyboard shortcut, or by RMB+click away from any geometry in the 3D viewport to open a context menu and selecting the 'Centers' option.

Pivot Selection Mode



The Pivot is a secondary center-like element that is defined by an offset away from the Center itself. Typically the Pivot and Center positions coincide precisely for any layer, if the user modifies the Pivot expressly, that then becomes the animation origin,

overriding the position of the Center. Pivots are differentiated from Centers in that their position and rotation can be keyframe animated, whereas a Center cannot. This allows certain interesting possibilities.

Pivot selection mode can be activated by LMB+clicking the 'Pivot' interface button in the modes toolbar (found in the flyout menu under the 'Items' button. LMB+click and hold to open). Alternatively, users can use the menu bar command "Select > Pivot", "6" keyboard shortcut, or by RMB+click away from any geometry in the 3D viewport to open a context menu and selecting the 'Pivots' option.

Implicit Selections for Component Modes

Beyond the direct selection of elements, all the selection modes implicitly select vertices, polygons and edges. In other words, when a polygon is selected, the four vertices that make up that polygon are also implicitly selected. In this way you can select a polygon and still perform a vertex bevel, for example, and the four vertices in that polygon will be the active mesh elements for that particular tool. The vertices will not appear as selected in the viewport, but you will be able to perform vertex operations on them just the same.

In all modes, if nothing is directly selected then all vertices, polygons or edges are implicitly selected.

Direct selections result in the following implicit selections:

Vertex mode
- selected vertices only

Edge mode
- vertices of selected edges

- selected edges

Polygon mode
- vertices of selected polygons
- contour edges of selected polygons
- selected polygons

Contour edge can be defined as follows: if you select a batch of polygons in polygon mode, and then try to use an edge tool, the action is applied to all the edges around your polygon group. This is a simple one-step method for selecting certain types of edge loops.

Using the 'Alt' key, users can convert implicit selections to direct selections. For

example, if you have 3 polygons selected, holding down the 'Alt' key and LMB+clicking on the Polygon selection button in the 'Modes' toolbar will change the viewport selection to edges. Doing the same for Edge selections will convert the selection to vertices. The 'Ctrl' key will change Edge selection button to a Boundary selection button where users can convert existing Polygon selections to boundary edge selections.

Selection Styles

Paint Select

Geometry selections can be painted by clicking on elements in 3D view:

Left mouse button+drag first deselects everything and then selects all elements under the mouse as it moves (clicking off any element deselects all). In this manner, you can quickly "paint" a selection by dragging over the desired component.

Shift+click adds more of the current selection mode elements to the selection, anything hit by the mouse pointer as it moves across the geometry. You can either shift+click on single elements or shift+drag across multiple elements to add to the selection with a paint selection style.

Ctrl+click removes items from the selection.

Lasso Select

Geometry selections can be made using a lasso in 3D view, activated using the right-mouse button. Geometry must be fully enclosed in the lasso to be selected.

In Shaded mode, the right-mouse button lasso uses a backface cull, so polygons, edges or vertices facing away from the view plane are not selected. Using a lasso activated with the middle-mouse button will select through the geometry, making selections on the backside as well. This is reversed in Wireframe mode; the right-mouse button selects through the mesh, the middle-mouse button uses the backface cull, only selecting facing geometry.

To add new geometry to a selection, hold down the SHIFT key and continue using the lasso selection. The new selection is added to the original.

To remove geometry from a selection, hold down the CTRL key and continue using the lasso selection. The newly selected geometry is removed from the existing selection.

You can also change the Lasso Style to a Rectangle, Circle or Ellipse by choosing "Select > Lasso Type" and picking an alternate style. These other types follow the same behavior as the default Lasso.

Lasso Style: Select Lasso Style allows you to choose the shape of lasso selection. The style defaults to a regular lasso, but can be changed to a rectangle, circle, or ellipse shape.

Dropping Selections

To deselect, or "drop" the current selection you can simply click anywhere in a 3D viewport, press the Space bar or ESC key. If you happen to click on another element of the current selection mode, the original selection will be dropped and the element you clicked on will become selected. Selections are an undoable action so you can easily undo back to a selection state if you accidentally drop a selection. When nothing is selected, everything is considered active by the tools and commands. This makes dropping selection a quick way to *select* everything.

Selection Modifiers

Convert Selection

This command sets a new selection mode and uses the content of the previous selection to set the new selection:

Vertex -> edge: selects edges defined by the selected points
Vertex -> polygon: selects any polygons defined by the currently selected points
Edge -> vertex: selects points of all selected edges
Edge -> polygon: selects polygons that are completely surrounded by selected edges
Polygon -> vertex: selects points of all selected polys
Polygon-> edge: selects all the edges of the currently selected polygons

It is also possible to convert from a Material selection to any of the component level selections or vice versa.

You can convert between selection types using the ALT key. For example, if you have polygons selected in the viewport, pressing the ALT key will change the selection popup buttons on the modo Modes toolbar to read Convert. Press the Convert button where Vertices was previously displayed, and the polygon selection will be converted to vertex selection. This way you can quickly access all the vertices that were part of the polygon selection.

Grow Selection (Shift+Up arrow)

Select > Grow expands or "grows" the selected elements, by selecting all unselected geometry elements directly contiguous with the selection.

Grow selection is mapped to SHIFT+Up arrow so you can quickly expand your selection set.

This command works with all elements of geometry (Vertices, Edges, and Polygons.)

This command is the functional opposite of the Select > Shrink command.

Shrink Selection (Shift+Down Arrow)

Select Shrink reduces the selection by deselecting elements that are at the "outside" of the currently selected elements. This has the affect of "shrinking" the selection set to a smaller size.

The keyboard equivalent of this tool is mapped to SHIFT+Down arrow. This works with all geometry elements (Vertices, Edges and Polygons.)

Select More/Less (Up/Down Arrows)

Selects next plausible element. If a polygon was skipped then the next plausible element select would also skip a polygon.

Select Loop ("L" key)

Selects a continuous span of quadrangular polygons, edges or vertices, depending on selection mode

Invert Selection ("[" -left square bracket)

Accessed through Select menu > Invert, this command switches the state of all polygons, edges or selected vertices.

The selected elements become unselected and the unselected become the selected.

Toggle the selection using the Invert quick key [(the left hand square bracket).

Select Connected ("]" right square bracket)

The Select Connected command lets you quickly expand geometry selections to include all elements physically contiguous with the selection. This is a very handy workflow for selecting mesh geometry. You simply click to select a single polygon, edge or vertex, then press the "]" key and the entire piece of geometry is selected. While in Polygon or Vertex selection mode you can also simply double-click on a vertex or polygon to select all connected elements.

Select Between (Shift+G)

The Select Between command requires that two polygons or edges are selected prior

to triggering the command. Once the command is triggered it completes the selection between the two based on the shortest distance between them. Only a single row is selected if the initial polygons or edges were on the same loop, polygons in adjoining rows or columns would create rectangular selection.

Boundary Edges

With polygons selected, this command selects all the edges around the boundary edge of the polygon set. If no polygons are selected, it basically selects all the non-manifold edges of the mesh, which can be useful for finding holes.

The mechanism is simply to select all edges bordered by an odd number of selected polygons.

To select a polygon boundary do the following:

1) Select the polygons on the mesh.
2) Press the CTRL key. Notice the Edge button now is labeled "Boundary".
3) Click the Boundary button.

This command is also available on the Select menu > Boundary or from the Statistic tab of the Info & Statistics form, by expanding the Edges > By Boundary entry.

Note that this command works on both continuous and discontinuous selection sets.

Select Ring (Alt+L)

Geometric selections can be expanded to include all vertices or edges adjacent to the selection across quads. For example, in a sphere, when a single vertical edge is selected, ring select will highlight all the vertical edges at the same "latitude" around the globe of the geometry. This is the counterpart to Loop selection, which will select all the vertical edges with the same "longitude".

Ring works with both vertex and edge selections.

Select Close Loop (Shift+])

The Select Close Loop command requires that two polygons are selected prior to triggering the command. Once the command is triggered it completes a loop selection using the two initial polygons to set direction and then selects all polygons on one side of the loop. The selection of the additional polygons is based on which side has fewer total polygons. For instance, to select all polygons on a finger from the knuckle to the finger tip the user would select two polygons just above the knuckle and fire the command. Since the finger itself has fewer polygons than the rest of the hand this is the side that is selected. This is generally the solution the user is looking for,

however, in the case that it is the opposite side that is of interest the Select Close Loop command can simply be followed by Select Invert.

Select Colinear

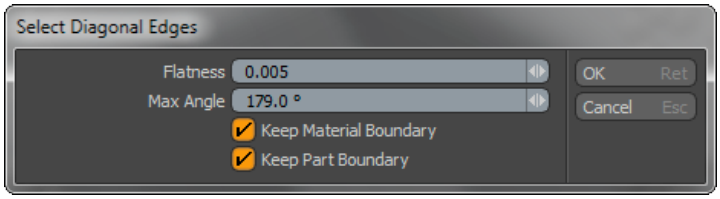
Select Colinear command will select any extraneous vertices across a polygons boundary, making for easy removal. To use, select the command from the menu bar 'Select > Colinear' and a popup dialog will ask you to define a distance. Click OK to execute the command resulting in the selection of all colinear vertices.

Select Coplanar

Select Coplanar command will select the edges between polygons that fall within a flatness range of a selected polygon. To use, select a polygon, and then select the command form the menu bar 'Select > Coplanar' and a dialog box will popup allowing the user to specify a flatness limit. Click OK to execute the command resulting in an edge selection based on the users input.

Select Diagonal

The select diagonal edges is the command to select edges to share triangle pairs. This is useful when you want to convert triangles to quadrangles. You can convert triangles to quadrangles by deleting the selected edges. Here are options for the command-



Flatness: The coplanar flatness threshold between opposing triangle pairs designating whether the edge will be selected or not.

Max Angle: The maximum of angle for the corners of quadrangle from the triangle pair.

Keep Material Boundary: When this option is enabled, edges that would fall along material boundary are not selected.

Keep Part Boundary: When this option is enabled, edges that fall along a part boundary are not selected.

Select by Info and Statistics

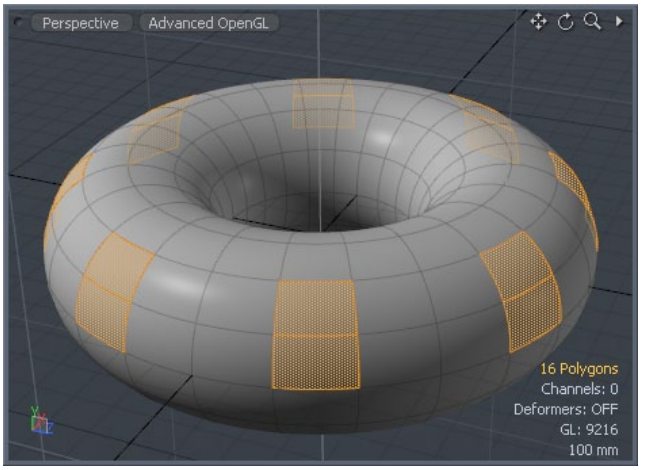
Users can also select elements by using the Info and Statistics panel.

Selection Sets

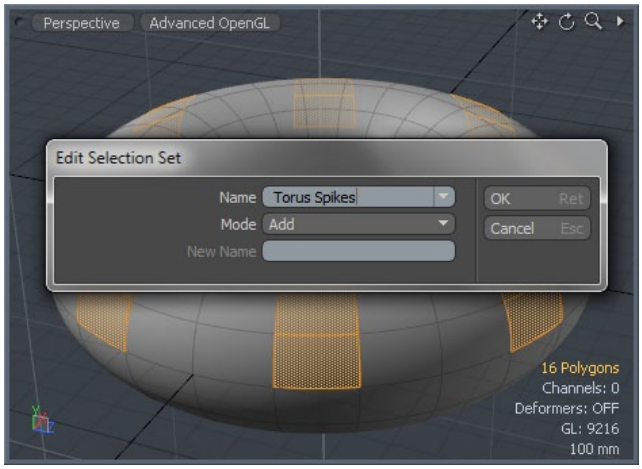
When editing geometry, modo offers a great many tools to make it easy to generate accurate selections. Repeatedly having to make identical selections for subsequent actions can be tedious. Selection Sets are a means of saving and editing component and item based selections for recall at a later time. Selection Sets can be assigned to vertex, edge, polygon and item selections for recall at a later time. Polygon selection sets can also be used for creating additional mask layers in the Shader Tree, allowing for all kinds of interesting layering possibilities. In order to recall or modify any exisiting selection sets, users must be in the same mode as when the selection set was created, meaning any vertex selection sets can only be recalled while in vertex mode. Selection sets are also flexible in how they can be edited; unlike a material or part tags where users can only assign a single given tag at a time, any component or item can belong to as many sets as the user wishes. There are three basic commands for working with selection sets that can be invoked from the menu bars 'Select' menu.

Defining Selection Sets

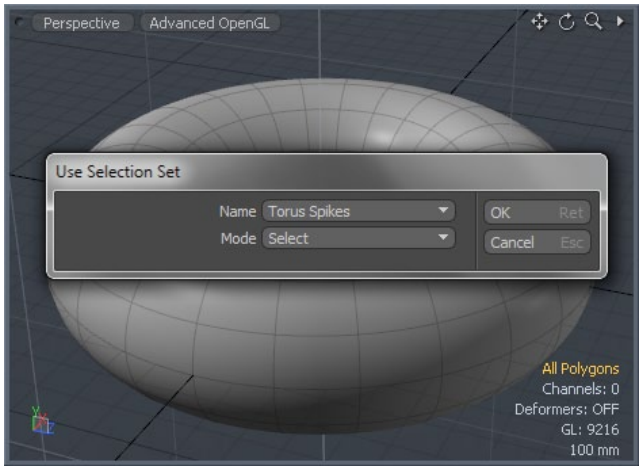
To assign a selection set, you would need to first define the selection you wish to save. For more on defining selections, please see the 'Selection modes and Types' page of the documentation. Selected (active) elements are hilighted an orange color.



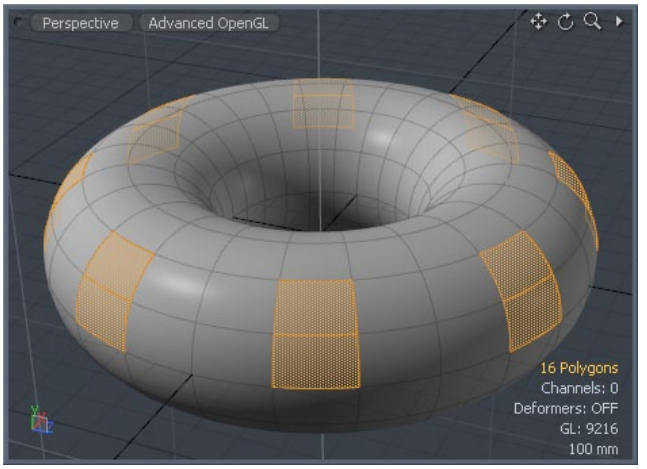
Once you have made the appropriate selection, choose the menu bar command "Select > Assign Selection Set..." to open the 'Edit Selection Set' dialog box. In this dialog, one can define a name for the set, additionally setting the mode to 'Add' or 'Remove' the selected polygons from the set.



To recall selection sets, select the menu bar command "Select > Use Selection Set..." to open the Use Selection Set' dialog box. From within this dialog, one can specify a selection set and choose to 'Select' or 'Deselect' the polygons contained in that grouping.

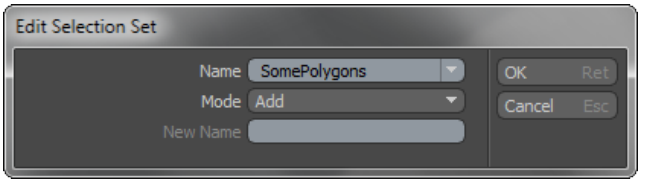


LMB-click the 'OK' button or press the 'Return' (Enter) key on the keyboard to make the selection.



Edit Selection Set

This dialog is multi purpose as it allows users to create a new selection set or add/remove selections from exisiting sets, as well as the ability to rename any existing set. Before creating a selection set, make sure there is an active selection appropriate to the type of set you wish to create, for example, having a polygon selection in 'Polygons' mode.



Name: The 'Name' option allows users the ability to create a new set by simply typing a new name in assigning it to the selection, while keeping the 'Mode' set to 'Add'. To edit existing sets, users may use the arrow widget to select exisiting sets for modifcation using the available 'modes'.

Mode: The 'Mode' option allows users options for modifying exisiting selection sets.

Add-- The 'Add' funtion allows the ability to add any active selections to an existing set.

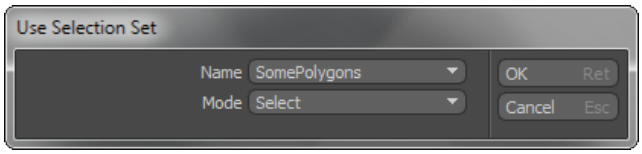
Remove-- The 'Remove' function conversely removes any active selections from an existing selection set.

Rename-- The 'Rename' function allows users the option to choose an exisiting selection set and assign a new name.

New Name: If the user wishes to rename an exisiting selection set, simply select the exisiting set under the 'Name' option, set 'Mode' to 'Rename' and then enter the new name here. LMB-click the 'OK' button or press Return/Enter key to accept.

Use Selection Set

The 'Use Selection Set' dialog window is used to recall selection sets created with the 'Edit Selection Set' dialog.

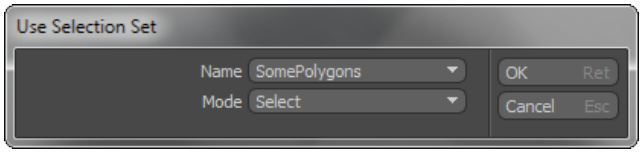


Name: The 'Name' options allows users the ability to select any existing selection set. LMB-click the arrow widget to open the list and make a selection. Only sets created withing the currently active edit mode will be available, such as 'Vertices','Edges', 'Polygons' or 'Items'.

Mode: Using the 'Mode' option, users can--

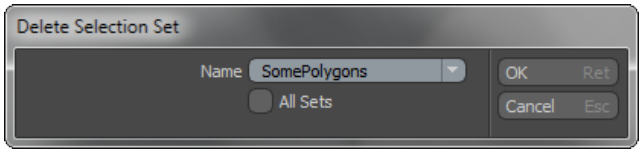
Select-- making the saved selection set active. The set will be added to any pre-exisiting active selections.

Deselect-- making the saved selection set inactive. The set will be removed from any pre-exisiting active selections.



Delete Selection Set

The 'Delete Selection Set' dialog window is used to remove selections sets created with the 'Edit Selection Set' dialog. Select a particular named set, or enable the 'All Sets' option and press 'OK' to delete the Selection Set(s).



Name: The 'Name' options allows users the ability to select any existing selection set. LMB-click the arrow widget to open the list and make a selection. Only sets created within the currently active edit mode will be available, such as 'Vertices','Edges', 'Polygons' or 'Items'.

All Sets: When this option is enabled, all Selection Sets will be deleted, regardless of the named set specified.

TIP: Selection sets can also be recalled using the Info and Statistics viewport. Each section having a listing of all the available names sets. Users can select the particular named set by pressing the '+' (plus) icon in the viewport, conversely users may de-select the geometry in a set by pressing the '-' (minus) icon.

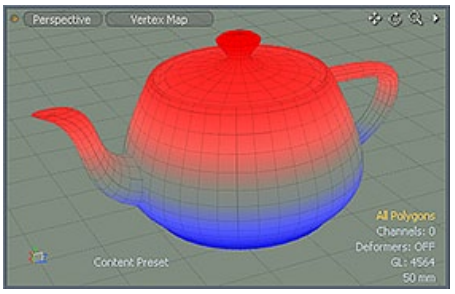
Vertex Maps

Working with Vertex Maps

As some users might assume, Vertex Mapping is not directly related to *texture mapping*, but can play an important role when texturing models for rendering. Put simply, Vertex Mapping is a means of storing vertex specific information within a model for variety of uses. Sometimes they act as instructions to tell a certain function how to act upon a model, as with Weight Maps. Sometimes they store positions offset information as in Morph maps. They can also store 2D coordinate information defining how textures apply to a surface by way of a UV map. Regardless of the map type, vertex maps have proven utility and are essential to working in 3D. As with modeling, there are a variety of options for creating, displaying and working with Vertex maps in modo.

Vertex Map Types

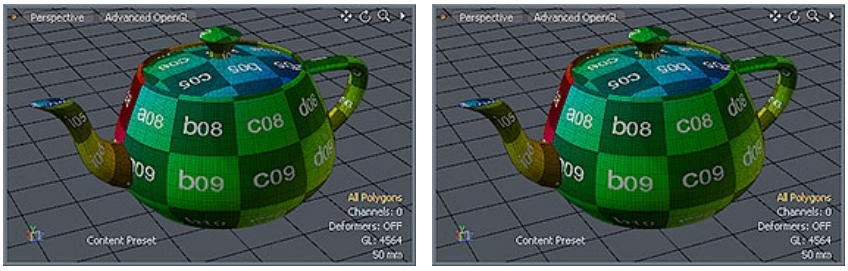
Weight Maps--



Used for a variety of purposes, a weight map can be used to modulate the strength of a tools effect when combined with a weight falloff operator, or used to control the

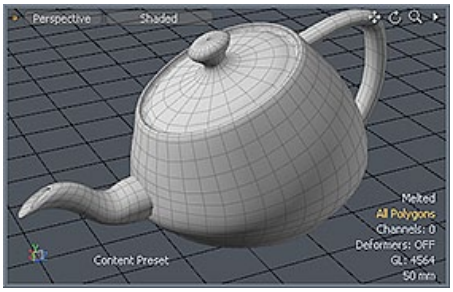
smoothing properties of a Subdivision Surfaces model by adjusting the subdivision weight map, in effect controlling how round or sharp a corner crease becomes when converted to Subdivision Surfaces. The Shader Tree can utilize Weight Maps for modulating surface attributes, with Masks and Gradients. They can also be exported to LightWave, Maya or other applications that support them for use in texture or deformation modulation. Weight maps are discontinuous one-dimensional vertex maps holding weight percentages. When a 3D viewport is set to the type 'Vertex Map' shading mode, a weight value of zero is represented with a neutral green shade while a negative weight value is visualized with a blue, "cool" shade and a positive value is displayed as a red, "warm" shade.

UV Maps--



UV maps are used to explicitly link vertices in the 3D mesh to locations on a 2D image for high accuracy texture mapping. UV Maps are stored as discontinuous 2D vertex maps values for 0-1 UV coordinate data. The UV map can be though of as a method to unfold any 3D mesh so that it lays flat, allowing the 2D image to be applied to the mesh more accurately.

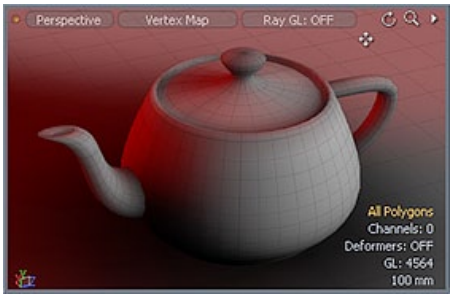
Morph Map--



Morph maps offer users a means of saving alternate deformations for any model. Individual morph map deformations can be animated over time providing the ability to animate say for instance facial expression. In order to animate any morph map, a

morph deformer must first be applied to the model. modo supports two varieties of morph maps, relative and absolute. Relative will offset a particular vertex relative to its current position, and absolute will position a vertex to a specific absolute position. Relative vertex maps are the more common variety. The main benefit to working with morph maps in modo is any topology changes made to a specific morph maps will be applied to all the associated maps.

Color Maps--

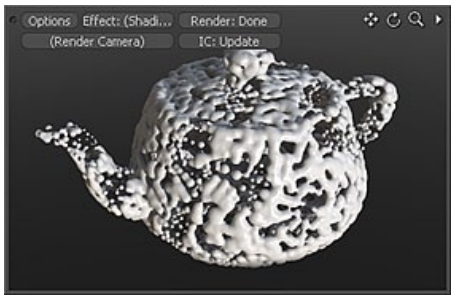


modo provides support for RGB and RGBA vertex color maps. Vertex color maps are often used in games to provide additional color variations in combination with texture maps simulating a fully shaded environment with little to no CPU shading overhead. Vertex color maps can be 'painted' by using the 'Vertex Map > Color Tool' combined with an 'Airbrush' type falloff.

Vertex Normal--

Vertex Normals work in tandem with the Smoothing function to shade polygons as either *faceted or smooth*. A *Vertex Normal map* is effectively the values of the Smoothing setting baked into the vertices. Most often, Vertex Normal maps are generated in an external CAD package when converting a Nurbs based model to polygons. The normal values are computed from the smooth nurbs surface eliminating smoothing errors of the highly triangulated polygonal surface when rendered elsewhere.

Particle Maps--



Particle Maps are useful in giving users control over particle and volume related items such as when using point clouds for Replicators and Texture Replicators and for use with Blobs, Volumes and Sprites ('point clouds' are just masses of particles). This includes 'Dissolve Maps' which control the opacity of individual *particles* and 'Size Maps' which control the scale of *individual particles*.

Transform Maps--

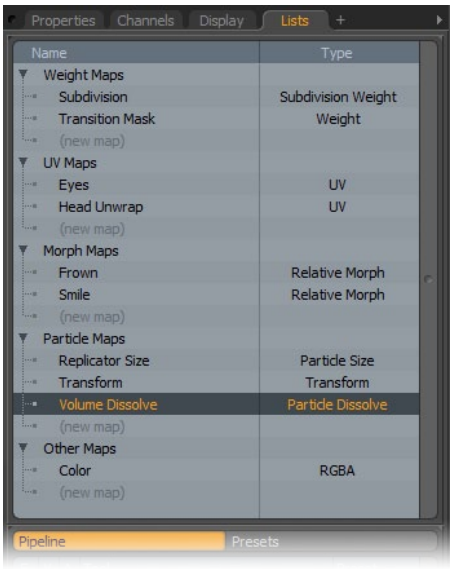
Vertex Transform Maps are useful as they give users individual control over Replicators, as users can apply a Transform Map, then then selecting a target vertex (with a Replicated element applied) and using the standard transform tools, users can adjust rotations, scale and positions on an individual particle basis.

Other Maps--

modo also offers support for various other maps types, such as Pick Maps. This is mainly for compatibility with other applications.

INFO: Particle Maps cannot be applied to procedural particle items, such as 'Particle Clouds' and the 'Particle Generator'.

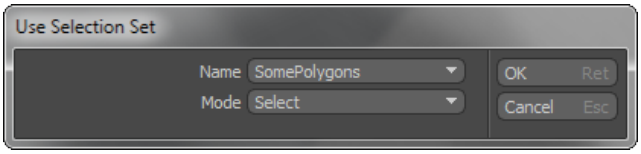
Lists viewport



While map specific information is stored within the model itself at the vertex level, Vertex maps, like Item layers, are organized within the Lists viewport panel. Within the viewport, users can select and deselect maps making them active for editing. Maps can also be created, deleted and renamed from within the Lists view. The various options are available by RMB+clicking on a vertex map name and choosing from one of the option in the pop-up context menu. The List view doesn't see as much action as the Item List or Shader Tree viewports, but the functions it provides are essential to working with vertex maps. All the attributes of the viewport are covered in the Vertex Map List viewport page of the documentation.

TIP: Vertex Maps are *item layer specific*, with each mesh item having its own dedicated maps. The only maps visible within the Lists viewport are for layers that are likewise visible (Controlled by the Item List). While different layers can share map names (and usually do if users stick with the default names modo generates), this can cause collision problems, producing incorrect results when baking out texture maps for rendering or export.

Managing Vertex Maps



Before adding a map to a mesh, users need to first determine the type of map required--

Map Type	Description
Weight Map	Weight Strength Values stored for Falloff and Texturing purposes
SubD Weight Map	Weight Strength Value influences edge creasing in SubDivision Surface geometry
UV Map	UV maps translates 3D vertex positions to flat 2D coordinates
Relative Morph Map	Vertex position offset, relative to the base vertex position
Absolute Morph Map	Vertex position offset to specific absolute position in 3D space
RGB Map	Vertex Color map defined by three R, G and B color values
RGBA Map	Vertex Color map defined by three R, G and B color and an additional Alpha value
Pick Map	Like a Selection Set, defines groupings of vertices
Vertex Normal Map	Surface Normal direction (Smoothing) values stored as fixed value
Edge Pick Map	Like a Selection Set, defines groupings of edges
Particle Size Map	Determines scale values for individual particles
Particle Dissolve Map	Determines transparency values for indvidual particles
Transform Map	Determines transform amounts for individual vertices/ particles

Creating a new map can be accomplished from the menu bar "Vertex Map > Create...". This opens the 'Create New Vertex Map' dialog where users can select the map type, define a name for the map, and set a base value. LMB+clicking OK creates the map for the currently selected mesh item layer(s), with the named map immediately available within the appropriate section of the Lists viewport. Users may also select any of the

'(new map)' options available directly in the Lists viewport under each map type, in turn opening the 'Create New Vertex Map' dialog with a pre selected map type defined by which section the map was created under.

Renaming of maps can be accomplished by either RMB+clicking on a map name and choosing 'Rename' from the popup context menu, or by selecting a map within the list, so it is highlighted, and choosing the menu bar command "Vertex Map > Name...". This opens a dialog box where users can enter the new name, making sure to choose the proper map type (unnecessary when using the rename option in the list). Additionally, users may also use the LMB+click pause click method to rename the layer inline. Removing maps can be done by selecting the target map and choosing the manu bar command "Vertex Maps > Delete". User may also simply select the layer and press the 'Delete' key.

Viewing and Editing Vertex Maps

How users go about displaying and editing Vertex Maps depends on the map type. For most types of maps, this is done directly in the 3D viewports, providing real time feedback to changes in value.

Weight Maps--

Weight Maps display directly in any 3D viewport, when the viewport display option are set as type 'Vertex Map'. Maps can be edited by selecting components and applying values, or using the 'Weight Tool' function. Combined with a 'Airbrush' falloff, interactive painting' of weight values can be accomplished.

UV Maps--

UV maps have their own 2D viewport, specific to editing maps of these types. 3D viewports only display the result of the UV map in the 'Texture', Texture Shaded' or 'Advanced OpenGL' modes. For user convenience modo has a dedicated UV editing layout, by default, with additional popup windows available under the "Layout > Windows" option in the menu bar.

Morph Maps--

When a Morph maps is enabled/selected in the Lists viewport, they are visible within any of the 3D viewports in all of the various modes. When animating morph maps, applied as a deformer, the morph itself will only be visible if the 'Enable Deformers' option is set as such within the viewport options (Access able by pressing the 'O' when hovering the mouse pointer over the target viewport). This is the default option for the 'Layout', 'Animation' and 'Render' layouts, but is disabled in the Modeling, Painting and UV view Layouts.

Color Maps--

When a Vertex Color Map is enabled/selected in the Lists viewport, it will display directly in any 3D viewport, when the viewport display option are set as type 'Vertex Map'. Maps can be edited by selecting components and applying values, or using the 'Color Tool' function. Combined with a 'Airbrush' falloff, interactive painting' of vertex color values can be accomplished.

Vertex Normals--

Vertex Normals, when present, override any smoothing settings, as defined by Material item settings in the Shader Tree. Vertex Normals are usually generated in an external application and used when rendering highly triangulated CAd in modo. However, modo offers the option to bake the Material Items smoothing results into a vertex Normal maps with the 'Set Vertex Normals' Command.

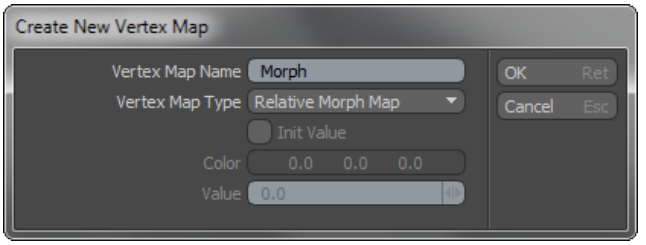
Vertex Mapping

Vertex Maps provide indispensable functions related to modeling, shading and rendering. The tools and commands modo provides are useful for generating a variety of maps and editing the results to fit the users particular need. The utility of each is covered further on the 'Working with Vertex Maps' page. For info on a specific function, click on the options below to be directed to the appropriate documentation page.

Vertex Map Commands and Tools			
Create...	Set Vertex...	Smooth Tool	Color Tool
Name...	Clamp...	Apply Morph...	Weight Tool
Copy...	Quantize...	BG Morph...	Edge Weight Tool
Paste...	Jitter...	Math...	Set Vertex Normals
Clear...	Unify...	Math Tool	Morph Tool
Delete	Cull...	Illumination Baking	BG Morph Tool
Set Value...	Transfer...	Convexity Baking	Show Vert Normal

Create New Vertex Map

This command lets users create a new vertex map, provide a unique name for the map and set an initial value. Once created, the named map immediately appears within the appropriate section of the Lists viewport.



Vertex Map Name: Users can enter a unique identifying name for the Vertex Map here.

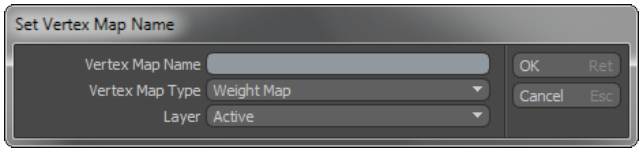
Vertex Map Type: Users can choose the type of map to create from the various options--

Map Type	Description
Weight Map	Weight Strength Values stored for Falloff and Texturing purposes
SubD Weight Map	Weight Strength Value influences creasing values in SubDivsion Surface geometry
UV Map	UV maps translates 3D vertex positions to flat 2D coordinates
Relative Morph Map	Vertex position offset, relative to the base vertex position
Absolute Morph Map	Vertex position offset to specific absolute position in 3D space
RGB Map	Vertex Color map defined by three R, G and B color values
RGBA Map	Vertex Color map defined by three R, G and B color and an additional Alpha value
Pick Map	Like a Selection Set, defines groupings of vertices
Vertex Normal Map	Surface Normal direction (Smoothing) values stored as fixed value
Edge Pick Map	Like a Selection Set, defines groupings of edges

Init Value: This toggle option provides user the means to define a initial value for the newly created vertex map. For Vertex Color maps, this would be a 'Color' RGB value, or for Weight maps, this would be a single numerical 'Value'.

Set Vertex Map Name

This option allows users to easily rename Vertex maps. Users simply need to select the target Vertex map, and choose the menu bar command "Name", where a dialog box will open providing several optiosn for users.



Vertex Map Name: Users can enter a unique identifying name for the Vertex Map here, this name will replace the current name for selected (*highlighted*) vertex maps.

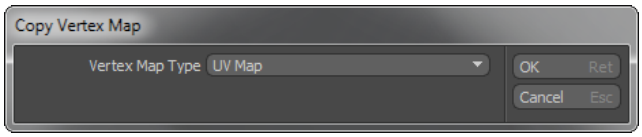
Vertex Map Type: When renaming a vertex map, modo automatically chooses the appropriate map type based on the map being renamed.

Layer: Users can choose wether to rename only the foreground items Vertex Map when set to 'Active', or the background items when set to 'Inactive'. Choosing the 'Both' option renames the current vertex map for all visible items (Only when layer Items share identical map names).

Copy

The 'Copy' command, when used in conjunction with the 'Paste' command, allows users to take the information from one vertex map and apply it to another.

You can access the Copy and Paste commands from the menu bar command "Vertex Map > Copy/Paste", or by selecting the map in the Vertex Map List, right-clicking on the map and choosing Copy. Select a different map in the same list, right-click again and choose Paste to complete the operation.



Vertex Map Type: Users should make sure when copying values with the menu bar

command "Vertex Map > Copy" to ensure that the proper vertex map type is defined for proper results.

Paste

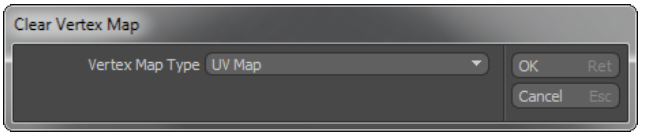
The 'Paste' command, when used in conjunction with the 'Copy' command, allows users to take the information from one vertex map and apply it to another.

You can access the Copy and Paste commands from the menu bar command "Vertex Map > Copy/Paste", or by selecting the map in the Vertex Map List, right-clicking on the map and choosing Copy. Select a different map in the same list, right-click again and choose Paste to complete the operation.

Vertex Map Type: Users should make sure when copying values with the menu bar command "Vertex Map > Copy" to ensure that the proper vertex map type is defined for proper results.

Clear

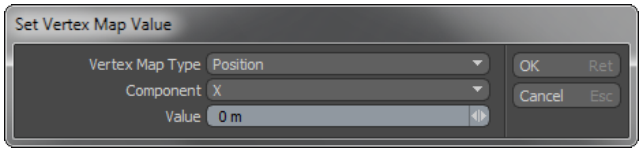
"Vertex Map > Clear" resets the values of the vertex map edits. If you make changes to your weight map, for example, then press Clear, those values are discarded and the weight map is returned to its initial state. Clear can be accessed on the Vertex Map menu by choosing Clear. You can also fire the Clear command by right-clicking on the Map name in the Vertex Map List and choosing Clear.



Delete

To completely remove a vertex map from the Vertex Map List and the current modo session, select the map in the list, then use the Delete command. You can right-click the entry in the list and select Delete, or from the menu bar choose Vertex Map > Delete.

Set Value



The Set Value tool is used to quickly and accurately set the value of a particular vertex map, or a group of selected vertex maps. Since much of the data in a 3D model is stored as a vertex map, this is quite a powerful tool. While vertex maps are typically referred to when discussing things such as morph maps, weight maps, or UV data, vertex maps are also used for storing the basic position information for all vertices in a mesh. When using Set Value, the dialog changes to reflect current conditions. So if there are no selected RGBA maps, then there is no RGBA choice in the popup. Also the component popup changes to match the type of the vertex map which has been selected.

To use Set Value, simply use the "Type" popup to choose the desired vertex map class. The contents of this pop-up may change depending on what types of vertex maps are active in the currently selected data. Now you can use the lower pop-up to choose the setting you wish to change. Finally, enter the desired value for the vertex map.

With this tool you can manage many different aspects of your model data quickly. For example, you could use this function to quickly set the UV coordinates to U = 0 for an entire set of selected vertices so that they snap to an edge. Or you could use the command to quickly set the selected vertices weight map to 100%. You could even use this tool to set all of the selected vertices to an absolute location such as X= 0.

Set Vertex

The Set Vertex command lets you define precise values for individual vertices in a vertex map. The numeric fields on the Set Vertex Value dialog include these options:

Component: Some vertex maps have multiple components. For example, a UV vertex map has two components U and V, an RGB map has three components R, G, and B while a Weight map has only a single component, the weights value. When using the Set Vertex command on a vertex map with multiple components it is important

to be able to choose which component will be set by the command. Using the Set Vertex command on a UV map with the component set to 0 will set the value on the U component of the map. Using a value of 1 in the component field will set the V value.

Index: Specifies which vertex is to be affected. If you turn on Show Indices under Visibility Options you can see the vertex indices in the viewport.

Value: Determines the numeric vertex value.

Clamp

Vertex Map Clamp command limits the values in a vertex map based on a Cutoff value.

Quantize

Vertex Map Quantize is a tool that reduces the vertex map selection values mathematically. It evaluates all the selected components and simplifies them based on the Amount values set in the Quantize Vertex Map floater. Increasing the Amount value will result in fewer diverse values in the Vertex Map. Quantize can be found on the Vertex Map menu.

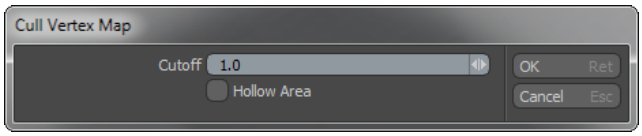
Jitter

Found on the Vertex Map menu, Vertex Map Jitter is a tool that procedurally affects selected vertex map values based on an Amount numeric field. This option provides soem randomization to the placement of the vertices in the vertex map.

Unify

A discontinuous vertex map is one where a single vertex has multiple UV map values. Vertex Map Unify merges continuous and discontinuous vertex map values if their difference is below the threshold value given. The difference is computed from the largest component value difference (as opposed to the difference of the magnitude of the entire vector). One way to visualize this operation is to make some discontinuous UVs that are very close to their continuous alter egos, then use the command to close the seam.

Cull

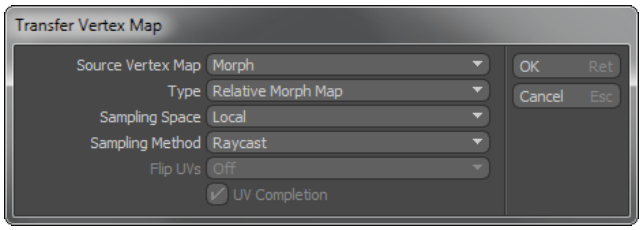


The Vertex Map Cull command removes extra vertex map values. It clears the values below the Cutoff setting. If Hollow Area is set, then only values that have non-culled neighbors are removed. It will limit culling to 'isolated' values, or values which are below the threshold and are not connected to any vertices with values above the threshold.

Transfer Vertex Map

modo has the ability to transfer any vertex map from one item to another arbitrary item. In a way, this is just like baking a texture map, but working with vertex data instead. This makes it easy to take a series of morph maps for different facial expressions and transfer them to another similar object. That's the key too; the objects do need to be similar for the best results. For the above result, I modified the old man figure using modos sculpting tools to loosely resemble the ogres facial size and position of features, matching lips to lips and brow to brow and so on. This gave me more predictable results as corresponding vertices were closer together.

To use, you must first have base vertex maps of the proper type present in both items before using the transfer function. Keep the item you want to transfer from as a background item (visible, but not selected); in the forground you have the item you're transfering to with the appropriate vertex map selected in the vmap list. Then select the command 'Transfer Vertex Map' from the menu bar under the 'Vertex Map' menu. This will open the Transfer tools dialog box.



Source Vertex Map: Here you select the appropriate vertex map from the background item that you are transfering the map from. Only the types of maps that correspond to the 'Type' will appear. If you have no corresponding maps associated with the foreground object, this item will not allow you to select a map.

Type: Choose from the various types fo Vertex Maps that modo supports. Both maps must be identical in function, it is not possible to transfer, say for instance, a color vertex map to a morph map.

Sampling Space: This is the coordinate system to sample positions. When "local" is set, the command samples position in the target item space, otherwise it samples in world space.

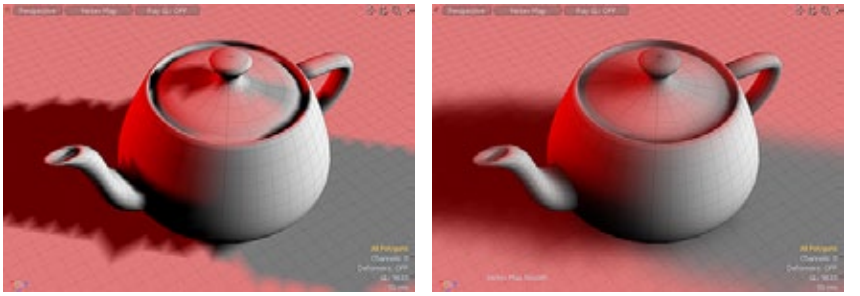
Sampling Method: The 'Sampling Method' allows the users to specify the method to sample position on the target mesh item from the source position. "Raycast" computes the hit position along the normal vector at the source position on the source mesh item. "Distance" finds the closest position on the target mesh item from the source position.

Flip UVs: The 'Flip UVs' option reverses the sampled UV values to horizontal or vertical. This is useful when making the mirrored UV polygons.

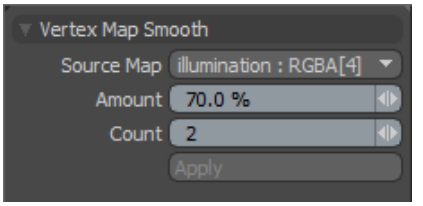
UV Completion: When the "UV completion" option is enabled, the command generates discontinuous UV polygons based on the sampling polygon information where each position is sampled from the different polygons over UV seams. Basically this should always be on, but there are rare instances where one would want it disabled, so the option is available should you find it necessary.

Smooth Vertex Map

modo provides a vertex map smoothing tool useful in softening/blurring the values in a vertex map. This is useful when applied to Weight Maps as well as Vertex Color maps. When applied to a Vertex Color map generated with illumination baking, this can produce a look similar to global illumination, but applicable to real time uses such as in games.



To apply, users must first have a base vertex maps of the proper type present, either a 'Weight Map' or a 'Vertex Color Map'. The smooth tool has no affect on either UV or Morph maps. Next, users need to make sure the viewport is set to the 'Vertex Map' display mode, with the target map selected, so that the affect of interactive editing is visible. Then activate the tool found in the menu bar under "Vertex Map > Smooth Tool". In the tools properties panel, select the named map to edit in the 'Source Map' and adjust the size of the blur with the 'Count' option. Hauling (LMB+click and dragging) in the viewport will interactively adjust the strength of the blur with the 'Amount' value. Simply drop the tool by pressing the 'Q' key to accept the changes when finished.



Vertex Map Smooth

Source Map: The 'Source Map' popup menu defines the target vertex map for editing.

Amount: The 'Amount' setting adjusts the overall strength of the blurring affect on the map.

Count: The 'Count' option defines the size of the blur, across the number of adjacent vertices. A value of '1' will average one adjacent vertices with each vertex, a value of '2' averages two vertices and so on.

Apply Morph

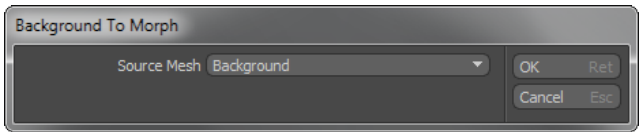
The 'Apply Morph' command allows users to apply an existing morph map into a new one, or directly onto the base mesh, often necessary to export a mesh to an external application. The command is found in the menu bar "Vertex Map > Apply Morph..." By selecting a named 'Morph' from the pop-up and defining a strength 'Amount', the map will move the vertices of the currently active map based on the values of the selected morph.

Morph: This pop-up choice reveals all available morph maps currently available in the active mesh. Simply click on the pop-up and select the desired map from the list.

Amount: This percentage entry allows you to set an exact numeric amount of the currently selected morph map to be applied. You can achieve more dramatic results by driving the value over 100% or completely different results by taking the value to a negative number.

Background to Morph

The 'Background to Morph' function can take geometry in a background layer and applies its vertex positions to the foreground layer as a morph map. For the best results, both the Target and Source meshes should have the same number of vertices in the same winding order. This function works great to combine a multi-mesh morph from an external app, into modo's native single mesh with a morph deformer. The single control allows users to choose the source mesh to base the morph off of. Users will need to *create* and *select* a Morph map before invoking the tool.



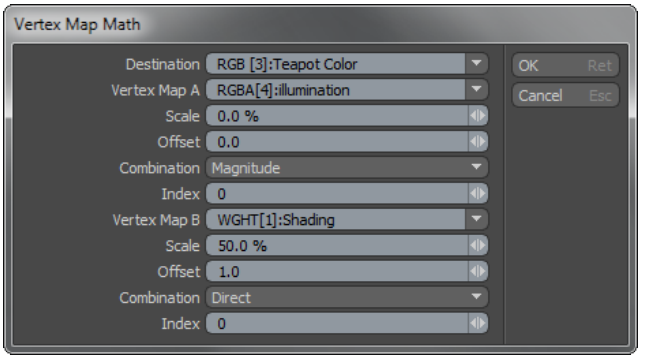
Source Mesh: This pop-up option allows users to choose either 'Background' where the visible background layer will be define the vertex positions for the morph map, or the layer directly below the active layer in the case of multiple backgrounds item layers. For 'Last Selected', modo takes the last selected item layer in the history stack into account when selecting the source.

Vertex Map Math

Vertex Map Math function is a generalized command that operates on multiple vertex maps. You can use it to do things like scaling values, composing maps in different ways and converting map types. Examples of things you do with this tool include: generating a weight map from a U or V coordinate; making a weight map from the magnitude of a morph map. You can also create a map with the max, min, sum or difference of two

maps, or clear one map's entries based on the values in another map.

The math tool makes a formula for the destination (assuming 'Amount' is 100% like: **dest = ((sourceA * scaleA) + offsetA) Operation ((sourceB * scaleB) + offsetB)** with Operation being '+", "-", etc.



The attributes for the Vertex Map Math Tool are as follows:

Destination: The name of the destination vertex map. The map will be created if needed

Amount: The amount (%) of the computed value that will go into the dest, blended like alpha

Combine: Specifies the math operation used to combine the terms (SourceA, SourceB)

Then for each source (#):

map# - the vmap index

scale# - a multiplier to apply to the offset vmap value: src# = scale# * (val# + offset#)

offset# - an amount to add to the vmap value

mode# - the way to deal with multi-dimensional vmap values. if src# is morf and dest is uv:

"Direct" - copy direct component to component, so morf.x->uv.u, morf.y->uv.v

"Component" - pick a single component (index#) so morf.index#->uv.u, morf.index#->uv.v

"Magnitude" - The 'vector length' of the source vmap value

"Maximum","Minimum" - The max. or min. of the components of the source vector

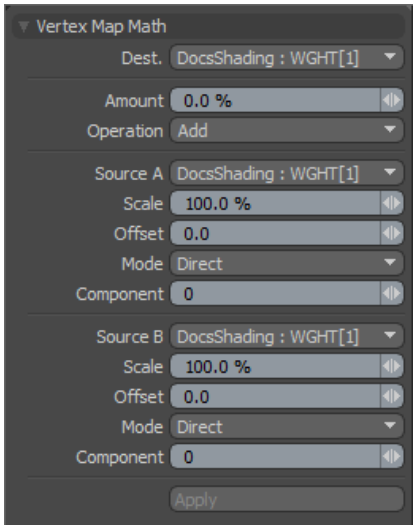
index# - the component of the source map that will be used

This tool can be used to make weight maps of morph magnitude, go between weights and UVs, invert vmaps, etc. The destination map can also be a/the source. If no map is specified, the value is 0, but the offset and scale are applied, so a constant value term is available.

Vertex Map Math Tool

Vertex Map Math Tool is a generalized tool that operates on multiple vertex maps. You can use it to do things like scaling values, composing maps in different ways and converting map types. Examples of things you do with this tool include: generating a weight map from a U or V coordinate; making a weight map from the magnitude of a morph map. You can also create a map with the max, min, sum or difference of two maps, or clear one map's entries based on the values in another map.

The math tool makes a formula for the destination (assuming 'Amount' is 100% like: **dest = ((sourceA * scaleA) + offsetA) Operation ((sourceB * scaleB) + offsetB)** with Operation being '+", "-", etc.



The attributes for the Vertex Map Math Tool are as follows:

Destination: The name of the destination vertex map. The map will be created if needed

Amount: The amount (%) of the computed value that will go into the dest, blended like alpha

Combine: Specifies the math operation used to combine the terms (SourceA, SourceB)

Then for each source (#):

map# - the vmap index

scale# - a multiplier to apply to the offset vmap value: src# = scale# * (val# + offset#)

offset# - an amount to add to the vmap value

mode# - the way to deal with multi-dimensional vmap values. if src# is morf and dest is uv:

"Direct" - copy direct component to component, so morf.x->uv.u, morf.y->uv.v

"Component" - pick a single component (index#) so morf.index#->uv.u, morf. index#->uv.v

"Magnitude" - The 'vector length' of the source vmap value

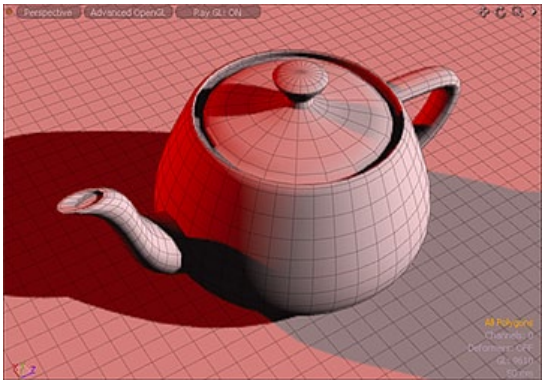
"Maximum","Minimum" - The max. or min. of the components of the source vector

index# - the component of the source map that will be used

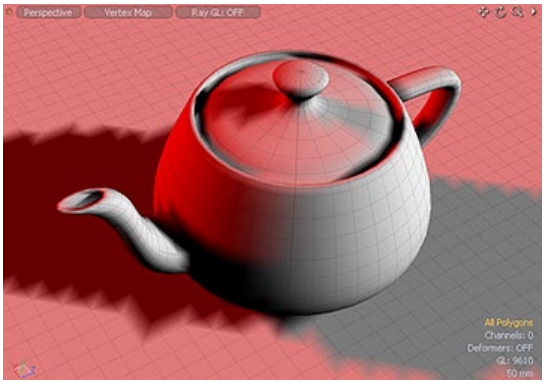
This tool can be used to make weight maps of morph magnitude, go between weights and UVs, invert vmaps, etc. The destination map can also be a/the source. If no map is specified, the value is 0, but the offset and scale are applied, so a constant value term is available.

Vertex Illumination Baking

The 'Vertex Illumination Baking' function provides a means for users to bake or burn-in the illumination colors of the scene directly to the vertices of the geometry as a Vertex Color Map. The results of the baking is a Vertex Color Map named *illumination* that will be immediately available in the Lists viewport. To produce a result, users only need to run the command, located in the menu bar under the "Vertex Map > Vertex Illumination Baking".



RayGL in-viewport render preview



Resulting baked Vertex Color Map

To view any Vertex Color Map, users need to set the viewport display option to 'Vertex Map' and then choose the named map from the 'Lists' viewport. To generate multiple maps, it is necessary for users to rename any previously generated map(s) before executing the command again, as subsequent baked maps will overwrite previous value to existing maps named 'illumination'.

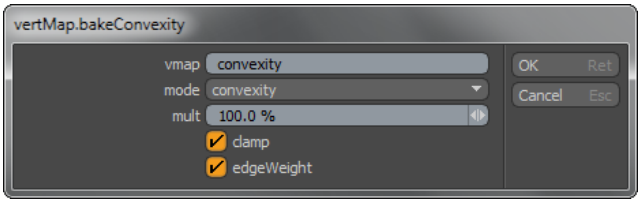
For best results, users need to keep in mind that the illumination colors are baked in to the individual vertices themselves, so geometry needs to be present for effect to work properly. To easily visualize the Illumination baking, users can enable the viewport RayGL rendering option, providing an in-viewport reference of the results. When the command is invoked, modo samples the color at each vertex point in 3D space,

including cast shadows and the effects of global illumination, and applies the color at that particular point to the vertex. With appropriately modeled geometry, this can produce strikingly realistic results, that take virtually zero time to render. Many real-time gaming engines then utilize these maps to simulate lighting in an environment without the additional overhead of actually calculating the lighting per frame. Depending on the particular lighting need, and the desired effect, users may also wish to apply 'Vertex Smoothing' to the Vertex Illumination result providing a softer, less chunky result.

TIP: Game engines likely display vertex color maps differently than modo does. It is therefore recommended users test geometry in their target application to ensure the expected results.

Vertex Convexity Baking

The 'Vertex Convexity Baking' function provides a means for users to generate vertex Weight Maps based on the Concavity or Convexity of of a surface. Surfaces that are Concave are the interior of a curve and those that are convex are on the exterior of a curve (think cave as being inside). The resulting maps produced by this command could be used as mask layers in the Shader Tree, when used in conjunction with the Weight Maps Texture layer. Users may also find it useful to convert the map to another type, such as a 'Vertex Color Map', by using the Vertex Math Tool.



Vmap: This input filed allows users to name the resulting vertex map.

Mode: Users can choose the type of map generated-

Convexity-- Outer areas of surfaces, especially curved.

Concavity-- Inner recessed areas of surfaces.

Both-- Generates map for both Convex and Concave surfaces.

Mult: This is a multiplier value to control the strength of the final result. If map seem too weak or strong, modifying this value will proportionally adjust all values in the map.

Clamp: This toggle, when enabled, clamps all values in the generated map to greater than 0, so there are no negative weight values.

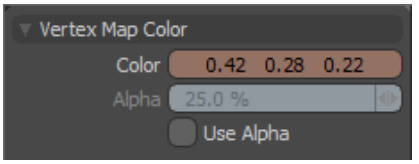
Edge Weight: Strengthens edge values.

To view any of the resulting Weight Maps, users need to set the viewport display option to 'Vertex Map' and then choose the named map from the 'Lists' viewport. When generating multiple maps, users should ensure to enter a unique name for each instance, as identical named maps will be overwritten by subsequent applications of the function.

TIP: Game engines likely display vertex color maps differently than modo does. It is therefore recommended users test geometry in their target application to ensure the expected results.

Vertex Color Tool

The Vertex Color Tool allows modo to assign RGB values to a vertex map for vertex map shading effects. The tool requires a vertex map of type RGB or RGBA be created and selected in the vertex map list prior to editing. The application of the tool is done on a selection basis with hauling in the viewport controlling the Alpha amount if enabled. Users can interactively apply color to a map using the Airbrush falloff option, providing a means to directly *paint* on color.



Vertex Color Map

Color: Users can define a RGB color value.

Alpha: When 'Use Alpha' is enabled, this value determines the amount of transparency.

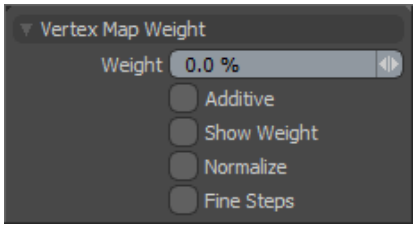
Use Alpha: This toggle option provides a means to apply an Alpha transparency value in additional to color values with the tool.

Vertex Weight Tool

The VertexThe Weight Adjust Tool lets users adjust the Vertex Map Weight percentage. This gives users control over edge weighting values. Weight maps are discontinuous 1D Vertex maps holding weight percentage values. These vertex maps can be used to modulate the strength of deformers when applied to animated characters, controlling the strength of tools when combined with a weight falloff operator and as a way to control edge creasing for Subdivsion Surface geometry. When a 3D viewport is set to a

Vertex Map shaded mode, weight shading (hot to cold=red>green>blue) is visible in the viewport.

In order to use the Weight Tool, user will first need to select the appropriate Map in the Vertex Map lists viewport. Adjustments will only apply to selected vertices.



Vertex Map Weight--

Weight: This value input field displays the currently applied Weight value. Users can adjust this interactively by hauling in the 3D viewport with the tool active, receiving real-time feedback of the adjusted values. Users amy also enter a Weight Value into the input field and simply LMB+click in the viewport to apply the value to the selected vertices.

Additive: When this option is *enabled*, Weight Value adjustments will be additive to those preexisting on the map. When *disabled*, values will replace those previously in the map.

Show Weight: When *enabled*, displays numeric readout of Weight Values for each vertex position.

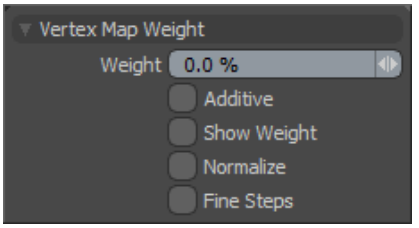
Normalize: When *enabled*, will normalize (equalize) weights between multiple weight maps assigned to bones.

Fine Steps: When *enabled*, controls the strength of the application of the weight tool allowing finer, more precise adjustments.

TIP: The Weight Tool plays nicely with all the Falloff types. Try using the Linear Falloff with the Weight Tool to blend your creases into the rest of the model.

Edge Weight Tool

The Edge Weight tool allows you to add creasing to subdivision surfaces without adding additional geometry by creating a "weight" on various edges. To use the tool simply select an edge or edges and click on the Edge Weight Tool button in the Vertex Map menu and then click and drag in the 3D viewport. Setting the value to 100% will create a completely hard edge whereas a value of 0% allows the default interpolation of the limit surface through the control vertices. It is also possible in modo to set edge weights to negative values which will push the limit surface away from the control vertices.



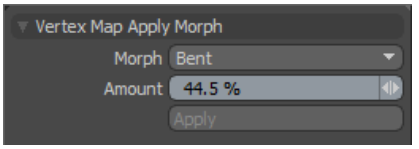
This tool is actually using the generic "Weight Tool" but first selects the Subdivision Vertex Map so that the appropriate vertex map is adjusted when using the tool.

Set Vertex Normals

modo has the ability to *freeze* the smoothing of a geometric surface by baking out the values into a vertex normal map. This smoothing value can be set in the 'Material -Reflective' properties of a surfaces material layer prior to baking. To create, simply select the item you wish to create the vertex map for and use the command 'Set Vertex Map Normals', under the Vertex Map menu of the menu bar. Once generated, the vertex map, called "vert_normals", can then be found in the section 'Other Maps' of the vertex map list. To see the result of the map, enable the 'Show Vertex Normals' function.

Important: When a vertex normal map is associated with a mesh item, modifying the smoothing value will no longer affect the actual smoothing of the surface as vertex normal maps supersede any other settings. To modify the values, you'll need to remove the map and generate a new one with the new values.

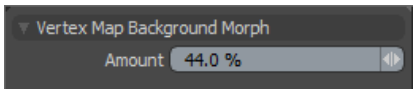
Morph Tool



This tool can be used to make weight maps of morph magnitude, go between weights and UVs, invert vmaps, etc. The destination map can also be a/the source. If no map is specified, the value is 0, but the offset and scale are applied, so a constant value term is available.

Background to Morph Tool

Using the background layer as the target shape, the Background Morph tool moves the vertices of the mesh layer toward those of the background geometry layer. The tool is very useful when combined with various falloff modifiers.



The single control is a percentage that determines how closely the foreground mesh should match the vertices in the background layer. This value can also be set to negative values or beyond 100% producing interesting effects.

Show Vertex Normals

The 'Show Vertex Normals' command displays small lines that represent the vertex normal facing directions for each vertex in a Vertex Normal map. Split lines represent creased hard edges, where single lines represent smoothed edges. USers must have a vertex map present to see anything using this commnd. Users can create a map based on the objects current smoothing using the 'Set Vertex Normals' command. The resulting map will appear in the vertex map lists viewport.

Working with UV Maps

Many users have a fear UV mapping, assuming a degree in black magic or voodoo is necessary to produce useful results. Fortunately, nothing is further from the truth. The basic concept is easy; UV mapping is simply the process of translating a 3D surface with volume and shape onto a flat 2D texture image. An easy way to visualize how that works is to imagine the object was accurately wrapped in gift wrap, and the UV map is the careful unwrapping of the object pressed flat, much like a geographical map of the earth, which suffers from the same constraints as a UV map. While it might be easy enough to understand, it's the actual execution of the process that requires a little science and whole lot of *art*, but like modeling, or texturing or even rendering, understanding the tools is an important part of understanding the process. modo provides an flexible range of functions that simplifies the process, making what many consider a tedious task a lot less painful, and possibly even *enjoyable*. As they say, practice makes *perfect*.

Understanding UVs

UV map values are simply position coordinate on a 2D texture stored within an associated vertex. In modo, this is a float value, meaning it is a number defined by a decimal point, chosen for the precision it provides. The UV Map viewport plots those 2D position values on a grid, including the edges and polygons, so its easier to reference how the 2D geometry map relates to its 3D counterpart. These values are typically confined to the 0-1 area of the grid as this is the normal space (*or home*) position of the texture; any texture applied using a UV map will always fill the entire 0-1 space; any part outside of this area will simply repeat the texture (how it gets tiled is controlled by the Texture Locator). By confining the map values within this single quadrant, it is easy for users to control positioning, avoiding any overlapping if necessary, as this causes

errors when *baking* textures. By scaling UV values *larger* than the 0-1 space, user can precisely control the *tiling* of the image on the surface. There are no specific contraints regarding the position of geometry within UV space.

One major point of confusion is the shape of the UV editing viewport and how it relates to the image aspect ratio. The UV window is square, so naturally users assume UV mapped images need to be square as well, but this is not the case at all. Images can be any size, any aspect ratio and any resolution, it will still always map to the 0-1 space of the UV grid. For very wide image or very tall, the values are positioned *proportionately* across the texture, which is why it can be advantageous to think of the values as *percentages*, a float value of .5 is effectively the 50% position, and will position a vertex directly in the center of the image, regardless of the images size or shape. It is that fact alone that makes UV mapping so precise and valuable. When displaying non-square images in the UV viewport, the image will simply be distorted to fill the 0-1 space. Users can also enable the 'Use Aspect Ratio' in the viewports options to distord the UV space view, in turn displaying the image aspect properly.

Making UV maps

For many instances, the texture projection modes of the 'Texture Locator' provide plenty of control for applying texture layers to a surface. UV maps are only necessary for surfaces that require *high precision* texture mapping such as for organic characters, applying a logo to a bottle in a specific way, or if the model has a peculiar shape. They are also a requirement of any real-time game engine and object to object baking. If there is no intention of applying a *texture* layer to your surface, then a UV map is unnecessary.

Generating and editing maps take place in the UV editing tab of the main (*default*) interface. This is where the dedicated UV editing viewport is located along with quick access to the most often used tools and commands. When working in the UV viewport, virtually all the same tools, modes and functions used in the 3D viewports apply; transform tools, modeling tools, even sculpting tools can be utilized to edit maps. Most operate the same as they do in any other viewport, but some tools will behave slightly differently. For instance, regardless of the geometric topology, function like 'Select Connected' will only select the connected area of a UV island when applied to the UV view, modo tools are just smart that way

To start making UVs, user would need to first select or define a vertex map to store values into. UVs are just another type of vertex map, managed within the Lists viewport whose use is covered on the Vertex Maps page of the documentation. Once the proper map type is selected in the Lists viewport, users can select the polygons they wish to map and invoke any of the tools or commands. For interactive tools such as Unwrap, users can haul directly in the 3D viewport, and see real time updates of the UV map generation in the UV viewport.

The tool or process used to produce the map is largely dependent on the shape of the surface. Geometric shapes, like a can of soda, or a box, will do best with the Projection tool, which generates UV values using options similar to those found in the texture locator, such as 'Spherical', and 'Planar'. For organic shapes, the Unwrap tool can produce great results right away, making what was once a tedious process, easy. Long, winding object (like a belt) can leverage the Peeler tools ability to produce precise maps with uniform spacing. Often times, the result produced by a given tool will be adequate and users can move to other tasks. Other times, the result will simply provide a starting point for further refinement. This can be the case for the default values generated by modo's *primitive* geometry. Depending on application, the default map may not be appropriate or optimum for the intended use. Lets take a look at a basic example--

As illustrated here, the layout of the map itself is largely dependant on how the geometry will be used, and there's no reason why users shouldn't set things up to make surfacing easy. It should also be comforting to know, there are no set rules to follow for layouts. Users can position UV *islands*, (grouped areas of geometry in a UV map) anywhere they wish in the UV map. It can help to keep in mind some guidelines, these are not hard and fast rules, but can be helpful to consider when creating a map.

Minimize Seams

Seams are the edges, or breaks in geometry between UV islands and are inevitable for nearly every model. Sometimes they can cause issue with painting or image based

Sculpting. If at all possible, its best to define seams in inconspicuous places. For human characters, this can be the inseam of the legs and thighs, under the arms, and behind the head and down the back. When viewed form the front, most of these seams would be hidden from view. When seams can't be hidden from view, image commands like 'Expand UV Border' can be useful in minimizing their appearance. It might seem obvious, but it should be said that having as few islands as necessary will reduce seams, so keep them to the minimum.

Managing Distortions

Two main goals when UV mapping are to eliminate or reduce seams, and to also minimize distortion from stretching. Unfortunately, these two are not mutually exclusive. Distortion is a fact of life with UV mapping. Completely eliminating distortion would require an increase in UV seams. Fortunately, modo has a number of tools and functions to help users manage distortion. First off, users should enable the 'Show Distortions' option of the UV view, this produces a color overlay on the geometry that signifys the amount a polygon is distorted, as to say scaled or stretched in relation to neighboring polygons. Polygons that are the closest relative size to others in the map with the least distortion will display a middle gray/green color. Distorted and smaller relative polygons will fade toward red, while distorted and larger relative polygons will display blue. Users can also display the distortion colors in the 3D viewport by selecting the 'Vertex Map' viewport style and then selecting the *UV Distortion* named vertex map generated under the 'Other Maps' heading of the vertex map list viewport. Once users can see where the problem areas are, they can take action to minimize issues. The UV Relax tool is the number one option for reducing distortions. Users can use it to massage a map, exactly as the namesake suggest, by *relaxing* the data, tugging and pulling neighboring polygons until a comfortable medium can be acheived. Users can limit the application of the relax using falloffs as well.

Consistent Texture Density

For the most part, models will look their best when an image maps *pixel density* is spread evenly across the surface. Take for instance a sphere, an image map would need to be twice as wide as it is high for even pixel density, because the circumference is twice the distance of how far the image wraps from top to bottom. This also means to compact UV islands as closely as possible to get the most out of the number of pixels in the image, large gaps between islands waste precious texture space. Quite often, professionals will map a temporary grid or checkerboard pattern onto the object to get direct visual feedback of how consistent the textures density is. There is a *UV Check* image included in the content download in the "Materials > Miscellaneous" folder.

Take the Image Map into Account

It's important to consider the texture map(s) that will be applied to the surface using the UV map. What will its size and aspect ratio be? Say if you were UV mapping a rectangular sign that was much wider than it was high. It would be far easier to make the final image the proper size and aspect ratio, especially if it needed further editing down the road (as opposed to artificially squeezing it in an image editor). In both these cases the geometry would appear squashed while displayed in the square UV viewport, but it will render exactly as expected.

Is Orientation Important

In most cases, orientation of individual islands is arbitrary, but in some cases it can become important. For example, when texturing a face directly in 3D, the paint tools take care of projecting the pixels into the image properly, so the UVs position within the map is inconsequential. But, if it becomes necessary to make further edits say in Photoshop, it will be far easier to accomplish when the face is straight up and down instead of at some odd angle.

Overlapping UVs

Overlapping UVs aren't always a bad thing. There are some instances when overlaps are OK such as when each overlapping part has a different material tag; this simplifies the model by reducing the total number of vertex maps. Barycentric maps (where each individual polygons is mapped to the entire 0-1 UV space) are useful in controlling the tiling of images. While texture memory in games continues to go up, it was a good idea to model and texture half of a character and then mirror it (making UV overlaps) effectively doubling the texture resolution, while halving the image size. When baking textures out, overlapping UVs will cause errors, so in those cases, avoid them.

Get to know the tools

modo has a huge toolbox, especially in the vertex mapping arena, and if you aren't at lease familiar with what it has to offer, you might be limiting yourself to what can be accomplished. This documentation is a great way to familiarize yourself with what modo has to offer, providing you the best leverage to get out of difficult situations. The functional operation of the individual tools is covered within each tools respective documentation page.

Applying the Map

UV maps are mainly employed when applying textures, so once the map has been created, it will first show up in the Vertex Maps window under the UV Maps section.

To actually utilize the map itself, one would need to first apply a texture layer, either as an 'Image Map' or a 'Procedural' texture. Once applied, users can access the associated 'Texture Locator' from the 'Item List', or the 'Shader Tree' (by LMB+clicking the ' + ' icon preceding the layer name). Within the locators properties, there is a setting 'Projection Type', when choosing the option 'UV Map', users may select the specific map from the selector below. modo will try to automatically select the proper map when applying an image map to any geometry that contains a UV, but it may be necessary at times to specify alternate maps when modo chooses incorrectly.

UV maps are also useful in the application of Anisotropic shading (which can be further controlled by an image map). When a map is applied, the 'U' horizontal direction (of U and V) defines the length that the anisotropic shading will follow. The particular map can be specified whenever the 'Anisotropy' setting of the Material item is set above 0%.

Lastly, UV maps can be leveraged to simplify certain modeling tasks. Often times it is easier to apply a UV maps to geometry, acting as a visual aid, the map can then be converted to a vertex map of another type, such as a weight map and applied as a falloff. One quick example is the possibility of using the Peeler option on a spiral. This creates a linear type uv that follows the contours of the form. Converting this to a weight map using the Math tool. Then applying the resulting weight as a falloff to the Push tool, allowing the interactive scaling of the spiral along its length.

TIP: Discontinuous vertices in a UV map are those that are single vertices within the model, but have two locations in a UV map (from a shared edge that are split into separate UV islands) modo highlights discontinuous vertices and edges in the UV viewport as a light lavender color to make it easier to locate the associated vertex or edge.

UV Mapping

UV mapping is an essential part of precision texture mapping. It can also be useful for controlling the application of procedural texture and gradient layers, as well as Anisotropic shading. Many functions and tools of modo generate automatic UV maps, but users will often find need to modify or replace those maps. The *tools* are useful in creating the initial maps, while *operators* modify existing maps. Using the maps in modo is covered on the 'Working with UV maps' page. For info on a specific function, click on the options below to be directed to the appropriate documentation page.

UV Map Tools	UV Map Operators		
Create UV Tool	Align...	Flip	Cut/Copy/Paste
View Projection	Fit...	Sew	Delete
Peeler Tool	Select Border	Move and Sew	Export UV to EPS
Unwrap Tool	Mirror...	Orient Pieces	
Relax Tool	Rotate	Pack...	

Create UV Tool

The 'Create UV' tool is used to generate UV maps and provides several methods and options for generating them. If there are no polygons selected in your mesh, the tool will use all visible foreground polygons to create the UV map. To isolate certain areas of your model simply select the desired polygons prior to using the tool. The Create UV tool works by projecting the polygons from 3D space into the 2D (UV) space. As such, it is very important to choose the most appropriate Projection Type; modo offers several. It should be noted that the 'Create UV' tools is often only a half step to a completed UV map as using this tool will generally result in UVs that need additional manual editing to complete.

UV Maps are stored as discontinuous 2D vertex maps for 0-1 UV coordinate data. UV maps are used to explicitly link vertices in the 3D mesh to locations on a 2D image for high accuracy texture mapping. The UV map can be though of as a method to lay out the 3D mesh so that it lays flat, allowing the 2D image to be applied to the mesh accurately. A UV View is provided to help visualize the UV's over a bitmap image. You can use the UV tab of the default layout to see mesh geometry in the viewport, and the UV map as well.

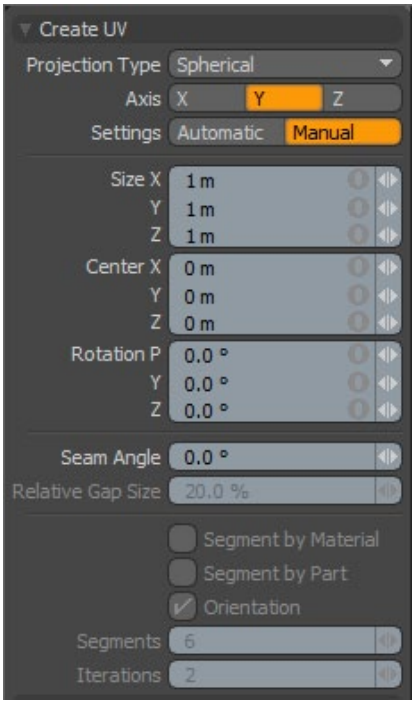
To use the 'Create UV' tool to generate UV's do the following:

1) Create some geometry, and select some polygons in the 3D Model View.

2) Change to the UV view layout.

3) On the UV menu, select the 'UV Projection' tool.

4) Click in the 3D Model View. The selected geometry will generate a UV map which will appear in the UV Texture View.



Create UV

Projection Type

Planar- The Planar mode projects the selected polygons from an orthogonal view determined by the Axis setting. The resulting UV map will actually look quite similar to viewing the model in a 3D viewport set to Top, Front or Side. This mode is very good for quickly assigning areas of a mesh that are all facing the same direction. For more complicated geometry such as a head, Planar will result in a significant amount of UV overlap as the projection will place polygons on both side of the head into the same UV coordinates. Planar Projection would be good for landscapes and some architectural or hard surface models.

Cylindrical- This mode projects the polygons into the UV map based on a virtual cylinder that surrounds the mesh. If you think of the cylinder as a soda can, setting the Axis to Y results in the can sitting on a table whereas setting it to X or Z will lay it on its side. Cylinder can be very useful for mapping specific areas of a model such as arms for organic meshes or tubes and pipe segments for hard surface meshes.

Spherical- Projects the UVs from a virtual sphere that surrounds the mesh. The poles of the sphere are placed based on the Axis setting. The Y axis setting will place the poles up and down where as X or Z puts them side to side or front to back.

Atlas- The Atlas setting uses an multi-faceted projection method to place all polygons into the UV map while maintaining relative scale based on the 3D volume of the polygons. This is designed to minimize any texture distortion from non-uniform UV data. However, the Atlas projection will result in a highly discontinuous

UV map which may only be useful for certain algorithmic uses such as texture baking.

Barycentric- This mode forces each individual polygon to fill the 0 to 1 UV space. In the case of quadrangular polys they will fit perfectly to the outer bounds of the UV area. Triangles and ngons are placed within the UV space as best they can. This is a very useful mode if you have a texture that you want to tile across every polygon in the mesh.

Atlas 2- Similar to Atlas mode Atlas 2 attempts to fill the UV space with all selected polygons in the most efficient way without distorting the relative scale of each polygon from 3D to 2D space.

Cubic- Projects the UVs from the polygons depending on their normal direction segmented by the 6 planar directions -Top,Bottom, Left, Right,Front,Back. Similar in a way to planar, however, it will avoid the common backwards texture errors that can result from planar projections.

Axis: Sets the main axis direction for generating the UV projection.

Automatic/Manual: When set to 'Automatic', modo will set the desired scale of the UV projection in order to best fill the total available UV space based on the selected mesh's bounding box. When set to manual, users can use the **Size/Center/Rotation** settings to define a specific projection that affects the overall size of the final UV map.

Seam Angle: For the Cylindrical and Spherical projections which have a seam, the seam angle specifies at what angle the seam appear in the UV map. This way if your face happens to be facing the direction that creates a seam across the face, you can adjust the seam angle to place in the back of the head without changing the geometry.

Relative Gap Size: For the Atlas mapping types that generate multiple pieces or *islands*, the relative gap size will produce greater distance between individual UV islands

Segment By Material: For the Atlas projection types, enabling 'Segment by Material' will create additional island segments so that individual islands never contain polygons from more than one 'Material' tag.

Segment By Part: For the Atlas projection types, enabling 'Segment by Part' will create additional island segments so that individual islands never contain polygons from more than one 'Part' tag.

Orientation: When used with the Atlas projection type, enabling the 'Orientation' setting will force UV islands to flip based on their axial direction, so for instance if you use this setting on a head, mapping for one side of the head will mirror the other side,

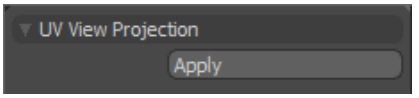
making it easier to distinguish one side from the other. For 'Atlas 2', this setting is fixed in the enabled state.

Segments: This setting can roughly control the number of generated UV islands when using the 'Atlas 2' projection type.

Iterations: When using the 'Atlas 2' projection type, it generates UV islands by grouping together polygons with similar normals to reduce overall distortion. Increasing the 'Iterations' value will increase modo's sensitivity to the normal direction to a finer degree generating less distortion, with more UV islands.

View Projection

Found in the Texture menu, the View Projection Tool creates a UV map from a mesh based on viewport projection.



To use this tool, do the following:

1) Setup a 3D Model View and UV viewport.

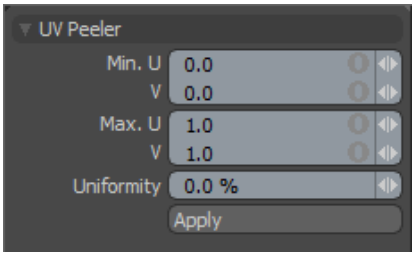
2) Create your mesh and choose View Projection Tool from the UV menu.

3) Click the 3D Viewport with the middle-mouse button and you will see the projected UV mesh update in the UV Viewport.

This works with selected polygons or the entire mesh.

UV Peeler

The Peeler Tool is used for manually fixing UV maps. User 'cuts' a seam of edges in a well-behaved quad mesh, the Peeler Tool spreads UVs out perpendicular to the seam, trying to loop back to the cut from the other side. The region of UVs produced can be controlled. This tool can be useful for tedious, manual-style atlas mapping, or other cylindrical unwrapping tasks.



Tool attributes include:

Min/Max UV: Normally 0,0,1,1, these define a box in UV space where all the grids will be positioned.

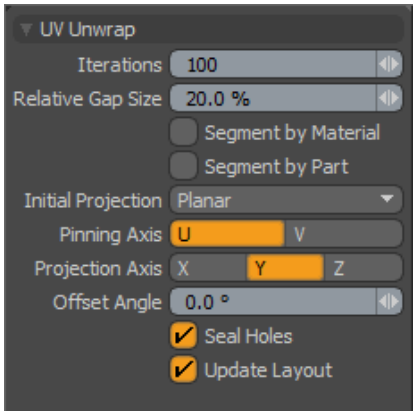
Uniformity: Controls the size of the UV polygons in proportion to the size of the 3D space polygons. At 0% every polygon's size in UV space is proportional to its size in 3D space. At 100% the

polygons all have the same size, making a nice regular UV grid.

The Peeler tool can work well on meshes that have uniform quadrangular polygons organized with adequate edge loop flow. For meshes that have more complex topology it is recommended to use the UV Unwrap tool which provides a far more sophisticated algorithm to untangle the UVs.

UV Unwrap

The UV Unwrap tool provides an elegant solution to creating UV maps from just about any kind of mesh from hard surface to organic. To understand the concept of UV Unwrap, simply think about skinning an animal. Sure it's a little gruesome but it's really the best way to think about it. The idea is that we need to cut seams into the skin of the subject so that when we lay the skin out onto the table it lays as flat as possible without much stretching of the skin. We have just described skinning an animal and UV Unwrap all at the same time. To use UV Unwrap you will define where the model will be "cut" by selecting edges. Don't worry, this will not effect your actual mesh - it only tells the Unwrap tool where to create UV boundaries. Once you have the seams defined via the selected edges, the tool is activated. It is important to choose a good initial projection type and axis before applying the tool. Once the tool is active, dragging in the viewport increases the amount of unwrapping. At this phase you are driving a global optimization of the UV polygons so that they unfold in a uniform manner.



Options for the tool include:

Iterations: This value determines the number of iterations for the global solver to run across the UV data. Basically this is the "amount" of unwrap. With the value set to zero the UV view will appear just as if you simply used a UV projection method based on the Initial Projection and Projection Axis settings for the tool. Once you increase the Iterations the UV data is unwrapped.

Relative Gap Size: If your UV data has multiple contiguous pieces there needs to be some amount of gap left between these UV "chunks". The Relative Gap Size dictates how much space will be forced between the UV boundaries. Increasing this value will push the UV data further apart as well as scale the UV data down to accommodate.

Segment by Material: If you are unwrapping a contiguous mesh that has different materials assigned to various polygon groups you can use these groups to create continuous UV sections while unwrapping. This option provides a quick way to unwrap

a single mesh into several UV segments which can either be left in a single UV map or pasted into separate UV maps.

Segment by Part: This option is identical to Segment by Material with the exception that it uses Part tags from the polygons.

Initial Projection: This popup choice determines how the initial UV projection will be created. The options are Planar, Cylindrical, Spherical, Use Current Map and Group Normal. The first three settings also rely on the Projection Axis.

Planar-- Planer projection will project the 3D model into UV space based on a simply plane that is projected across the model in the user defined axis.

Cylindrical-- Cylindrical creates a virtual Cylinder around the mesh

Spherical-- Spherical creates a virtual sphere for the projection. For more information about the Projection Modes see UV Create Tool.

Use Current Map-- The Use Current Map option uses the current state of the UV map as a starting point rather than projecting new UVs. This is very useful if you already have created some reasonable UV map and want to use Unwrap to further refine the UVs without destroying your hard work.

Group Normal-- Group Normal uses the average normal of the selected polygons to determine the initial projection plane. This can be useful on geometry that is mostly facing the same direction but will give less than useful results when the mesh is complex and has many overlapping areas. If you want to see in 3D the axis that would be chosen for the Group Normal option you can hover your mouse over the 3D viewport and press Ctrl-A. This will align the viewport to the average normal of the selected polygons.

Pinning Axis: The Pinning Axis determines the UV axis to favor when shuffling the UV boundaries during the unwrap. U is the horizontal axis and V is vertical.

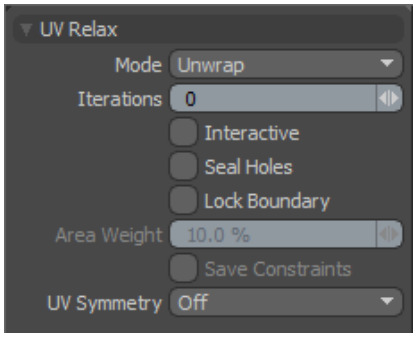
Offset Angle: This value allows you to add some rotation to the resulting unwrapped UVs. This can be very useful if your original mesh is not world axis aligned. By adjusting this value you can see in real time the effect on the map.

Seal Holes: If your mesh is not "water tight" (such as a head with open eye sockets) the Unwrap algrithm can sometimes have difficulty resolving these areas. The Seal Holes option will create a virtual polygon to close the holes and improve the unwrapping effect.

Update Layout: This option forces the UV layout to remain within the 0 to 1 UV coordinate space.

Relax UV

The Relax Tool is very useful for smoothing out overlapping UV data, particularly in combination with the Unwrap tool. The tool has three modes, 'Unwrap', 'Adaptive' and 'Spring'. The Unwrap option uses the same method for untangling UVs as the Unwrap tool but in this context you can use the method to untangle existing UVs as opposed to only creating new ones. The 'Unwrap' option is the default and for most cases will be the best option.



Adaptive method is also very useful. The Adaptive method attempts to maintain the relative volume of the polygon in 3D space when it relaxes the mesh in UV space. This means that polygons the cover a larger area in 3D space will also cover a larger area in UV space. Adaptive mode is designed to assist in the reduction of texture distortion.

The Spring setting is a global optimization solver that treats the UV data like a spring system and iterates through UV positions to try and find an optimal global solution. Spring solvers can be somewhat finicky about the scale of the data and the number of iterations which can result in an unstable solution in certain circumstances.

Perhaps the more interesting use of the Relax tool is with the Pinning option. Pinning allows you to define specific vertices on the UV map that will act as control points or "pins". When the relax iterations are set above 0 the free vertices will smooth themselves out between the pinned vertices. You can drag the pinned vertices in UV space to adjust the map leveraging a sort of UV IK. To set new control points simply click directly on the vertex and a light blue handle will appear. Once the handle is in place you can click to drag it in UV space. Ctrl-Click any pins to remove them. Use the 'Save Constraints' checkbox if you wish modo to remember which vertices were set as pins each time the tool is used.

Using the pinning option is very useful in coordination with the UV overlap highlighting. When you see areas in Red (overlapping UVs) you can simply set a pin in that region and with other critical areas of the mesh pinned, drag the overlap area until the red zones melt away leaving you with a cleaner UV map.

Align UV

The Align command on the Texture menu takes the bounding box of the selected UV

vertices, and moves all of them either in U or in V, to the line in the box that's indicated by the other argument; the 'Align To' field.

The parameter choices on the Align UVs dialog include:

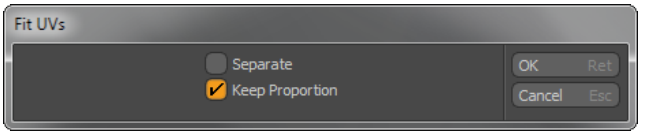
Direction: Up and Down both move the vertices in the V direction. The Left and Right directions both move vertices in U.

Align to: 'Low' is the minimum value of the box. 'Average' is the average of all the values, 'Center' is the center of the bounding box. 'High' is the max value of the box.

The UV Tools tool bar also contains easy to use preset buttons that will quickly align UVs up, down, left or right.

Fit UV

The Fit UVs command will scale the currently selected UV data to fill the 0 to 1 UV space. This is a very useful command for optimizing UVs. There are two options for this command, 'Separate' and 'Keep Proportion'.



Fit UVs

Keep Proportion: When this option is active the UV data will scale proportionally so that the resulting UV data has the same ratio of height to width. The data will scale until one of the maximum UV sizes is fit on one or both axes but the data will not deform to accomplish this. The following images show the UV fit tool with and without the Keep Proportion toggle active.

Separate: The Separate option will force each polygon to become discontinuous in UV

space and fill the UV 0 to 1 coordinates. This is similar to the Barycentric UV projection type and is very useful for forcing a texture to completely repeat from polygon to polygon.

Select Border

The UV Select Border command is similar to the Boundary Edges selection, only for UV mapping. It takes the UV's that appear in the UV Texture window and selects the Border around the UV island sections, and also selects the border edges on the geometry in the 3D Model View. This makes it easy to see the way the flattened sections fit together on the mesh geometry.

To use Select Border follow these steps:

1) Create UV's from mesh geometry using the Create UV's tool.

2) On the UV menu, choose Select Border.

3) The borders surrounding the UV's are highlighted in orange on the mesh geometry, and also in the UV view.

Mirror UVs

The UV Mirror command lets you quickly flip the orientation of the UV selection in the UV Texture View. The Mirror UV's dialog gives you the option to choose which Axis you wish to mirror around (U or V) as well as a numeric field to precisely set the position of the Center of the mirror operation.

To mirror UV's do the following:

1) In the UV Texture View, select the UV's you want to mirror.

2) On the UV menu, choose Mirror.

3) On the Mirror UV's dialog, choose the Axis and Center position, then click OK.

The UV"s will flip to their new position and orientation in the UV Texture View.

Rotate UV

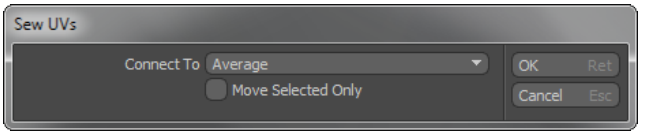
The Rotate UVs command "turns" the UV data around each selected UV polygon resulting in a rotation of the UV order. This is a very useful command when working on low polygon meshes where the orientation of the texture is critical.

Flip UV

The Flip UVs command changes the vertex order of the UV map at the individual polygon level. This is very useful when precise control over textures on low polygon models.

Sew UVs

The Sew UV command allows you to quickly connect matching pairs of discontinuous UVs. To use the command, select an edge or vertex that is currently discontinuous and press the Sew button. The command will move both matching UVs to an averaged UV coordinate and convert them in to a single UV. This is very useful for mending UV maps that have been broken apart either manually, or via mapping tools such as an Atlas Map projection.



When working with Edge selections you can simply select one of the two corresponding UV pairs to use the command. However, to use the sew command on vertex selections you must have both vertices selected before firing the command.

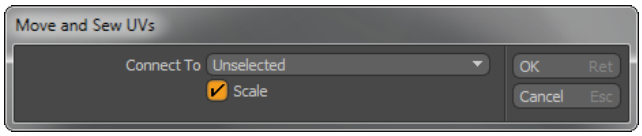
It is very helpful to massage the UVs somewhat prior to using the command so that the UVs are not too far apart before they are "sewn" back together.

Move and Sew UVs

The Move and Sew command is used for merging UV pieces to one another. The "sew" portion represents the action of converting discontinuous UV data into continuous data. In other words, the Sew command portion works to stitch UVs together. The "Move" portion idicates that the entire UV island will be translated to facillitate the merger.

For instance, here we see a UV map with two islands. You can see the "shared" edges via selection. The blue highlit edges indicate that they are the geometric equivalent of those that are selected in yellow. In this way we know where the sew will occur.

By using the Move and Sew with the Unselected option (Move to unselected) the UV island is properly oriented, translated and joined together with the unselected counter-part.

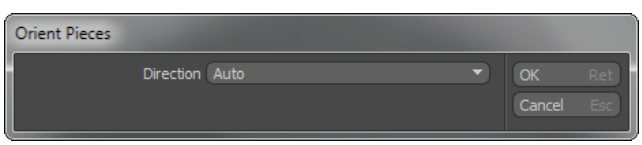


The Move and Sew command has several options all which are made quickly accesible via the Move and Sew subsection of the UV tab. The first three options quicky determine which portion of the UV map will be translated to match the other island. "Selected" will move the unselected geometry to match the position and orientation of the Selected geometry in UV space. "Unselected" indicates that the selected geometry will match the position and orientation of the unselected geometry and Average will move and rotate both pieces to an averaged transform. In all cases the geometry it also scaled and the result is a stitched continuous UV map.

There is also a pop-up for triggering the command with additional options. The options include the ability to Sew only which will move only the selected edges leaving the rest of the island in place, an option to disable scaling of the UV islands and the various "Connect to" options for Selected, Deselected and Average. In most cases you will find that they Selected and Unselected quick options will be the version you use.

Orient Pieces

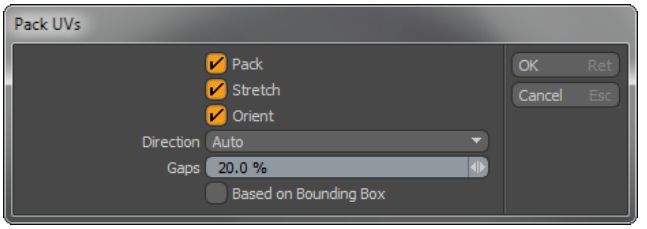
This command automates the process of organizing UV Islands. UV islands are continuous areas of polygons in UV space that are disconnected from any other UV data. This is a common occurrence in Atlas mapped UV sets as well as UV unwrapped data that was accomplished by isolating areas of the mesh for focused unrwaps.



The Orient Pieces command has a single pop-up choice for chosing the Direction of to orient each island. The options are Auto, Horizontal and Perpindicular. The default, Auto, will automatically rotate the pieces in either direction according to which organization creates the algorithically optimal layout. Horizontal will force the islands into a more horizontal orientation based on the lengths of the islands and Perpendicular will force the islands in an orientation that is perpendicular to the Horizontal method.

Pack UVs

The Pack UVs tool can automate the process of organizing UV data to fit into the 0 to 1 UV space. The command includes options for Pack, Stretch and Orient as well as a Gaps percentage setting. You can think of this command as an automated puzzle solver. With the default settings the command will move, scale and rotate each UV island until they all fit within the 0 to 1 UV grid as if solving a complex jigsaw puzzle.



The various toggle options include Pack, Stretch and Orient. Pack enables basic positional transforms of UV islands, Stretch enables the UV data to be scaled and Orient enables rotation of the islands.

Cut/Copy/Paste UVs

The Cut, Copy and Paste commands work together providing users a means to easily move UV values between selections. When the 'Cut' command is invoked, the UV values are removed from the map, and stored in a temporary buffer. Users may make a subsequent selection and apply the 'Paste' command, copying the UV values to the new selection. The 'Copy' command also stores values in a temporary buffer, but are not removed from the initial map.

Delete UVs

The UV Delete tool lets you delete UV selections from your UV map. This doesn't affect the mesh selection in the viewport, it only deletes the UV information from the UV map presently displayed in the UV Texture view.

To delete UV selections do the following:

1) In the 3D Model View, select the geometry.

2) Change the layout to UV_3D_Split.

3) In the UV Texture window, select the UV map components you wish to delete.

4) On the Texture menu in UV Operators, choose Delete.

Export UVs to EPS

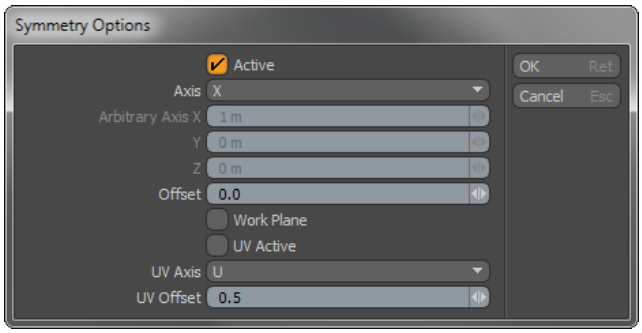
This option provides users the ability to export a UV map to an external application for additional processing in the resolution independent EPS vector format. The resulting file will be useful as a reference point when opened in an Image Editing application (such as Photoshop) when creating bitmap textures.

Precision

Symmetry

Symmetry allows users to apply mirrored actions across a chosen axis simplifying the creating of symmetrical objects. Once Symmetry is enabled, any operations performed in the *component* modes (vertex, edge, polygon) such as selecting polygons or deforming edges will be likewise mirrored across the chosen axis. However, prior to enabling symmetry, users will need to make sure the model itself is perfectly symmetrical as modo applies symmetry operations to corresponding vertex *positions*. Users also need to ensure that the model is located perfectly centered at the item layers world origin (0,0,0 location). This can be accomplished by applying one of the Center commands available in the toolbox of the two modeling layout tabs, or in the menu bar under the "Edit > Center Selected" menu. Additionally, users can use the Symmetry Tool to force a models topology to match symmetrically across an axis, resolving any issues that may cause symmetry to break.

Symmetry can be enabled using the Symmetry pop-up on the modo Modes Toolbar or within the menu bar under "Edit > Symmetry" and choosing an axis to mirror across. Symmetry always works in model space, by default, in relation to the Center position of the item layer. The user can edit the settings of Symmetry using the 'Symmetry Options' available in the menu bar under "Edit > Symmetry Options...".



Active: Enabled/Disables the symmetry function.

Axis: Choose which planar axis that Symmetry works over or 'Arbitrary' to define your own custom axis.

Arbitrary Axis: The symmetry axis is defined by a *vector* and *center position*, 'Arbitrary Axis' allows the user to set a custom axis vector.

Offset: Defines a symmetry position away from the origin, parallel to the base symmetry plane.

Workplane: When 'Enabled, symmetry will respect the Work Plane, but keep in mind as the Work Plane position is modified, the mesh must still have corresponding symmetrical geometry.

UV Active: Enabled/Disables the application of symmetry in the UV Viewport

UV Axis: Choose which axis that Symmetry works across, 'U' is vertical, 'V' is horizontal.

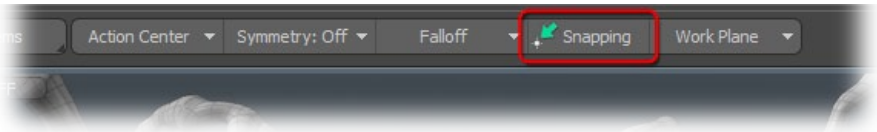
UV Offset: Defines a symmetry position away from the UV origin.

TIP: In certain instances of extreme asymmetry on a model, a quick and dirty way to make certain a model is symmetrical is to simply delete half of the model (making sure symmetry is *disabled* for this operation) and then use the Mirror command with an Action Center set to 'Origin'.

Snapping

Snapping in modo is the action where two elements, when coming in close proximity with one another, the moving element jumps to the fixed element making it possible to very accurately align geometry components in relation to one another, or in relation to

3D space. modo offers a variety of snapping options that can be set accordingly, based on user intent.

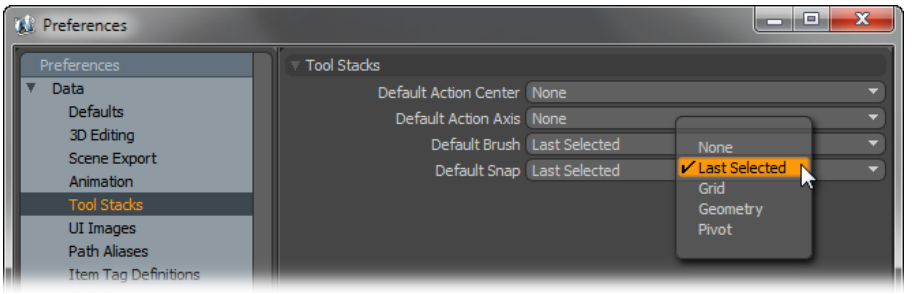


Snapping is a global state in modo, when it is enabled, all actions will obey to the current definition, as specified in the snapping options. Users can open the options panel one of two ways. In the tool properties viewport, there is a persistent 'Snapping' button, LMB+click to open, or simply press the 'F11' keyboard shortcut to open a viewport popover under the mouse pointer; move the mouse away from the form to dismiss it. Users can temporarily enable snapping by pressing and holding the 'X' key on the keyboard, or fix the state by tapping the 'X' key toggling it either *enabled* or *disabled*. When the Snapping state is enabled, the snapping button (illustrated above) will highlight orange.

One of the most important things to keep in mind regarding snapping in modo is that the snapping function snaps the tools *handle*, the widget (or gizmo if you prefer) that appears in the viewport when applying a transform. The position and orientation of said tool handle is controlled by the Action Center and Axis, so these settings are important. By default modo has no defined action center, which is similar to the automatic action center except no axis can be explicitly defined (see the 'Action Center' page of the documentation for more). What happens in this state is modo will move the tool handle to the intersection point of the Work Plane and the mouse cursor click. If the user tried to click over a specific component, they might be exact (unlikely), they might be close or they might not be close at all. Then when snapping the element, the result would be very inaccurate and it would appear as if snapping failed. It actually didn't, It just wasn't told the correct thing to do.

For example, to properly snap a single vertex to another single vertex, users would first need to set the snapping mode to 'Geometry' and choose Vertex as the type. Then set the action center to 'Selection Center Auto Axis'. Then selecting a single vertex, invoke the 'Move' tool (W) and then moving the vertex, hold down 'X' to enable snapping temporarily, the single vertex will then properly snap to whatever new vertex the mouse hovers over. Releasing the mouse button and then 'X' key will set the element at the snapped position. modo has lots of other snapping abilities but just understanding the basic concept of the function will go a long ways to getting it to work the way that is expected.

Setting Defaults



Different users have different ways of working. Snapping in modo is designed to be as flexible as possible. However, with its various functions, most users will primarily use same options over and over. ArchVis users will preferring the grid options, precision modelers preferring the geometry snapping modes. Realizing this is the case, users can set defaults behaviors. eliminating the need to constantly set preferred values. The defaults can be set in the System preferences, or directly in the Snapping options popover using the 'Default Snap Tool' option. Within the Preferences (System > Preferences), under "Data > Tool Stacks", users can choose from several snapping modes, Grid, Geometry or Pivot. With 'None' selected, Snapping is disabled completely and 'Last Selected' keeps the most recent settings as the default.

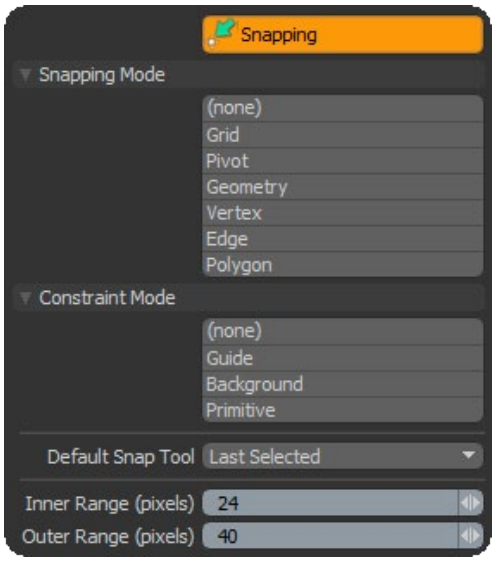
With a specific mode defined as the default ('Grid', 'Geometry' or 'Pivot') whenever the users enables snapping, by pressing the "X' key, that particular mode is activated. Users can press the 'F11' key to open the snapping options popover and modify the default state. However, the new settings are transient. Once snapping is applied and the tool dropped. Subsequent snapping applications will revert to the default state (hence why it's called the default). If snapping is used a lot, users may also wish to assign default 'Action Center' and 'Axis' states. Tools that utilize specific snapping features will override any default values ensuring the tools act as expected. Dropping the tool will return modo to the default state.

Snapping Options

The 'Snapping Options' menu offers options that allow users to choose from the various snapping modes. When a particular mode is selected, additional related options appear in the menu and within the tool properties panel as an adjunct to the tools attributes. Specific mode settings are covered below.

Snapping Mode--

None: Disables Snapping.



Grid: When Snapping is enabled, snapped elements will be compelled toward a user defined fixed grid.

Pivot: When Snapping is enabled, snapped elements will be compelled toward the Pivot position of each visible item layer.

Geometry: When Snapping is enabled, snapped elements will be compelled toward user defined geometry components.

Vertex: Same as 'Geometry Snap' set to Vertex as component type.

Edge: Same as 'Geometry Snap' set to Edge as component type.

Polygon: Same as 'Geometry Snap' set to Polygon as component type.

Constrain Mode--

None: Disables Constraints.

Guide: When selected, users can define custom guide lines to enhance snapping.

Background: When selected, users can constrain or compel vertices of a foreground object against those of the background geometry during a transform or translation event.

Primitive: When selected, users can constrain or compel vertices of a foreground object against those of a user defined primitive shape, during a transform or translation event..

Default Snap Tool: Determines the default tool to use when activating snapping.

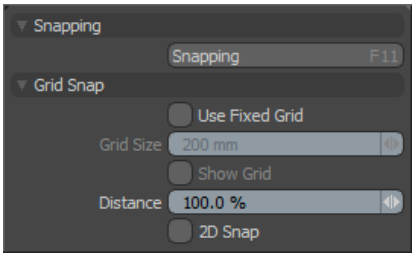
Inner Range: Determines the range, in pixels, where a selected element will snap to the target element.

Outer Range: Determines the range, in pixels, where pre-highlighting will occur. When snap element and target element are within this proximity range, the target element will highlight.

Snapping Modes

Grid Snap Mode--

Snapping of selected element is fixed to a user defined grid. This setting can be helpful in keeping values to whole numbers (eliminating numbers with decimals) and can make the construction of geometry that conforms to a grid far easier and more precise to create.



Use Fixed Grid: When this toggle is *enabled*, users can define a fixed grid with the 'Grid Size' setting, which sets the snapping interval. When *disabled*, the grid snapping defaults to the dynamic grid displayed in the 3D viewports.

Grid Size: When the 'Use FIxed Grid' option is enabled, this value input filed defines the interval for the fixed grid, originating at the 0,0,0 world origin position.

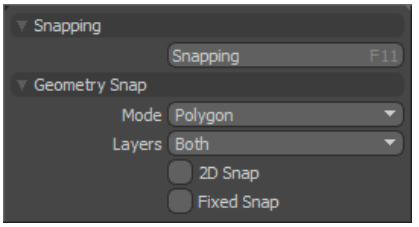
Show Grid: The 'Show Grid' toggle displays the grid as a light magenta overlay in the viewport, which dynamically updates similar to the workplane, in the center of the viewport as a visual reference to the 'Grid Size' setting.

Distance: This option defines the snapping range, the distance when modo will snap to the closest grid position. When set at the default 100%, modo will always keep values fixed to the grid, reducing this value would require the cursor to be closer to the grid positions before snapping occurs.

2D Snap: The 2D snap function relates to the orthographic viewports (top, front, side etc.) that allow snapping on fixed planes parallel to the viewport window, this allows users to move points so they snap into position in the viewport, but they don't necessarily snap into a co-located position in 3D space.

Geometry Snap Mode--

Snapping to specific component elements. These snapping options allow users to very precisely position component elements in relation to one another.



Mode: Users can select which component elements to base the snapping on-

Auto-- modo snaps to the nearest plausible component element.

Auto Center-- modo snaps the center of the nearest plausible component.

Vertex-- modo snaps to the nearest vertex position.

Edge-- modo snaps to the nearest edge.

Edge Center-- modo snaps to the center of the nearest edge.

Polygon-- modo snaps to the nearest polygon.

Polygon Center-- modo snaps to the center of the nearest polygon (average of all the polygons vertices)

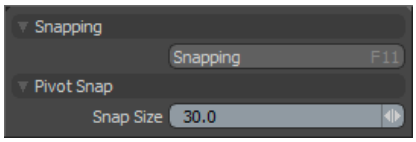
Layers: Instructs modo as to which layers it should consider for snapping, 'Background' only takes into account layers that are visible but not selected, 'Active' only takes into account selected visible layers (also known as foreground layers) and 'Both' which takes into account all visible geometry in the Item List.

2D Snap: The 2D snap function relates to the orthographic viewports (top, front, side etc.) and allow snapping on fixed planes parallel to the viewport window, thus allows users to move points so they *snap* into position in the viewport, but they don't necessarily snap into a co-located position in 3D space.

Fixed Snap: This toggle, when enabled forces modo to only position the moving element to snapping positions, normally moving elements would move, and then snap as they neared an element, but 'Fixed Snap' jumps the elements between snap-able positions.

Pivot Snap Mode--

Snaps selected element to items Pivot elements. This option can be helpful when several items need to pivot rotate from the same location.



Snap Size: This values defines how close an element must be to snap to the pivot position. Larger values will more easily snap elements to the pivot position of an item layer.

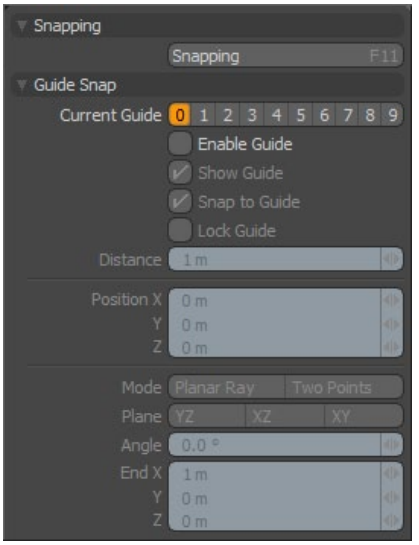
Constrain Modes

Like snapping, the 'Constrain To' options helps to position component elements in a scene, but where Snapping collocates elements, altering their position, Constraints restricts the movement of vertices, using any of the available options.

Guide Snap--

modo allows the creation of Guides for snapping. You can create lines in 3D space and then use them as a snapping tool. To create Guide Snaps, choose the Guide Snap option form the Snap Options menu, and LMB+click in any 3D Model Viewport to create

the Snap Guide at the intersection of the cursor and the Work Plane. Once created, use the blue handle to reposition the guide, and the blue circle to rotate it in 3D Space. When Guide Snaps are being used, the Tool Properties panel displays controls for the Guide. You can scroll through the various guides in the viewport using the Current Guide field. You can precisely control the position, plane, angle and distance of the guide using the numeric fields, as well as turn off the display of the guide, enable the guide and turn the snapping on and off.



Current Guide: modo provides the option to create ten unique guides, each with independent settings. The 'Current Guide' option allows users to choose which guide to create or edit by choosing from one of the ten button options.

Enable Guide: This toggle determines if the current guide is enabled or disabled for snapping.

Show Guide: This toggle determines the visibility of the current guide.

Snap to Guide: This toggle determines

Lock Guide: Once users have set the appropriate guide options, it can be helpful to lock the guide with this toggle to avoid accidentally changing the guides path. When set, the main guide area will change from yellow to cyan.

Distance: Determines the length of the path.

Position: When using the 'Two Points' mode of the Guide function, the position option determines 'Start Point' of the guide. A straight line will be drawing between this value and the 'End' value the represents the guide line itself. When using the 'Planar Ray' option, the 'Position' value determines the home position for the guide.

Mode: Users can define guides in one of two ways, Planar Ray for only planar facing guides, or two points for guides of any angle-

> **Planar Ray--** Users can define a single point, a 'Plane' on which the line sits, and an angle to determine the guideline. Users can use the handle in the viewport to rotate the guideline.
>
> **Two Points--** Users can define two single points in 3D space and the guideline will be created through the intersection of those two points. Users can precisely position the line by dragging the two handles in the 3D viewport.

Plane: When choosing the 'Planar Ray' guide mode, the 'Plane' option sets which

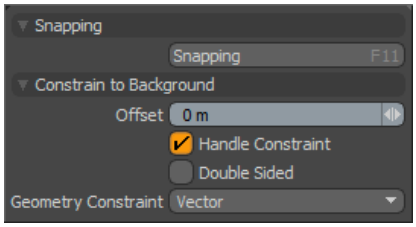
plane (of the 3 axial planes) that the ray rests upon.

Angle: When choosing the 'Planar Ray' guide mode, the 'Guide' option determines the rotation of the guide around the 'Position' point.

End: When choosing the 'Two Point' guide mode, the 'End' option determines the position of the second point.

Constrain to Background--

Geometry can be constrained to background layer polygons during transform/translation events. Constrain to background will constrain the movement or creation of objects from passing through a mesh sitting in a background layer. Constraints can be activated under the 'Snapping' menu using the Constraint popup of the modo Modes toolbar. Any Item layer that is visible, but not selected (highlighted) for editing is considered a background layer. Layer visibility can be easily toggled by clicking the eye icons in the left column of the items list. When active, the following properties appear under the active tool with a selection of options.



Offset: The 'Offset' value allows the user to define a distance away from a surface for constraint. This allows something like a tube being draw over a background object to rest on the surface. If the tube has a diameter of 10 cm, applying a 5cm offset will rest the tube right on the surface as the tube is generated outward from its center.

Handle Constraint: For tools where a curve of some type is drawn, such as the tube tool, curves, bezier, or the pen tool, the 'Handles' of these tools can be constrained as well, by enabling the 'Handle Constraints' setting, otherwise, only polygonal geometry will be constrained.

Double Sided: When enabled, this setting treats background polygons as double sided, so geometry can be constrained regardless of the constraining surfaces normal facing direction.

Geometry Constraint: Several options are available to define how modo searches for the background geometry to constrain against.

Off- This setting disables the constraint.

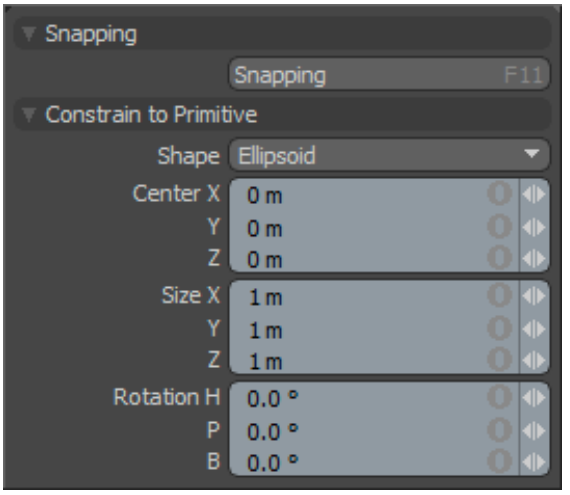
Screen Axis- 'Screen Axis' will simply find the geometry by searching in a perpendicular direction to the current view position, stopping once it hits any geometry.

Vector- 'Vector' uses a direction set by the active tool. If the user has the 'Move' tool active and starts dragging the handle on X axis, the constraint will "look" to that direction to see what background elements it hits and then stop. If performing a bend, vertices will move in many directions, as each vertex will have its own "vector", which is effectively the direction the vertex is currently moving.

Point- 'Point' constrains each individual vertex against the background geometry ensuring it is always resting on its surface.

Constrain to Primitive--

The 'Constrain to Primitive' modifier creates a virtual primitive shape onto which tool handles can be snapped. Similar in concept to 'Constrain to Background', however, no physical geometry need to be present for 'Constrain to Primitive'. Once activated, the following properties will appear below the selected tool.



Shape: Users can choose from the various primitive shapes available for constraint- Plane, Ellipsoid, Cone, Cylinder and Cube.

Center: Defines the center position of the virtual constraint element, based on the center of the bounding box.

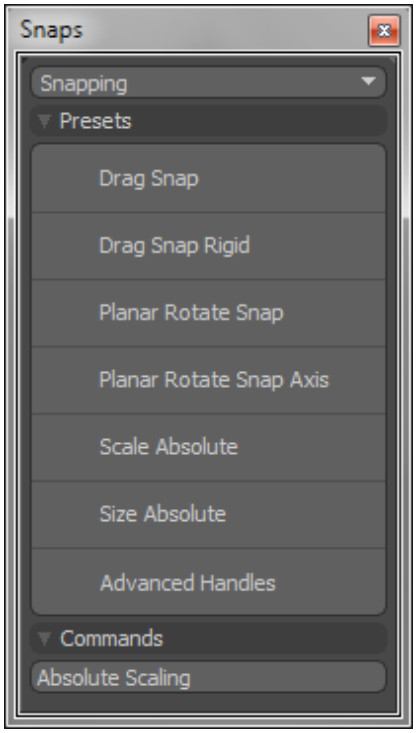
Size: Defines the bounding box, which sets the size of the virtual constraint element.

Rotation: Defines the rotation amount for the virtual constraint element.

Snaps Palette

The 'Snaps Palette' provides a handy floating palette that holds all the settings and tools related to snapping in a single viewport. The palette can be initiated from the menu bar command "Edit > Snapping > Snaps Palette". Once opened, the visibility of all floating viewports can be toggled on and off by pressing the ' ` ' (*grave accent, same as tilde* ~) key. At the top of the palette is the same 'Snapping' options available in the interface snapping menu or from the menu bar. The information on snapping is covered on the snapping page of the documentation. Below are several tool options users can use for precision transforms. Created with the power of the 'Tool Pipe', they are automated combinations of transform actions, snap settings and Action Centers.

Snaps Palette - Presets



Drag Snap

The Drag Snap tool combines the Element Move transform with Geometry Snap. This allows components of a mesh to be quickly "dragged" and "snapped" to other components in the project. By default the Element Move transform is set to "Auto" which allows you to click and drag a vertex, edge or polygon regardless of the current selection mode. (See Element Falloff and/or Geometry Snap for more information.)

Drag Snap Rigid

The Drag Snap Rigid tool is very useful for moving connected geometry by dragging it from a single component such as a vertex, edge or polygon. The tool combines with snapping so that you can quickly snap the current component to another component in the project while dragging the rest of the connected polygons along in a rigid manner.

Planar Rotate Snap

The Planar Rotate Snap tool is excellent for precisely aligning elements. To use the

tool you simply click on any component in the scene. This initial click determines the element upon which the tool handles will be centered. Next you use the outer most handle to establish the resting orientation for the rotation control by dragging that handle to any component in the project. The handle will snap to vertices, edges or polygons. Once the rest orientation has been determined, use the inner round handle to apply some rotation. This rotation action can be freehand or you can once again leverage geometry snapping to very accurately set the rotation.

Planar Rotate Snap Axis

This tool is similar to the Planar Rotate Snap tool with the extension of a second axis handle for controlling the plane of rotation. This is an advanced tool that is invaluable in certain complex rotation scenarios that require absolute precision. Like the Planar Rotate Snap tool the first click sets the center and axis for the tool based on the component under the mouse.

Scale Absolute

Scale Absolute combines the scale tool (Absolute Handles option active) with the Geometry Snap modifier. The tool utilizes a two step process. In the first step you use the Absolute Handles to set the starting scale point for the tool. Next you use the Scale handles to actually modify the geometry. During both steps the handles can/will snap to the geometry thus allowing you to use one mesh component to set the initial scale and a second component to create the Absolute scale.

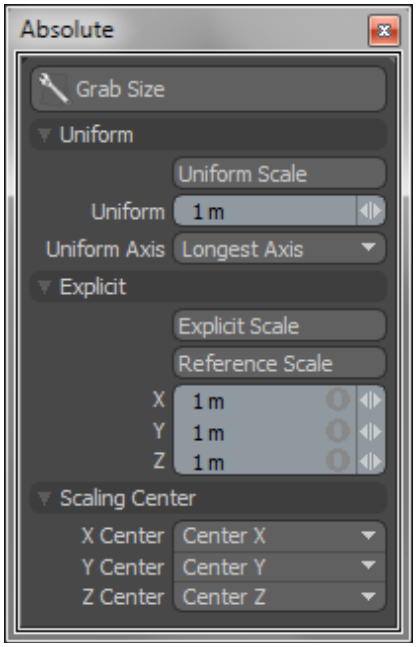
Size Absolute

The size absolute option opens a absolute sizing option

Advanced Handles

This option toggles the visibility of Advanced Handles for the selected tools. This often places additional controls into the screen viewport making it easier to produce more accurate results.

Commands--



Absolute Scaling

Pressing the 'Absolute Scaling' button of the Snaps palette, opens yet another palette with a useful set of utilities that go well beyond basic scaling. Allowing users to explicitly define a size and scale on an items bounding box, and modify the center axis position when assigning the transform.

Grab Size: This command 'grabs' the actual bounding box size and inputs it into the 'Uniform' and 'Explicit' data fields as a point of reference for sizing the selection.

Uniform--

The 'Uniform Scale' options provide a means to uniformly (equally) scale all axes of a selection in a single step. To use, first choose the 'Grab Size' command, this captures the Longest Axis length and inserts it into the Uniform value input field as a point of reference. Next, select a 'Scaling Center' if you wish it to be other than the default bounding box center. Enter a new explicit value into the 'Uniform' input field. In Uniform Axis, define the Axis you wish to scale absolutely, and then finally LMB+click the 'Uniform Scale' button to perform the scale operation.

Uniform Scale: When this button is pressed, the bounding box for the selected item is uniformly scaled by the value in the 'Uniform' input field, based on the 'Uniform Axis' field.

Uniform Axis: When scaling the bounding box, the 'Uniform Axis determines on which axis the scaling command will operate, with the other two axes scaled in proportional amount in relation to the defined axis.

Explicit--

The 'Explicit Scale' option provides a means for users to define a specific size to the bounding box of the selected elements. To use, first choose the 'Grab Size' command, this captures the bounding box volume size of the entire selection. Next, select a 'Scaling Center' if you wish it to be other than the default bounding box center. Enter a any new Size values into the X,Y & Z input fields,and press the 'Explicit Scale' option to scale the object to the specified size.

Explicit Scale: When this button is pressed, the bounding box for the selected item is

scaled by the value in the X,Y and Z input fields.

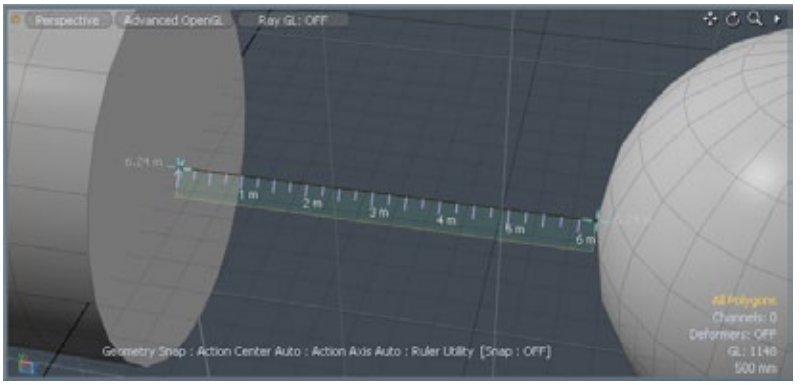
Reference Scale: When this button is pressed, the selected section of geometry (assuming its a smaller section of a whole) defines the scaling amount as a reference, but all connected geometry is scaled proportionately the the selected area. Say for instance if you knew the size of the cockpit of a race car, one could scale the entire car by the proportional amount based on just the selected cockpit area.

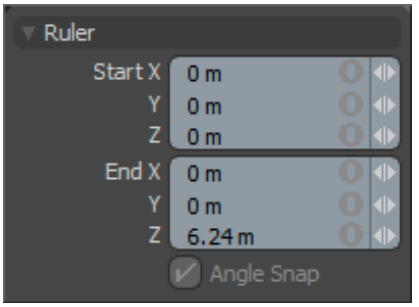
Scaling Center--

Scaling Center X/Y/Z: When scaling a selected area, the 'Scaling Center' values define the origin of the scaling operation as defined from the drop down box selections. The 'Low' options defines the side toward the negative end of the axis, 'Center' the averaged center position of all vertices, and 'High' being the side of the bounding box nearest the positive values of the axis.

Utilities

Ruler Tool

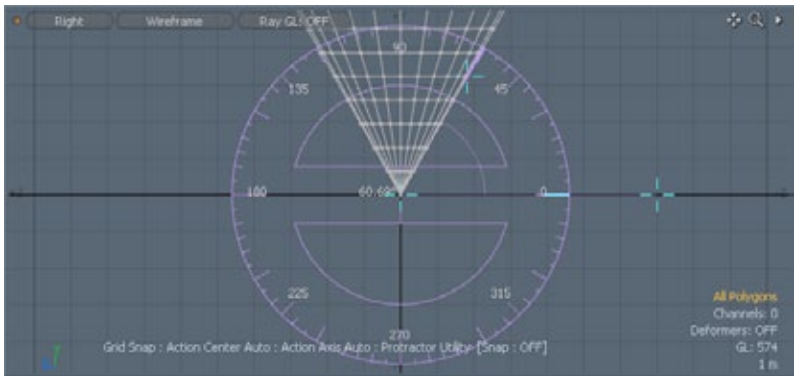


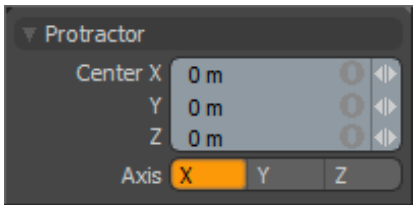


The 'Ruler' command creates a virtual measuring device in the viewport that lets users quickly ascertain the distance between two locations. Users can activate the tool using the menu bar command "View > Ruler Tool". Once activated the Ruler will become visible in the 3D viewport adding an additional tool properties panel. Users can hover the mouse cursor directly over the handles at either end and LMB+click and drag to position the handles in the scene. The values on the ruler will interactively update as

the ruler is positioned in the scene. The value displays are view dependant, so moving the view position of the viewport will adjust the values for ease of reading. At either end will be a readout of the rulers total length. To reposition the ruler itself, users can simply LMB+click and drag the ruler element repositioning it within the scene. Users can also use the Ruler Start and End fields of the property form to define a fixed size ruler within the scene. When combined with the Snapping function, the ruler is a powerful option for measuring elements in a scene.

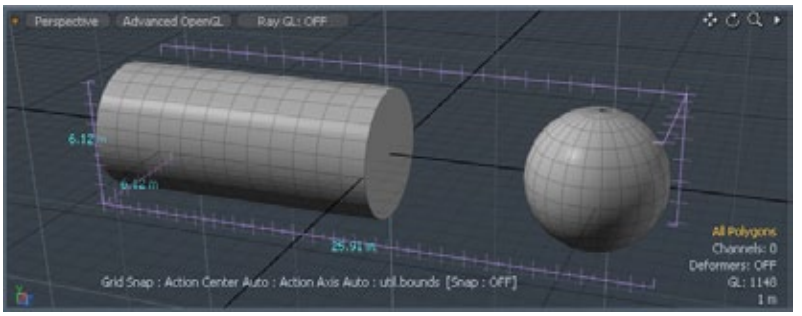
Protractor Tool



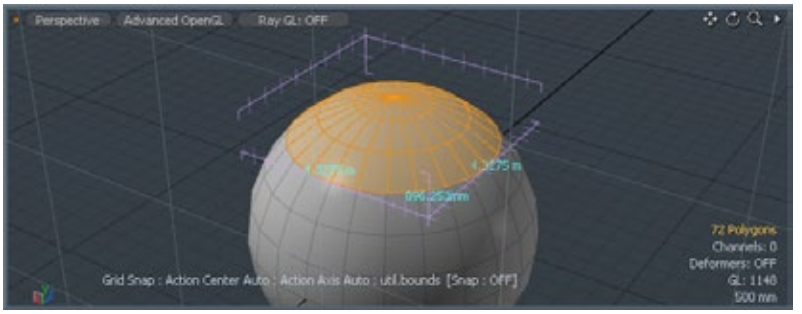


The Protractor Tool provides users with a simple visual method for accurately measure angles in the 3D viewport. Users can activate the tool with the menu bar command "View > Protractor Tool". Once activated, the Protractor tool will become visible in the 3D viewports. There are three handles the user can use to position and measure elements in the scene. The handle at the center of the Protractor itself will position the tool within the scene. The second tool handle controls the rotation of the tool representation and the third handle measures the angle, producing the readout display. TO adjust a handle, users can simply hover the mouse pointer over the handle representation and LMB+click and drag to adjust. Users can also use the tools property form to position the center of the Protractor. When combined with the Snapping function, the Protractor is a powerful option for measuring angles in a scene.

Dimension Tool



The Dimensions tool will display the bounding box dimensions of any object or component selection. When activated through the menu command "View > Dimensions Tool" the dimensions become visible within the 3D viewport displaying the size of the element in all three axes. The dimension tool requires no user input or interaction outside of making a selection.



Work Plane

modo's Work Plane

Using a two dimensional (*flat*) screen to create and edit content in 3D space can sometimes feel awkward and unnatural. Software has tried to resolve the disconnect using different solution over the years. The most popular is the four viewport method. Each orthogonal viewport only provides two axes at any given time, eliminating surprises to users as they create and modify geometry in a given scene. Another solution is to require the user to work with tool handles constraining operations performed in a perspective viewport to the same flat 2D axes of the four viewport method. Both of these solutions are reasonable and available in modo. However, modo also employs a third method called the 'Work Plane'.

The Work Plane is a white grid that appears in the 3D openGL viewports. In this example above, the Work Plane has been highlighted for emphasis.

The Work Plane is a white grid that appears in the 3D openGL viewports. In this example above, the Work Plane has been highlighted for emphasis.

The Work Plane is an adaptive modeling aid that automatically adjusts itself to match the two major axes most closely aligned to the users screen axis, represented by a white grid within any of the openGL viewports. When you rotate the Perspective viewport the Work Plane will snap to one the dominant planes, XY, XZ or XY. Action Centers often use the Work Plane position to determine the center and axis of tools. When hauling, many tools also use the work plane to determine the axes of operation as well when this information cannot be derived from a specific tool handle.

While the default mode for the work plane is to automatically adjust to the screen rotation, the work plane can also be locked and used as a construction plane, when applied as such, the Work Plane effectively positions the entire modo universe to that fixed angle and position. There are a number of methods for manually setting the Work Plane to a specific center and axis. The work plane can be easily locked to a specific major axis and position, or you can snap the Work Plane to selected geometry or even just to the polygons directly under the mouse. This level of flexibility and control provides users with an incredibly fluid and rich set of options for editing their workspace.

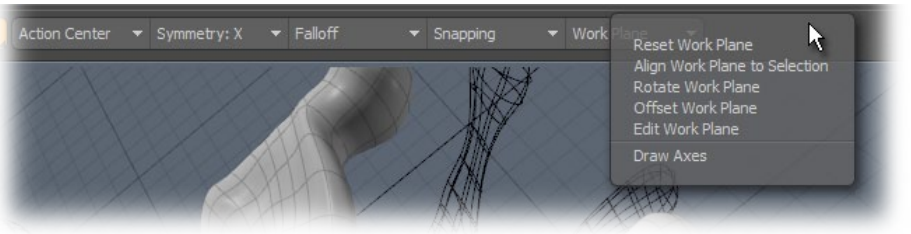
Mastering the Work Plane will allow users the freedom to do most, if not all, of their

modeling in a single perspective viewport. That means more screen real estate for viewing the geometry and performing the various modeling tasks allowing users to work more fluidly and accurately. Users are encouraged to experiment with the Work Planes when using tools and hauling, so they can get a better grasp of its behaviors and use them to their advantage.

TIP: Users can quickly offset the position of the Work Plane using a keyboard shortcut. By hovering the mouse pointer over some geometry in the scene where the Work Plane should move to, and then pressing 'Alt/+O' ('Option+O' on a Mac), the Work Plane will retain its orientation, but move to the location where the mouse pointer intersects with the geometric element.

Work Plane Controls

The Work Plane is a persistent function of the 3D viewports, and is visible by default, users can toggle its display under the visibility controls of the viewport options panel that pops up when hovering the mouse pointer over a viewport and pressing the 'o' key. Additionally, users can toggle the Work Plane visibility by pressing the ' * ' (asterisks) key on the number pad. The controls for the Work Plane are conveniently located in the interface under the Work Plane menu button, or within the menu bar under the "Edit > Work Plane" menu. Users can also use the 'Home" key to dynamically position the Work Plane to any geometry directly under the mouse pointer, centering under where the pointer tip intersects the polygons surface and aligning it to the polygons normal direction.



Reset Work Plane

Invoking this command returns the Work Plane to it's default behavior. Only applies if the Work Plane has been aligned as a fixed construction plane.

Align Work Plane to Selection

Aligns the Work Plane to the current component selection as described below. If nothing is selected, the Work Plane will be aligned to the average normal of all foreground polygons.

When selecting vertices--

If a single vertex is selected, only the center of the Work Plane is changed.

If two vertices selected, the first vertex sets the center and the workplane will be altered to align the Z axis with the second vertex.

If three vertices are selected, it uses the first vertex to set the center, the second to set the Z axis and the third will define the XZ plane.

If more than three vertices are selected, the center is located at the average of their positions, and the plane is angled to put all vertices as close as possible to the XZ plane.

When selecting edges--

If a single edge is selected, it is treated the same way as two vertices using the ends of the edge (above).

If two edges are selected, they are treated the same as three vertices (above).

If three or more edges are selected. this results in the same solution as vertices, using the end points of the edges as the vertices

When selecting polygons--

If a single polygon is selected, the workplane is centered on that polygon with the Y axis aligned to the polygon normal. The longest edge of the polygon will be along the Z axis.

If multiple polygons are selected, the average normal is along the Y axis, and the system uses a "best guess" to determine correct orientation.

Rotate Work Plane

The Rotate Work Plane function provides users a means to accurately rotate the Work Plane a defined number of degrees away from its current position. Invoking the command opens this dialog pop up-

Axis: This option defines around which axis the Work Plane will rotate, originating at the Work Plane origin (0,0,0) position.

Angle: Defines the number of degrees to rotate the Work Plane away from its home position.

Offset Work Plane

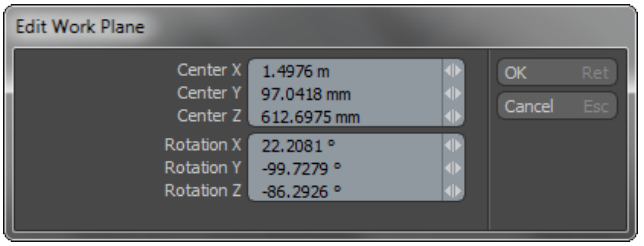
The Offset Work Plane function provides users a means to accurately position the Work Plane by way of an offset from its current position. Invoking the command opens this dialog pop up-

Axis: This option defines around which axis the Work Plane offset its position along.

Offset: Defines the distance to offset the Work Plan away from its current position.

Edit Work Plane

Displays the current values of the work plane as offset from the true world origin position at 0,0,0. Users can edit the values to precisely position and rotate the Work Plane.



Draw Axes

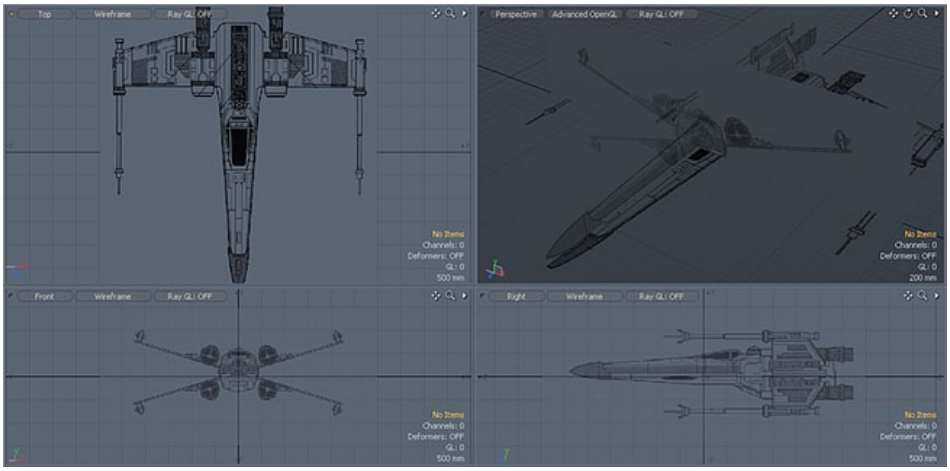
This toggle will enable/disable the drawing of an axes handle widget in the viewport at the origin position of the Work Plane making it easier to recognize the center and axis directions.

Backdrops

Modeling with Images

Backdrop items are terrific modeling aides allowing users to add images into a background plane that acts as reference for geometry creation. This can be a photograph, a line drawing or even blueprints. As Backdrop Items are only meant for reference they will not contribute to rendered output. Best of all, since they are items in

the item list, they are saved with the scene and can be manipulated with the standard transform tools.



To add a background image users can use the menu bar command "Item > Create Backdrop" which creates a new Background item layer in the Item List. Users can then select the newly created layer and further define the backdrop settings within its associated properties panel, where users set the image, orientation and adjust its image properties (detailed below). Once users have defined a number of different views, it is handy to group them together using the 'Ctrl+G' keyboard shortcut providing users the ability to toggle the all of the Backdrop images visibility with a single action.

Backdrop Item

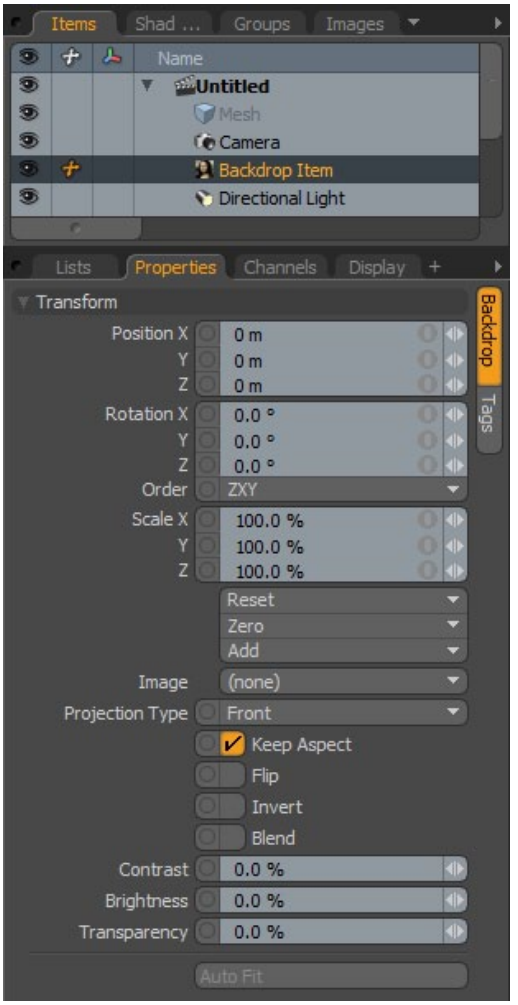
Transform--

Position: The 'Position' setting allows the user to specify a numeric XYZ values for the placement of the backdrop image within the viewport. All transforms originate from the planes center point.

Rotation: With 'Rotation' you can adjust a backdrop items rotation in the viewport, however in most instances, it would be best to simply adjust the 'Projection Type' to position the image toward a specific viewport.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: The 'Scale' function allows the user to adjust the size of the backdrop items.



Reset: Resets the selected transform values to (0,0,0) returning the items back to the world space center position.

Zero: Returns the items Center position to the world space center (0,0,0) without changing the position of the backdrop item itself.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms. Since Backdrops don't render, this setting is unnecessary.

Image: Click in the image drop down to specify an image or use the load command to load an image from disc.

Projection Type: The 'Projection Type' setting allows the user to specify which viewport the backdrop image will face toward as well as a 'Camera' projection mode that appears automatically in the background of any Viewport set to 'Camera' as the type. For multiple viewports, such as front, side and back, three different backdrop items would need to be created with the appropriate image and projection type specified for each.

Keep Aspect: The 'Keep Aspect' function will retain the aspect ratio of an image when it is scaled.

Flip: The 'Flip' command will mirror the images pixels across the vertical axis, useful for using the same image for front and back pairs or left and right pairs.

Invert: The 'Invert' toggle will invert the RGB values for the backdrop image, useful when inverting a black and white line drawing to make the lines more visible.

Blend: The 'Blend' function will produce a small amount of smoothing between neighboring pixels, reducing the jagged appearance of low resolution images.

Contrast: Contrast is the apparent visual different between light and dark values in an image. Values greater than 0% will increase the contrast of the backdrop image, while values lower than 0% will decrease it.

Brightness: Brightness is the degree or amount of light or luminosity in an image. Values greater than 0% will increase the brightness of the backdrop image, while values lower than 0% will decrease it.

Transparency: Transparency determines the opacity of the backdrop image. Values greater than 0% will increase the transparency, ramping toward 100% where the image would become completely invisible.

Auto Fit: The 'Auto Fit' command will automatically scale the Backdrop item to match the *height* of the currently active mesh items, or the height and width when 'Keep Aspect' is disabled. However, this will only work when 'Display Backdrop Items in Uniform Size' is enabled in the preferences file. To find it, click in the menu bar "System > Preferences' and then in the preferences window, select the 'Display > OpenGL' section, backdrop items are listed toward the bottom.

TIP: Within modo's 'Preferences', there are a few options related to Backdrop items and how modo scales images when placing them. For more information, please reference the appropriate Preferences page of the documentation.

Painting and Sculpting

Painting and Sculpting in modo is designed to be simple to use, while providing users with powerful and intuitive toolset that empowers artists to realize their vision with the least amount of strain. The 'Workflow' section provides the overview to how modo's brushing engine works, while the subsequent sections detail the individual brushes and related tools for texturing and detailing your creation.

Working with the Tools

modo provides users with the ability to sculpt their models in a variety of ways, and then texture them easily in one integrated environment, allowing user to work non-linearly. This flexibility means users can model, sculpt, apply a basic texture and then go back and model some more. As with all the tools in modo, the functions are deeply integrated and take advantage of the core functionality providing an intuitive and powerful set of options. Combining a natural blend of 2D and 3D painting techniques, users can just grab a brush and begin painting in any viewport, including UV views; yes sculpting tools can be used to adjust UVs! While modo will try to choose the best option automatically when painting or sculpting, there are a couple requirements necessary for the best results. modo's default interface provides a layout tab called 'Paint' dedicated to painting, sculpting and vertex map editing tools, with a provided toolbox containing all the related tools.

While not a strict requirement, users will benefit greatly from the use of a pressure sensitive tablet when painting or sculpting in modo. Most users find them to be much more natural way to work with their pen based input, as opposed to strictly mouse based. Regardless, the additional amount of control offered by the pressure sensitive stylus can provide users a greater degree of precision and subtlety when modeling and texturing their creations.

Sculpting

There are 3 types of sculpting in modo- 'Mesh', 'Image Based', and 'Multi-Res'. 'Mesh' sculpting utilizes the raw geometry, great for building out rough forms, but not so good at producing fine details. 'Image Based' sculpting utilizes image maps that are great at producing fine details. The images can either be grayscale *displacement*, where shades of gray push vertices in or out a specific distance based on their shade or a 32bit *vector* image that allows not only in and out movement, but side to side as well, allowing for cavities and undercuts. Image based sculpting is a powerful way to generate fine details for models but suffer from some limitations, like the need for fixed resolution bitmap images and of course the related UV map. Finally, modo offers 'Multi-Res' sculpting, which requires no images or UV mapping, allowing users to sculpt on Multi Resolution meshes providing the flexibility and immediacy of mesh sculpting, and the detail of image based sculpting. It's only drawback being the greater amount of resources necessary to calculate them. It is up to users to understand the methods and how to best apply them for their intended purposes. Each function has it's own set of requirements covered below.

Mesh Sculpting

Mesh sculpting is a simple affair in modo, all that is required is some geometry and a brush. Users can grab any of the Sculpting tools, available under the 'Paint' tab of the interface, and just start sculpting. modo doesn't care if the model is straight polygons or a Subdivision Surface, and will push and pull the vertices as the user wishes. The only caveat to sculpting in raw polygons is the fact that it can be quite easy to generate grossly non-planar polygons, which in turn can cause rendering errors. While these can generally be easily resolved by triangulating the model or applying Subdivision Surfaces to the mesh, it is something to keep in mind.

Benefits- Easy setup, can work across multiple surfaces, not limited by material assignment, CPU friendly, low memory requirements

Disadvantage- no high-frequency detail, only works on vertices

Image Base Sculpting

modo's displacement mapping methods both require subdivision surfaces models, either regular or Pixar's 'Catmull-Clark' versions will work. They also need an image map applied to all surfaces you wish to sculpt. For best results with displacement maps, 16bit images are recommended, for vector displacement based images, 32bit formats such as EXR are required. The images also need to be applied to the Surface via the Shader Tree using a UV map. This can be specified in the texture locator. While defining a UV map is often considered a tedious process, modo provides tools like 'Unwrap' and 'UVRelax' that make it easy to produce quality results in a short amount of time.

Once these requirements are met, users can begin sculpting in the same way as for straight mesh sculpting. Grab a brush and a tip from the toolbox and begin to sculpt the model.

Benefits- Good high frequency detail, low polygon requirements until rendertime

Disadvantages- setup required with UVs, video card display limitation for openGL viewports

Multi-Res Sculpting

The new Multi-Res sculpting is simpler to set up than image based sculpting, but users do need to ensure it is done properly. First, Multi-Res requires the newly introduced Pixar 'Catmull-Clark' Subdivision Surfaces geometry, so if your model has regular SubDs, then they must be converted to Pixar's 'Catmull-Clark' subdivision surfaces. This can be accomplished by selecting all the SubD surfaces, pressing 'Tab' to disable SubD mode, and then pressing "Shift+Tab" to apply Pixars 'Catmull-Clark' SubDs. Next, users will need to enable the 'Multi-Res' checkbox in the mesh item properties.

Additionally, user may wish to introduce some additional Subdivision levels using the 'Subdivision Level' option. After these requirements are met, users can simply grab a sculpting tool and begin to sculpt. Users can interactively add or remove subdivision level by pressing the '+' and '-' keys on the numeric keypad. Additionally, users can *walk though* the subdivision level using 'Ctrl+Plus Key' and Ctrl+Minus Key'.

Benefits- Easy setup, no images or UV maps, Multi-Res feature allows powerful editing methodology

Disadvantages- Requires more RAM and processing power than other methods, result must be *frozen* for export to other app's

Painting

In order to paint in modo, all you need is a 3D model and a bitmap texture. You will need to apply the texture map to some or all of your objects polygons via the Shader Tree, but that's it, you're ready to paint! modo has several tools to allow you to quickly meet all of these requirements including extensive UV mapping tools and functions to quickly add a texture to your object.

There are two different painting technologies you can use when painting in modo. There is standard '3D Painting' which paints directly onto the 3D model and there is 'Projection Painting'. Standard painting keeps the brush stroke size consistent with the canvas, while projection painting keeps it consistent with the viewport, that's why standard 3D painting often results in oddly shaped brush strokes when painting on complex objects, such as a character. Think of Projection Painting as painting onto a plane of glass in your viewport and like a movie theater screen, each stroke is *projected* onto your 3D model eliminating the distortions inherent in standard 3D painting.

IMPORTANT: While possible, painting on *complex objects* with any of the various non-UV texture projections such as 'Planar', 'Cubic' or 'Spherical' will usually yield unsatisfactory results.

Vertex Map Tools

Users can create and edit many of the different Vertex Maps available in modo using the intuitive brush based input method. The brushes and techniques used for regular image based painting apply as well to Vertex Maps. While having a variety of uses, the most common will be for users wishing to adjust Weight Maps used in relation to Deformers. Users may also wish to use the tools to create and edit Vertex Color Maps, often used to add depth and variety to low-resolution game assets.

Prior to editing any Vertex Maps, it will be necessary to select the 'Vertex Map' viewport display style, this will allow selected maps to be visible. For information on working with display styles for the 3D viewport, please reference that page of the documentation. When working with any Vertex Maps, users will need to first select the target 'Mesh' item in the Items list. Next, the appropriate Vertex Map needs to be *selected* in the Vertex Maps list viewport. Each Item layer can have its own Vertex Maps, and if the target layer isn't selected the associated maps will not be available within the vertex maps list (If no map of the intended type is available, it may be necessary to create the map using the '(new map)' option in the vertex map list viewport). Once the Item and Vertex Map is selected, users can simply choose any of the tools and proceed to use the mouse or table to apply the action.

TIP: The 3D viewport has an additional 'Show Weight Value' overlay mode option making it possible to view and edit Weight Maps while in any of the Viewport Display styles. Press 'O' while hovering the mouse pointer over the viewport to open the display options.

Particle Paint Tools

Particles have many uses in modo, mostly as point-sources for Replicators and Texture Replicators, as well as for Blobs, Sprites and Volumes. Previously, these particles were either created procedurally with 'Surface Generators', or had to be created indirectly by generating geometry and them removing the polygon edges between the vertices. The Particle Paint tools provide users a means to create and edit particle point sources directly using the familiar brush based workflow.

Creating the particles is a simple procedure; the only necessary preparation is having a geometry surface to spray the particles onto. The 'Particle Paint' tool is the main tool to create the initial particles, then the Editing tools can be used to further mold and shape the particles for the intended use. When using the 'Constrain' actions, the particles will always remain conformed to the target surface, and the beauty of modo is that it's smart enough to know what a particle is versus regular polygons, so there's no fear of accidentally editing the base geometry to which the particles are being applied.

Customizing Brushes

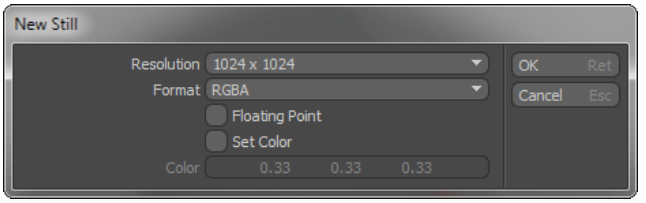
Like the rest of the modo toolset, the paint and sculpting tools are modular and composed via the Tool Pipe. If you are not familiar with the Tool Pipe and don't care to be, do not worry about it. You can get rolling with 3D paint and sculpt and not even have to think about it. Simply use any of the provided tools; that should fulfill most (if not all) of your modeling and texturing requirements. For those who understand the power of the Tool Pipe, you will understand immediately the flexibility and power this brings to

the painting system. The main point here is that there are components that users can add together to create various custom tools. These components include the paint tool, a brush tip, custom inks, nozzles and even falloffs. For basic painting and sculpting, only a tool and a brush tips are required; the inks, nozzles and falloffs can be added or removed depending on the intended effect. The variety of combinations available provide a dizzying number of brush customizing opportunities, that should keep users busy for a long time.

Add Image

Bitmap images are an essential component of texturing, painting and image based sculpting. To simplify this common requirement, saving users a trip to an image editing application, modo allows users to create *blank* images directly in modo, easily and quickly for a variety of applications using the 'New Image' command found in all image browsing palettes and in the image selector drop downs of many panel options. Additionally, users can use the 'New Image' command in the menu bar under "File > New Image" menu. Once an image is created, it must be properly placed in the Shader Tree, so that it is applied to the surface the user wishes to affect. It also may be necessary to set its layer 'Effect' to the proper option, such as 'Displacement' for image based displacement. For more information on working with layers in the Shader Tree and assigning their 'Effect', please reference the Shader Tree page of the documentation. Modo also has a variety of utilities to make it easier to add specific types of new images to the Shader Tree which are covered below.

When the 'New Image' command is invoked, users will first be presented with a OS specific dialog window where user specify the saved files name, location and format. Once that requirement is filled, the following modo panel opens up to designate specifics of the image itself.



Resolution: When painting in 3D the images are displayed via OpenGL. Performance of GL texture rendering is superior when working with square images. As a result the choices for Resolution are based on standard GL texture sizes (64x64, 128x128, 256x256, 512x512, 1024x1024, 2048x2048, 4096x4096 and 8192x8192)

Format: The created image can be gray, RGB color or RGBA color with an alpha

channel. Gray images are typically smaller in file size and are very useful for textures that will be used for displacement or bump maps. Typically the best choice for color textures is RGBA so that the new image texture does not completely obscure the materials and textures layered beneath it. RGB images are opaque.

Floating Point: Also known as HDRI (High Dynamic Range Images), floating point images give a much broader range of potential color values. Whereas an integer (default) image defines colors in 8 bits per channel (0 to 255) a floating point image can use values many times larger. This can be incredibly useful when using images as light sources or to drive finely detailed displacement maps.

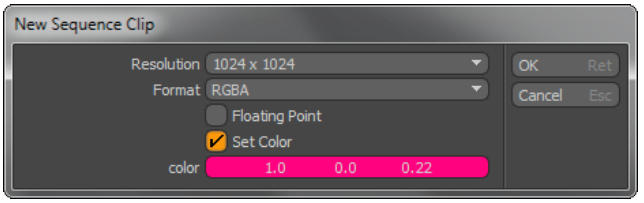
Set Color: This toggle defaults to off so that new images are completely "blank" meaning they have no color or alpha pixels. Turning this option on will create an image with a opaque alpha so all pixels are visible and a base color for the image. The color is set by the color component under this toggle option.

Color: The color swatch is only applicable when the Set Color toggle is active. This color value will be used to fill the new image.

TIP: To save an image as 16bit, users must enable the 'Floating Point' option or images will default to 8 bit regardless of the chosen image format.

Add Image Sequence

Users can add image sequences as easily as adding still images. Sequences are simply a series of still frames stored as consecutivly numbered image files. These can be useful for baking out animated textures, or painting animated sequences. When applied to a surface, the sequence frame number will equal the rendered frame number. When frames go missing, the last plausible frame will be repeated until another frame is found. To create a 'New Sequence' users can use the menu bar command "File > New Image Sequence" or use the 'New Sequence' option in the clips browser. When invoked, an OS specific dialog will open for users to define the name, location and format of the files. Then click 'OK' for the modo dialog where users define the resolution and other aspects of the frames themselves.



Resolution: When painting in 3D the images are displayed via OpenGL. Performance of GL texture rendering is superior when working with square images. As a result the choices for Resolution are based on standard GL texture sizes (64x64, 128x128, 256x256, 512x512, 1024x1024, 2048x2048, 4096x4096 and 8192x8192)

Format: The created image can be gray, 'RGB' color or 'RGBA' color with an alpha channel. 'Gray' images are typically smaller in file size and are very useful for textures that will be used for displacement or bump maps. Typically the best choice for color textures is RGBA so that the new image texture does not completely obscure the materials and textures layered beneath it. RGB images are opaque.

Floating Point: Also known as HDRI (High Dynamic Range Images), floating point images give a much broader range of potential color values. Whereas an integer (default) image defines colors in 8 bits per channel (0 to 255) a floating point image can use values many times larger. This can be incredibly useful when using images as light sources or to drive finely detailed displacement maps.

Set Color: This toggle defaults to off so that new images are completely "blank" meaning they have no color or alpha pixels. Turning this option on will create an image with a opaque alpha so all pixels are visible and a base color for the image. The color is set by the color component under this toggle option.

Color: The color swatch is only applicable when the Set Color toggle is active. This color value will be used to fill the new image.

Saving Images

Once users have modified their images or sequences in modo in any way, through baking, sculpting or painting, the images themselves will need to be saved. This can be done several ways. Typically, users can select a particular image in the 'Images' viewport, RMB+click to open the context menu and then select the 'Save' option. Users can also select the particular image they wish to save and use the menu bar commands "File > Save Image/Save Sequence". If users want to simply save all changes, they can use the menu bar command "File Save All Images". When closing a modo session, users will be prompted by a dialog to save any unsaved images if not saved prior to closing the application.

Add Texture Commands

The Add Texture commands appear in the main 'Paint' interface tab, under the 'Utilities' subtab of the toolbox, with some additional options available under the 'Hair Tools' subtab. Users can use these as shortcuts to quickly add the appropriate image type to the Shader Tree for the most common Image Mapping options.

Add Color Texture--

This command automatically creates a blank new 'Image Map' texture layer in the Shader Tree, set as 'Diffuse Color' for the 'Effect'. This is the necessary option for painting a surface with the Paint tools. The image is added to whatever Material Group is currently selected in the Shader Tree, regardless of the current foreground item selections. To ensure proper Shader Tree placement, users should first select the Material Group Item where they want the texture layer to reside. It may also be necessary to adjust the projection method in the Texture Locator. The image format will default to whatever is specified by the 'Default Image Format' defined by the user Preference setting but can be changed in the 'New Image' dialog window that opens when the command is invoked.

Add Displacement Texture--

This command automatically creates a blank new OpenEXR format 'Image Map' texture layer in the Shader Tree, set as 'Vector Displacement' for the 'Effect'. This is the most flexible option for image based sculpting with the Sculpting tools. The image is added to whatever Material Group is currently selected in the Shader Tree, regardless of the current foreground item selections. To ensure proper Shader Tree placement, users should first select the Material Group Item where they want the texture layer to reside. It may also be necessary to adjust the projection method in the Texture Locator. For best results, users should accept the default values of the 'New Still' popup window, only adjusting the resolution as necessary.

Add Mask Texture--

This command automatically creates a blank new 'Image Map' texture layer in the Shader Tree, set as 'Layer Mask' for the 'Effect'. Mask Textures are useful in limiting the affect another texture layer has on a shaded surface. The image is added to whatever Material Group is currently selected in the Shader Tree, regardless of the current foreground item selections. To ensure proper Shader Tree placement, users should first select the Material Group Item where they want the texture layer to reside. This layers position in the Shader Tree is important as a 'Layer Mask' will only mask the texture layer directly above it in the tree. It may also be necessary to adjust the projection method in the Texture Locator. The image format will default to whatever is specified by the 'Default Image Format' defined by the user Preference setting but

can be changed in the 'New Image' dialog window that opens when the command is invoked.

Add Fur Texture--

There are four 'Add Fur' commands that automatically creates new, blank OpenEXR format 'Image Map' texture layers in the Shader Tree. The 'Add Fur Texture' option, once added to the Shader Tree, is set as 'Fur Vector' for the 'Effect'. The 'Fur Direction' and Fur Length' options are set to 'Fur Direction' and 'Fur Length' as their 'Effect' respectively. These options provide the most flexible method of styling Fur material layers using the Styling tools. The images are added to whatever Material Group is currently selected in the Shader Tree, regardless of the current foreground item selections. To ensure proper Shader Tree placement, users should first select the Material Group Item where they want the texture layer to reside, such as the group with the 'Fur Material' item. It may also be necessary to adjust the projection method in the Texture Locator. For best results, users should accept the default values of the 'New Still' popup window, only adjusting the resolution as necessary.

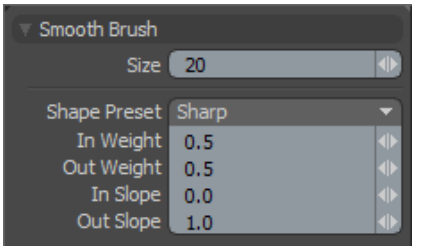
Tool Options

Brush Tips

When users choose a tool for sculpting and painting, it is also necessary to choose an appropriate brush 'Tip' which defines the way color or form is applied to the surface. By default, modo will auto select the 'Smooth Brush' making it easier for users to just start painting or sculpting their surface. Users can easily select an alternate Tip form the selection available. Custom Tool and Tip combinations can also be saved using the 'Save Preset Brush' option found under the 'Brush Tips' icons section of the toolbox. When selected, each Tip has it own set of attributes covered below.

Smooth Brush

The Smooth Brush creates a nice feathered airbrush effect on the canvas. There are two main options for the Smooth Brush, Size and Shape Preset.



Size: This value sets the radius of the brush in pixels from screen space. This is important to understand as zooming the view will change the result of the brush. With the brush size fixed zooming in on the model will decrease the relative scale the the brush whereas zooming out will increase it. Users can interactively adjust the Tip size value by RMB+click and haul in the viewport, adjusting an on-screen display representing the brushes size.

Shape Preset: The popup option includes several preset brush options for varying effects. This setting drives the profile of the airbrush. Users may also adjust the In and Out Slope and Weight setting to produce further customized brushes.

Linear-- will create a straight ramp from the center to the edge which results in a brush profile that looks like a cone.

Smooth-- will create a brush stroke that is opaque in the center and falls off smoothly to the edge of the brush with an S curve shape.

Bulge-- changes the bias of the curve to start out slowly and then ramp more harshly towards the middle, resulting in a brush profile that looks similar to a dome.

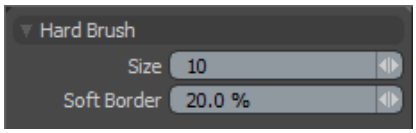
Fat-- brush starts out soft and ramps harshly at the outer edge producing a soft 'U' shaped brush that varies the most under light tablet input pressure.

Sharp-- the falloff is very harsh and steep at the center producing a fine stroke.

Very Sharp-- this preset is nearly identical to the 'Sharp' version with a more extreme falloff for even finer strokes.

Hard Brush

The Hard Brush creates a circular brush with a hard outside edge so that the paint is consistent from the center to the outside edge. There are two settings for the Hard Brush; Size and Soft Border.



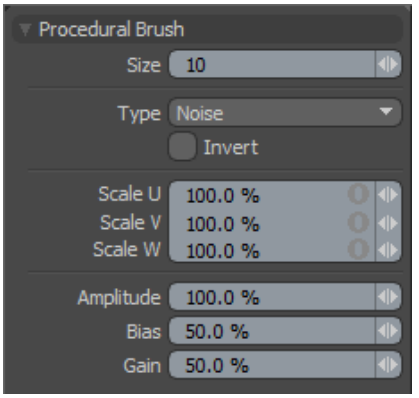
Size: This value sets the radius of the brush in pixels from screen space. This is important to understand as zooming the view will change the result of the brush. With the brush size fixed, zooming in on the model will decrease the relative scale the the brush whereas zooming out will increase it. Users can interactively adjust the Tip size value by RMB+click and haul in the viewport, adjusting an on-screen display representing the brushes size.

Soft Border: This value sets a small number of pixels around the outside of the brush for the falloff to occur. This results in a much more pleasing brush result. The default value of 20% yields a brush stroke that has the appearance of being completely hard edged while providing the paint system a small buffer in which to anti alias the stroke.

Driving the number higher can result in interesting brushes that have a solid inner area and a linear falloff to the edge of the stroke.

Procedural Brush

The Procedural Brush uses mathematical textures to change the amount of paint applied to the surface as the paint stroke is drawn. The options include a pop-up choice to select the procedural texture, a check box to invert the values of the procedural, Scale u, Scale v percentages and an Amplitude setting to control the strength of the effect.



Size: This value sets the radius of the brush in pixels from screen space. This is important to understand as zooming the view will change the result of the brush. With the brush size fixed zooming in on the model will decrease the relative scale the the brush whereas zooming out will increase it. Users can interactively adjust the Tip size value by RMB+click and haul in the viewport, adjusting an on-screen display representing the brushes size.

Type: This popup has choices for Noise, Cellular and Dots. Each of these settings creates a different pattern for the brush based on the respective procedural texture.

Invert: This checkbox option will simply invert the values of the procedural.

Scale U: Changes the size of the procedural pattern along the U axis in UV space.

Scale V: Changes the size of the procedural pattern along the V axis in UV space.

Scale W: Changes the size of the procedural pattern along the W axis in UV space (along normal direction of the UV plane).

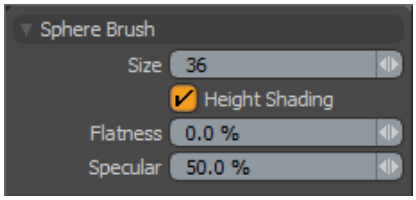
Amplitude: Changes the overall *strength* of the procedural

Bias: The 'Bias' option is like a curve adjustment to the procedural that favors one side of the range over the other, acting much like a gamma curve would affect the shades of an image.

Gain: The 'Gain' option in essence tightens the curve of the bias reducing the shades of gray in the transition. At 0%, there are the maximum number of transition values, attenuating toward 100% where the procedural would be completely hard edged.

Sphere Brush

The Sphere brush creates a gradient effect inside the brush that uses the FG color in the center of the brush fading toward black at the edges with a smooth falloff between them. The resulting effect is that of painting a rounded brush stroke. This can be very useful when working with bump or displacement textures. There are two main modes for the Sphere Brush. Both are useful for color painting but one is intended mostly for displacement or bump painting.



Size: This value sets the radius of the brush in pixels from screen space. This is important to understand as zooming the view will change the result of the brush. With the brush size fixed zooming in on the model will decrease the relative scale the the brush whereas zooming out will increase it. Users can interactively adjust the Tip size value by RMB+click and haul in the viewport, adjusting an on-screen display representing the brushes size.

Height Shading: When using the Sphere brush tip on a flat image map, enabling the 'Height Shading' option shades the brush tip the corresponding shades of gray representing the height of the sphere, producing a cylindrical shading to the brush application.

Flatness: The default value of 0% produces a fully round sphere attenuating toward 100% producing a completely flat brush, just like the normal Hard Brush.

Specular: The 'Specular' option controls the amount of faux specular highlight on the brush tips shading. At 0% there is no specular, attenuating toward 100% with full specular shading.

Image Brush (*and* Stamp Brush)

Like applying a sticker, users can use the Image Brush to apply bitmap texture images to a surface when either painting or sculpting. Depending on mode, users can control the placement of those images. When using the Image Brush in Stamp mode, the brush no longer uses a procedural stroke, rather the click and drag controls the diameter and rotation of the image inside the brush. To apply the *image* simply release the mouse or tablet button. When dragging the Image brush moving the tablet or mouse away from the original click location will increase the size of the brush. Dragging around in a circular motion adjusts the rotation of the brush. When you first click and drag a small handle appears. This handle is the visual cue as you edit the size and rotation of the brush. When in Roller mode, modo continuously tiles the image in the direction of the stroke. Normal mode sprays images, similar to the behavior of other brush tips.

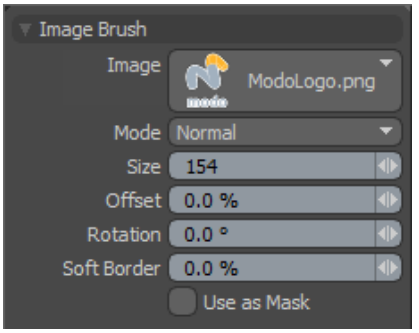


Image: This popup allows the user to select the bitmap image to be used as the brush tip.

Mode: Options for 'Normal', 'Roller' and 'Stamp'.

Normal-- Normal mode acts as any other paint brush with the image controlling the effect.

Roller-- Roller mode forces the brush to act similar to a roller painting tool where the image is revealed the further you drag the brush. This mode also causes the brush to rotate with your stroke.

Stamp-- Stamp mode causes the brush to "stick" down where you click. Dragging the mouse then controls the size and rotation of the brush. When you release the mouse the image brush is "stamped" down at that location scale and orientation.

Size: This value sets the radius of the brush in pixels from screen space. This is important to understand as zooming the view will change the result of the brush. With the brush size fixed zooming in on the model will decrease the relative scale the the brush whereas zooming out will increase it. Users can interactively adjust the Tip size value by RMB+click and haul in the viewport, adjusting an on-screen display representing the brushes size.

Offset: This control is particularly useful when working with the sculpting tools. The percentage value allows you to re-map the "zero-point" of the image. For instance, if your image was a grayscale ramp from mid-gray to full-white, the edges of the brush would create a "step" since the sculpting tools would have a 50% gray input at the edges of the brush. In this case you can set the Offset value to 50% and now the mid-gray value is considered the zero-point and will no longer cause any step effect.

Rotation: Use this control to effect the orientation of the brush.

Soft Border: This causes a smooth fade of the image to the edges of the brush. The percentage value determines how far from the edge of the brushes diameter the fade will begin. If you set a value of 10% the fade will begin 10% in from the outermost edge of the brush. When this value is non-zero a second ring will appear on the brush to indicate where the soft border begins.

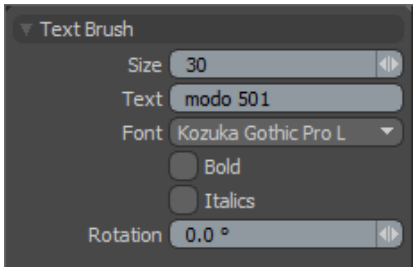
Use as Mask: This control will use the dark areas of an image as alpha.

Text Brush

The Text brush provides a quick and easy way to use many of the fonts on your system as a brush while painting or sculpting, easily placing words, letters and numeric call-outs on textured surfaces. The tool includes a pop-up for choosing the desired font as

well as a text input field to set the text string to be used as the brush.

To use the Text brush, first enter the desired text string and choose your font from the font popup. At this time you can also choose Bold or Italic styles. Make sure to have a painting or sculpting tool selected as well as the brush and then LMB+click and drag in the 3D view. As you drag to the right or left the 'Size' value of the brush will increase. Dragging around in a circle sets the 'Rotation' (orientation) of the brush. At this time users can continue to scale the text brush by simply moving the mouse further away from the center of the brush. When you release the mouse the brush will be *projected* onto the surface.



Size: The size of the projected Text string is determined interactively by the user dragging in the viewport. The Size *value* represents the radius of the circle generated by the tool when dragging, calculated in screen pixels. The Text tool always creates the text string from its center.

Text: This input field allows users to enter text that will be generated when the users drags in the viewport. While many the entry field will hold many characters, the text tool is best suited to no more than a few words.

Font: The 'Font' field is a button that, when LMB+clicked, displays all currently active TrueType and OpenType fonts on the users system. Users can choose a specific font from the fly-out menu. Once selected, the chosen face will be used with the Text brush.

Bold/Italics: These checkboxes allow users to produce faux 'Bold' and 'Italic' text faces when fonts of that type are not directly available.

Rotation: The rotation angle of the text (like the size) is determined interactively by the user dragging in the viewport. The angle is calculated from the point of the initial mouse click in the viewport to the position of the cursor when the user releases the button projecting the text block onto the texture.

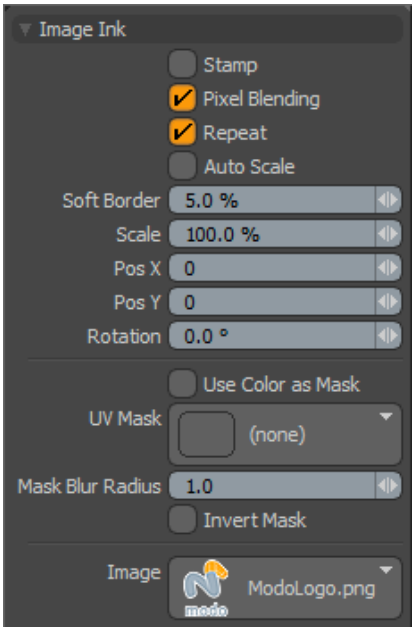
Inks

The 'Ink' functions in modo provide a way for users to modulate brush strokes as they are applied. This can apply to affecting colors when painting, or affecting the strength of the sculpting tools. All three options can be utilized individually, or simultaneously in combination to create interesting effects. Found in the 'Paint' and 'Sculpt' toolboxes of the 'Paint' interface tab. To use, simply select the 'Ink' button in the interface and adjust the settings in the viewport or properties panel as necessary.

Image Ink

The Image Ink function allows user the ability to place a bitmap image in the viewport (sort of like a floating window) and then using the brush to control the application, users can paint through the image, applying it to their objects surface wherever they stroke. When Image Ink is first enabled, users will need to LMB+click in the viewport to activate the tools widget. If an image were selected prior to this action, its height would fill the viewport with the image. The widget has a number of handles represented as small circles that allow users to adjust the projection plane directly in the viewport. The center handle positions it, while the right handle scales the widget, and the top handle will rotate it. Simply LMB+click on the handle and drag to modify. Users may also use the keyboard shortcuts 'E' to rotate clockwise, and 'R' to rotate counter-clockwise. Scaling is done by pressing 'D' to go down and 'F' to go up. Interactive scaling with the viewport is achieved with the 'Auto Scale' option retaining the scale of the widget over the geometry as the viewport scale itself is modified.

Image Ink also allows users to adjust its standard controls for 'Scale', 'Rotate' and 'Position' as well as a Soft Border percentage and other options by selecting the options found in the tool properties viewport. Users can gain quick access to tool properties by pressing the 'K' key to activate the properties pop-over.



Stamp: Stamp mode checkbox will apply the entire image ink image in one quick stamp. This option requires the use of Projection Paint. This can be a very useful method for applying decal textures in very precise positions on a 3D mesh as well as for stamping down texture elements such as scars. The stamp option can even be useful for quickly stamping down an image repeatedly to texture a model completely. By using some Soft Border and changing the rotation of the image ink you can create very interesting texture effects quite quickly.

Pixel Blending: The Pixel Blending option will smooth your image when it is scaled beyond its maximum native resolution, reducing visibly pixelated edged.

Repeat: This option will cause the image ink to continually repeat once painting has extended beyond the image boundary.

Auto Scale: The Auto Scale option will scale the image ink frame relative to the

viewport being scaled. Useful for applying a consistent texture across the surface of the model. Whenever you zoom in and out of the object, Auto Scale will make sure that relationship remains the same.

Soft Border: The Soft Border option is a percentage that creates an opacity ramp from the center of the image ink to the edges. As you paint through the image plane (Projection Mode) the Soft Border amount will cause the edges to feather, creating a soft blend from the edge of the ink boundary. This is also very useful when working in Stamp mode. With a tiling image as you ink and a soft border value of approximately 50%, you can repeatedly stamp down a texture across the surface of the mesh to create very quick detail mapping.

Scale: The Scale control allows you to change the size of the image relative to the object in your viewport. Reducing the value results in the image appearing smaller relative to the model and increasing the value will increase the relative scale of the image. When using Projection Paint the scale handle is the square handle on the right-hand side of the image plane.

Pos X/Pos Y: These two values represent the position of the center of the image ink frame within the viewport. Values are pixel coordinates with 0,0 being the upper left corner. Each time the Image Ink function is activated, your image frame will be centered within the viewport. The can also interactively position the image ink frame using the circle in the center.

Rotation: The Rotate value tilts the image. In the case of standard 'Fast UV' painting, this is essentially rotating the image in UV space. With Projection Painting the image plane is rotated in screen space. The circle handle at the top of the image plane can be dragged right and left to adjust rotation interactively.

Use Color as Mask: The 'Use Color as Mask' option will use the brightness (Luminosity) of the current Image Ink image as an alpha mask when painting onto the object. So instead of painting the image itself onto your object, the image becomes like a virtual silkscreen allowing the color of the brush to only go through the dark areas of the image.

UV Mask: The UV Mask option provides users the ability to use a secondary image as a painting mask. The only requirement is the masking image must use the same UV Map as the image being painted. This provides users the ability, for example, to bake an Occlusion map image and then use that as a mask to paint rust into the crevices of a old vehicle.

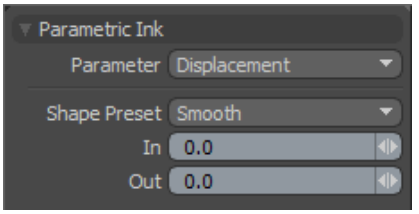
Mask Blur Radius: This option allows users to blur or soften the image applied as a UV Mask. The greater the value the softer the mask transition between light and dark.

Invert Mask: This option inverts the RGB or Gray values of the mask reversing the application of the mask, once enabled, areas that were masked will become paintable, and areas that were paintable will become masked.

Image: Use this popup image loader to load and select images for use with the Image Ink function.

Parametric Ink

The Parametric Ink provides a method to modulate the ink between the foreground (FG) and background (BG) values based on parameters including Slope, Altitude, Incidence Angle, Pressure and Tangent Pressure. The first four parameters are attributes of the surface on which you are painting and the last two are based on tablet input device feedback.



Parameter: The various parameter options allow uses to choose how the Parametric function modulates the painted brush stroke--

Displacement- The Displacement setting allows users to modulate between the foreground and background color based on the amount of displacement applied to a surface, providing users a means to paint in the cavities of surfaces. When working with procedural texture displacement, users will first need to convert the procedural to an Image Map using the modo baking functions to produce results with this option.

Slope- The slope setting changes the ink from the FG value to the BG value as the geometric normals change. If we look at a sphere in "world coordinates" the top of the sphere has normals that point directly upward, the bottom points directly down and at the equator normals point straight out. These "normals" would range from 0 to 90.

Altitude- Based on the bounding area of the mesh the colors will change using the FG color at the very bottom of the model and the BG color as you reach the top of the mesh.

Incidence Angle- This is the angle of the surface of the model as it relates to the viewport in which you are painting. Polygons that face the view will shade using the FG color, whereas polygon normals that are perpendicular to the view will shade with the BG ink color.

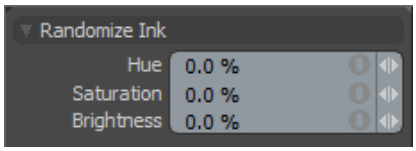
Pressure- When painting with a tablet, 100% pressure will use the FG color for the ink and as you use less pressure it will fade to the BG color.

Tangent Pressure- This allows you to change between FG and BG colors by rotating the tablet stylus.

Shape Preset: Similar in option to the 'Smooth Brush', the Shape Preset options allow users to fine tune the application of the strokes when using the Parametric ink function. Users can select from the various presets, or adjust the 'In' and 'Out' values to further customize the brush shape.

Randomized Ink

The Random Ink option allows the paint tools ink color to randomly vary as you paint on the mesh. There are three input fields for the Random Ink; Hue, Saturation and Brightness. The default values of 0% result in no random variation to the ink. By increasing any of the values you can randomize the ink in Hue, Saturation, and/or Brightness or any combination therein. Setting any of these values to 100% will cause the ink to randomly swing plus or minus 100% from its original value. Setting 'Brightness' and 'Saturation' options to 10-30% can be very useful when creating noise maps to *rough up* a surface.



Hue: Modifying this value will result in variations of color for the individual paint dabs when stamped onto the surface. 0% effectively disables this setting, ramping toward 100% with the maximum amount of hue variation in the stroke.

Saturation: Modifying this value will result in variations of color intensity for the individual paint dabs when stamped onto the surface. 0% effectively disables this setting, ramping toward 100% with the maximum amount of saturation variation in the stroke.

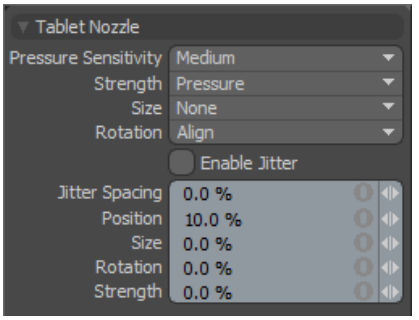
Brightness: Modifying this value will result in variations of luminosity for the individual paint dabs when stamped onto the surface. 0% effectively disables this setting, ramping toward 100% with the maximum amount of brightness variation in the stroke.

Tablet Nozzle

Tablet Nozzle

If you had a virtual microscope in modo that would allow you to look at how brush strokes are created, they aren't actually continuous lines, but a series of tiny 'dabs' sprayed across the canvas. The 'Tablet Nozzle' option gives users greater control over these dabs, as well as over a tools pressure sensitivity when using a Pen and Tablet as an input device. (Make sure to set the input preference to 'Tablet' if you wish to work this way. The setting can be found in the menu bar under "System > Preferences" in the "Input > Remapping > Input Device" option field). The Tablet Nozzle also provides options to vary the positions and scales of the paint dabs when stamped down, providing some randomization to that spray of dabs from your brush. There are five

parameters; Spacing, Position, Size, Rotation and Strength.



Tablet Nozzle--

Pressure Sensitivity: This options controls the brushs sensitivity to changes in tablet pressure when painting strokes on the canvas, the 'Low' option is less sensitive to variation of pressure, while 'Medium' and 'High' are increasingly more sensitive.

Strength: The 'Strength' option defines what action controls the *strength* (and therefore density) of the stroke. 'None' disables strength modulation, the 'Pressure' option controls the density of the stroke based on the pens pressure input value. The 'Speed' option controls the density of the stroke based on how fast or slow the cursor moves over the surface. The 'Both' option combines the pressure sensitivity of the tables input with the speed of the cursor to control the final strokes strength.

Size: The 'Size' options defines what action controls the *scaling* of the stroke dabs providing users the ability to vary strokes width. 'None' disables scaling of the tip, the 'Pressure' option controls the scaling of the stroke based on the pens pressure input value. The 'Speed' option controls the scaling of the stroke based on how fast or slow the cursor moves over the surface. The 'Both' option combines the pressure sensitivity of the tables input with the speed of the cursor to control the final strokes size.

Rotation: The 'Rotation' option controls the facing direction of paint dabs when placed on the canvas. 'None' disables rotation of the dabs, aligning them vertically to the viewport when brushed to the canvas. The 'Align' option aligns the dab to the vector direction the stroke is traveling, meaning each dab is rotated to flow along the direction the brush stroke is moving.

Enable Jitter: When this option is enabled, uses can adjust the values for the various options to control randomizations the the brush dabs.

Jitter Spacing: modo brush dabs spray with tiny but consistent spaces between each dab. This value will randomize that spacing. Higher values will cause the dabs to clump together leaving noticeable gaps, while smaller values will vary the positions only slightly.

Position: This value randomize the placement of the dabs away from the brushstroke. Higher values will cause the dabs to 'splatter' around the stroke, while smaller values will vary the positions only slightly creating uneven natural-looking strokes.

Size: This value will randomize the size of each dab as it is placed on the canvas. Higher values will create blobby clumped strokes while smaller values will reduce the clumping making strokes with randomly varying thickness.

Rotation: This value will randomly rotate each dab. Higher values will randomize the rotation of each dab to a higher degree, and obviously lower values vary the rotation in smaller increments. While not too useful for the perfectly round dabs of the smooth and hard brushes, when used with image maps and the preset brush, one can quickly add an overall texture to a surface with only one image, avoiding obvious visual repetition.

Strength: This value will randomize the strength of each dab, essentially varying the pressure applied to each one. This will result in different degrees of opacity for each dab. Higher values will very the pressure to a higher degree, while lower values will vary the pressure in smaller amounts.

Play around with different combination's and you can create dramatic variations of simple brushes with just a few settings easily creating special effect brushes that will complement the many already present in modo.

Sculpting Tools

Sculpting Brush Reference

Sculpting is an fast and intuitive method for building and refining 3D models. The brushes for sculpting can be found in the 'Paint' tab of the interface, where a focused collection of tools and useful commands are located dedicated to the painting and sculpting functionality. The tools may also be called from the menu bar, under the "Texture > Sculpt Tools" menu. Invoking any of the tools will display the associated properties in the appropriate viewport panel. To sculpt, users will need to select both a Tool (*brush*) and a 'Tip' which designates how the tool affects the geometry. When working with a mouse, users can simply LMB+drag over the surface to apply the stroke. The LMB click is applied automatically when using a pressure sensitive stylus and tablet. Users may also RMB+click and drag in the viewport to interactively scale the size of the brush tip.

Sculpting Tools			Sculpting Options
Push	Inflate	Spin	Lock Borders
Smooth	Smudge	Emboss	

Carve

Move

Attenuate

Flatten

Center Pinch

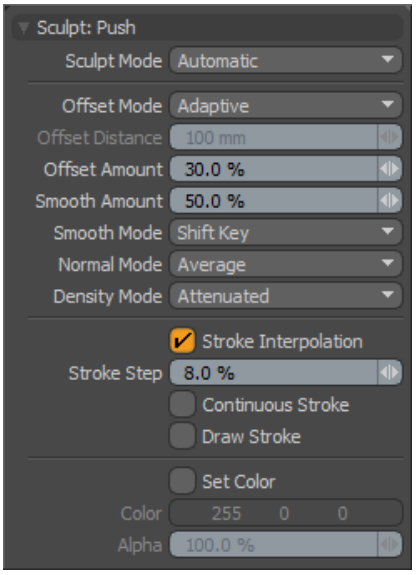
Mask

Fold

Tangent Pinch

Push

The Push tool moves vertices away from the surface along the calculated normal direction of the polygon. The default method is to use the normal calculated from the center of the brush stroke. The effect is that of pulling the surface of the model outward from the surface as if the sculpt tool were simply moving vertices outward according to the profile of the brush. Pressing the 'Ctrl' key down while using the tool will send the surface downwards creating a valley rather than a hill. Pressing 'Shift' while using the tool will enable the 'Smooth' function. The 'Push' tool can be found under the 'Paint' tab for the interface in the 'Sculpt' submenu of the toolbox. The tool can also be called from the menu bar under the "Texture > Sculpt Tools" menu. Additionally, the tool can be combined with the various 'Tips' as well as the 'Inks' and 'Nozzle' to further customize the way the brush affects the surface. When activated the following attributes are displayed in the Properties panel.



Sculpt: Push--

Sculpt Mode: The 'Sculpt Mode' determines which method of sculpting is to be used. By default the setting is 'Automatic' which will perform mesh based sculpts when there are no displacement images and automatically switch to image based sculpting when there an image is available. There are times when you may want to force modo into mesh based sculpting even if you have an image applied for displacement. This can be very helpful for blocking out shapes and form, as mesh sculpting is only effecting the cage vertices of the base geometry.

Automatic-- chooses between mesh and image sculpting depending on availability of displacement image map texture.

Mesh-- forces the tools to perform mesh sculpting only.

Image Map (Details)-- forces the tools to perform image sculpting only.

Offset Mode: modo provides two methods for calculating the way the tool moves the surface when brushing.

Adaptive: The adaptive mode will adapt to the surface that is being brushed on based on the setting for 'Offset Amount' and the size of the brush.

Absolute: When in 'Absolute' mode, the 'Offset Distance' setting will become active and users may enter a specific distance value the brush will conform to regardless of brush scale.

Offset Distance: This control is disabled unless the 'Offset Mode' is set to 'Absolute'. When active, the value determines specifically how far to displace the surface.

Offset Amount: Active only when the 'Offset Mode' is set to 'Adaptive', this percentage value modulates the extent of the brushstroke. The total scale of the sculpt deformation will be equal to the brush size times the offset amount percentage.

(Note: This is a very important value for controlling sculpted detail and as such it can be adjusted interactively by using the MMB+click and hauling left or right.)

Smooth Amount: The Smooth tool is such a useful method for controlling sculpt detail it is available with almost every sculpting tool in modo and can be accessed by simply holding the Shift key down while sculpting. The Smooth amount attribute allows you to control the amount of smoothing applied when using the Shift qualifier with that specific tool. This is a convenience feature implemented specifically to enhance the sculpting workflow.

Smooth Mode: modo provides alternate methods for applying smoothing while sculpting-

Shift Key-- The smooth mode is enabled when the 'Shift' key is depressed.

Low Pressure-- When using a pressure sensitive tablet, users may apply the 'Low Pressure' option, where light sculpting strokes will have a smoothing affect on the surface. Once the pressure threshold is exceeded, the smoothing effect will attenuate toward the full pressure effect where the brush will work normally as determined by the tool providing a dual function to the brush.

Uniform-- When set to 'Uniform', smoothing is always on while sculpting, thus making softer strokes.

Normal Mode: This option controls the method used for displacing the sculpted vertices. The various options include:

Brush Center-- uses the geometry at the very center of the brush to calculate the direction along which to displace the mesh.

Vertex-- displaces each vertex along its own normal.

Average-- calculates an average normal for all the vertices inside the brush.

Screen-- displaces the vertices directly towards the active viewport.

Mouse Down-- uses the normal of the mesh surface directly under the mouse when the LMB is clicked. All displacement is then carried out according to that vector.

X,Y,Z-- allows you to pick a world axis along which to displace the vertices.

Density Mode: This control determines how the sculpt stroke effects areas of the mesh that it has already covered. For instance, brushing back and forth will cause the sculpt tool to run back over itself which can have a number of different effects. This controls the sculpt during a single stroke. Once the stylus comes off the tablet or you release the LMB the density control is reset.

Additive-- continues to add to the displacement each time the brush crosses over its own path with no maximum cap to govern the sculpt.

Attenuate-- continues to add to the displacement as the brush crosses over its own stroke until the displacement reaches the Offset Distance or Amount value.

Max Stroke-- enforces a maximum offset based on the largest displacement value of the current stroke.

Max Map-- caps the sculpt effect at the maximum amount in the current displacement image.

Stroke Interpolation: When the option 'Interpolation' is enabled, modo will add additional sample points and filter out small variances in pressure and position resulting in smoother strokes.

Stroke Step: Generally, paint strokes are not a continuous lines, but a series of dots or dabs placed on the canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to 8%. So for an 64 pixel sized brush, dabs would be placed approx every 7 pixels. Users can increase or reduce this value for a variety of effects. Increasing it toward 100% will produce individual dots, rather than what looks like a stroked line.

Continuous Stroke: When this option is enabled, modo will continuously draw a stroke like a line, instead of as individual dabs. modo will draw the stroke as a linear segments between each interpolation step dab so increasing the scale of the interpolation steps will reveal the linear nature of the continuous stroke option.

Default 10% 'Interpolation Step' produces smooth stroke

Interpolation Step set to 200% reveals individual dabs

200% Interpolation Step with Continuous Stroke enabled

Draw Stroke: When *enabled*, this option simply draws a line in the interface defined by the path for the brush stroke.

Set Color: When utilizing Multi-Res sculpting, and enabling the 'Set Color' option, users can simultaneously sculpt the mesh and paint into an Image Map layer using the same settings of the sculpting tool itself.

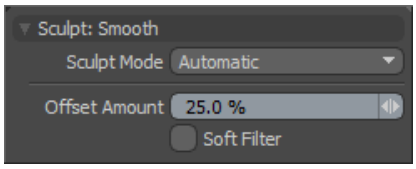
Color: This value determines the color applied to the image map when sculpting.

Alpha: This value determines the opacity of the stroke applied to the image map.

Smooth

When sculpting, the 'Smooth' tool provides an important function that evens out bumpy, irregular surfaces. By averaging the positions of vertices under the brush tip, surface inconsistencies and detail will disappear producing a result similar to melting (as if the surface were made of wax). The Smooth tool is essential to the sculpting workflow, so much so that it is always available when using any of the other sculpting tools. By simply holding down the Shift key when sculpting, users can access the Smooth tool while any other tool is active. The 'Smooth' tool itself can be found under the 'Paint' tab for the interface in the 'Sculpt' submenu of the toolbox. The tool can also be called from the menu bar under the "Texture > Sculpt Tools" menu. Additionally, the tool can be combined with the various 'Tips' as well as the 'Inks' and 'Nozzle' to further customize the way the brush affects the surface. When activated the following attributes are displayed in the Properties panel.

Sculpt: Smooth--



Sculpt Mode: The 'Sculpt Mode' determines which method of sculpting is to be used. By default the setting is 'Automatic' which will perform mesh based sculpts when there are no displacement images and automatically switch to

image based sculpting when there an image is available. There are times when you may want to force modo into mesh based sculpting even if you have an image applied for displacement. This can be very helpful for blocking out shapes and form, as mesh sculpting is only effecting the cage vertices of the base geometry.

Automatic-- chooses between mesh and image sculpting depending on availability of displacement image map texture.

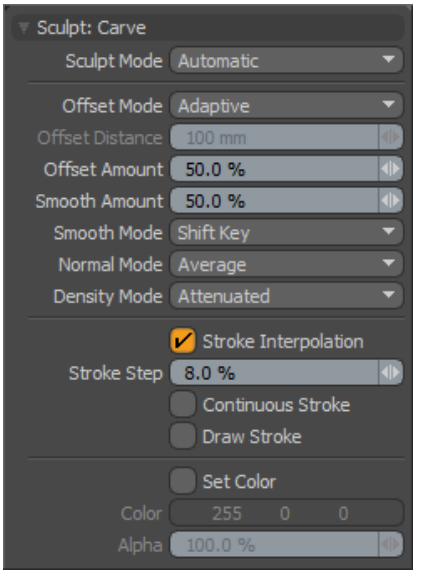
Mesh-- forces the tools to perform mesh sculpting only.

Image Map (Details)-- forces the tools to perform image sculpting only.

Offset Amount: The 'Offset Amount' value determines the strength of the smoothing applied to the surface.

Soft Filter: This option only applies to image based sculpting and will *soften* the image map by filtering out small details producing a different result over the standard 'Smooth' function.

Carve



The Carve tool works by pushing the surface inward creating a valley of a fixed maximum amount per stroke. This simulates carving away from the surface and is very useful for scoring or scratching a surface. Using a 'Sharp' brush tip will also provide users the ability to easily create wrinkles. Pressing the 'Ctrl' key down while using the tool will send the surface upwards creating a hill rather than a valley. Pressing 'Shift' while using the tool will enable the 'Smooth' function. The 'Carve' tool can be found under the 'Paint' tab for the interface in the 'Sculpt' submenu of the toolbox. The tool can also be called from the menu bar under the "Texture > Sculpt Tools" menu. Additionally, the tool can be combined with the various 'Tips' as well as the 'Inks' and 'Nozzle' to further customize the way the brush affects the surface. When activated the following attributes are displayed in the Properties panel.

Sculpt: Carve

Sculpt Mode: The 'Sculpt Mode' determines which method of sculpting is to be used. By default the setting is 'Automatic' which will perform mesh based sculpts when there are no displacement images and automatically switch to image based sculpting when

there an image is available. There are times when you may want to force modo into mesh based sculpting even if you have an image applied for displacement. This can be very helpful for blocking out shapes and form, as mesh sculpting is only effecting the cage vertices of the base geometry.

Automatic-- chooses between mesh and image sculpting depending on availability of displacement image map texture.

Mesh-- forces the tools to perform mesh sculpting only.

Image Map (Details)-- forces the tools to perform image sculpting only.

Offset Mode: modo provides two methods for calculating the way the tool moves the surface when brushing.

Adaptive: The adaptive mode will adapt to the surface that is being brushed on based on the setting for 'Offset Amount' and the size of the brush.

Absolute: When in 'Absolute' mode, the 'Offset Distance' setting will become active and users may enter a specific distance value the brush will conform to regardless of brush scale.

Offset Distance: This control is disabled unless the 'Offset Mode' is set to 'Absolute'. When active, the value determines specifically how far to displace the surface.

Offset Amount: Active only when the 'Offset Mode' is set to 'Adaptive', this percentage value modulates the extent of the brushstroke. The total scale of the sculpt deformation will be equal to the brush size times the offset amount percentage.

(Note: This is a very important value for controlling sculpted detail and as such it can be adjusted interactively by using the MMB+click and hauling left or right.)

Smooth Amount: The Smooth tool is such a useful method for controlling sculpt detail it is available with almost every sculpting tool in modo and can be accessed by simply holding the Shift key down while sculpting. The Smooth amount attribute allows you to control the amount of smoothing applied when using the Shift qualifier with that specific tool. This is a convenience feature implemented specifically to enhance the sculpting workflow.

Smooth Mode: modo provides alternate methods for applying smoothing while sculpting-

Shift Key-- The smooth mode is enabled when the 'Shift' key is depressed.

Low Pressure-- When using a pressure sensitive tablet, users may apply the 'Low Pressure' option, where light sculpting strokes will have a smoothing affect on the surface. Once the pressure threshold is exceeded, the smoothing effect will

attenuate toward the full pressure effect where the brush will work normally as determined by the tool providing a dual function to the brush.

Uniform-- When set to 'Uniform', smoothing is always on while sculpting, thus making softer strokes.

Normal Mode: This option controls the method used for displacing the sculpted vertices. The various options include:

Brush Center-- uses the geometry at the very center of the brush to calculate the direction along which to displace the mesh.

Vertex-- displaces *each vertex* along its own normal.

Average-- calculates an average normal for all the vertices inside the brush.

Screen-- displaces the vertices directly towards the active viewport.

Mouse Down-- uses the normal of the mesh surface directly under the mouse when the LMB is clicked. All displacement is then carried out according to that vector.

X,Y,Z-- allows you to pick a world axis along which to displace the vertices.

Density Mode: This control determines how the sculpt stroke effects areas of the mesh that it has already covered. For instance, brushing back and forth will cause the sculpt tool to run back over itself which can have a number of different effects. This controls the sculpt during a single stroke. Once the stylus comes off the tablet or you release the LMB the density control is reset.

Additive-- continues to add to the displacement each time the brush crosses over its own path with no maximum cap to govern the sculpt.

Attenuate-- continues to add to the displacement as the brush crosses over its own stroke until the displacement reaches the Offset Distance or Amount value.

Max Stroke-- enforces a maximum offset based on the largest displacement value of the current stroke.

Max Map-- caps the sculpt effect at the maximum amount in the current displacement image.

Stroke Interpolation: When the option 'Interpolation' is *enabled*, modo will add additional sample points and filter out small variances in pressure and position resulting in smoother strokes.

Stroke Step: Generally, paint strokes are not a continuous lines, but a series of *dots* or *dabs* placed on the canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to 8%. So for an 64 pixel sized brush, dabs would be placed approx every 7 pixels. Users can increase or reduce this

value for a variety of effects. Increasing it toward 100% will produce individual dots, rather than what looks like a stroked line.

Continuous Stroke: When this option is enabled, modo will continuously draw a stroke like a *line*, instead of as individual dabs. modo will draw the stroke as a linear segments between each interpolation step dab so increasing the scale of the interpolation steps will reveal the linear nature of the continuous stroke option.

Default 10% 'Interpolation Step' produces smooth stroke

Interpolation Step set to 200% reveals individual dabs

200% Interpolation Step with Continuous Stroke enabled

Draw Stroke: When enabled, this option simply draws a line in the interface defined by the path for the brush stroke.

Set Color: When utilizing Multi-Res sculpting, and enabling the 'Set Color' option, users can simultaneously sculpt the mesh and paint into an Image Map layer using the same settings of the sculpting tool itself.

Color: This value determines the color applied to the image map when sculpting.

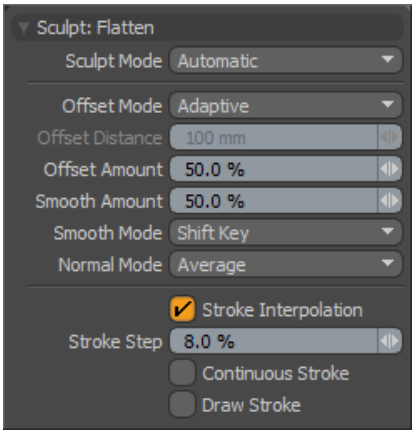
Alpha: This value determines the opacity of the stroke applied to the image map.

Flatten

The Flatten tool massages the mesh to a common flat plane. By default the tool works based on the center of the brush (Normal Mode). This creates an ever adapting plane that changes orientation as the brush moves over the surface of the mesh. In this mode, using deliberate brush strokes or multiple strokes over the same area works well to create more consistent surfaces. If you wish to create a plateau based on an exact position, set the 'Normal Mode' to 'Mouse Down'. With this setting on, the tool will read the orientation of the surface when you click down and use that as reference when moving the surrounding geometry. Pressing the 'Ctrl' key down while using the tool will send the surface upwards instead of flattening it. Pressing 'Shift' while using the tool will enable the 'Smooth' function. The 'Flatten' tool can be found under the 'Paint' tab for the interface in the 'Sculpt' submenu of the toolbox. The tool can also be called from the menu bar under the "Texture > Sculpt Tools" menu. Additionally, the tool can be combined with the various 'Tips' as well as the 'Inks' and 'Nozzle' to

further customize the way the brush affects the surface. When activated the following attributes are displayed in the Properties panel.

Sculpt: Flatten--



Sculpt Mode: The 'Sculpt Mode' determines which method of sculpting is to be used. By default the setting is 'Automatic' which will perform mesh based sculpts when there are no displacement images and automatically switch to image based sculpting when there an image is available. There are times when you may want to force modo into mesh based sculpting even if you have an image applied for displacement. This can be very helpful for blocking out shapes and form, as mesh sculpting is only effecting the cage vertices of the base geometry.

Automatic-- chooses between mesh and image sculpting depending on availability of displacement image map texture.

Mesh-- forces the tools to perform mesh sculpting only.

Image Map (Details)-- forces the tools to perform image sculpting only.

Offset Mode: modo provides two methods for calculating the way the tool moves the surface when brushing.

Adaptive: The adaptive mode will adapt to the surface that is being brushed on based on the setting for 'Offset Amount' and the size of the brush.

Absolute: When in 'Absolute' mode, the 'Offset Distance' setting will become active and users may enter a specific distance value the brush will conform to regardless of brush scale.

Offset Distance: This control is disabled unless the 'Offset Mode' is set to 'Absolute'. When active, the value determines specifically how far to displace the surface.

Offset Amount: Active only when the 'Offset Mode' is set to 'Adaptive', this percentage value modulates the extent of the brushstroke. The total scale of the sculpt deformation will be equal to the brush size times the offset amount percentage.

(Note: This is a very important value for controlling sculpted detail and as such it can be adjusted interactively by using the MMB+click and hauling left or right.)

Smooth Amount: The Smooth tool is such a useful method for controlling sculpt detail

it is available with almost every sculpting tool in modo and can be accessed by simply holding the Shift key down while sculpting. The Smooth amount attribute allows you to control the amount of smoothing applied when using the Shift qualifier with that specific tool. This is a convenience feature implemented specifically to enhance the sculpting workflow.

Smooth Mode: modo provides alternate methods for applying smoothing while sculpting-

Shift Key-- The smooth mode is enabled when the 'Shift' key is depressed.

Low Pressure-- When using a pressure sensitive tablet, users may apply the 'Low Pressure' option, where light sculpting strokes will have a smoothing affect on the surface. Once the pressure threshold is exceeded, the smoothing effect will attenuate toward the full pressure effect where the brush will work normally as determined by the tool providing a dual function to the brush.

Uniform-- When set to 'Uniform', smoothing is always on while sculpting, thus making softer strokes.

Normal Mode: This option controls the method used for displacing the sculpted vertices. The various options include:

Brush Center-- uses the geometry at the very center of the brush to calculate the direction along which to displace the mesh.

Vertex-- displaces *each vertex* along its own normal.

Average-- calculates an average normal for all the vertices inside the brush.

Screen-- displaces the vertices directly towards the active viewport.

Mouse Down-- uses the normal of the mesh surface directly under the mouse when the LMB is clicked. All displacement is then carried out according to that vector.

X,Y,Z-- allows you to pick a world axis along which to displace the vertices.

Stroke Interpolation: When the option 'Interpolation' is *enabled*, modo will add additional sample points and filter out small variances in pressure and position resulting in smoother strokes.

Stroke Step: Generally, paint strokes are not a continuous lines, but a series of dots or dabs placed on the canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to 8%. So for an 64 pixel sized brush, dabs would be placed approx every 7 pixels. Users can increase or reduce this value for a variety of effects. Increasing it toward 100% will produce individual dots, rather than what looks like a stroked line.

Continuous Stroke: When this option is enabled, modo will continuously draw a stroke like a *line*, instead of as individual dabs. modo will draw the stroke as a linear segments between each interpolation step dab so increasing the scale of the interpolation steps will reveal the linear nature of the continuous stroke option.

Default 10% 'Interpolation Step' produces smooth stroke

Interpolation Step set to 200% reveals individual dabs

200% Interpolation Step with Continuous Stroke enabled

Draw Stroke: When *enabled*, this option simply draws a line in the interface defined by the path for the brush stroke.

Fold

The 'Fold' tool creates a "hill and valley" effect by pushing vertices at the center of the brush in and pulling vertices at the edge of the brush outward. As with all sculpting tools the profile of the brush is very important in the final look of the surface. Pressing the 'Ctrl' key down while using the tool will invert the effect forcing the center outward and the edges inward. Pressing 'Shift' while using the tool will enable the 'Smooth' function. The 'Fold' tool can be found under the 'Paint' tab for the interface in the 'Sculpt' submenu of the toolbox. The tool can also be called from the menu bar under the "Texture > Sculpt Tools" menu. Additionally, the tool can be combined with the various 'Tips' as well as the 'Inks' and 'Nozzle' to further customize the way the brush affects the surface. When activated the following attributes are displayed in the Properties panel.

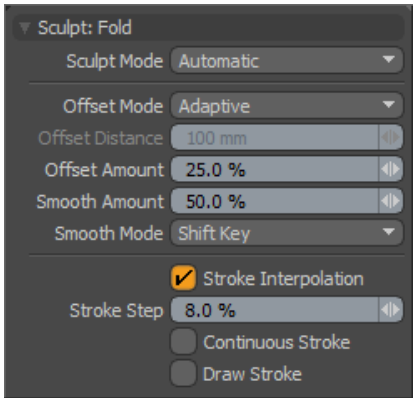
Sculpt: Fold--

Sculpt Mode: The 'Sculpt Mode' determines which method of sculpting is to be used. By default the setting is 'Automatic' which will perform mesh based sculpts when there are no displacement images and automatically switch to image based sculpting when there an image is available. There are times when you may want to force modo into mesh based sculpting even if you have an image applied for displacement. This can be very helpful for blocking out shapes and form, as mesh sculpting is only effecting the cage vertices of the base geometry.

Automatic-- chooses between mesh and image sculpting depending on availability of displacement image map texture.

Mesh-- forces the tools to perform mesh sculpting only.

Image Map (Details)-- forces the tools to perform image sculpting only.



Offset Mode: modo provides two methods for calculating the way the tool moves the surface when brushing.

Adaptive: The adaptive mode will adapt to the surface that is being brushed on based on the setting for 'Offset Amount' and the size of the brush.

Absolute: When in 'Absolute' mode, the 'Offset Distance' setting will become active and users may enter a specific distance value the brush will conform to regardless of brush scale.

Offset Distance: This control is disabled unless the 'Offset Mode' is set to 'Absolute'. When active, the value determines specifically how far to displace the surface.

Offset Amount: Active only when the 'Offset Mode' is set to 'Adaptive', this percentage value modulates the extent of the brushstroke. The total scale of the sculpt deformation will be equal to the brush size times the offset amount percentage.

(Note: This is a very important value for controlling sculpted detail and as such it can be adjusted interactively by using the MMB+click and hauling left or right.)

Smooth Amount: The Smooth tool is such a useful method for controlling sculpt detail it is available with almost every sculpting tool in modo and can be accessed by simply holding the Shift key down while sculpting. The Smooth amount attribute allows you to control the amount of smoothing applied when using the Shift qualifier with that specific tool. This is a convenience feature implemented specifically to enhance the sculpting workflow.

Smooth Mode: modo provides alternate methods for applying smoothing while sculpting-

Shift Key-- The smooth mode is enabled when the 'Shift' key is depressed.

Low Pressure-- When using a pressure sensitive tablet, users may apply the 'Low Pressure' option, where light sculpting strokes will have a smoothing affect on the surface. Once the pressure threshold is exceeded, the smoothing effect will attenuate toward the full pressure effect where the brush will work normally as determined by the tool providing a dual function to the brush.

Uniform-- When set to 'Uniform', smoothing is always on while sculpting, thus making softer strokes.

Stroke Interpolation: When the option 'Interpolation' is *enabled*, modo will add additional sample points and filter out small variances in pressure and position resulting in smoother strokes.

Stroke Step: Generally, paint strokes are not a continuous lines, but a series of *dots* or *dabs* placed on the canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to 8%. So for an 64 pixel sized brush, dabs would be placed approx every 7 pixels. Users can increase or reduce this value for a variety of effects. Increasing it toward 100% will produce individual dots, rather than what looks like a stroked line.

Continuous Stroke: When this option is enabled, modo will continuously draw a stroke like a *line*, instead of as individual dabs. modo will draw the stroke as a linear segments between each interpolation step dab so increasing the scale of the interpolation steps will reveal the linear nature of the continuous stroke option.

Default 10% 'Interpolation Step' produces smooth stroke

Interpolation Step set to 200% reveals individual dabs

200% Interpolation Step with Continuous Stroke enabled

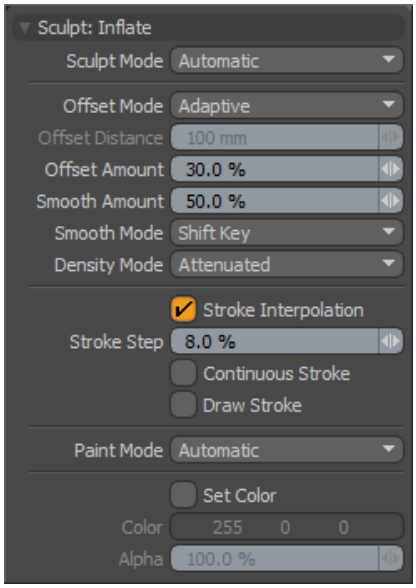
Draw Stroke: When *enabled*, this option simply draws a line in the interface defined by the path for the brush stroke.

Inflate

The Inflate tool works by pushing all vertices out along their normals with the push strength modulated by the brush. On a flat surface this will appear to be very similar to the Push tool, however, on surfaces with curvature the effect is very different and has gives the appearance that the model is being *inflated* from the inside (like a balloon). This is a great tool for growing proportions very quickly. Pressing the 'Ctrl' key down while using the tool will send the surface downward, with an effect similar to *deflating*. Pressing 'Shift' while using the tool will enable the 'Smooth' function. The 'Inflate' tool can be found under the 'Paint' tab for the interface in the 'Sculpt' submenu of the toolbox. The tool can also be called from the menu bar under the "Texture > Sculpt Tools" menu. Additionally, the tool can be combined with the various 'Tips' as well

as the 'Inks' and 'Nozzle' to further customize the way the brush affects the surface. When activated the following attributes are displayed in the Properties panel.

Sculpt: Inflate--



Sculpt Mode: The 'Sculpt Mode' determines which method of sculpting is to be used. By default the setting is 'Automatic' which will perform mesh based sculpts when there are no displacement images and automatically switch to image based sculpting when there an image is available. There are times when you may want to force modo into mesh based sculpting even if you have an image applied for displacement. This can be very helpful for blocking out shapes and form, as mesh sculpting is only effecting the cage vertices of the base geometry.

Automatic-- chooses between mesh and image sculpting depending on availability of displacement image map texture.

Mesh-- forces the tools to perform mesh sculpting only.

Image Map (Details)-- forces the tools to perform image sculpting only.

Offset Mode: modo provides two methods for calculating the way the tool moves the surface when brushing.

Adaptive: The adaptive mode will adapt to the surface that is being brushed on based on the setting for 'Offset Amount' and the size of the brush.

Absolute: When in 'Absolute' mode, the 'Offset Distance' setting will become active and users may enter a specific distance value the brush will conform to regardless of brush scale.

Offset Distance: This control is disabled unless the 'Offset Mode' is set to 'Absolute'. When active, the value determines specifically how far to displace the surface.

Offset Amount: Active only when the 'Offset Mode' is set to 'Adaptive', this percentage value modulates the extent of the brushstroke. The total scale of the sculpt deformation will be equal to the brush size times the offset amount percentage.

(Note: This is a very important value for controlling sculpted detail and as such it can be adjusted interactively by using the MMB+click and hauling left or right.)

Smooth Amount: The Smooth tool is such a useful method for controlling sculpt detail it is available with almost every sculpting tool in modo and can be accessed by simply holding the Shift key down while sculpting. The Smooth amount attribute allows you to control the amount of smoothing applied when using the Shift qualifier with that specific tool. This is a convenience feature implemented specifically to enhance the sculpting workflow.

Smooth Mode: modo provides alternate methods for applying smoothing while sculpting-

Shift Key-- The smooth mode is enabled when the 'Shift' key is depressed.

Low Pressure-- When using a pressure sensitive tablet, users may apply the 'Low Pressure' option, where light sculpting strokes will have a smoothing affect on the surface. Once the pressure threshold is exceeded, the smoothing effect will attenuate toward the full pressure effect where the brush will work normally as determined by the tool providing a dual function to the brush.

Uniform-- When set to 'Uniform', smoothing is always on while sculpting, thus making softer strokes.

Density Mode: This control determines how the sculpt stroke effects areas of the mesh that it has already covered. For instance, brushing back and forth will cause the sculpt tool to run back over itself which can have a number of different effects. This controls the sculpt during a single stroke. Once the stylus comes off the tablet or you release the LMB the density control is reset.

Additive-- continues to add to the displacement each time the brush crosses over its own path with no maximum cap to govern the sculpt.

Attenuate-- continues to add to the displacement as the brush crosses over its own stroke until the displacement reaches the Offset Distance or Amount value.

Max Stroke-- enforces a maximum offset based on the largest displacement value of the current stroke.

Max Map-- caps the sculpt effect at the maximum amount in the current displacement image.

Stroke Interpolation: When the option 'Interpolation' is *enabled*, modo will add additional sample points and filter out small variances in pressure and position resulting in smoother strokes.

Stroke Step: Generally, paint strokes are not a continuous lines, but a series of *dots* or *dabs* placed on the canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to 8%. So for an 64 pixel sized brush, dabs would be placed approx every 7 pixels. Users can increase or reduce this

value for a variety of effects. Increasing it toward 100% will produce individual dots, rather than what looks like a stroked line.

Continuous Stroke: When this option is enabled, modo will continuously draw a stroke like a *line*, instead of as individual dabs. modo will draw the stroke as a linear segments between each interpolation step dab so increasing the scale of the interpolation steps will reveal the linear nature of the continuous stroke option.

Default 10% 'Interpolation Step' produces smooth stroke

Interpolation Step set to 200% reveals individual dabs

200% Interpolation Step with Continuous Stroke enabled

Draw Stroke: When *enabled*, this option simply draws a line in the interface defined by the path for the brush stroke.

Set Color: When utilizing Multi-Res sculpting, and enabling the 'Set Color' option, users can simultaneously sculpt the mesh and paint into an Image Map layer using the same settings of the sculpting tool itself.

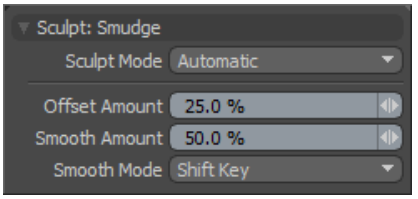
Color: This value determines the color applied to the image map when sculpting.

Alpha: This value determines the opacity of the stroke applied to the image map.

Smudge

The Smudge tool pulls vertices along with the brush briefly and then leaves them to rest as the brush passes on. This creates a smudging effect and is very useful for quickly adding some organic imperfections in a surface. Pressing the 'Ctrl' key down while using the tool will invert the effect sliding the surface toward the opposite direction. Pressing 'Shift' while using the tool will enable the 'Smooth' function. The 'Smudge' tool can be found under the 'Paint' tab for the interface in the 'Sculpt' submenu of the toolbox. The tool can also be called from the menu bar under the "Texture > Sculpt Tools" menu. Additionally, the tool can be combined with the various 'Tips' as well as the 'Inks' and 'Nozzle' to further customize the way the brush affects the surface. When activated the following attributes are displayed in the Properties panel.

Sculpt: Smudge--



Sculpt Mode: The 'Sculpt Mode' determines which method of sculpting is to be used. By default the setting is 'Automatic' which will perform mesh based sculpts when there are no displacement images and automatically switch to image based sculpting when there an image is available. There are times when you may want to force modo into mesh based sculpting even if you have an image applied for displacement. This can be very helpful for blocking out shapes and form, as mesh sculpting is only effecting the cage vertices of the base geometry.

Automatic-- chooses between mesh and image sculpting depending on availability of displacement image map texture.

Mesh-- forces the tools to perform mesh sculpting only.

Image Map (Details)-- forces the tools to perform image sculpting only.

Offset Amount: This percentage value modulates the strength of the brushstroke. The total scale of the sculpt deformation will be equal to the brush size times the offset amount percentage.

(Note: This is a very important value for controlling sculpted detail and as such it can be adjusted interactively by using the MMB+click and hauling left or right.)

Smooth Amount: The Smooth tool is such a useful method for controlling sculpt detail it is available with almost every sculpting tool in modo and can be accessed by simply holding the Shift key down while sculpting. The Smooth amount attribute allows you to control the amount of smoothing applied when using the Shift qualifier with that specific tool. This is a convenience feature implemented specifically to enhance the sculpting workflow.

Smooth Mode: modo provides alternate methods for applying smoothing while sculpting-

Shift Key-- The smooth mode is enabled when the 'Shift' key is depressed.

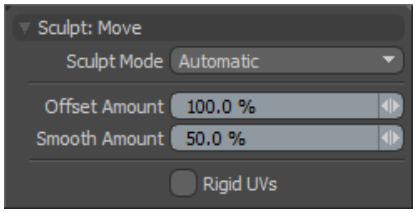
Low Pressure-- When using a pressure sensitive tablet, users may apply the 'Low Pressure' option, where light sculpting strokes will have a smoothing affect on the surface. Once the pressure threshold is exceeded, the smoothing effect will attenuate toward the full pressure effect where the brush will work normally as determined by the tool providing a dual function to the brush.

Uniform-- When set to 'Uniform', smoothing is always on while sculpting, thus making softer strokes.

Move

The Move sculpt tool simply *grabs* the mesh inside the brush and drags the surface in the direction of the mouse. The tool leverages screen orientation to derive the perpendicular axis for the move. For instance, if the 3D viewport is rotated such that you are looking directly downward at the surface, the resulting move effect would translate the mesh on the same XZ plane. If the intended result is to "lift" the surface, the 3D viewport should be rotated so that the tool is working at more of a perpendicular (glancing) angle to the surface of the mesh. Pressing the 'Ctrl' key down while using the tool will send the surface moving in the opposite direction that the mouse moves. Pressing 'Shift' while using the tool will enable the 'Smooth' function. The sculpt 'Move' tool can be found under the 'Paint' tab for the interface in the 'Sculpt' submenu of the toolbox. The tool can also be called from the menu bar under the "Texture > Sculpt Tools" menu. Additionally, the tool can be combined with the various 'Tips' as well as the 'Inks' and 'Nozzle' to further customize the way the brush affects the surface. When activated the following attributes are displayed in the Properties panel.

Sculpt: Move--



Sculpt Mode: The 'Sculpt Mode' determines which method of sculpting is to be used. By default the setting is 'Automatic' which will perform mesh based sculpts when there are no displacement images and automatically switch to image based sculpting when there an image is available. There are times when you may want to force modo into mesh based sculpting even if you have an image applied for displacement. This can be very helpful for blocking out shapes and form, as mesh sculpting is only effecting the cage vertices of the base geometry.

Automatic-- chooses between mesh and image sculpting depending on availability of displacement image map texture.

Mesh-- forces the tools to perform mesh sculpting only.

Image Map (Details)-- forces the tools to perform image sculpting only.

Offset Amount: This percentage value modulates the strength of the move as a percentage of the mouse movement. At 100% the points will move the same distance as the mouse, at 50%, they points will only move half as far and so on. (Note: This is a very important value for controlling sculpted detail and as such it can be adjusted interactively by using the MMB+click and hauling left or right.)

Smooth Amount: The Smooth tool is such a useful method for controlling sculpt detail it is available with almost every sculpting tool in modo and can be accessed by simply

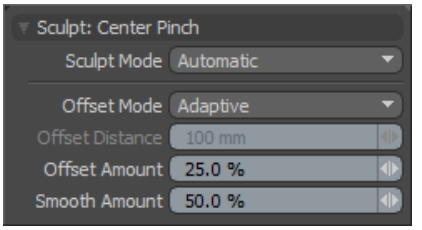
holding the Shift key down while sculpting. The Smooth amount attribute allows you to control the amount of smoothing applied when using the Shift qualifier with that specific tool. This is a convenience feature implemented specifically to enhance the sculpting workflow.

Rigid UVs: When using the sculpt tools to edit a UV map, the function of the move tool will attenuate based on the 'Tip' size and shape. By enabling the 'Rigid UVs' option, users can easily move entire UV islands as solid rigid pieces by simply moving the brush tip over an area and dragging.

Center Pinch

The Center Pinch tool pulls geometry in towards the center of the brush. As the components gravitate towards the epicenter of the brush, they tend to stay on the original surface, producing a sort of sliding effect. Pressing the 'Ctrl' key down while using the tool will invert the effect forcing the center area outward. Pressing 'Shift' while using the tool will enable the 'Smooth' function. The 'Center Pinch' tool can be found under the 'Paint' tab for the interface in the 'Sculpt' submenu of the toolbox. The tool can also be called from the menu bar under the "Texture > Sculpt Tools" menu. Additionally, the tool can be combined with the various 'Tips' as well as the 'Inks' and 'Nozzle' to further customize the way the brush affects the surface. When activated the following attributes are displayed in the Properties panel.

Sculpt: Center Pinch--



Sculpt Mode: The 'Sculpt Mode' determines which method of sculpting is to be used. By default the setting is 'Automatic' which will perform mesh based sculpts when there are no displacement images and automatically switch to image based sculpting when there an image is available. There are times when you may want to force modo into mesh based sculpting even if you have an image applied for displacement. This can be very helpful for blocking out shapes and form, as mesh sculpting is only effecting the cage vertices of the base geometry.

Automatic-- chooses between mesh and image sculpting depending on availability of displacement image map texture.

Mesh-- forces the tools to perform mesh sculpting only.

Image Map (Details)-- forces the tools to perform image sculpting only.

Offset Mode: modo provides two methods for calculating the way the tool moves the surface when brushing.

Adaptive: The adaptive mode will adapt to the surface that is being brushed on based on the setting for 'Offset Amount' and the size of the brush.

Absolute: When in 'Absolute' mode, the 'Offset Distance' setting will become active and users may enter a specific distance value the brush will conform to regardless of brush scale.

Offset Distance: This control is disabled unless the 'Offset Mode' is set to 'Absolute'. When active, the value determines specifically how far to displace the surface.

Offset Amount: Active only when the 'Offset Mode' is set to 'Adaptive', this percentage value modulates the extent of the brushstroke. The total scale of the sculpt deformation will be equal to the brush size times the offset amount percentage.

(Note: This is a very important value for controlling sculpted detail and as such it can be adjusted interactively by using the MMB+click and hauling left or right.)

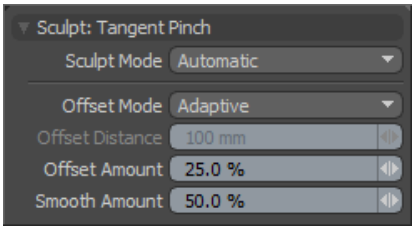
Smooth Amount: The Smooth tool is such a useful method for controlling sculpt detail it is available with almost every sculpting tool in modo and can be accessed by simply holding the Shift key down while sculpting. The Smooth amount attribute allows you to control the amount of smoothing applied when using the Shift qualifier with that specific tool. This is a convenience feature implemented specifically to enhance the sculpting workflow.

Tangent Pinch

The Tangent Pinch tool works to pull geometry in towards the axis perpendicular to the brush stroke. If the brush stroke is moving up and/or down, the geometry will pull inwards from the right and left. This behavior makes it possible to quickly create or accentuate grooves and/or creases. Pressing the 'Ctrl' key down while using the tool will invert the effect forcing the center area outward. Pressing 'Shift' while using the tool will enable the 'Smooth' function. The 'Tangent Pinch' tool can be found under the 'Paint' tab for the interface in the 'Sculpt' submenu of the toolbox. The tool can also be called from the menu bar under the "Texture > Sculpt Tools" menu. Additionally, the tool can be combined with the various 'Tips' as well as the 'Inks' and 'Nozzle' to further customize the way the brush affects the surface. When activated the following attributes are displayed in the Properties panel.

Sculpt: Tangent Pinch--

Sculpt Mode: The 'Sculpt Mode' determines which method of sculpting is to be used.



By default the setting is 'Automatic' which will perform mesh based sculpts when there are no displacement images and automatically switch to image based sculpting when there an image is available. There are times when you may want to force modo into mesh based sculpting even if you have an image applied for displacement. This can be very helpful for blocking out shapes and form, as mesh sculpting is only effecting the cage vertices of the base geometry.

Automatic-- chooses between mesh and image sculpting depending on availability of displacement image map texture.

Mesh-- forces the tools to perform mesh sculpting only.

Image Map (Details)-- forces the tools to perform image sculpting only.

Offset Mode: modo provides two methods for calculating the way the tool moves the surface when brushing.

Adaptive: The adaptive mode will adapt to the surface that is being brushed on based on the setting for 'Offset Amount' and the size of the brush.

Absolute: When in 'Absolute' mode, the 'Offset Distance' setting will become active and users may enter a specific distance value the brush will conform to regardless of brush scale.

Offset Distance: This control is disabled unless the 'Offset Mode' is set to 'Absolute'. When active, the value determines specifically how far to displace the surface.

Offset Amount: Active only when the 'Offset Mode' is set to 'Adaptive', this percentage value modulates the extent of the brushstroke. The total scale of the sculpt deformation will be equal to the brush size times the offset amount percentage.

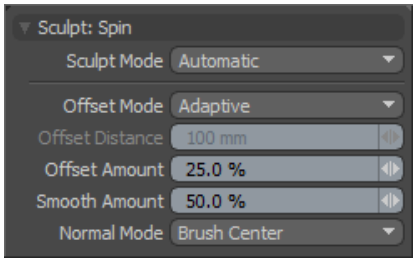
(Note: This is a very important value for controlling sculpted detail and as such it can be adjusted interactively by using the MMB+click and hauling left or right.)

Smooth Amount: The Smooth tool is such a useful method for controlling sculpt detail it is available with almost every sculpting tool in modo and can be accessed by simply holding the Shift key down while sculpting. The Smooth amount attribute allows you to control the amount of smoothing applied when using the Shift qualifier with that specific tool. This is a convenience feature implemented specifically to enhance the sculpting workflow.

Spin

The Spin tool creates a vortex of vertices around the center of the brush using the 'Normal Mode' to determine the axis around which the vortex will occur. This is very useful for creating swirls in a surface or simply adding some organic perturbation. The default normal mode is Brush Center which causes all vertices within the brush to rotate around that center point using the normal of the geometry at that point as the axis of rotation. It is also useful to change the Normal Mode to Screen which allows you to control the axis of rotation by rotating the view. The default method moves the vertices mostly along the existing surface while Screen allows the creation of overhangs and more intense swirls. Pressing the 'Ctrl' key down while using the tool will twirl the surface the opposite direction. Pressing 'Shift' while using the tool will enable the 'Smooth' function. The 'Spin' tool can be found under the 'Paint' tab for the interface in the 'Sculpt' submenu of the toolbox. The tool can also be called from the menu bar under the "Texture > Sculpt Tools" menu. Additionally, the tool can be combined with the various 'Tips' as well as the 'Inks' and 'Nozzle' to further customize the way the brush affects the surface. When activated the following attributes are displayed in the Properties panel.

Sculpt: Spin--



Sculpt Mode: The 'Sculpt Mode' determines which method of sculpting is to be used. By default the setting is 'Automatic' which will perform mesh based sculpts when there are no displacement images and automatically switch to image based sculpting when there an image is available. There are times when you may want to force modo into mesh based sculpting even if you have an image applied for displacement. This can be very helpful for blocking out shapes and form, as mesh sculpting is only effecting the cage vertices of the base geometry.

Automatic-- chooses between mesh and image sculpting depending on availability of displacement image map texture.

Mesh-- forces the tools to perform mesh sculpting only.

Image Map (Details)-- forces the tools to perform image sculpting only.

Offset Mode: modo provides two methods for calculating the way the tool moves the surface when brushing.

Adaptive: The adaptive mode will adapt to the surface that is being brushed on based on the setting for 'Offset Amount' and the size of the brush.

Absolute: When in 'Absolute' mode, the 'Offset Distance' setting will become active and users may enter a specific distance value the brush will conform to regardless of brush scale.

Offset Distance: This control is disabled unless the 'Offset Mode' is set to 'Absolute'. When active, the value determines specifically how far to displace the surface.

Offset Amount: Active only when the 'Offset Mode' is set to 'Adaptive', this percentage value modulates the extent of the brushstroke. The total scale of the sculpt deformation will be equal to the brush size times the offset amount percentage.

(Note: This is a very important value for controlling sculpted detail and as such it can be adjusted interactively by using the MMB+click and hauling left or right.)

Smooth Amount: The Smooth tool is such a useful method for controlling sculpt detail it is available with almost every sculpting tool in modo and can be accessed by simply holding the Shift key down while sculpting. The Smooth amount attribute allows you to control the amount of smoothing applied when using the Shift qualifier with that specific tool. This is a convenience feature implemented specifically to enhance the sculpting workflow.

Normal Mode: This option controls the method used for displacing the sculpted vertices. The various options include:

Brush Center-- uses the geometry at the very center of the brush to calculate the direction along which to displace the mesh.

Vertex-- displaces each vertex along its own normal.

Average-- calculates an average normal for all the vertices inside the brush.

Screen-- displaces the vertices directly towards the active viewport.

Mouse Down-- uses the normal of the mesh surface directly under the mouse when the LMB is clicked. All displacement is then carried out according to that vector.

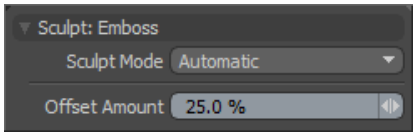
X,Y,Z-- allows you to pick a world axis along which to displace the vertices.

Emboss

Similar to the Move tool, the Emboss tool is not a stroke based deformer. When you click down with the mouse (or tablet) Emboss moves the surface of the mesh that is within the brush at the time of the click in or out along the local normals. Moving the brush to the right moves out from the surface and moving to the left pushes down inward. The extent of the emboss is modulated by the value of the brush. In the case of the Smooth Brush the Emboss tool simple creates single dimples or bumps on the

surface. This can be useful, but the Emboss tool works most effectively when combined with more interesting brushes such as the image based Preset brush. By using the 'Geometry to Brush' option, even more interesting brushes can be created, such as ears, and other 'parts'. By pressing the 'Ctrl' key when applying an image, this will invert the image, reversing the displacement direction. Additionally, the tool can be combined with the various 'Tips' as well as the 'Inks' and 'Nozzle' to further customize the way the brush affects the surface. When activated the following attributes are displayed in the Properties panel.

Sculpt: Emboss--



Sculpt Mode: The 'Sculpt Mode' determines which method of sculpting is to be used. By default the setting is 'Automatic' which will perform mesh based sculpts when there are no displacement images and automatically switch to image based sculpting when there an image is available. There are times when you may want to force modo into mesh based sculpting even if you have an image applied for displacement. This can be very helpful for blocking out shapes and form, as mesh sculpting is only effecting the cage vertices of the base geometry.

Automatic-- chooses between mesh and image sculpting depending on availability of displacement image map texture.

Mesh-- forces the tools to perform mesh sculpting only.

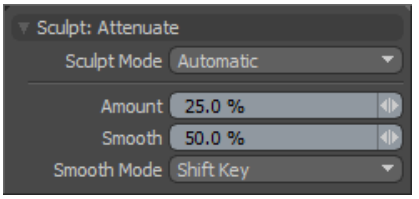
Image Map (Details)-- forces the tools to perform image sculpting only.

Offset Amount: The 'Offset Amount' value determines the strength of the embossment applied to the surface.

Attenuate

Attenuate works specifically with image based sculpting and will have no effect on mesh based sculpting data. The tool works by diminishing the strength of the sculpted data stored in the current displacement or vector displacement image texture. The effect is similar to melting away the sculpted detail and can be very useful for healing areas of your sculpt. Pressing 'Shift' while using the tool will enable the 'Smooth' function. The 'Attenuate' tool can be found under the 'Paint' tab for the interface in the 'Sculpt' submenu of the toolbox. The tool can also be called from the menu bar under the "Texture > Sculpt Tools" menu. Additionally, the tool can be combined with the various 'Tips' as well as the 'Inks' and 'Nozzle' to further customize the way the brush affects the surface. When activated the following attributes are displayed in the Properties panel.

Sculpt: Attenuate--



Sculpt Mode: The 'Sculpt Mode' determines which method of sculpting is to be used. By default the setting is 'Automatic' which will perform mesh based sculpts when there are no displacement images and automatically switch to image based sculpting when there an image is available. There are times when you may want to force modo into mesh based sculpting even if you have an image applied for displacement. This can be very helpful for blocking out shapes and form, as mesh sculpting is only effecting the cage vertices of the base geometry.

Automatic-- chooses between mesh and image sculpting depending on availability of displacement image map texture.

Mesh-- forces the tools to perform mesh sculpting only.

Image Map (Details)-- forces the tools to perform image sculpting only.

Amount: This percentage value modulates the strength of the attenuation.

(Note: This is a very important value for controlling sculpted detail and as such it can be adjusted interactively by using the MMB+click and hauling left or right.)

Smooth: The Smooth tool is such a useful method for controlling sculpt detail it is available with almost every sculpting tool in modo and can be accessed by simply holding the Shift key down while sculpting. The 'Smooth' attribute allows you to control the amount of smoothing applied when using the Shift qualifier with that specific tool. This is a convenience feature implemented specifically to enhance the sculpting workflow.

Smooth Mode: modo provides alternate methods for applying smoothing while sculpting-

Shift Key-- The smooth mode is enabled when the 'Shift' key is depressed.

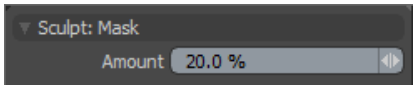
Low Pressure-- When using a pressure sensitive tablet, users may apply the 'Low Pressure' option, where light sculpting strokes will have a smoothing affect on the surface. Once the pressure threshold is exceeded, the smoothing effect will attenuate toward the full pressure effect where the brush will work normally as determined by the tool providing a dual function to the brush.

Uniform-- When set to 'Uniform', smoothing is always on while sculpting, thus making softer strokes.

Mask

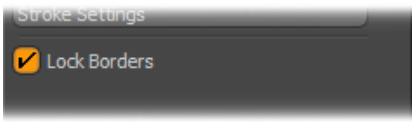
The 'Sculpt Mask' tool requires the use of image based sculpting, as it is meant to specifically paint into an 'Image Map' layer defined as a *layer mask* to the Displacement layer. The tool is simply painting into an image using grayscale values which are used to hide or reveal the sculpted data, without affecting the displacement layer itself. To use the tool, users will need an additional 'Image Map' layer *below* the Displacement layer defined as a 'Layer Mask'. For the users convenience, the "Add Mask Texture" on the 'Utilities' subtab of the Paint interface's toolbox will add an image texture with the appropriate effect selected in the Shader Tree, at which point the user can begin to use the 'Sculpt Mask' feature by simply brushing in any of the 3D viewports. The main benefit of the tool is that it will eliminate the need to select the appropriate layer in the shader tree to Displace or Mask; users can simply sculpt and then use the mask tool to hide and/or reveal areas of the displacement.

Sculpt: Mask--



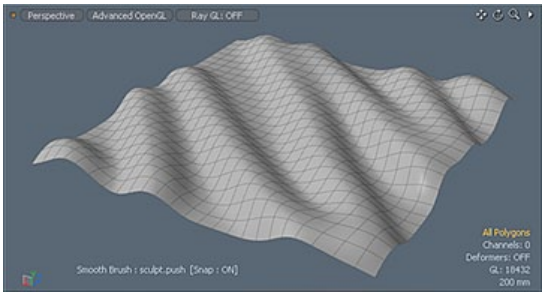
Amount: The 'Amount' value determines the strength of the brush stroke when masking the displacement. Higher values will mask the displacement layer more quickly. Pressing the 'Ctrl' when masking the layer will remove the areas of the mask, revealing the displacement.

Lock Borders

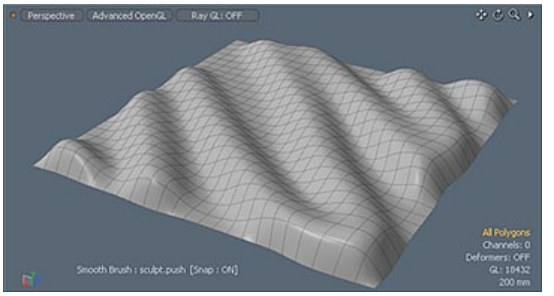


Located in the 'Paint' interface tab at the bottom of the sculpting toolbox, the 'Lock Borders' option when *enabled*, will lock the position of any *open* edges, making them immovable by the sculpting tools. While having a variety of uses, this function is mostly useful for sculpting on a mesh where the edges will need to seamlessly meet up with another mesh. Simply LMB+click the checkbox to toggle.

Open edges can be easily highlighted by selecting all the geometry in a layer (Alt+Shift+A) in 'Polygons' mode, then press and hold 'Ctrl', which changes the 'Edges' mode button to the 'Boundary' button, LMB+click the Boundary option to highlight all open edges in the selection. If no edges are visible, then the geometry has no open edges and is considered *water-tight*.



'Lock Borders' disabled during sculpting.



'Lock Borders' enabled during sculpting.

Painting Tools

Painting Brush Reference

Painting in modo is an intuitive blend of 2D and 3D painting techniques, where users can paint onto any bitmap image map within any of the GL viewports. The brushes for painting can be found in the 'Paint' tab of the interface, where a focused collection of tools and useful commands are located dedicated to the painting and sculpting functionality. The tools may also be called from the menu bar, under the "Texture > Paint Tools" menu. Invoking any of the tools will display the associated properties in the appropriate viewport panel. To paint, users will need to select both a Tool (*brush*) and a 'Tip' which designates how the brush itself applies the color. When working with a mouse, users can simply LMB+drag over the surface to apply the stroke. The LMB click is applied automatically when using a pressure sensitive stylus and tablet. Users

may also RMB+click and drag in the viewport to interactively scale the size of the brush tip.

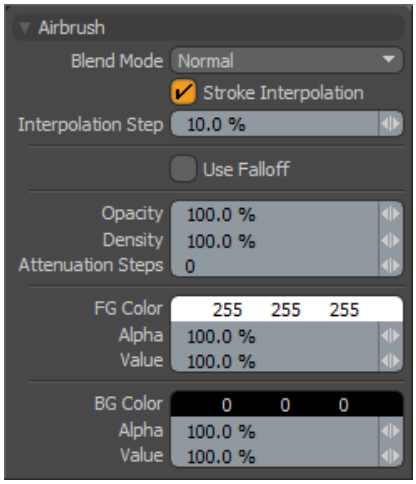
Painting Tools

Airbrush	Clone	Radial Gradient	Blur
Paintbrush	Line	Linear Gradient	Sharpen
Eraser	Fill	Smudge	Lasso

Airbrush

The Airbrush tool, with the proper 'Tip' can be used to mimic a real airbrush which excels at creating soft even strokes. Users can select the tool from the 'Paint' interface tab under the 'Paint Tools' subtab of the toolbox. The 'Airbrush' tool can also be activated from the the menu bar option under the "Texture > Paint Tools" menu. When selected, the tool will also auto-select a Tip allowing users to begin painting directly into the active image map, defined by the image in the shader tree with the small paintbrush icon next to it ( ). Simply select a different target layer if you wish to paint onto a different layer. Users may also select different Tips and additionally add Ink and Nozzle effects to the airbrush to further customize the way the brush applies to the surface. When selected, the associated attributes display in the associated Properties panel.

Airbrush--



Blend Mode: The Blend Mode defines how the strokes will blend into the existing canvas. The default 'Normal' setting will apply opaque strokes (as defined by the 'Opacity' option) over the top of the existing canvas. The alternate options will work in the same respect as the layer blending options documented with examples on the Blend Modes page of the documentation.

Stroke Interpolation: When the option 'Interpolation' is *enabled*, modo will add additional sample points and filter out small variances in pressure and position resulting in smoother strokes.

Interpolation Step: Generally, paint strokes are not a continuous lines, but a series of *dots* or *dabs* placed on the canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to 10%. So for an 80 pixel sized brush, dabs would be placed every 8 pixels (10% of 80 = 8). Users can increase or reduce this value for a variety of effects. Increasing it toward 100% will produce individual dots, rather than what looks like a stroked line.

Use Falloff: The 'Use Falloff' option will allow any of the various falloffs to mask the application of brush strokes. Since the RMB brush sizing function supersedes the RMB function to define the falloff, ensure the falloff is defined prior to activating the tool.

Opacity: This value determines how the total opacity of the ink distributed by the brush. If the value is less than 100% the ink will max out at that percentage and never become fully opaque no matter how many times you paint over the stroke.

Density: This value determines the amount of ink that comes out during a paint stroke. Unlike Opacity, when Density is below 100% painting over the existing stroke repeatedly will continue to increase the amount of paint on the map.

Attenuation Steps: This value will cause the paint to "fade out" after the set number of steps is reached in the stroke. This setting is somewhat related to the speed with which you create the stroke as painting quickly will spread out the "steps" used to paint the stroke. Moving the brush slowly with this value set above 0 will cause the paint to run out in a shorter stroke. This is a very useful setting to use if you do not have a tablet to paint with. Tablet users would be better suited to modulate the paint opacity based on the tablet pressure so as to have greater control.

FG Color: Sets the foreground color for various paint tools. This can be set via the color component on the tool properties or by clicking on the Color Picker while the tool is active.

Alpha: This option allows an alpha value to be associated with the FG paint color.

Value: This percentage can be used to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).

BG Color: Sets the background color for paint tools that use it. This can be set via the color component on the tool properties or by clicking on the Color Picker while the tool is active.

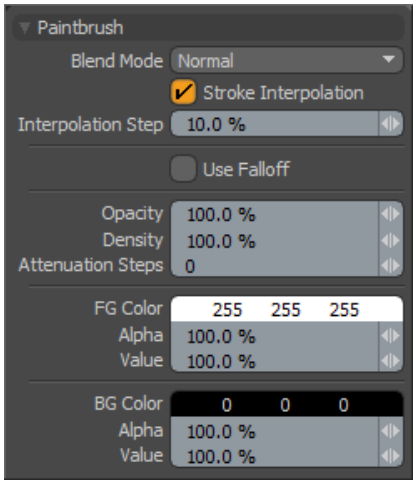
Alpha: This option allows an alpha value to be associated with the BG paint color.

Value: This percentage can be used to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).

Paintbrush

The Paintbrush tool, combined with a 'Tip' applies solid even strokes to the canvas. The main difference between the *Paintbrush* and the *Airbrush* tools is that the Paintbrush applies the paint with no internal blending, which is to say the paint comes out at full strength whereas the Airbrush *layers* on the paint daubs as you repeat strokes. Users can select the tool from the 'Paint' interface tab under the 'Paint Tools' subtab of the toolbox. The 'Paintbrush' tool can also be activated from the the menu bar option under the "Texture > Paint Tools" menu. When selected, the tool will also auto-select a Tip allowing users to begin painting directly into the active image map, defined by the image in the shader tree with the small paintbrush icon next to it (). Simply select a different target layer if you wish to paint onto a different layer. Users may also select different Tips and additionally add Ink and Nozzle effects to the paintbrush to further customize the way the brush applies to the surface. When selected, the associated attributes display in the Properties panel.

Paintbrush--



Blend Mode: The Blend Mode defines how the strokes will blend into the existing canvas. The default 'Normal' setting will apply opaque strokes (as defined by the Opacity option) over the top of the existing canvas. The alternate options will work in the same respect as the layer blending options documented with examples on the Blend Modes page of the documentation.

Stroke Interpolation: When the option 'Interpolation' is *enabled*, modo will add additional sample points and filter out small variances in pressure and position resulting in smoother strokes.

Interpolation Step: Generally, paint strokes are not a continuous lines, but a series of *dots* or *dabs* placed on the canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to 10%. So for an 80 pixel sized brush, dabs would be placed every 8 pixels (10% of 80 = 8). Users can increase

or reduce this value for a variety of effects. Increasing it toward 100% will produce individual dots, rather than what looks like a stroked line.

Use Falloff: The 'Use Falloff' option will allow any of the various falloffs to mask the application of brush strokes. Since the RMB brush sizing function supersedes the RMB function to define the falloff, ensure the falloff is defined prior to activating the tool.

Opacity: This value determines how the total opacity of the ink distributed by the brush. If the value is less than 100% the ink will max out at that percentage and never become fully opaque no matter how many times you paint over the stroke.

Density: This value determines the amount of ink that comes out during a paint stroke. Unlike Opacity, when Density is below 100% painting over the existing stroke repeatedly will continue to increase the amount of paint on the map.

Attenuation Steps: This value will cause the paint to "fade out" after the set number of steps is reached in the stroke. This setting is somewhat related to the speed with which you create the stroke as painting quickly will spread out the "steps" used to paint the stroke. Moving the brush slowly with this value set above 0 will cause the paint to run out in a shorter stroke. This is a very useful setting to use if you do not have a tablet to paint with. Tablet users would be better suited to modulate the paint opacity based on the tablet pressure so as to have greater control.

FG Color: Sets the foreground color for various paint tools. This can be set via the color component on the tool properties or by clicking on the Color Picker while the tool is active.

Alpha: This option allows an alpha value to be associated with the FG paint color.

Value: This percentage can be used to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).

BG Color: Sets the background color for paint tools that use it. This can be set via the color component on the tool properties or by clicking on the Color Picker while the tool is active.

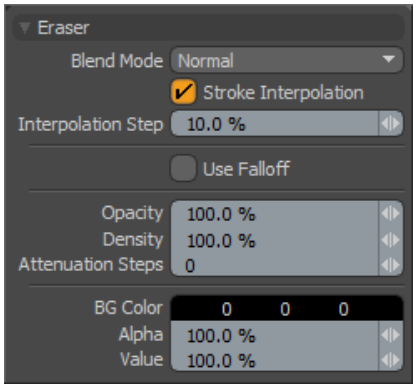
Alpha: This option allows an alpha value to be associated with the BG paint color.

Value: This percentage can be used to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).

Eraser

The Eraser tool works by modifying the alpha channel of the currently selected image texture to create the effect of erasing the paint. When utilizing an image with no 'Alpha' the Eraser tool produces no effect. Users can select the tool from the 'Paint' interface tab under the 'Paint Tools' subtab of the toolbox. The 'Eraser' tool can also be activated from the the menu bar option under the "Texture > Paint Tools" menu. When selected, the tool will also auto-select a Tip allowing users to begin painting directly into the active image map, defined by the image in the shader tree with the small paintbrush icon next to it ( ). Simply select a different target layer if you wish to edit a different layer. Users may also select different Tips and additionally add Ink and Nozzle effects to the brush to further customize the way the tool applies to the surface. When selected, the associated attributes display in the Properties panel.

Eraser--



Blend Mode: The Blend Mode defines how the strokes will blend into the existing canvas. The default 'Normal' setting will apply opaque strokes (as defined by the Opacity option) over the top of the existing canvas. The alternate options will work in the same respect as the layer blending options documented with examples on the Blend Modes page of the documentation.

Stroke Interpolation: When the option 'Interpolation' is *enabled*, modo will add additional sample points and filter out small variances in pressure and position resulting in smoother strokes.

Interpolation Step: Generally, paint strokes are not a continuous lines, but a series of *dots* or *dabs* placed on the canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to 10%. So for an 80 pixel sized brush, dabs would be placed every 8 pixels (10% of 80 = 8). Users can increase or reduce this value for a variety of effects. Increasing it toward 100% will produce individual dots, rather than what looks like a stroked line.

Use Falloff: The 'Use Falloff' option will allow any of the various falloffs to mask the application of brush strokes. Since the RMB brush sizing function supersedes the RMB function to define the falloff, ensure the falloff is defined prior to activating the tool.

Opacity: This value determines how the total opacity of the ink distributed by the brush. If the value is less than 100% the ink will max out at that percentage and never become

fully opaque no matter how many times you paint over the stroke.

Density: This value determines the amount of ink that comes out during a paint stroke. Unlike Opacity, when Density is below 100% painting over the existing stroke repeatedly will continue to increase the amount of paint on the map.

Attenuation Steps: This value will cause the paint to "fade out" after the set number of steps is reached in the stroke. This setting is somewhat related to the speed with which you create the stroke as painting quickly will spread out the "steps" used to paint the stroke. Moving the brush slowly with this value set above 0 will cause the paint to run out in a shorter stroke. This is a very useful setting to use if you do not have a tablet to paint with. Tablet users would be better suited to modulate the paint opacity based on the tablet pressure so as to have greater control.

BG Color: Sets the background color for paint tools that use it. This can be set via the color component on the tool properties or by clicking on the Color Picker while the tool is active.

Alpha: This option allows an alpha value to be associated with the BG paint color.

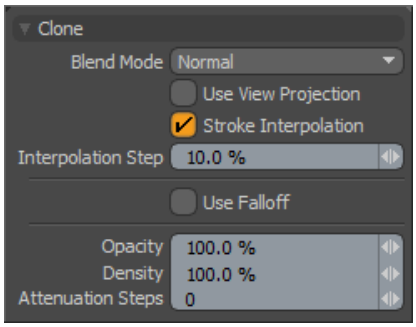
Value: This percentage can be used to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).

Clone

The Clone tool provides a function to duplicate an area of a texture to another area. Useful for quickly covering imperfection is a photographic image by sampling from another like area of the image or for simply enlarging areas of useful *texture*. Before applying a stroke, users must define a clone source area. This can be done by holding down the 'Ctrl' key and LMB+click the area of the image where the center of the cloning will originate from. Each stroke will not clone from that center source until a new source is defined.

Users can select the tool from the 'Paint' interface tab under the 'Paint Tools' subtab of the toolbox. The 'Clone' tool can also be activated from the the menu bar option under the "Texture > Paint Tools" menu. When selected, the tool will also auto-select a Tip allowing users to begin painting directly into the active image map, defined by the image in the shader tree with the small paintbrush icon next to it ( ). Simply select a different target layer if you wish to paint onto a different layer. Users may also select different Tips and additionally add Ink and Nozzle effects to the brush to further customize the way the tool applies to the surface. When selected, the associated attributes display in the Properties panel.

Clone--



Blend Mode: The Blend Mode defines how the strokes will blend into the existing canvas. The default 'Normal' setting will apply opaque strokes (as defined by the Opacity option) over the top of the existing canvas. The alternate options will work in the same respect as the layer blending options documented with examples on the Blend Modes page of the documentation.

Use View Projection: Normally when painting, samples take place in the *flat image* space, enabling 'Use View Projection' forces modo to sample directly in *screen* space. This option avoids seam issues, but the image will be distorted by the models view and projection (which isn't necessarily bad), and limits the sampling resolution to that of the screen at the particular zoom level.

Stroke Interpolation: When the option 'Interpolation' is *enabled*, modo will add additional sample points and filter out small variances in pressure and position resulting in smoother strokes.

Interpolation Step: Generally, paint strokes are not a continuous lines, but a series of dots or dabs placed on the canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to 10%. So for an 80 pixel sized brush, dabs would be placed every 8 pixels (10% of 80 = 8). Users can increase or reduce this value for a variety of effects. Increasing it toward 100% will produce individual dots, rather than what looks like a stroked line.

Use Falloff: The 'Use Falloff' option will allow any of the various falloffs to mask the application of brush strokes. Since the RMB brush sizing function supersedes the RMB function to define the falloff, ensure the falloff is defined prior to activating the tool.

Opacity: This value determines how the total opacity of the ink distributed by the brush. If the value is less than 100% the ink will max out at that percentage and never become fully opaque no matter how many times you paint over the stroke.

Density: This value determines the amount of ink that comes out during a paint stroke. Unlike Opacity, when Density is below 100% painting over the existing stroke repeatedly will continue to increase the amount of paint on the map.

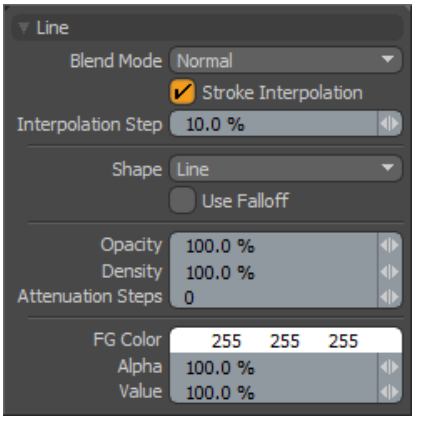
Attenuation Steps: This value will cause the paint to "fade out" after the set number of steps is reached in the stroke. This setting is somewhat related to the speed with which you create the stroke as painting quickly will spread out the "steps" used to paint the stroke. Moving the brush slowly with this value set above 0 will cause the paint to run

out in a shorter stroke. This is a very useful setting to use if you do not have a tablet to paint with. Tablet users would be better suited to modulate the paint opacity based on the tablet pressure so as to have greater control.

Line

The Line paint tool allows users to quickly create straight lines by simply LMB+clicking a start point, dragging to the end point position and releasing the mouse button to set. The look of the line is defined by the 'Tip' shape and size. Users can select the tool from the 'Paint' interface tab under the 'Paint Tools' subtab of the toolbox. The 'Line' tool can also be activated from the the menu bar option under the "Texture > Paint Tools" menu. When selected, the tool will also auto-select a Tip allowing users to begin creating lines directly into the active image map, defined by the image in the shader tree with the small paintbrush icon next to it ( ). Simply select a different target layer if you wish to paint onto a different layer. Users may also select different Tips and additionally add Ink and Nozzle effects to the paintbrush to further customize the way the tool applies to the surface. When selected, the associated attributes display in the Properties panel.

Line--



Blend Mode: The Blend Mode defines how the strokes will blend into the existing canvas. The default 'Normal' setting will apply opaque strokes (as defined by the Opacity option) over the top of the existing canvas. The alternate options will work in the same respect as the layer blending options documented with examples on the Blend Modes page of the documentation.

Stroke Interpolation: When the option 'Interpolation' is *enabled*, modo will add additional sample points and filter out small variances in pressure and position resulting in smoother strokes.

Interpolation Step: Generally, paint strokes are not a continuous lines, but a series of *dots* or *dabs* placed on the canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to 10%. So for an 80 pixel sized brush, dabs would be placed every 8 pixels (10% of 80 = 8). Users can increase or reduce this value for a variety of effects. Increasing it toward 100% will produce individual dots, rather than what looks like a stroked line.

Use Falloff: The 'Use Falloff' option will allow any of the various falloffs to mask the application of brush strokes. Since the RMB brush sizing function supersedes the RMB function to define the falloff, ensure the falloff is defined prior to activating the tool.

Opacity: This value determines how the total opacity of the ink distributed by the brush. If the value is less than 100% the ink will max out at that percentage and never become fully opaque no matter how many times you paint over the stroke.

Density: This value determines the amount of ink that comes out during a paint stroke. Unlike Opacity, when Density is below 100% painting over the existing stroke repeatedly will continue to increase the amount of paint on the map.

Attenuation Steps: This value will cause the paint to "fade out" after the set number of steps is reached in the stroke. This setting is somewhat related to the speed with which you create the stroke as painting quickly will spread out the "steps" used to paint the stroke. Moving the brush slowly with this value set above 0 will cause the paint to run out in a shorter stroke. This is a very useful setting to use if you do not have a tablet to paint with. Tablet users would be better suited to modulate the paint opacity based on the tablet pressure so as to have greater control.

FG Color: Sets the foreground color for various paint tools. This can be set via the color component on the tool properties or by clicking on the Color Picker while the tool is active.

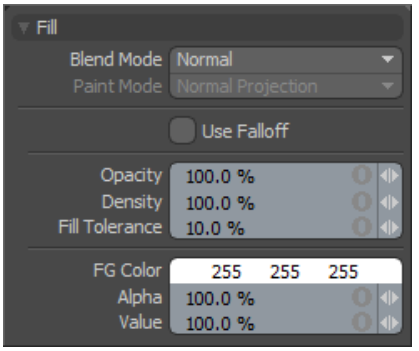
Alpha: This option allows an alpha value to be associated with the FG paint color.

Value: This percentage can be used to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).

Fill

The Fill tool tool will color all the pixels in the image that fall within the 'Fill Tolerance' according to the color of the pixel that was clicked on when initiating the fill. Users can select the tool from the 'Paint' interface tab under the 'Paint Tools' subtab of the toolbox. The 'Fill' tool can also be activated from the the menu bar option under the "Texture > Paint Tools" menu. Once selected, users can begin painting directly into the active image map, defined by the image in the shader tree with the small paintbrush icon next to it ( ). Simply select a different target layer if you wish to paint onto a different layer. When selected, the associated attributes display in the Properties panel.

Fill--



Blend Mode: The Blend Mode defines how the filled area will blend into the existing canvas. The default 'Normal' setting will apply opaquely (as defined by the Opacity option) over the top of the existing canvas. The alternate options will work in the same respect as the layer blending options documented with examples on the Blend Modes page of the documentation.

Paint Mode: Has no effect over the Fill tool (as it always works in *image space*) and is therefore disabled.

Use Falloff: The 'Use Falloff' option will allow any of the various falloffs to mask the application of fill effect.

Opacity: This value determines how the total opacity of the ink distributed by the brush. If the value is less than 100% the ink will max out at that percentage and never become fully opaque no matter how many times you paint over the stroke.

Density: This value determines the amount of ink that comes out during a paint stroke. Unlike Opacity, when Density is below 100% painting over the existing stroke repeatedly will continue to increase the amount of paint on the map.

Fill Tolerance: This value defines the threshold area that is flooded with color by the fill tool.

FG Color: Sets the foreground color for various paint tools. This can be set via the color component on the tool properties or by clicking on the Color Picker while the tool is active.

Alpha: This option allows an alpha value to be associated with the FG paint color.

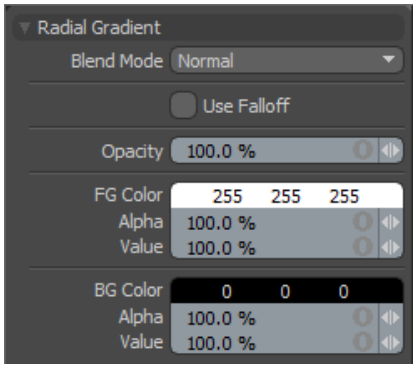
Value: This percentage can be used to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).

Radial Gradient

The Radial Gradient tool will apply a flood fill of color that ramps from the FG color to the BG color in a circular fashion. LMB+clicking and dragging in the 3D view will create

a line from the cursor. This line is used to define the ramp distance from the FG to BG color. The color ramp is calculated in a radial direction with the FG color at the center according to the user defined line. The ramp can be created in 3D views or directly in the UV view. Users can select the tool from the 'Paint' interface tab under the 'Paint Tools' subtab of the toolbox. The 'Radial Gradient' tool can also be activated from the the menu bar option under the "Texture > Paint Tools" menu. Once selected, new gradients are created in the active image map, defined by the layer in the shader tree with the small paintbrush icon next to it ( ). Simply select a different target layer if you wish to paint onto a different layer. When selected, the associated attributes display in the Properties panel.

Radial Gradient--



Blend Mode: The Blend Mode defines how the filled area will blend into the existing canvas. The default 'Normal' setting will apply opaquely (as defined by the Opacity option) over the top of the existing canvas. The alternate options will work in the same respect as the layer blending options documented with examples on the Blend Modes page of the documentation.

Use Falloff: The 'Use Falloff' option will allow any of the various falloffs to mask the application of fill effect.

Opacity: This value determines how the total opacity of the ink distributed by the gradient. If the value is less than 100% the ink will max out at that percentage and never become fully opaque no matter how many times you paint over the stroke.

FG Color: Sets the foreground color for the gradient. This can be set via the color component on the tool properties or by clicking on the Color Picker while the tool is active.

Alpha: This option allows an alpha value to be associated with the FG paint color.

Value: This percentage can be used to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).

BG Color: Sets the background color for the gradient. This can be set via the color component on the tool properties or by clicking on the Color Picker while the tool is active.

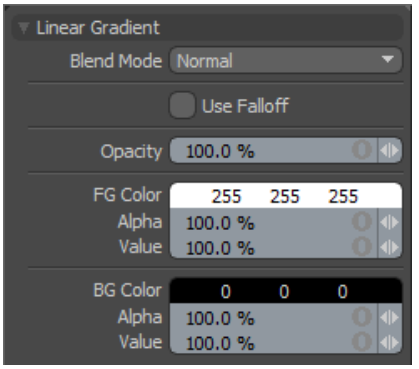
Alpha: This option allows an alpha value to be associated with the BG paint color.

Value: This percentage can be used to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).

Linear Gradient

The Linear Gradient paint tool will apply a flood fill of ink that ramps from the FG color to the BG color. Clicking and dragging in the 3D view will create a line from the cursor. This line is used to define the ramp distance from the FG to BG color. The color ramp is calculated in a linearly according to the user defined line. The ramp can be created in 3D views or directly in the UV view. Users can select the tool from the 'Paint' interface tab under the 'Paint Tools' subtab of the toolbox. The 'Linear Gradient' tool can also be activated from the the menu bar option under the "Texture > Paint Tools" menu. Once selected, new gradients are created in the active image map, defined by the layer in the shader tree with the small paintbrush icon next to it (). Simply select a different target layer if you wish to paint onto a different layer. When selected, the associated attributes display in the Properties panel.

Linear Gradient--



Blend Mode: The Blend Mode defines how the filled area will blend into the existing canvas. The default 'Normal' setting will apply opaquely (as defined by the Opacity option) over the top of the existing canvas. The alternate options will work in the same respect as the layer blending options documented with examples on the Blend Modes page of the documentation.

Use Falloff: The 'Use Falloff' option will allow any of the various falloffs to mask the application of fill effect.

Opacity: This value determines how the total opacity of the ink distributed by the gradient. If the value is less than 100% the ink will max out at that percentage and never become fully opaque no matter how many times you paint over the stroke.

FG Color: Sets the foreground color for the gradient. This can be set via the color component on the tool properties or by clicking on the Color Picker while the tool is active.

Alpha: This option allows an alpha value to be associated with the FG paint color.

Value: This percentage can be used to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).

BG Color: Sets the background color for the gradient. This can be set via the color component on the tool properties or by clicking on the Color Picker while the tool is active.

Alpha: This option allows an alpha value to be associated with the BG paint color.

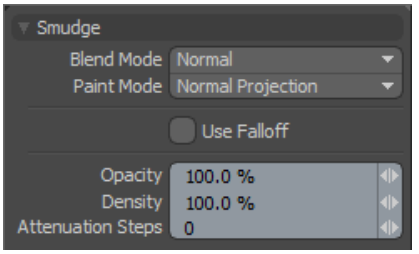
Value: This percentage can be used to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).

Smudge

The Smudge tool shift pixels from under the brush in the direction of the mouse or stylus' movement creating the visual effect of smearing the paint together. Users can select the tool from the 'Paint' interface tab under the 'Paint Tools' subtab of the toolbox. The 'Smudge' tool can also be activated from the the menu bar option under the "Texture > Paint Tools" menu. When selected, the tool will also auto-select a Tip allowing users to begin painting directly into the active image map, defined by the image in the shader tree with the small paintbrush icon next to it (). Simply select a different target layer if you wish to paint onto a different layer. Users may also select different Tips and additionally add Ink and Nozzle effects to the paintbrush to further customize the way the brush applies to the surface. When selected, the associated attributes display in the Properties panel.

Smudge--

Blend Mode: The Blend Mode defines how the strokes will blend into the existing canvas. The default 'Normal' setting will apply opaque strokes (as defined by the Opacity option) over the top of the existing canvas. The alternate options will work in the same respect as the layer blending options documented with examples on the Blend Modes page of the documentation.



Stroke Interpolation: When the option 'Interpolation' is *enabled*, modo will add additional sample points and filter out small variances in pressure and position resulting in smoother strokes.

Interpolation Step: Generally, paint strokes are not a continuous lines, but a series of *dots* or *dabs* placed on the canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to 10%. So for an 80 pixel sized brush, dabs would be placed every 8 pixels (10% of 80 = 8). Users can increase or reduce this value for a variety of effects. Increasing it toward 100% will produce individual dots, rather than what looks like a stroked line.

Use Falloff: The 'Use Falloff' option will allow any of the various falloffs to mask the application of brush strokes. Since the RMB brush sizing function supersedes the RMB function to define the falloff, ensure the falloff is defined prior to activating the tool.

Opacity: This value determines how the total opacity of the ink distributed by the brush. If the value is less than 100% the ink will max out at that percentage and never become fully opaque no matter how many times you paint over the stroke.

Density: This value determines the amount of ink that comes out during a paint stroke. Unlike Opacity, when Density is below 100% painting over the existing stroke repeatedly will continue to increase the amount of paint on the map.

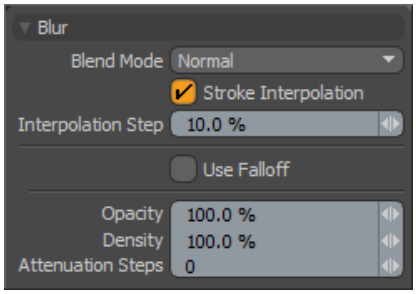
Attenuation Steps: This value will cause the paint to "fade out" after the set number of steps is reached in the stroke. This setting is somewhat related to the speed with which you create the stroke as painting quickly will spread out the "steps" used to paint the stroke. Moving the brush slowly with this value set above 0 will cause the paint to run out in a shorter stroke. This is a very useful setting to use if you do not have a tablet to paint with. Tablet users would be better suited to modulate the paint opacity based on the tablet pressure so as to have greater control.

Blur

The Blur tool averages pixels from under the brush to create the effect of softening any apparent detail. Users can select the tool from the 'Paint' interface tab under the 'Paint Tools' subtab of the toolbox. The 'Blur' tool can also be activated from the the menu bar option under the "Texture > Paint Tools" menu. When selected, the tool will also auto-select a Tip allowing users to begin painting directly into the active image map, defined by the image in the shader tree with the small paintbrush icon next to it (). Simply select a different target layer if you wish to paint onto a different layer.

Users may also select different Tips and additionally add Ink and Nozzle effects to the brush to further customize the way the tool applies to the surface. When selected, the associated attributes display in the Properties panel.

Blur--



Blend Mode: The Blend Mode defines how the strokes will blend into the existing canvas. The default 'Normal' setting will apply opaque strokes (as defined by the Opacity option) over the top of the existing canvas. The alternate options will work in the same respect as the layer blending options documented with examples on the Blend Modes page of the documentation.

Stroke Interpolation: When the option 'Interpolation' is *enabled*, modo will add additional sample points and filter out small variances in pressure and position resulting in smoother strokes.

Interpolation Step: Generally, paint strokes are not a continuous lines, but a series of *dots* or *dabs* placed on the canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to 10%. So for an 80 pixel sized brush, dabs would be placed every 8 pixels (10% of 80 = 8). Users can increase or reduce this value for a variety of effects. Increasing it toward 100% will produce individual dots, rather than what looks like a stroked line.

Use Falloff: The 'Use Falloff' option will allow any of the various falloffs to mask the application of brush strokes. Since the RMB brush sizing function supersedes the RMB function to define the falloff, ensure the falloff is defined prior to activating the tool.

Opacity: This value determines how the total opacity of the ink distributed by the brush. If the value is less than 100% the ink will max out at that percentage and never become fully opaque no matter how many times you paint over the stroke.

Density: This value determines the amount of ink that comes out during a paint stroke. Unlike Opacity, when Density is below 100% painting over the existing stroke repeatedly will continue to increase the amount of paint on the map.

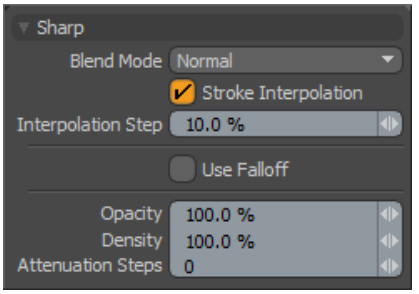
Attenuation Steps: This value will cause the paint to "fade out" after the set number of steps is reached in the stroke. This setting is somewhat related to the speed with which you create the stroke as painting quickly will spread out the "steps" used to paint the stroke. Moving the brush slowly with this value set above 0 will cause the paint to run out in a shorter stroke. This is a very useful setting to use if you do not have a tablet to

paint with. Tablet users would be better suited to modulate the paint opacity based on the tablet pressure so as to have greater control.

Sharpen

The Sharpen tool increases the edge contrast between colors producing an effect that looks like focusing, making images appear sharper. Users can select the tool from the 'Paint' interface tab under the 'Paint Tools' subtab of the toolbox. The 'Sharpen' tool can also be activated from the the menu bar option under the "Texture > Paint Tools" menu. When selected, the tool will also auto-select a Tip allowing users to begin painting directly into the active image map, defined by the image in the shader tree with the small paintbrush icon next to it ( ). Simply select a different target layer if you wish to paint onto a different layer. Users may also select different Tips and additionally add Ink and Nozzle effects to the brush to further customize the way the tool applies to the surface. When selected, the associated attributes display in the Properties panel.

Sharpen--



Blend Mode: The Blend Mode defines how the strokes will blend into the existing canvas. The default 'Normal' setting will apply opaque strokes (as defined by the Opacity option) over the top of the existing canvas. The alternate options will work in the same respect as the layer blending options documented with examples on the Blend Modes page of the documentation.

Stroke Interpolation: When the option 'Interpolation' is *enabled*, modo will add additional sample points and filter out small variances in pressure and position resulting in smoother strokes.

Interpolation Step: Generally, paint strokes are not a continuous lines, but a series of *dots* or *dabs* placed on the canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to 10%. So for an 80 pixel sized brush, dabs would be placed every 8 pixels (10% of 80 = 8). Users can increase or reduce this value for a variety of effects. Increasing it toward 100% will produce individual dots, rather than what looks like a stroked line.

Use Falloff: The 'Use Falloff' option will allow any of the various falloffs to mask the application of brush strokes. Since the RMB brush sizing function supersedes the RMB function to define the falloff, ensure the falloff is defined prior to activating the tool.

Opacity: This value determines how the total opacity of the ink distributed by the brush.

If the value is less than 100% the ink will max out at that percentage and never become fully opaque no matter how many times you paint over the stroke.

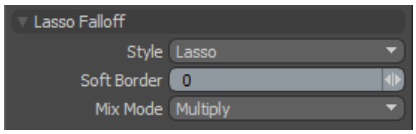
Density: This value determines the amount of ink that comes out during a paint stroke. Unlike Opacity, when Density is below 100% painting over the existing stroke repeatedly will continue to increase the amount of paint on the map.

Attenuation Steps: This value will cause the paint to "fade out" after the set number of steps is reached in the stroke. This setting is somewhat related to the speed with which you create the stroke as painting quickly will spread out the "steps" used to paint the stroke. Moving the brush slowly with this value set above 0 will cause the paint to run out in a shorter stroke. This is a very useful setting to use if you do not have a tablet to paint with. Tablet users would be better suited to modulate the paint opacity based on the tablet pressure so as to have greater control.

Lasso

The Lasso falloff option provides users a means to draw selection areas directly on a viewport that then limit the application of paint strokes to the surface. The Lasso option is simply a standard falloff type, as documented in the modeling section under Falloffs.

Lasso



Several styles of Lasso selections, drawn in screen space projecting outward toward infinity, producing Falloffs that are very specific. Settings allow users to then to define areas of influence around the lasso shape with the 'Soft Border' option. All areas within the Lasso receive 100% of the tools effect; the attenuation of the falloff occurs across the 'Soft Border' distance only outside the Lasso area.

Style: Users can choose from the various Lasso styles-

Lasso- Allows users to draw freeform shapes onto the screen defining the Falloff .

Rectangle- Users can drag out rectangular shapes to define the Falloff area.

Circle- Users can drag out perfectly circular shapes to define the Falloff area. Shape originates from center of the circle.

Ellipse- Users can drag out elliptical shapes to define the Falloff area. Shape originates from the upper or lower corner of the ellipse.

Soft Border: The 'Soft Border' option defines the actual Falloff radius around the Lasso area that attenuates across the distance defined. Calculated in screen-space pixels from the lasso border.

Mix Mode: In instances where there are multiple falloffs applied to a transform (using the 'Add' option of the Falloff menu), the mix mode defines how each Falloff interacts with the other.

Tool Falloffs

Because of modo's generalized nature, the same Falloff options that are used for geometric edits can be applied to most of the paint tools. In order to utilize falloffs, each paint tool has a 'Use Falloffs' option that must be enabled. Using this option, it is easy to confine paint strokes to specific areas of the model (or viewport). Of course, there are some falloffs that are more useful than others in this regard. For instance, the 'Screen' and 'Airbrush' falloff operators would not be very useful during painting since they have similar attributes to the paintbrushes themselves. Falloffs such as 'Linear', 'Cylinder', 'Radial' and of course 'Lasso' are quite useful when working with the Paint tools.

Typically Falloff operators use the RMB for setup. However, since the paint tools use RMB to set the brush size you must select the Falloff in the Tool Pipe and then use the LMB to define the falloff area *before* painting. Make sure to select the paint tool prior to painting. Once the Falloff has an initial location you can also easily use the falloff handles to further edit their positions without changing tool selections via the tool pipe.

Hair Styling Tools

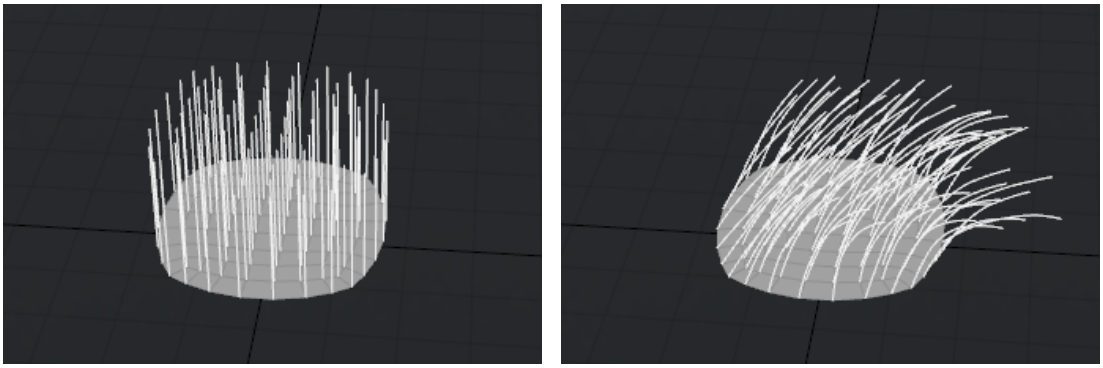
Hair Tools Reference

Much like the sculpt and painting tools, modo offers a variety of tools meant specifically for styling hair and its associated *guides*. Located in the 'Paint' tab of the default workspace, in the 'Hair Tools' subtab of the toolbox, giving you everything you need for combing and sculpting your fur into the proper *do*. The tools all work double duty styling both guides, and when 'Edit Fur Map' is enabled in the tools properties, you can paint on image maps meant to control the styling of a regular fur layer.

Hair Styling Tools			Hair Styling Options
Push	Move	Stretch	Tip Edit Mode
Smooth	Curl	Root Puff	Set Guide Color
Comb	Separate		

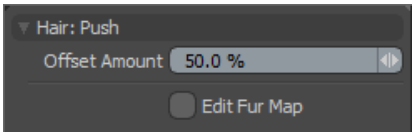
Push

The 'Push' styling brush pushes and pulls fibers as the brush moves over them the direction that the cursor moves. The Push brush is similar to combing the fibers with the greatest strength at the tips of the fiber tapering off toward the root. The Fur brush can be combined with the 'Tips' to further customize how the brush styles the fur. The 'Push' tool can be found in the 'Hair Tools' subtab of the toolbox, found in the 'Paint' tab of the interface. When selected, the following attributes display in the Properties viewport.



On the left, a group of guides, on the right the 'Push' tool applied.

Hair: Push--

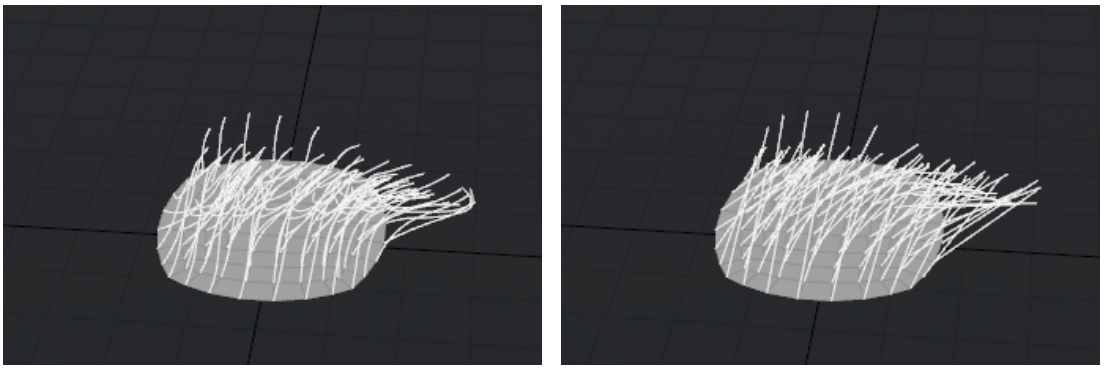


Offset Amount: The 'Offset Amount' value determines the strength of the brush stroke. Higher values will have a greater influence over the fibers than lower values.

Edit Fur Map: Users can style fur by way of a Fur Vector Map, providing a means to control combing direction and length with an image map, producing very detailed results. All that is necessary is a Fur Layer in the Shader Tree and a Floating Point image (such as EXR) with the layer effect set to 'Fur Vector'. Generally, the fur styling tools are meant to edit Guides, in order to edit the associated fur map, users need to enable the 'Edit Fur Map' option.

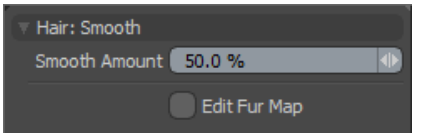
Smooth

The "Smooth' brush straightens and smooth's bent and kinked fibers. The effect is similar to aligning the all points into a straight line between the root and tip of each individual fiber. The Fur brush can be combined with the 'Tips' to further customize how the brush styles the fur. The 'Smooth' tool can be found in the 'Hair Tools' subtab of the toolbox, found in the 'Paint' tab of the interface. When selected, the following attributes display in the Properties viewport.



On the left, a group of combed guides, on the right the 'Smooth' tool applied.

Hair: Smooth--

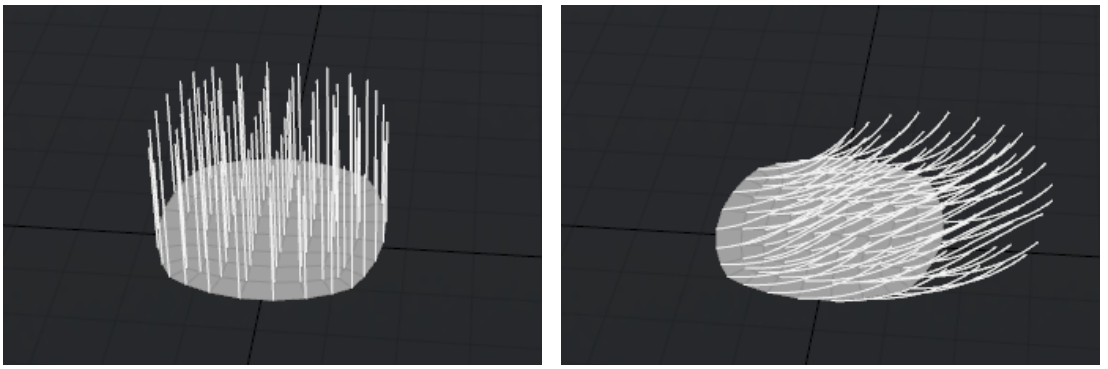


Smooth Amount: The 'Smooth Amount' value determines the strength of the brush stroke. Higher values will have a greater influence over the fibers than lower values.

Edit Fur Map: Users can style fur by way of a Fur Vector Map, providing a means to control combing direction and length with an image map, producing very detailed results. All that is necessary is a Fur Layer in the Shader Tree and a Floating Point image (such as EXR) with the layer effect set to 'Fur Vector'. Generally, the fur styling tools are meant to edit Guides, in order to edit the associated fur map, users need to enable the 'Edit Fur Map' option.

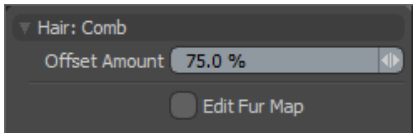
Comb

The 'Comb' brush works much like a virtual comb, dragging fibers in the direction of the cursor. Similar to the 'Push' tool, but the comb brush exerts its greatest influence toward the roots of the fibers. The Fur brush can be combined with the 'Tips' to further customize how the brush styles the fur. The 'Comb' tool can be found in the 'Hair Tools' subtab of the toolbox, found in the 'Paint' tab of the interface. When selected, the following attributes display in the Properties viewport.



On the left, a group of guides, on the right the 'Comb' tool applied.

Hair: Comb--

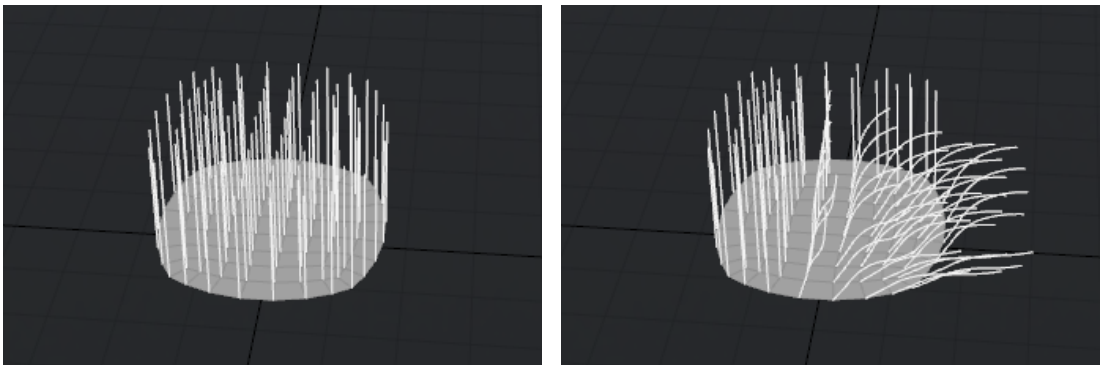


Offset Amount: The 'Offset Amount' value determines the strength of the brush stroke. Higher values will have a greater influence over the fibers than lower values.

Edit Fur Map: Users can style fur by way of a Fur Vector Map, providing a means to control combing direction and length with an image map, producing very detailed results. All that is necessary is a Fur Layer in the Shader Tree and a Floating Point image (such as EXR) with the layer effect set to 'Fur Vector'. Generally, the fur styling tools are meant to edit Guides, in order to edit the associated fur map, users need to enable the 'Edit Fur Map' option.

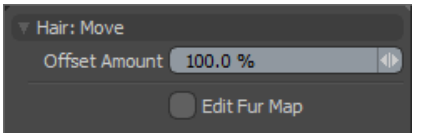
Move

The 'Move' brush grabs the fibers like a magnet and moves them in the direction of the mouse pointer. The tool exerts its greatest influence over the tips of the fibers with an overall gradual falloff based on the brushes size and shape. The Fur brush can be combined with the 'Tips' to further customize how the brush styles the fur. The 'Move' tool can be found in the 'Hair Tools' subtab of the toolbox, found in the 'Paint' tab of the interface. When selected, the following attributes display in the Properties viewport.



On the left, a group of guides, on the right the 'Move' tool applied.

Hair: Move--

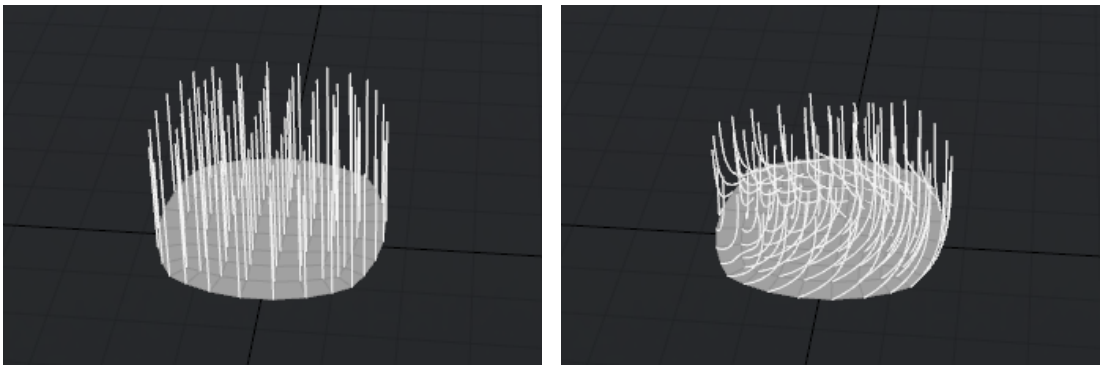


Offset Amount: The 'Offset Amount' value determines the strength of the brush stroke. Higher values will have a greater influence over the fibers than lower values.

Edit Fur Map: Users can style fur by way of a Fur Vector Map, providing a means to control combing direction and length with an image map, producing very detailed results. All that is necessary is a Fur Layer in the Shader Tree and a Floating Point image (such as EXR) with the layer effect set to 'Fur Vector'. Generally, the fur styling tools are meant to edit Guides, in order to edit the associated fur map, users need to enable the 'Edit Fur Map' option.

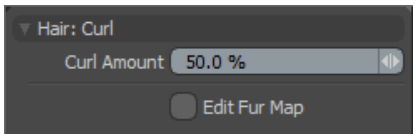
Curl

The 'Curl' brush adds swirls to fibers rotating them based on the size of the brush tip. The effect if similar to the 'Spin' sculpting brush with the curl effect projecting straight out from the brush tip through the viewport, so one can easily control its application by rotating the guides orientation within the viewport. The Fur brush can be combined with the 'Tips' to further customize how the brush styles the fur. The 'Curl' tool can be found in the 'Hair Tools' subtab of the toolbox, found in the 'Paint' tab of the interface. When selected, the following attributes display in the Properties viewport.



On the left, a group of guides, on the right the 'Curl' tool applied.

Hair: Curl--

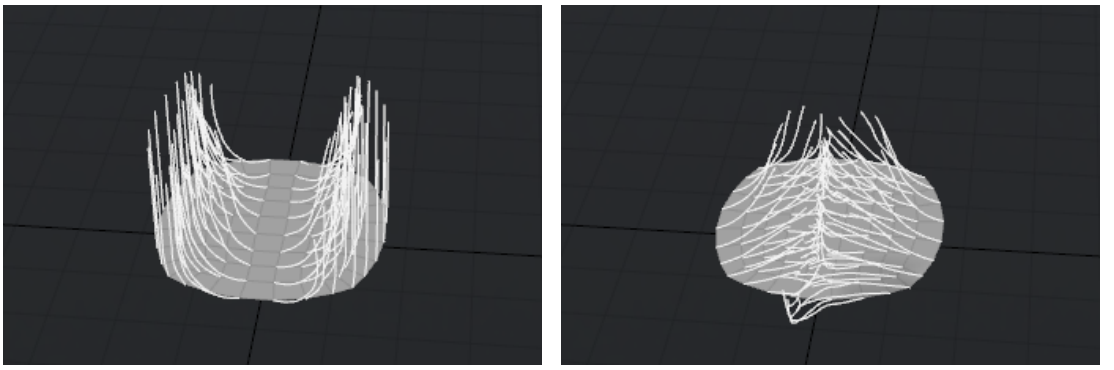


Curl Amount: The 'Curl Amount' value determines the strength of the brush stroke. Higher values will have a greater influence over the fibers than lower values.

Edit Fur Map: Users can style fur by way of a Fur Vector Map, providing a means to control combing direction and length with an image map, producing very detailed results. All that is necessary is a Fur Layer in the Shader Tree and a Floating Point image (such as EXR) with the layer effect set to 'Fur Vector'. Generally, the fur styling tools are meant to edit Guides, in order to edit the associated fur map, users need to enable the 'Edit Fur Map' option.

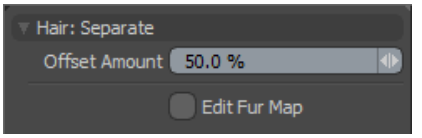
Separate

The 'Separate' brush moves adjoining fibers or guides apart from each other with a falloff based on the brush shape and strength. By holding down the 'Crtl' key, the user can reverse the separate function and gather the fibers closer together. The Fur brush can be combined with the 'Tips' to further customize how the brush styles the fur. The 'Separate' tool can be found in the 'Hair Tools' subtab of the toolbox, found in the 'Paint' tab of the interface. When selected, the following attributes display in the Properties viewport.



On the left, guides are separated by the brush, on the right the guides are *gathered*.

Hair: Separate--

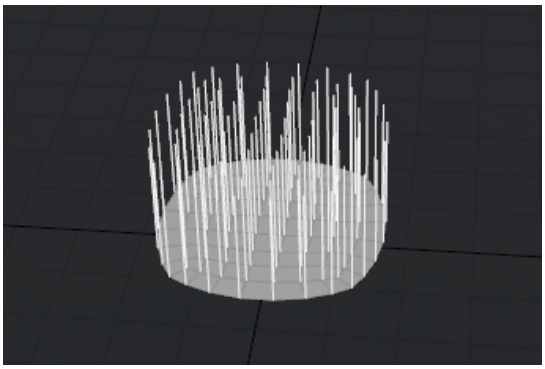


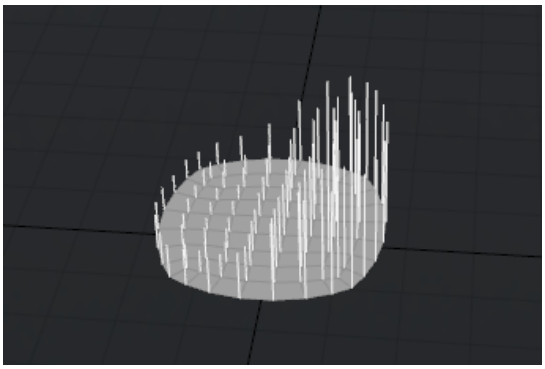
Offset Amount: The 'Offset Amount' value determines the strength of the brush stroke. Higher values will have a greater influence over the fibers than lower values.

Edit Fur Map: Users can style fur by way of a Fur Vector Map, providing a means to control combing direction and length with an image map, producing very detailed results. All that is necessary is a Fur Layer in the Shader Tree and a Floating Point image (such as EXR) with the layer effect set to 'Fur Vector'. Generally, the fur styling tools are meant to edit Guides, in order to edit the associated fur map, users need to enable the 'Edit Fur Map' option.

Stretch

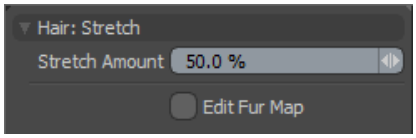
The 'Stretch' brush scales the individual fibers, reducing their overall length. By holding down the 'Ctrl' key, the users can increase a fibers size growing the overall length. The Fur brush can be combined with the 'Tips' to further customize how the brush styles the fur. The 'Stretch' tool can be found in the 'Hair Tools' subtab of the toolbox, found in the 'Paint' tab of the interface. When selected, the following attributes display in the Properties viewport.





On the left, a group of guides, on the right the 'Stretch' tool applied.

Hair: Stretch--

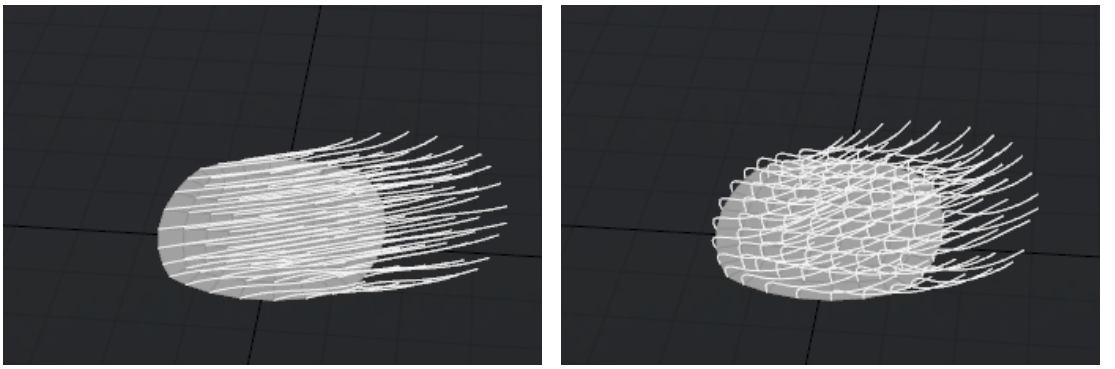


Stretch Amount: The 'Stretch Amount' value determines the strength of the brush stroke. Higher values will have a greater influence over the fibers than lower values.

Edit Fur Map: Users can style fur by way of a Fur Vector Map, providing a means to control combing direction and length with an image map, producing very detailed results. All that is necessary is a Fur Layer in the Shader Tree and a Floating Point image (such as EXR) with the layer effect set to 'Fur Vector'. Generally, the fur styling tools are meant to edit Guides, in order to edit the associated fur map, users need to enable the 'Edit Fur Map' option.

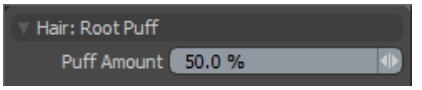
Root Puff

The 'Root Puff' brush will rotate the root of the fibers or guides toward the direction of the surface normal. Especially useful when fibers have been combed to the point where the roots are rotated parallel to the surface. The Fur brush can be combined with the 'Tips' to further customize how the brush styles the fur. The 'Root Puff' tool can be found in the 'Hair Tools' subtab of the toolbox, found in the 'Paint' tab of the interface. When selected, the following attributes display in the Properties viewport.



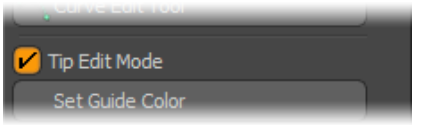
On the left, a group of combed guides, on the right the 'Root Puff' tool applied.

Hair: Root Puff--



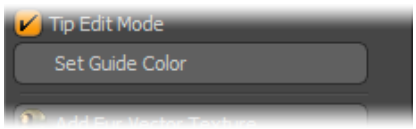
Puff Amount: The 'Puff Amount' value determines the strength of the brush stroke. Higher values will have a greater influence over the fibers than lower values.

Tip Edit Mode

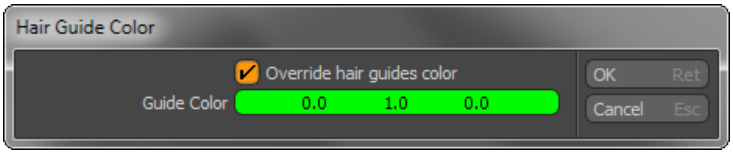


Located in the 'Hair Tools' subtab of the 'Paint' toolbox, the 'Tip Edit Mode' option helps when styling fur guides. *Enabled* by default, the option attenuates the application of brush strokes along the length of the guide providing a smoother, more natural means to working with the guides. When *disabled*, the styling tools affect the curves evenly in strength from the root to the tip.

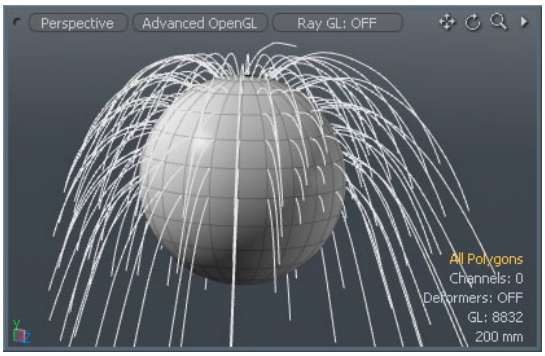
Set Guide Color



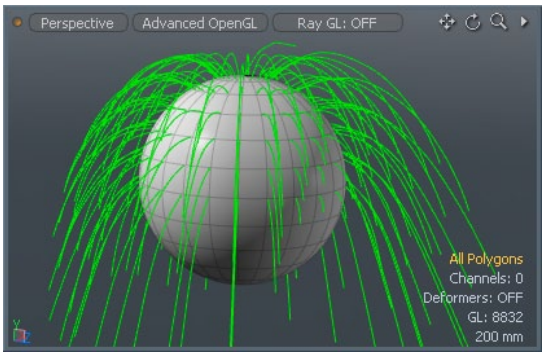
When styling fur, depending on the source surfaces coloration, it can be difficult to see the guides clearly. The 'Set Guide Color' function allows users to override the default guide color (temporarily if you wish) and set the guides color to any arbitrary value desired. To apply, simply press the 'Set Guide Color' button, located below the the 'Hair Tools' subtab of the 'Paint' toolbox to open the 'Hair Guide Color' dialog.



Enable the 'Override hair guides color' option, define a color and press 'OK'. All guides will use this color for display. To disable, open the 'Hair Guide Color' dialog again and disable the 'Override' option and press 'OK'. As the color override is system wide, this can also be useful when working with guides for other means as well. Users should note that the color display is only visible while in the Shaded and Advanced OpenGL display modes.



Hair Guide set at default color.

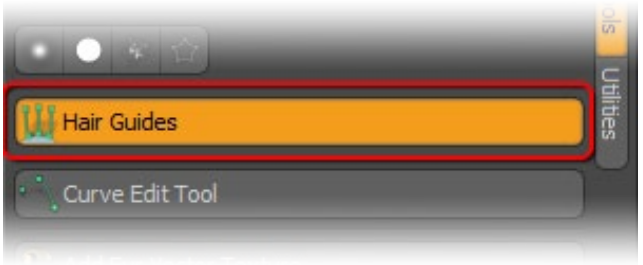


Hair Guide Color Override enabled.

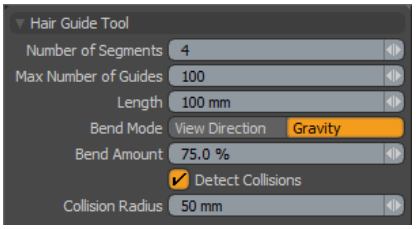
Hair Guides

Much like the sculpt and painting tools, modo offers a variety of tools meant specifically for styling Hair and its associated guides. Located in the 'Paint' tab of the default workspace, the 'Hair Tools' tab (along the edge of the toolbox) give you everything you need for combing and sculpting your fur into the proper 'do'. Many longer hair

styles will require the use of guides to properly control the shape of the rendered hair. Generating these guides is simple when using the 'Hair Guides' tool.



Hair guides can be used to control any fur layer added to a material, granted the texture layer is enabled to do so. When using the 'Hair Guides' tool, the generated curves inherit the material tag from the selected surface, so it is best to name your fur surface with a proper material before activating it. If your polygons aren't already selected, highlight the polygons you want your fur to grow from and keep them highlighted. With the tool, activated, you get this properties panel allowing you to click in your 3D viewport to actually generate the fur guides. Clicking and hauling left of right will grow or shrink the guides in length. You can also manually set the length directly in the tool properties.



Hair Guide Tool--

Number of Segments: This setting is independent of the fur segments, specifying the number of segments used to create the guides themselves. Increasing the number of guides can allow for more detailed control, but watch out as manipulating them with too many segments can become cumbersome and may produce less than satisfactory results. Add just as many as you need.

Max Number of Guides: Generally, with this tool, a guide is created at each vertex position on the surface. If your surface has a lot of polygons, you may wish to control the number of guides it creates using the 'Max Number of Guides'. If your selection has less than the max number of vertex points, only that many guides will be created.

Length: You can use this setting to manually set the guide length to a specific value.

Bend Mode: With 'Bend Mode' you can choose how the generated guides will bend based on 'View Direction' or 'Gravity'. With 'View Direction', the guides will grow straight from the normal direction of the polygon, but bend based on the rotation of the currently active viewport. This is cool because you can select a few polygons from your

fur surface and create the guides with the view rotated one direction. Then select some other polygons and create the guides with a different rotation, getting you half way there with the styling. 'Gravity' will bend the guides toward the negative 'Y' direction.

Bend Amount: The 'Bend Amount' controls the how far a guide will bend when it is created. A setting of 0% would grow guides perfectly straight in the direction of the vertex normal at its root, increasing this value will increase the intensity of the bend.

Detect Collisions: When creating the guides, you might find some are intersecting with surrounding surfaces. Enabling 'Detect Collisions' will avoid this situation by detecting surrounding surfaces realistically drooping and bending the guides around them. Best example is generating guides for long hair, you don't want the hair to intersect the shoulders, so enable this setting to drape the guides over the shoulder for a more realistic look

Collision Radius: You still might find that fur generated from guides intersects with a surface. When 'Detect Collisions' is enabled, the collision radius allows the user to offset the guides from the surface by the specified amount to avoid the intersection. Quite often guides will be used in 'Range' mode, generating fur fibers within a certain range around the guide. It may be helpful to set the collision radius to half the range (as range is a diameter) to effectively avoid fibers intersecting the surfaces as well.

Vertex Map Tools Reference

Vertex Map painting tools are useful for a variety of functions. Most commonly users working with game assets can apply Vertex Color maps in addition to textures for added variety and depth, and animators can use Weight Maps to control how deformers influence deformations applied to geometry. The Vertex Map tools gives users added control and finesse in either situation. Located in the 'Painting ' interface tab under the 'Vertex Map Tools' subtab of the toolbox. Invoking any of the tools will display the associated properties in the appropriate viewport panel. To paint, users will need to select both a Tool (*brush*) and a 'Tip' which designates how the brush itself applies to the vertex map. It occasionally will be necessary to select the target Vertex Map within the 'Lists' viewport. When working with a mouse, users can simply LMB+drag over the surface to apply the stroke. The LMB click is applied automatically when using a pressure sensitive stylus and tablet. Users may also RMB+click and drag in the viewport to interactively scale the size of the brush tip. For more information of working with Vertex Maps, please reference that section of the documentation.

Vertex Map Tools			
Paint	Scale Down	Erase	Set Value
Smooth			

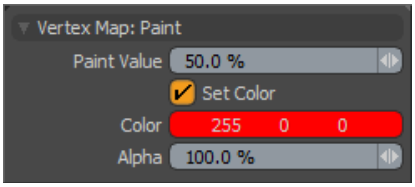
Vertex Map Paintbrush

The Vertex Paint tool provides users a way to define and adjust 'Vertex Color' maps and 'Vertex Weight' maps interactively. Using the familiar painting workflow already established for creation of bitmap based textures, users can select the 'Vertex Paint' tool and using either a tablet based input or a mouse and paint directly onto the surface in the 3D viewport.

Using the tool

The Vertex Paint tool is located in the 'Paint' interface tab, under the 'Vertex Map Tools' subtab of it's toolbox. In order to use the Vertex Paint tool, users will first need to select the target Mesh item layer in the Items List. Users will also need to create and/or select a 'Vertex Color Map' or a 'Weight Map' in the Vertex Map list viewport. Please reference the 'Working with Vertex Maps' page of the documentation for information on creating vertex maps.

Upon selecting the tool, its attributes appear in the tool properties viewport. When first selected, the tool will also automatically select the brush tip, by default the 'Smooth Brush'. Users can select one of the alternate brush tips before proceeding or RMB+click in the 3D viewport to interactively scale the brush size. Users can then simply LMB+click and drag in the viewport to apply the selected values to the designated map. Pressing and holding down the 'Ctrl' key will negate the values applied to Weight Maps, reducing the Weight Amount instead of increasing it. "Pressing and holding down the 'Shift' key will apply the Smooth VMap function.



Paintbrush--

Paint Value: The 'Paint Value' option determines the the strength of the Painting action. Higher values will saturate the canvas more quickly than lower values. Set to 100% to quickly overlay an area, and to 5% or under to make subtle adjustments.

Set Color: When applying values to a Vertex Color Map, the 'Set Color' option *enabled* automatically. Users can then define a 'Color' value and apply it to the target map.

Color: Defines an RGB value that is applied to the target Vertex Color Map.

Alpha: This value determines how the total opacity of the color applied to the Vertex Color Map.

TIP: Vertex Map values are stored by the vertices themselves of the mesh's surface. Therefor it is not possible to resolve any finer detail than the resolution of the target geometry (un-subdivided). If subdividing the mesh is undesirable, then the use of a bitmap based texture would be a better option to assign colors to the surface.

Vertex Map Smooth

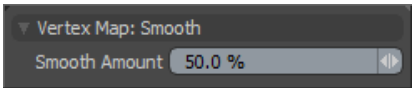
The Vertex Map Smooth tool provides users a way to adjust 'Vertex Color' maps and 'Vertex Weight' maps, interactively softening and blurring the target maps as the brush is dragged over the surface.

Using the tool

The Vertex Map Smooth tool is located in the 'Paint' interface tab, under the 'Vertex Map Tools' subtab of it's toolbox. In order to use the Vertex Map Smooth tool, users will first need to select the target Mesh item layer in the Items List. Users will also need to create and/or select a 'Vertex Color Map' or a 'Weight Map' in the Vertex Map list

viewport. Please reference the 'Working with Vertex Maps' page of the documentation for information on creating vertex maps.

Upon selecting the tool, its attributes appear in the tool properties viewport. When first selected, the tool will also automatically select the brush tip, by default the 'Smooth Brush'. Users can select one of the alternate brush tips before proceeding or RMB+click in the 3D viewport to interactively scale the brush size. Users can then simply LMB+click and drag in the viewport to apply the smoothing to the designated map. Pressing and holding down the 'Ctrl' key will negate the smoothing, producing a sharpening like effect.



Vertex Map Smooth--

Smooth Amount: The 'Smooth Amount' option controls the strength of the smoothing applied to the Vertex Map. Higher values (>50%) will produce increasingly softer results, where lower values (<50%) will produce less noticeable softening.

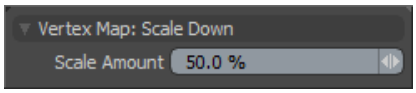
Vertex Map Tools

The Vertex Map tools provides users a way to adjust 'Vertex Color' maps and 'Vertex Weight' maps, interactively adjusting the target maps as the brush is dragged over the surface.

Using the tools

The Vertex Map tools is located in the 'Paint' interface tab, under the 'Vertex Map Tools' subtab of it's toolbox. In order to use the Vertex Map tools, users will first need to select the target Mesh item layer in the Items List. Users will also need to create and/or select a 'Vertex Color Map' or a 'Weight Map' in the Vertex Map list viewport. Please reference the 'Working with Vertex Maps' page of the documentation for information on creating vertex maps.

Upon selecting the tool, its attributes appear in the tool properties viewport. When first selected, the tool will also automatically select the brush tip, by default the 'Smooth Brush'. Users can select one of the alternate brush tips before proceeding or RMB+click in the 3D viewport to interactively scale the brush size. Users can then simply LMB+click and drag in the viewport to apply the brushes effect to the designated map. Pressing and holding down the 'Ctrl' key will typically negate the effects of the brush.



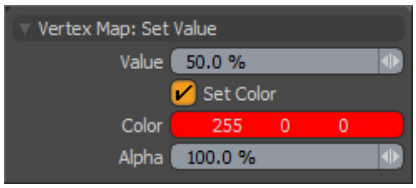
Scale Down

The "Scale Down' tool will interactively adjust the values in a map down toward '0' (zero), sort of like slowly reducing the values toward '0' (zero). Pressing and holding the 'Ctrl' key will Scale Up the values. The overall effect is sort of similar to dodging and burning in an image editor.

Blend Mode: The 'Scale Amount' determines the strength of the scaling applied to the values.

Erase

The 'Erase' tool will remove any values in the target map that the brush touches. This isn't just lowering the value of the map to 0%, but completely removes the value for the vertex. No property form for this tool as there are no user adjustable values necessary.



Set Value

The 'Set Value' tool is sort of like the paint tool combined with the erase too, in that it affects any vertex it touches 100% with no fading or attenuation. Pressing and holding the 'Ctrl' key will erase the value.

Value: When editing Weight Map values, this option determines the Weight Value applied.

Set Color: When painting onto Vertex Color Maps, this option is enabled automatically allowing users to specify a color to apply to the map.

Color: Defines an RGB value that is applied to the target Vertex Color Map.

Alpha: Defines an Alpha value that is applied to the target RGBA Vertex Color Map.

Particle Tools

Particle Tools Reference

Particles are simply single vertices that provide important functions, acting as point sources for Replicators and Texture Replicators, as well as for Volumes, Blobs and Sprites. The Particle Tools allow users to create and edit particle point clouds using the familiar brush based workflow. Users can additionally edit Vertex Maps using the 'Size' and 'Dissolve' tools directly modifying the associated 'Particle Vertex Maps'.

Particle Creation	Particle Editing		
Particle Paint	Smooth	Pinch	Flatten
	Smudge	Tangent Pinch	Size
	Move	Spin	Dissolve

Particle Paint Tool

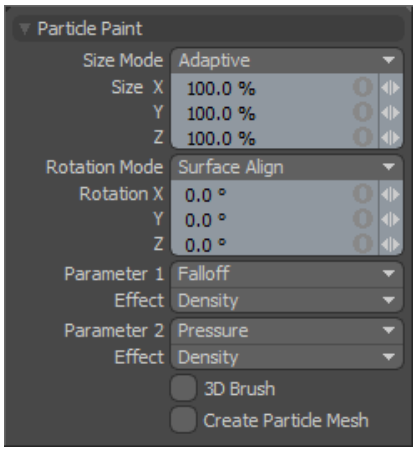
Much like modo's painting tool allow users to apply color onto a surface, the 'Particle Paint' tool allows users to spray particles onto a surface in an intuitive and controllable manner. Particles are simply single vertices that can be used to control a number of other functions in modo, such as point sources for goemetry Replicators, and the Texture Replicator, as well as point clouds for Blobs, Volumes and Sprites. Once the particles have been created, users can continue editing them by using the Particle Paint tools; combined with their 'Particle Constraint' option, users are afforded ultimate control over the placement of the individual particles. Of course the generated Particles can also take advantage of all the additional controls available with Particle Vertex Maps, such as the Transform Map, providing extensive controls for the most demanding applications.

Using the Tool

The Particle Paint Tool is found in the 'Paint' interface tab, under the 'Particle Tools' subtab of it's toolbox. Particles will only spray onto polygon surfaces, so users will need to select the appropriate target layer to spray onto prior to activating the tool. Upon selecting the tool, its attributes appear in the tool properties viewport. When first selected, the tool will also automatically select the Brush Tip, by default the 'Smooth Brush'. Users can select one of the alternate Brush Tips before proceeding or RMB+click in the 3D viewport to interactively scale the brush size. Users can then

simply LMB+click and drag in the viewport to start creating particles. Addition control is added by applying the 'Parameter 1 & 2' options.

Particle Paint--



Size Mode: When creating the Particles, a 'Particle Size' Vertex Map can is generated automatically, storing additional information that can be used to control the scaling of the eventual clones or volume elements. The results of the scaling are dependant on the following settings-

Uniform-- The 'Uniform' option results in identically sized elements, controlled by the 'Size X,Y & Z' setting values.

Adaptive-- The 'Adaptive' option will scale elements larger or smaller based on the size of the Brush Tip used.

Random-- The 'Random' option will randomly scale the elements up the the maximum scaling amount, determined by the 'Scale X, Y & Z' values.

Adaptive+Random-- The 'Adaptive+Random' option will randomly scale the elements in relation to the size of the Brush Tip, with the maximum scaling amount determined by the 'Scale X,Y &Z' values.

Rotation Mode: When creating the Particles, a 'Particle Transform' Vertex Map can is generated automatically, storing additional information that can be used to control the rotation of the eventual cloned or volume elements. The results of the rotation are dependant on the following settings-

Uniform-- The 'Uniform' option results in identically oriented elements, controlled by the 'Rotation X,Y & Z' setting values.

Surface Align-- The 'Surface Align' option results in elements that are oriented to align with the surface polygons normal direction.

Random-- The 'Random' option results in randomized rotations for each element, up to the maximum rotation amount determined by the 'Rotation X,Y &Z' setting values.

Surface Align+Random-- The 'Surface Align+Random' first orients the element to the surfaces normal direction, and then offsets it with an additional random rotation up to the maximum amount determined by the 'Rotation X,Y &Z' setting values.

Surface Align+Tablet-- The 'Surface Align+Tablet' first orients the element to the

surfaces normal direction, and then offsets it based on the angle of a pen on a tablet.

Surface Align+Tablet (No Pitch)-- The 'Surface Align+Tablet' first orients the element to the surfaces normal direction ignoring the sloping angle (pitch), and then offsets it based on the angle of a pen on a tablet

Screen Align+Tablet-- Description coming.

Screen Align+Tablet (No Pitch)-- Description coming.

Parameter 1/2: The 'Parameter' options provider users with two additional controls in creating particles. It is a two-step process. First the controlling input is defined as the 'Parameter' option, and the the result of that control is defined as the 'Effect'. Both options are in combination with the 'Brush Tip' input. Some options relate to using a Pen and Tablet input combination to control elements.

Falloff-- The 'Falloff' option allows modo's many 'Falloffs' to be used to determine the 'Effect'. Since the Brush Size action superseded the falloff set with the RMB< make sure to define the falloff prior to activating the tool.

Pressure-- The 'Pressure' of input related to how hard the pen is pressed against the tablets surface controls the 'Effect'.

Slope-- The downward 'Slope' of the target polygon surface controls the 'Effect'.

Altitude--

1.0 Slope--

1.0 Altitude--

Effect: The 'Effect' is simply the results that is controlled by the chosen 'Parameter' input option.

(none)-- The 'None' option disabled the Parameter/Effect option.

Density-- The 'Density' option controls the number of sprayed particles as determines by the input 'Parameter' option.

Size-- The 'Size' option controls the scale of sprayed particles as determines by the input 'Parameter' option.

Rotation-- The 'Rotation' option controls the orientation of sprayed particles as determines by the input 'Parameter' option.

Size+Density-- The 'Size+Density' option controls the number and scale of sprayed particles as determined by the input 'Parameter' option.

3D Brush: When '3D Brush' is enabled, particels will be created within the 3D volume of the brush tip, adding depth to the particles, so they arent placed directly against the constraining background surface.

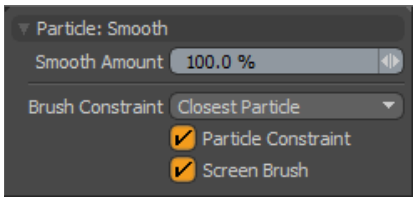
Create Particle Mesh: When the 'Create Particle Mesh' option is enabled, an additional 'Mesh' item layer is created containing all the particles as a child item of the target layer. This eliminates the need to copy and paste the vertices into a new mesh layer manually when used as a Point Source.

Particle Paint Smooth

The Particle Tools ('Smooth', 'Smudge', 'Move', 'Pinch', 'Tangent Pinch', 'Spin' & 'Flatten') are meant as a way to modify particle positions within a point cloud. Much like the sculpt tools are for shaping surfaces, the particle tools allow users to control, position and direct their particles, even individually, to produce the desired results. Particles can be any number of things, a Mesh layer with single point vertices or a collection of single point polygons (also called 'point clouds'). While there are a number of ways users could approach the creation of a point cloud - the most direct would be the use of the 'Particle Paint' tool. Note that the Particle tools cannot be used to modify procedurally generated particles such as those of the 'Particle Cloud' or 'Particle Generator' items.

Using the Tool

The Particle Tools are found in the 'Paint' interface tab, under the 'Particle Tools' subtab of it's toolbox. Upon selecting the tool, its attributes appear in the tool properties viewport. When first selected, the tool will also automatically select the brush tip, by default the 'Smooth Brush'. Users can select one of the alternate brush tips before proceeding or RMB+click in the 3D viewport to interactively scale the brush size. Users can then simply LMB+click and drag in the viewport to affect the particles. The Smooth tool will gather particles toward the brushes center, the speed at which they move is dependant on the 'Smooth Amount'. Pressing and holding down the 'Ctrl' key will negate the affects of the tool, spreading the particles outward, away from its center.



Particle Smooth--

Smooth Amount: Controls the strength of the 'Smoothing' which is essentially the speed at which the affected particles will move inward or outward.

Brush Constraint: Description coming-

Closest Particle-- Description coming.

Grid-- Description coming.

Center Particle-- Description coming.

Particle Constraint: When the 'Particle Constrain' option is enabled, each individual vertex is constrained against any foreground polygons ensuring it is always resting on its surface as it moves. Work in the same manner as the 'Constrain to Background' - 'Point' mode, but is not dependant upon background geometry.

Screen Brush: Description coming.

TIP: When working with the 'Particle Constraint' option, constraining geometry doesn't necessarily have to be in the same layer as the point cloud. Users can hold down the 'Ctrl' key and select additional Item layers, bringing them into the foreground. The edited particles will conform to the foreground polygons surface but remain within their initial layer.

Particle Paint Smudge

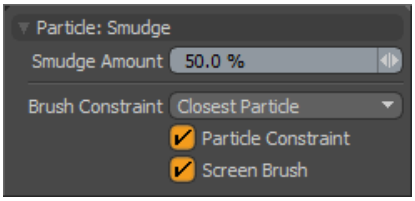
The Particle Tools ('Smooth', 'Smudge', 'Move', 'Pinch', 'Tangent Pinch', 'Spin' & 'Flatten') are meant as a way to modify particle positions within a point cloud. Much like the sculpt tools are for shaping surfaces, the particle tools allow users to control, position and direct their particles, even individually, to produce the desired results. Particles can be any number of things, a Mesh layer with single point vertices or a collection of single point polygons (also called 'point clouds'). While there are a number of ways users could approach the creation of a point cloud - the most direct would be the use of the 'Particle Paint' tool. Note that the Particle tools cannot be used to modify procedurally generated particles such as those of the 'Particle Cloud' or 'Particle Generator' items.

Using the Tool

The Particle Tools are found in the 'Paint' interface tab, under the 'Particle Tools' subtab of it's toolbox. Upon selecting the tool, its attributes appear in the tool properties viewport. When first selected, the tool will also automatically select the brush tip, by default the 'Smooth Brush'. Users can select one of the alternate brush tips before proceeding or RMB+click in the 3D viewport to interactively scale the brush size. Users can then simply LMB+click and drag in the viewport to affect the particles. The Smudge tool will drag the particles in the direction of the brush, like they are being smeared. The distance in which they are dragged is dependant on the 'Smudge Amount'. Pressing and holding down the 'Ctrl' key will negate the affects of the tool, dragging the particles in the opposite direction of the brushes movement.

Particle Smooth--

Smooth Amount: Controls the strength of the 'Smudge' which is essentially the distance the affected particles will move when dragged.



Brush Constraint: Description coming-

Closest Particle-- Description coming.

Grid-- Description coming.

Center Particle-- Description coming.

Particle Constraint: When the 'Particle Constrain' option is *enabled*, each individual vertex is constrained against any foreground polygons ensuring it is always resting on its surface as it moves. Work in the same manner as the 'Constrain to Background' - 'Point' mode, but is not dependant upon background geometry.

Screen Brush: Description coming.

TIP: When working with the 'Particle Constraint' option, constraining geometry doesn't necessarily have to be in the same layer as the point cloud. Users can hold down the 'Ctrl' key and select additional Item layers, bringing them into the foreground. The edited particles will conform to the foreground polygons surface but remain within their initial layer.

Particle Paint Move

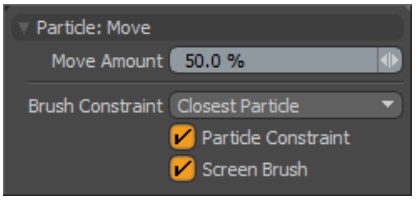
The Particle Tools ('Smooth', 'Smudge', 'Move', 'Pinch', 'Tangent Pinch', 'Spin' & 'Flatten') are meant as a way to modify particle positions within a point cloud. Much like the sculpt tools are for shaping surfaces, the particle tools allow users to control, position and direct their particles, even individually, to produce the desired results. Particles can be any number of things, a Mesh layer with single point vertices or a collection of single point polygons (also called 'point clouds'). While there are a number of ways users could approach the creation of a point cloud - the most direct would be the use of the 'Particle Paint' tool. Note that the Particle tools cannot be used to modify procedurally generated particles such as those of the 'Particle Cloud' or 'Particle Generator' items.

Using the Tool

The Particle Tools are found in the 'Paint' interface tab, under the 'Particle Tools' subtab of it's toolbox. Upon selecting the tool, its attributes appear in the tool properties viewport. When first selected, the tool will also automatically select the brush tip, by default the 'Smooth Brush'. Users can select one of the alternate brush tips before proceeding or RMB+click in the 3D viewport to interactively scale the brush size. Users can then simply LMB+click and drag in the viewport to affect the particles. The Move tool will drag the particles the distance that the brush travels, how much they are affected is controlled by the 'Move Amount'. Pressing and holding down the 'Ctrl'

key will negate the affects of the tool, moving the particles the opposite direction of the mouse pointer.

Particle Move--



Move Amount: Controls the strength of the brush's influence in moving the particles. A value of 100% will move the particles the same distance the mouse travels, attenuating downward toward 05 where they wouldn't move at all.

Brush Constraint: Description coming-

Closest Particle-- Description coming.

Grid-- Description coming.

Center Particle-- Description coming.

Particle Constraint: When the 'Particle Constrain' option is *enabled*, each individual vertex is constrained against any foreground polygons ensuring it is always resting on its surface as it moves. Work in the same manner as the 'Constrain to Background' - 'Point' mode, but is not dependant upon background geometry.

Screen Brush: Description coming.

TIP: When working with the 'Particle Constraint' option, constraining geometry doesn't necessarily have to be in the same layer as the point cloud. Users can hold down the 'Ctrl' key and select additional Item layers, bringing them into the foreground. The edited particles will conform to the foreground polygons surface but remain within their initial layer.

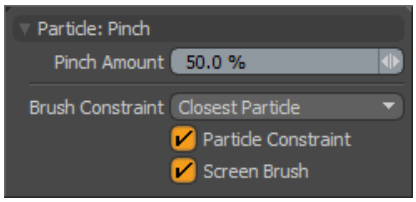
Particle Paint Pinch

The Particle Tools ('Smooth', 'Smudge', 'Move', 'Pinch', 'Tangent Pinch', 'Spin' & 'Flatten') are meant as a way to modify particle positions within a point cloud. Much like the sculpt tools are for shaping surfaces, the particle tools allow users to control, position and direct their particles, even individually, to produce the desired results. Particles can be any number of things, a Mesh layer with single point vertices or a collection of single point polygons (also called 'point clouds'). While there are a number of ways users could approach the creation of a point cloud - the most direct would be the use of the 'Particle Paint' tool. Note that the Particle tools cannot be used to modify procedurally generated particles such as those of the 'Particle Cloud' or 'Particle Generator' items.

Using the Tool

The Particle Tools are found in the 'Paint' interface tab, under the 'Particle Tools' subtab of it's toolbox. Upon selecting the tool, its attributes appear in the tool properties viewport. When first selected, the tool will also automatically select the brush tip, by default the 'Smooth Brush'. Users can select one of the alternate brush tips before proceeding or RMB+click in the 3D viewport to interactively scale the brush size. Users can then simply LMB+click and drag in the viewport to affect the particles. The Pinch tool will gather particles toward the brushes center, the speed at which they move is dependant on the 'Pinch Amount'. Pressing and holding down the 'Ctrl' key will negate the affects of the tool, spreading the particles outward, away from its center.

Particle Smooth--



Pinch Amount: Controls the strength of the 'Pinching' which is essentially the speed at which the affected particles will move inward or outward.

Brush Constraint: Description coming-

Closest Particle-- Description coming.

Grid-- Description coming.

Center Particle-- Description coming.

Particle Constraint: When the 'Particle Constrain' option is *enabled*, each individual vertex is constrained against any foreground polygons ensuring it is always resting on its surface as it moves. Work in the same manner as the 'Constrain to Background' - 'Point' mode, but is not dependant upon background geometry.

Screen Brush: Description coming.

TIP: When working with the 'Particle Constraint' option, constraining geometry doesn't necessarily have to be in the same layer as the point cloud. Users can hold down the 'Ctrl' key and select additional Item layers, bringing them into the foreground. The edited particles will conform to the foreground polygons surface but remain within their initial layer.

Particle Paint Tangent Pinch

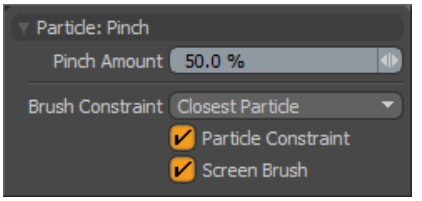
The Particle Tools ('Smooth', 'Smudge', 'Move', 'Pinch', 'Tangent Pinch', 'Spin' & 'Flatten') are meant as a way to modify particle positions within a point cloud. Much like the sculpt tools are for shaping surfaces, the particle tools allow users to control,

position and direct their particles, even individually, to produce the desired results. Particles can be any number of things, a Mesh layer with single point vertices or a collection of single point polygons (also called 'point clouds'). While there are a number of ways users could approach the creation of a point cloud - the most direct would be the use of the 'Particle Paint' tool. Note that the Particle tools cannot be used to modify procedurally generated particles such as those of the 'Particle Cloud' or 'Particle Generator' items.

Using the Tool

The Particle Tools are found in the 'Paint' interface tab, under the 'Particle Tools' subtab of it's toolbox. Upon selecting the tool, its attributes appear in the tool properties viewport. When first selected, the tool will also automatically select the brush tip, by default the 'Smooth Brush'. Users can select one of the alternate brush tips before proceeding or RMB+click in the 3D viewport to interactively scale the brush size. Users can then simply LMB+click and drag in the viewport to affect the particles. The Tangent Pinch tool will gather particles toward the brushes center, the speed at which they move is dependant on the 'Tangent Pinch Amount'. Pressing and holding down the 'Ctrl' key will negate the affects of the tool, spreading the particles outward, away from its center.

Particle Smooth--



Tangent Pinch Amount: Controls the strength of the 'Pinching' which is essentially the speed at which the affected particles will move inward or outward.

Brush Constraint: Description coming-

Closest Particle-- Description coming.

Grid-- Description coming.

Center Particle-- Description coming.

Particle Constraint: When the 'Particle Constrain' option is *enabled,* each individual vertex is constrained against any foreground polygons ensuring it is always resting on its surface as it moves. Work in the same manner as the 'Constrain to Background' - 'Point' mode, but is not dependant upon background geometry.

Screen Brush: Description coming.

TIP: When working with the 'Particle Constraint' option, constraining geometry doesn't necessarily have to be in the same layer as the point cloud. Users can hold down the 'Ctrl' key and select additional Item layers, bringing them into the foreground. The edited particles will conform to the foreground polygons surface but remain within their initial layer.

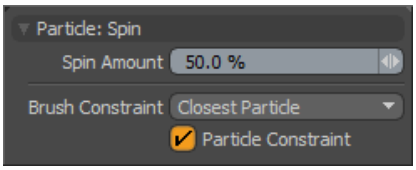
Particle Paint Spin

The Particle Tools ('Smooth', 'Smudge', 'Move', 'Pinch', 'Tangent Pinch', 'Spin' & 'Flatten') are meant as a way to modify particle positions within a point cloud. Much like the sculpt tools are for shaping surfaces, the particle tools allow users to control, position and direct their particles, even individually, to produce the desired results. Particles can be any number of things, a Mesh layer with single point vertices or a collection of single point polygons (also called 'point clouds'). While there are a number of ways users could approach the creation of a point cloud - the most direct would be the use of the 'Particle Paint' tool. Note that the Particle tools cannot be used to modify procedurally generated particles such as those of the 'Particle Cloud' or 'Particle Generator' items.

Using the Tool

The Particle Tools are found in the 'Paint' interface tab, under the 'Particle Tools' subtab of it's toolbox. Upon selecting the tool, its attributes appear in the tool properties viewport. When first selected, the tool will also automatically select the brush tip, by default the 'Smooth Brush'. Users can select one of the alternate brush tips before proceeding or RMB+click in the 3D viewport to interactively scale the brush size. Users can then simply LMB+click and drag in the viewport to affect the particles. The Spin tool will rotate the particles around the brushes center, the speed at which they rotate is dependant on the 'Spin Amount'. Pressing and holding down the 'Ctrl' key will negate the affects of the tool, rotating the particles the opposite direction.

Particle Smooth--



Spin Amount: Controls the strength of the 'Smoothing' which is essentially the speed at which the affected particles will move inward or outward.

Brush Constraint: Description coming-

Closest Particle-- Description coming.

Grid-- Description coming.

Center Particle-- Description coming.

Particle Constraint: When the 'Particle Constrain' option is *enabled*, each individual vertex is constrained against any foreground polygons ensuring it is always resting on its surface as it moves. Work in the same manner as the 'Constrain to Background' - 'Point' mode, but is not dependant upon background geometry.

TIP: When working with the 'Particle Constraint' option, constraining geometry doesn't necessarily have to be in the same layer as the point cloud. Users can hold down the 'Ctrl' key and select additional Item layers, bringing them into the foreground. The edited particles will conform to the foreground polygons surface but remain within their initial layer.

Particle Paint Flatten

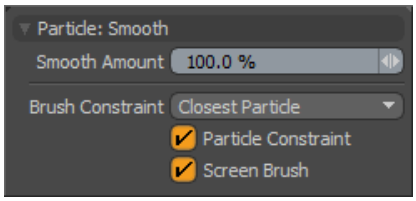
The Particle Tools ('Smooth', 'Smudge', 'Move', 'Pinch', 'Tangent Pinch', 'Spin' & 'Flatten') are meant as a way to modify particle positions within a point cloud. Much like the sculpt tools are for shaping surfaces, the particle tools allow users to control, position and direct their particles, even individually, to produce the desired results. Particles can be any number of things, a Mesh layer with single point vertices or a collection of single point polygons (also called 'point clouds'). While there are a number of ways users could approach the creation of a point cloud - the most direct would be the use of the 'Particle Paint' tool. Note that the Particle tools cannot be used to modify procedurally generated particles such as those of the 'Particle Cloud' or 'Particle Generator' items.

Using the Tool

The Particle Tools are found in the 'Paint' interface tab, under the 'Particle Tools' subtab of it's toolbox. Upon selecting the tool, its attributes appear in the tool properties viewport. When first selected, the tool will also automatically select the brush tip, by default the 'Smooth Brush'. Users can select one of the alternate brush tips before proceeding or RMB+click in the 3D viewport to interactively scale the brush size. Users can then simply LMB+click and drag in the viewport to affect the particles. The Flatten tool will gather particles toward the brushes center, the speed at which they move is dependant on the 'Flatten Amount'. Pressing and holding down the 'Ctrl' key will negate the affects of the tool, spreading the particles outward, away from its center.

Particle Smooth--

Flatten Amount: Controls the strength of the 'Smoothing' which is essentially the speed at which the affected particles will move inward or outward.



Brush Constraint: Description coming-

Closest Particle-- Description coming.

Grid-- Description coming.

Center Particle-- Description coming.

Particle Constraint: When the 'Particle Constrain' option is *enabled*, each individual vertex is constrained against any foreground polygons ensuring it is always resting on its surface as it moves. Work in the same manner as the 'Constrain to Background' - 'Point' mode, but is not dependant upon background geometry.

Screen Brush: Description coming.

TIP: When working with the 'Particle Constraint' option, constraining geometry doesn't necessarily have to be in the same layer as the point cloud. Users can hold down the 'Ctrl' key and select additional Item layers, bringing them into the foreground. The edited particles will conform to the foreground polygons surface but remain within their initial layer.

Particle Paint Size

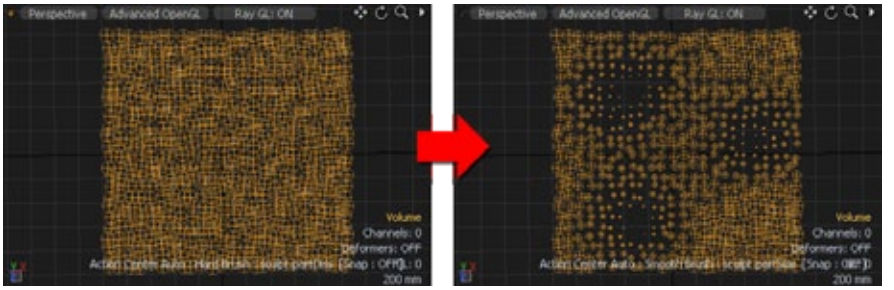
The Particle Tools ('Size' & 'Dissolve') are meant as a way to modify particle related attributes, giving users control over settings that would be difficult if not impossible to control otherwise. The 'Particle Size' function gives users direct control of the scale of 'Volumes', 'Blobs' and 'Sprites' on an individual particle basis, when applied as a 'Particle Size' vertex map. Particles can be any number of things, a Mesh layer with single point vertices or a collection of single point polygons (also called 'point clouds'). While there are a number of ways users could approach the creation of a point cloud - the most direct would be the use of the 'Particle Paint' tool. Note that Particle Vertex Maps cannot be used to control procedural particles such as those of the 'Particle Cloud' or 'Particle Generator' items.

Using the Tool

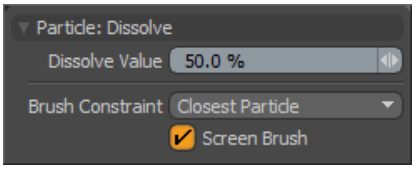
The Particle Tools are found in the 'Paint' interface tab, under the 'Particle Tools' subtab of it's toolbox. In order to use the Particle Size tool, users will first need to

select the target point cloud Mesh item layer in the Items List. Users will also need to create and select a 'Particle Size Map' in the Vertex Map list viewport. Please reference the 'Working with Vertex Maps' page of the documentation for information on creating vertex maps. It will also be better to have already applied the 'Volume', 'Blob' or 'Sprite' items to the point cloud as their viewport representations will provide interactive feedback to the application of the Size and Dissolve functions (see sample below).

Upon selecting the tool, its attributes appear in the tool properties viewport. When first selected, the tool will also automatically select the brush tip, by default the 'Smooth Brush'. Users can select one of the alternate brush tips before proceeding or RMB+click in the 3D viewport to interactively scale the brush size. Users can then simply LMB+click and drag in the viewport to affect the particles. Pressing and holding down the 'Ctrl' key will negate the affects of the tool, decreasing the particle size instead of increasing it.



Particle: Size--



Dissolve Value: Controls the strength of scaling amount applied to the particles, lower values will scale the particles more slowly than higher values will.

Brush Constraint: Description coming-

Closest Particle-- Description coming.

Grid-- Description coming.

Center Particle-- Description coming.

Screen Brush: Description coming.

TIP: When working with the 'Particle Constraint' option, constraining geometry doesn't necessarily have to be in the same layer as the point cloud. Users can hold down the 'Ctrl' key and select additional Item layers, bringing them into the foreground. The edited particles will conform to the foreground polygons surface but remain within their initial layer.

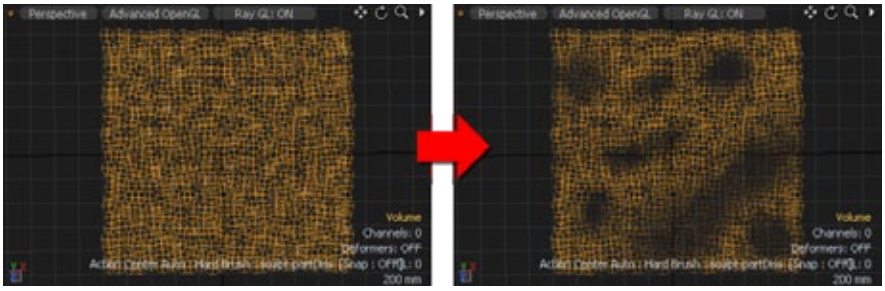
Particle Paint Dissolve

The Particle Tools ('Size' & 'Dissolve') are meant as a way to modify particle related attributes, giving users control over settings that would be difficult if not impossible to control otherwise. The 'Particle Dissolve' function gives users direct control of the opacity of 'Volumes' and 'Sprites' on an individual particle basis, when applied as a 'Particle Dissolve' vertex map. Particles can be any number of things, a Mesh layer with single point vertices or a collection of single point polygons (also called 'point clouds'). While there are a number of ways users could approach the creation of a point cloud - the most direct would be the use of the 'Particle Paint' tool. Note that Particle Vertex Maps cannot be used to control procedural particles such as those of the 'Particle Cloud' or 'Particle Generator' items.

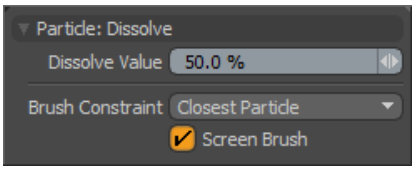
Using the Tool

The Particle Tools are found in the 'Paint' interface tab, under the 'Particle Tools' subtab of it's toolbox. In order to use the Particle Dissolve tool, users will first need to select the target point cloud Mesh item layer in the Items List. Users will also need to create and select a 'Particle Dissolve Map' in the Vertex Map list viewport. Please reference the 'Working with Vertex Maps' page of the documentation for information on creating vertex maps. It will also be better to have already applied the 'Volume' or 'Sprite' items to the point cloud as their viewport representations will provide interactive feedback to the application of the Size and Dissolve functions (see sample below).

Upon selecting the tool, its attributes appear in the tool properties viewport. When first selected, the tool will also automatically select the brush tip, by default the 'Smooth Brush'. Users can select one of the alternate brush tips before proceeding or RMB+click in the 3D viewport to interactively scale the brush size. Users can then simply LMB+click and drag in the viewport to affect the particles. Pressing and holding down the 'Ctrl' key will negate the affects of the tool, adding opacity instead of removing it.



Particle: Dissolve--



Dissolve Value: Controls the strength of dissolve amount applied to the particles, lower values will affect the transparency more slowly than higher values will.

Brush Constraint: Description coming-

Closest Particle-- Description coming.

Grid-- Description coming.

Center Particle-- Description coming.

Screen Brush: Description coming.

TIP: If any Shader Tree layers defined as a 'Particle Dissolve' effect are applied to the particle item, these values will supercede those of the Particle Dissolve vertex map.

Commands

Image Adjustment Commands

The image adjustment commands are located in the menu bar under the "Texture > Image Processing" menu. If the commands are inactive, this indicates that no image is currently selected. Use the 'Shader Tree' or 'Images' viewport tab (or Clip List) to select an image and enable the commands for use.

Invert Image

This command will simply invert the RGB color values of the currently selected image map producing a negative of the original.

Clear Image

This option deletes all the pixels in a given image making it effectively a blank image, while retaining the image size, resolution and filename.

Expand UV Seams

This command analyzes the currently selected image map and the UV map to which it is associated and expands the pixels outward beyond the UV map boundary. This can

be very useful for repairing texture boundaries when aggressive texture antialiasing or displacements can cause seams to become visible. The amount of expansion is based on the 'Bake UV Border Size' preference setting.

Convert to Vector Displacement

This command converts a grayscale height-based displacement map into a vector displacement image. This is useful when importing sculpt data from other applications as they are only capable of generating grayscale height-fields. To take advantage of modo's vector displacement capability, users must either sculpt on a separate vector displacement image layer or convert the height-field to vector displacement using this command.

Set Resolution

This command simply changes the currently selected image to a user defined resolution. This is very useful if you are working on a texture or sculpt and decide you need more detail as you can simply res-up the image with this command. The command is based on power of 2 images (64, 128, 256, 512, 1024, 2048, 4096) as most OpenGL hardware is optimized for square aspect image maps.

Flatten Layers

This command works on a group of selected image texture layers in the Shader Tree. All images in use must be of the same type and resolution. The command will flatten the layers into a single image texture layer using a single image map that is the composite of the original images selected. Before running the command, select the layers to flatten by pressing the 'Ctrl' key and LMB+clicking each layer in the Shader Tree you wish to combine.

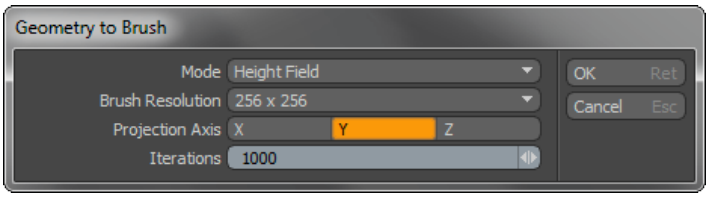
Sculpting Commands

modo provides a few commands to ease the use of sculpting. Found in the 'Utilities' subtab of the default Painting menu.

Geometry to Brush

This is a powerful feature that allows you to convert any selected geometry into a brush for painting and/or sculpting. The command can create height-field (grayscale

image) or vector displacement (32-bit color image) brushes from the currently selected geometry. In order for the command to work properly, there must be open edges surrounding the geometry to be converted. Enclosed objects (sometimes referred to as *water tight*) such as a sphere or torus will produce no results.



Mode: Users can choose between two options; 'Height Field' creates a grayscale image that is useful for displacing geometry, only taking in to account the actual height of the geometry, ignoring insets and such. The 'Vector Displacement' option is more accurate in capturing the contours of the geometry, accommodating insets and undercuts in the geometry.

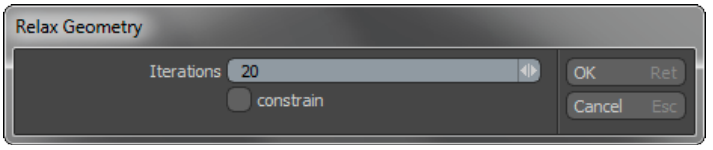
Brush Resolution: This option defines the final resolution of the generated image map. Higher resolutions may be necessary to capture finer detail.

Projection Axis: Projection Axis defines the facing direction of the geometry,

Iterations: Higher iterations produce more accurate results, but increasingly take longer to calculate. The default setting of '1000' should be fine in most cases. If you feel the shape isn't being captured properly of the base geometry, you might try increasing the 'Iterations' value.

Relax Mesh

This command uses the same mesh smoothing algorithm leveraged for the Geometry to Brush function. When fired the command requests a number of iterations (default is 5) this will affect the strength of the smoothing performed. This is a very useful command for converting a mesh with detail modeled in to a sculpt-ready mesh. It can also be very useful when creating UV maps for very detailed objects when applied to a Morph map. Users can simply create a morph vertex map, apply the Relax Mesh option to the new Morph, and then UV map the simplified mesh. Once finished, users can delete the additional morph and the UV map will be retained for the original mesh.



Iterations: This value defines how many times the relax operation is applied to the model. Higher values will produce increasingly smoother geometric surfaces.

Constrain: When the 'Constrain' Option is selected, the relaxed mesh with take into account additional unselected geometry in a mesh item layer. This function gives users an additional degree of control over smoothing, where geometry can be placed inside a mesh, then select the mesh that is to be smoothed, select the Constrain option and apply. The relaxed mesh will conform around the unselected geometry. This function works much like the Background Constrain function, but requires the additional geometry to be in the same mesh layer.

Copy to Morph

This is a convenience command that will quickly copy the current state of your mesh into an 'Absolute Morph' vertex map. This allows you to quickly store the current sculpt state of your mesh which you can restore at any time via the Morph Tool. It is important to note that this is only relevant to the changes made directly to the mesh and is not useful when working with image based sculpting.

Morph Map: This option allows users to either create a new 'Morph' vertex map, or to store the sculpt information in an existing Morph Map by selecting it from the option list.

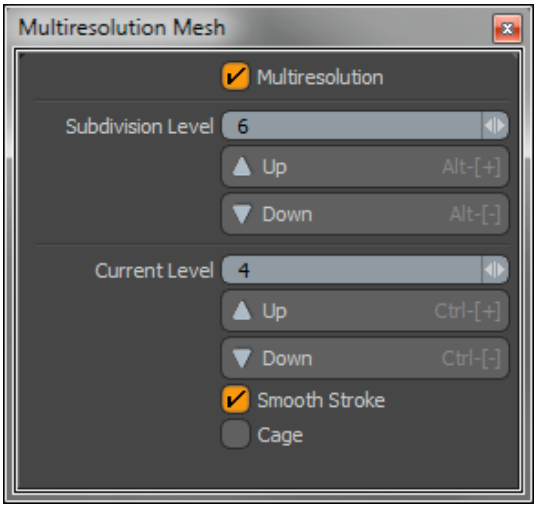
Windows

Multi Resolution Palette

A 'Multiresolution Mesh' is a special mode of the Pixar Subdivision Surfaces geometry type meant specifically for sculpting, allowing users to interactively step up and down through multiple subdivision levels. Users can easily sculpt a form at a lower subdivision level, step up the level to add fine details, then return to a lower subdivision level to adjust the overall shape without affecting the fine details developed at the higher level.

When dealing with a Multi Resolution mesh for sculpting, the 'Multi Resolution Mesh' palette helps to give users a visual reference to the number of subdivisions, the current subdivision level as well as controls to easily step up and down through the levels.

The basic options are part of the 'Sculpting' toolbox found in the 'Paint' interface tab, but users may prefer to access the palette directly, located in the menu bar, under the "Layout > Palettes > Multiresolution Mesh" menu.



In order to properly set-up a mesh for Multi Resolution, users need to ensure the mesh is first in Pixar's 'Catmull-Clark' Subdivision mode. Catmull-Clark SubDs are differentiated by their rollover color, regular SubDs are a pale lavender color on rollover, while Catmull-Clark Subds are a pale olive color. If that's too subtle, users may also use the 'Statistics' viewport panel, and under "Polygon > By Type > Catmull-Clark", users can use the ' + ' and ' - ' buttons to *select* and *de-select* respectively, while in 'Polygons' mode, and see what surfaces are designated as such. If you find your surface is a regular Subdivision Surface, it can be converted by again using the statistics panel to select just the SubD geometry "Polygon > By Type > SubDivs", press the ' + ' to select, then press 'Tab' to disable, then press 'Shift+Tab' to convert the selection to Catmull-Clark.

Multi Resolution: Once 'Catmull-Clark' SubDivs have been specified, user need to activate the 'Multi-Resolution' checkbox in the palette, enabling the function. From there users can specify the maximum number of Subdivisions with the 'Subdivision Level' option and can then use the 'Current Level' up and down buttons to step through the levels when sculpting.

Subdivision Level: This value determines the maximum number of subdivision levels available. Higher values allow for greater sculpting detail, but keep in mind that Catmull-Clark SubDs have a greater memory and processing requirement. Levels in the range of 6 to 8 are all that should be necessary for most properly prepared models.

Unlike regular SubDs, each additional level is 4x more polygons than the previous producing a exponential scale. Adding or removing levels can be done by pressing the 'Alt+(num pad plus)+' Key and 'Alt+(num pad minus)-' keys.

Current Level: When Multi Resolution is enabled, the 'Current Level' value shows the current subdivision level for sculpting. This valuable option allows users to make broad changes at lower subdivision levels affecting fewer polygons, while retaining fine details sculpted at higher levels. Stepping through the levels can be done by pressing the 'Ctrl+(num pad plus)+' Key and 'Ctrl+(num pad minus)-' keys.

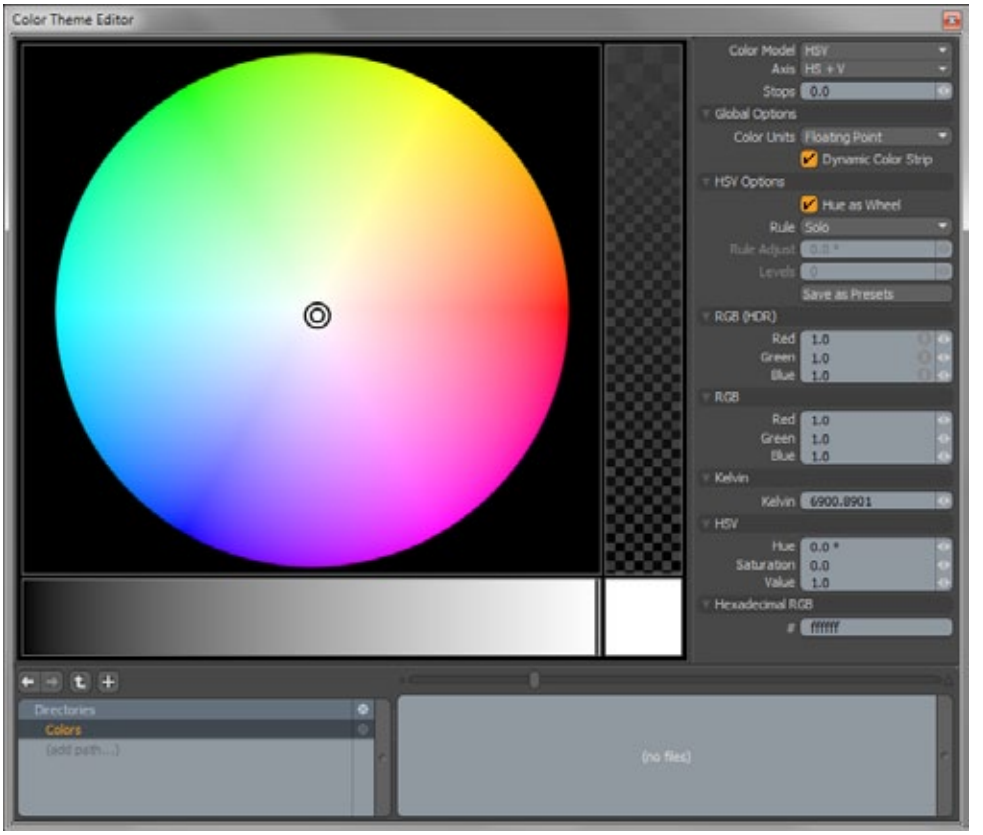
Smooth Stroke: The 'Smooth Stroke' option filters variances in pressure when sculpting on a Multi Resolution mesh.

Cage: When the 'Cage' option toggle is *enabled*, the display of Pixar Subdivisions is temporarily disabled showing the base geometry cage.

Color Theme Editor

Color is an integral function when painting and surfacing for 3D. Color sets the mood, leads a viewers eye, and provide a sense of cohesion to a scene. modo provides a dedicated window for editing colors and creating schemes for scenes, that can then be easily applied when surfacing, setting up lights and any other color related operation. The window itself is really the standard color picker popup that appears when any color option is clicked, however, for the Theme Editor, it is persistent, instead of disappearing as soon as the cursor moves away from it. It can be recalled from the menu bar, under "Texture > Open Color Theme Editor".

The viewport is dominated by the color picker with its color wheel on the left, the color options on the right, joined by a preset browser across the bottom. Combined with its many options, the color wheel is a full-featured color scheme generator, with options for creating a variety of automatically coordinated schemes based of various rules users can adjust.



The wheel itself is straight forward, users only need to LMB click over the color they wish to choose, while LMB clicking and dragging, users will update the value in real-time, releasing the mouse pointer when the proper color is arrived at will select it. Depending on the color model, the bottom color strip will update accordingly, but users can LM B+click and drag in that window as well. The right hand side of the color picker offers users the ability to define alpha (transparency) value, with the box in the lower right hand corner displaying the current color. The properties panel on the far right side controls the display of the picker, and allows users to define numerical values as well set the scheme rules, and save presets to the browser below.

Color Model: This option allows users to choose the color model that suits their needs best, there are three current options-

HSV-- Standing for 'Hue, Saturation and Value'; it is considered the most intuitive color model, users can adjust colors by their 'Hue', the apparent color itself, saturation, which controls the intensity of the color, and value, which controls how light or dark the color is. This is also the only model that is displayed as the familiar color wheel.

Kelvin-- The Kelvin scale is based on thermal dynamics, and considers colors as *temperatures*. This single axis scale is useful in describing the color of light sources, as the hotter they burn, the brighter, and more white they become. The goes from warm colors, reds and oranges, through yellow to white, and then extending to a slight blue as it increases to the maximum value.

RGB-- Standing for Red, Green and Blue, the model itself is based on the additive property of light, and is the primary model for electronic displays and sensors (such as your monitor and digital camera). If pure versions of each color are combined, they produce white light. While it is a common model, its not very intuitive, as users often don't have a good idea about what will happen if they reduce 'Red' by 50%. Still, it is the most common format for colors when exchanging data between applications.

Axis: Since the color picker is two dimensional, and the various color spaces are multi dimensional, the additional axes options modify how users interact with the picker providing additional options for customizing the picker and color strip to suit the users particular needs.

Stops: This option is related the f-stops, familiar to photographers. It is intended to aid in *previewing* and selecting HDR (High Dynamic Range) colors, whose brightness goes beyond what typical computer monitors are capable of displaying. At the default 1.0 value, the color picker works like any other, maxing out at pure white. Changing the stops value to 1.4 allows users to define colors whose brightness is twice that of the 1.0

setting. A stop of 2.0 would again be twice as bright (or four times brighter than 1.0). These values work the same as those in photography, where each additional *stop* value lets in twice as much light as the previous. The standard photographic progression of stops is 1, 1.4, 2, 2.8, 4, 5.8, 8, 11, 16, & 22. It should be noted that the 'Stops' value only changes how modo displays the colors in the picker and allows users the ability to choose over bright values in the color wheel, and doesn't change the color value itself.

Global Options--

Color Units: The color units option changes the global preference for how color values are entered into modo color fields. There are four options-

Floating Point: For this model, standard color values exist between 0 and 1, with HDR colors values being those beyond 1.0

Percentage: Color values are defined as percentages, similar to Floating point values, where up to 100% would be normal color range, and values beyond 100% extend into the HDR colors.

Integer: This is the standard format for most image editing applications, where standard color values exist in the 0 to 255 range.

Hexadecimal: Hexadecimal colors are the standard for defining color on the web as they are natural values for computers to understand, similar to binary, where the two values represent 255 different levels.

Dynamic Color Strip: When this option is checked, the Color Strip bar will update dynamically showing the the color selection as a component of the bar itself. When unchecked, the bar will simply show its single color axis, as defined by the 'Axis' option.

HSV Options--

Hue as Wheel: When the 'Color Model Options' of 'HSV' is selected, enabling the 'Hue as Wheel' option will display the picker as a continuous wheel. When displayed as such, several options related to making rule based color schemes becomes available.

Wheel: This option defines the type of HSV wheel to display, users can choose the standard 'Red Green Blue' (RGB) option common to many image editing applications, or the more familiar and artist friendly 'Red Yellow Blue' version. Unlike the RGB version, the RYB version works like the color wheel many artists learn in art school, where red & green, blue & orange, and yellow & purple oppose each other on the wheel.

Rule: modo allows multiple options for generating automatic schemes based on common color rules-

Solo-- The default mode, only a single color is defined.

Complementary-- Two colors whose positions oppose each other on the color wheel.

Analogous-- Three colors that contain a similar hue, appearing close to each other on the color wheel.

Triadic-- A three color scheme, where colors appear equidistant from each other on the color wheel.

Tetradic-- A four color scheme, where colors appear equidistant from each other on the color wheel (dual complementary).

Compound-- A five color scheme, with a main color and two split complementary colors.

Tints--Tints appear as graduated value (brightness) variations of the foreground color toward white.

Shades-- Shades appear as graduated value (brightness) variations of the foreground color toward black.

Rule Adjust: Users can adjust the angle of degrees between scheme rules that adjust the scheme to suit their own taste.

Levels: This option becomes active for the 'Tints' and Shades' rules, defining how many steps will be in the scheme.

Save as Presets: Pressing this button will save the current foreground color into the 'Presets > Colors' folder of the preset content. If the default content hasn't been installed, users will need to manually define a directory and add it to the preset browser using the 'Add Path..' option. When defining color schemes using the rule options, all the color defined in the swatch patch will be added to the preset browser simultaneously.

Models--

The five following sections, 'RGB(HDR)', 'RGB', 'Kelvin', 'HSV' and 'Hexadecimal RGB' are simply numeric value input fields for displaying and entering colors using alternate color models. Displayed values will always represent the current foreground color and will update accordingly as the users interact with the picker. All models, except 'RGB (HDR) are not capable of defining or displaying HDR colors, and will clip at the maximum values for the selected model (showing whatever the equivalent color is); the

maximum value will be different for each of the various 'Color Units' options as defined by the user.

TIP: Users can access their system level (OS) color picker at any time by LMB+clicking on the color swatch in the lower right corner of the color picker itself. Users should note that any colors selected in this way will also be affected by the 'Stops' value.

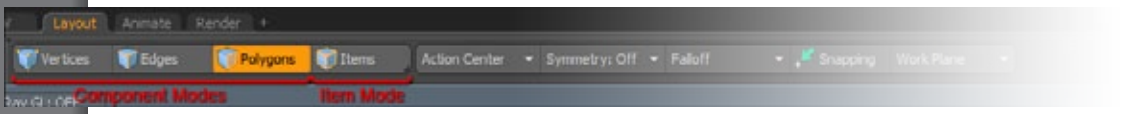
Scene Layout

Once elements are modeled and textured, transforming a series of models into a scene requires some amount of layout, the placement and preparation for final rendering. These steps can also be useful in preparation for animation. modo provides a useful layout interface tab, this mode collects important item transform and duplication tools, and presents a Preset browser to speed up many tasks related to working with the many items within modo.

Items vs. Components

One of the areas that is often confusing to new users of modo are the various selection modes, causing a variety of seemingly baffling issues. It's a very important concept to understand (which is why it is mentioned so often here in these docs). When in their respective selection modes, only selections of that type may be made, keeping in mind that when nothing specific is selected, then everything in a layer is selected (another important concept).

The first three modes 'Vertices', 'Edges' or 'Polygons' are called the component modes. Users can press any of the interface buttons to enter these modes, and press the space bar to toggle through them sequentially. The next mode is called 'Item' or 'Items' mode, users can enter this mode by LMB+clicking the button directly within the interface. It can also be handy for users to know the mode keyboard shortcuts, pressing '1', '2' or '3' respectively enter 'Vertices', 'Edges' or 'Polygons' modes. Pressing '5' will enter 'Items' mode. These aren't the only modes, but they are the most important.



Users can transform their geometry while in *any* of these modes (transform as in *move*, *scale* and *rotate*). When applying these kinds of changes, the big difference between what happens with each lies in the layers center point. When in a component mode, applying any type of transform, users are editing the raw geometry that makes up any model; these are called *component level transforms*. Regardless of the position of the geometry in the mesh item layer, the center position for that item in the properties panel will always remains at the world origin- 0,0,0.

Now when entering 'Items' mode, and applying a transform, users are applying what is called an *item level transform*. Moving the geometry in 3D space, the values in the items properties panel are now changing from something other than 0,0,0, moving them away from the world origin. Users can no longer affect individual vertices (or edges, or polygons) in 'Items' mode, as the layer as a whole is being modified. This is how objects are animated (and positioned for layout) in modo, by applying item level transforms. (For more on the Mesh Items properties panel, please reference that page of the documentation) It can be said that modifications made in the Vertex, Edge and Polygon modes happen *around the layers center point*, while transforms applied while in Item mode move the entire item *including* its center point.

It can become confusing as the same tools for move, scale and rotate are used for transforming items in component *and* item mode, but if you keep an eye out for which mode you are in (usually based on what is being selected), it will soon become second nature. It should be noted that the component modes only apply specifically to mesh item layers, other layers such as lights, texture locators, replicators or cameras can only be edited while in Items mode.

Other Modes

While on the subject, the other selection modes not mentioned above are the 'Center', 'Pivot' and 'Material' modes. While in the prior two types, users can change the position of a 'Center' or 'Pivot' relative to the geometry or item, essentially modifying the layers origin position. Because of Action Centers, the positions of 'Centers' and 'Pivots' aren't all that important until an item needs to be animated, then it becomes very important because the Center represents where the animation transforms originate from. Rotating a tire when the Center is positioned below the tire would not produce the desired results. But when the tire is already in the proper position relative to a car body, its much easier to just transform the 'Center' itself moving it to the proper position at the axle so the wheel can spin smoothly. The 'Animate' section has a number of setup tools that can make it much easier to modify geometry and Centers relative to each other. In order to manipulate a Center or a Pivot, users must first select the item while in the particular mode, by LMB+clicking on the Center or Pivot icons in the 3D viewport and then transforming them. This extra step makes it more difficult to accidentally change their respective positions.

The final selection mode is 'Material' mode. While users can select and transform geometry based on Material assignments (actually tags), it is mostly useful for easily selecting geometry based on particular tag types. Users can change precisely what type of tag the mode specifically selects by adjusting the settings in the "Select > Polygon Tag Type" menu. In Materials mode (the default), the selection function will not only highlight all the geometry in the scene with a specific material tag, but it will highlight the associated Shader Tree layers making it easier to pinpoint a particular layer in a complex tree. In the 'Part' and Smoothing Group' options, either Part tags or Smoothing Group assignments are selected respectively.

modo provides users the flexibility to work in the way that is most comfortable. Take for instance these two examples, the user that wants to Radial Sweep a tire model around the origin and position it in the wheel well, or the user that wants to Radial Sweep the tire in position and adjust the center point after the fact; either way will work. Users should always be aware of an objects Center and Pivot position, so when issues arise, such as when working with Replicators, it will be easy to spot the fix.

Working with Items

modo scenes are basically collections of items- Cameras, Lights, Meshes, Locators, etc. These items are collected and cataloged either in the 'Items List' or in the 'Shader Tree', where each layer represents an entire object unto itself. Users tell modo what particular item they wish to edit through the selection of these items, also referred to as a layer. When selected, their associated attributes appear in an Properties viewport panel where users can manipulate their values directly, or easily manipulate items directly in the 3D viewports.

Creating Items

While modo starts with a number of basic item layers, such as a camera necessary for rendering and an empty mesh layer for building geometry, there are a variety of ways to add additional item layers to the scene. Users will do most of their scene item management from the 'Items List'. In the upper right area of the viewport is an 'Add Item' button users can LMB+click to open the popup menu, holding down the mouse button users can drag over the menu to make a selection, once over appropriate item release the mouse button to select and add to the list. New items are added directly above the current selection. Users may also add items from the menu bar, found under the "Item" menu, there are a variety of commands that add the appropriate item-

Create Camera: This command adds an additional camera item with default settings at the origin of the scene. If an item exists with an identical name, modo will add an appropriate number to the new camera to differentiate it from the existing camera. Cameras are necessary for setting point of view and render settings specific to the frame. The Camera item is covered fully in the 'Shading, Lighting and Rendering' section of the documentation.

Create Light: This menu opens options for adding any of the variety of direct light sources modo supports, including 'Area', 'Cylinder', 'Directional', 'Dome', 'Photometric', 'Point' and 'Spot' light types. If an existing item in the scene has an identical name to the added item, modo will add an appropriate number to the new lights name to differentiate it from the existing light(s).

Create Locator: This command adds a locator item, also sometimes referred to as a null object. Locators are useful as reference objects, and can aide in the creation of hierarchies. Locators do not contribute to the rendered image but do have a 3D viewport representation useful for applying transforms. Locators are covered in the Items reference page of the documentation.

Create Texture Locator: This command adds a Texture Locator item. Users generally

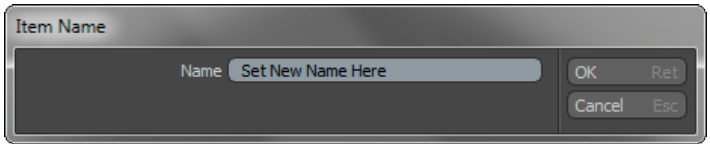
wouldn't add a texture locator manually, as they are added automatically whenever an Image Map or Procedural texture layer is added. Texture Locators define how textures are applied to surfaces at render time. The Texture Locator is covered in the 'Shading, Lighting and Rendering' section of the documentation.

Create Backdrop: This command creates a Backdrop item. Backdrop items allow users to display images directly in the 3D viewports, useful as modeling reference. These image however do not contribute in final rendered images. Backdrop items are covered in the 'The modo Toolbox' section of the documentation.

Create Environment: This command will add an additional Environment item to the Shader Tree. Environments represent the background areas of the rendered scene. The Environment item is covered in the 'Shading, Lighting and Rendering' section of the documentation.

Primitive: This menu allows users to choose from and add a variety of unit primitives, basic shaped geometry scaled approximately to 1 cubic meter in size. These *primitive* objects can act as stand-in proxy geometry, the basis for more complex objects, or as final geometry requiring only simple forms. The primitive object tools that the geometry is based upon is covered on the 'Creating Geometry' page of the documentation. Users can also add unit primitives by holding the 'Ctrl' kay and LMB+clicking on the appropriate tool in the modeling toolbox. Pressing 'Shift' will add the unit primitives as a new mesh item layer.

Renaming Items



Users can rename items by first selecting the target layer, then by RMB+clicking directly on the item opening a context menu and selecting the 'Rename' option form the menu. This opens a popup dialog box where users enter the actual new name and apply it when LMB+clicking 'OK'. Users can also inline edit the name by LMB+click pause then click again to change the field to an edit field. Once changed users can LMB+click on the edit field to set cursor focus and then use the keyboard to enter the new name. User can also select the target item in either the 'Items List' of the 'Shader Tree' and select the command "Item > Rename" from the menu bar.

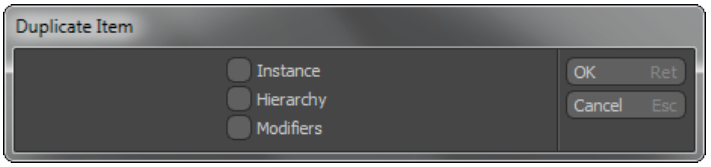
Duplicating Items

Users can clone items in a variety of ways, depending on the desired outcome (and number of clones). The easiest route is the direct duplicate. Users can first select the target layer and then RMB+click directly on the layer and choose the 'Duplicate' command from context menu, making a copy of the item with a number value appended to it, differentiating it from its source. Users can also select the target item and press the 'Ctrl+D' keyboard shortcut to quickly duplicate items. The duplicate item will be an identical copy of the source item, inheriting all its attributes. Users can then modify and/or transform the duplicate to make it a unique object.

When a greater number of duplicates is desired, users can use any of the Clone and Array commands found in the 'Duplicate' tab of the modo toolbox. These same options are also available under the "Geometry > Duplicate" and are covered in the 'Duplicating Geometry' page of the documentation. Replicators are also an excellent way to generate many duplicates of an item providing a procedural means to clone objects over a surfacing by specifying another item that acts as a point source. The application and use of Replicators is covered on its respective page of the documentation.

modo also provides users the ability to create *Instances* of items. Instances are virtual duplicates that require no additional memory as they only are references back to their source, and update automatically when any component edits are applied to the source geometry. Instances of items can be positioned, scaled and rotated as it they were their own item, however users cannot make direct component level edits to Instances. Many of the Clone and Array tools provide options for users to generate Instances instead of geometric clones, greatly reducing scene overhead.

Duplicate Options: The duplicate options command provides users with a few options for the duplicated items. By first selecting the target items, users can access the command from the RMB+click context menu as before and choosing the 'Duplicate...' option. Users can also find the command in the menu bar under "Item > Duplicate Options"-



Instance: When this option is *enabled*, modo will generate instances instead of duplicates.

Hierarchy: When this option is *enabled*, all child items of the source item will be duplicated as well, retaining their hierarchy.

Modifiers: When this option is *enabled*, any related modifiers associated with the source will be duplicated as well.

Converting Items

Users can convert any type of item into any other type of item using the 'Change Type' command. Found in the menu bar under "Item > Change Type" and then selecting from the various item options, this command will convert whatever is currently selected into the type of item selected. Users can also select a layer and RMB+click opening the context menu and selecting the "Change Type" option opens another menu from which to chose the item type you wish to convert to. The standard transform properties of the item (position, scale and rotation) will be retained, but in unrelated items, the converted object will be based on default item values.

Changing a mesh item into a light can be useful from time to time, but the more important function lies in converting heavy mesh items into a 'Static Mesh'. Since a converted mesh item will be frozen and triangulated it is not always the most desirable option, but in cases where items are simply to large to manage otherwise, it is a good option to have available. For more on Static Meshes, please reference that page of the documentation.

Removing Items

Users can remove an item at any time by selecting the target items in the 'Item List' or 'Shader Tree' and RMB+clicking directly on the target layer to open the contextual menu and selecting the 'Delete' command. With the target items selected, users may also invoke the menu bar command "Edit > Delete" to remove item layers. Finally, users can remove selected items by simply pressing the 'Delete' key on the keyboard.

Items Reference

modo scenes are basically collections of items- Cameras, Lights, Meshes, Locators, etc. Users tell modo what particular item they wish to edit through the selection of these items, also referred to as layers. When selected, their associated attributes appear in an Properties viewport panel where users can manipulate their values directly. For information on a specific Item, click on any of the link options below to be directed to the proper page. For more information on the various 'Light' items, please reference that page of the documentation in the 'Shading and Rendering' section.

Item Reference	Deformers		Duplication
Scene Item	Bend Effector	Morph Influence	Replicator Item
Camera Item	Lag Effector	Vortex Effector	
Mesh Item	Magnet Effector	Weight Container	

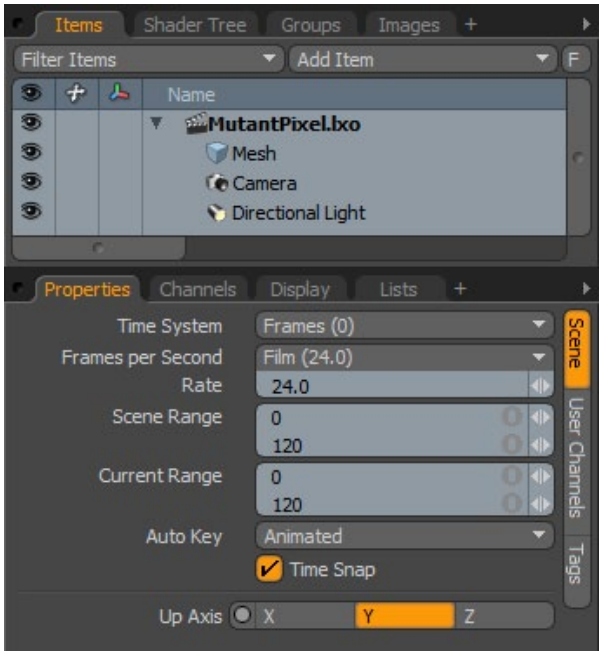
Falloffs	Locators	Particles	Procedural
Capsule Falloff	Locator	Particle Cloud	Fractal Gasket
Linear Falloff	Group Locator	Particle Generator	Gear
Radial Falloff	Texture Locator	Realflow Particles	RPC Mesh

Volumes		Other	Misc
Blob	Sprite	Spline Falloff	Static Mesh
Ground Plane	Volume		Instance Item
Render Boolean			Backdrop

Scene Item

The Scene Item is the top most layer of any file within modo, as such, all items within the scene will be children of the scene itself. A scene item will always be named by the file it represents. To change the name of a scene, users would need to use the 'Save As'

command to assign a new file name to the scene. There are a number of attributes that are associated to scenes, users can select the scene item in the 'Items List' (although it will not highlight as normal item layers do) and access them in the Properties viewport-



Time System: Allows the user to specify which standard is used to display individual frame numbers

Seconds- Indicates frames as a passage of time, using the "0.0s" format.

Frames- Indicates frames as sequential numbers, using standard numbering format.

SMPTE- SMPTE time code specifies frames utilizing a shorthand method "00:00:00:00"; representing hours:minutes:seconds:frames

Film Code- Specifies frames by the length of the film in feet (linear measurement) using "0.0 + 0" format.

Frames per Second: Users can specify a frame rate by standard definitions, such as Film (24 fps), PAL (25 FPS), and NTSC (29.97 fps). Frames per Second, usually referred to as FPS, is the number of still images that playback in sequence in one second of time.

Rate: The actual frame rate value is displayed here, change this value for specific nonstandard frame rates.

Scene Range: This value specifies the total length for animated scenes.

Current Range: When animating, users may wish to concentrate on a single area of a scene for fine tuning, the current range setting allows the users to essentially temporarily crop out the rest of the scene so they can concentrate on the one section.

Auto Key: When items are moved in a scene, the auto key function can automatically create keyframes eliminating the need to manually create them. The different settings specify how keyframes are created, 'All' will generate keyframe data for all channels associated with the modified item, 'Animated' will only keyframe those values that have

been edited or modified and 'None' disabled the auto keyframe functionality.

Time Snap: The time snap toggle will 'snap' the current time indicator as well as keyframes to whole frame values.

Up Axis: By default, the 'Y' axis represents *up* in modo's virtual world. The up axis indicator allows users to specify alternative axis.

Note: These setting can also be modified in the timeline options dialogue box.

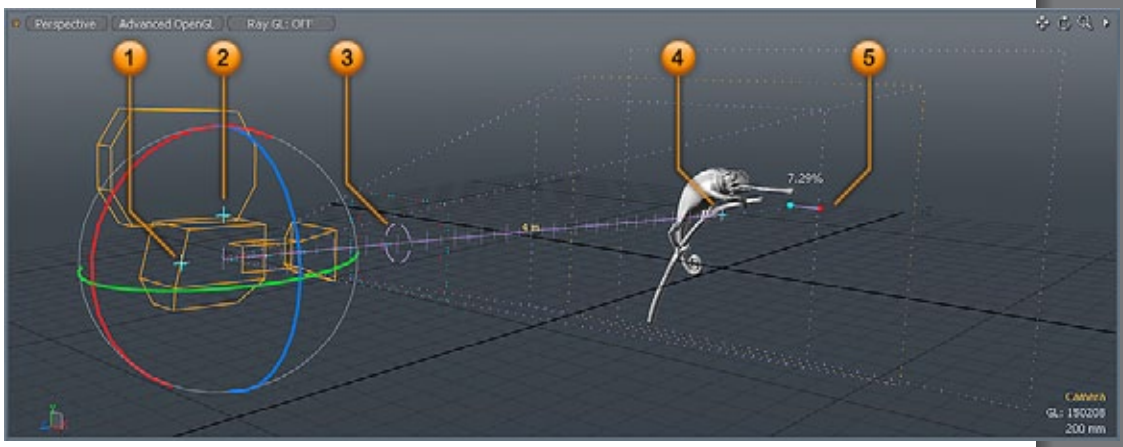
Camera Item

The Camera Item is a virtual lens into the 3D world inside modo. Everything rendered is seen from this point of view, as such, these settings determine *how* your camera sees your 3D world. Much like how a real camera operates, the camera *captures* your image, and works in conjunction with the 'Render Item' making your virtual snapshot viewable in the 'Render Display'. What is displayed depends on your 'Render Output(s)', which acts as virtual *developer* determining what actually gets rendered when 'F9' is pressed (or any render command is invoked). While the Camera Item contains the properties related to the projection, frame aspect ratio, DOF, etc., the actual frame resolution is set within the 'Render Item'.

Each new modo scene by default has a single camera item, additional cameras may be added to the scene with the menu bar command 'Item > Create Camera' or within the 'Item List' viewports 'Add Item' function; LMB-click the add item button and select 'Camera' from the popup menu. Additional cameras are helpful in providing alternate views of a scene, easing navigation, or for providing close-up views of a particular area, especially in situations where the *hero* camera is animated and locked, avoiding accidental changes. Only a single camera may be 'rendered' from at a time, the particular camera used is specified in the 'Render Item's 'Frame' subtab under the 'Render Camera' option. Additionally, alternate camera may be selected within the 'Render Preview' viewport using the select camera option available, without affecting the 'Render Camera' used for *final rendering*.

Working with the Camera

The cameras attributes, such as Position and Rotation, can be modified in the properties panel whenever the camera item is selected in either the "Item List' or the 'Shader Tree'. By default, the camera icon is only visible in the 'Animate' and 'Render' interface tabs, but by changing the visibility in the viewport options, they can be made viewable in any viewport; hover the mouse pointer over the 3D viewport you wish to modify, and press 'o' on the keyboard to open the visibility options panel.



The camera item may also be interactively modified in the 3D viewport while in 'Items' mode (camera items have no editable components). Use the standard transform tools to move (W) and rotate (E) the camera. When making fine adjustments, users may find it easier to see how the camera is viewing the scene. This can be done by changing the viewport view button in the upper left to 'Camera'. When set as such, black bars may be drawn over the viewport if the *aspect ratio* of the 'Frame Size' is different from the viewports aspect ratio; the area inside the bars represents the area of the scene that will render when a render command is invoked. To position the camera while in this view mode, various keyboard/mouse options are available--

Shift+Alt+LMB Drag	=	Move camera item
Shift+Alt+RMB Drag	=	Move Up/Down direction only
Alt+LMB Drag	=	Rotate around Target Distance
Alt+RMB Drag	=	Rotate around Camera center
Ctrl+MMB Drag	=	Rotate around 'Z' axis
Ctrl+Alt+LMB Drag	=	Move Forward/Backward only
Ctrl+Alt+RMB Drag	=	Adjust Focal Length
Roll Mouse Wheel	=	Move Forward/Backward only

When tool handles are set to 'Advanced' (within the 'Preferences' or 'Tool Pipe'), additional handles become visible allowing users to interactively adjust camera setting in the viewport (mouse over the above image for handle reference). The handles are--

1.) Adjust 'Focal Length'/'Field of View'.

2.) Adjust F-Stop for Depth of Field.

3.) Adjust 'Convergence Distance'.

4.) Adjust 'Focus Distance'.

5.) Parallax Offset Difference value (distance between items for left & right eye in final rendered frame) expressed as percentage of frame width. Users can drag the handle around the scene to get an idea how far apart elements will appear on the screen when rendered as a stereo image. Generally speaking, for comfortable viewing users will want to keep the sum of the values for the closest elements in frame to the farthest elements 5% and under. Adjustments can be made to 'Interoccular Distance' and the 'Convergence Distance' options to help in obtaining this result.

TIP: User will likely find it more intuitive to adjust the camera when the 'Action Center' is set to 'Local', keeping the transform handles aligned to the camera item itself.

On occasion, users may wish to modify the size of the camera icon within the 3D viewport, this can be accomplished with the 'Size' setting in the 'Display' viewport, which offers additional settings for modifying the cameras iconic representation including the ability to label the item. Users may also adjust the 'Scale' option of the cameras in the 'Properties' panel. Note that changing the size of the camera icon in no way affects the rendering outcome or the overall scale of the scene.

The attributes of the Camera item found in the properties panel are documented on the Camera Item page of the Shading Lighting and Rendering section of the documentation, please reference that page for additional information.

Mesh Item

Mesh Items can also be considered geometry *layers* as this is how they appear in the Item List. Mesh Items contain vertices, edges and polygons and are usually the hero of your project. These are your models. As with any Item, Meshes have Item level transforms for Position, Rotation and Scale. Unlike any other Item, the Mesh item also has transforms at the component level which is how you typically do modeling operations. The difference between component edits and Item level edits is very important.

Component transforms are simply editing vertices, edges and polygon information on the mesh. If nothing is selected, modo considers everything to be selected so that you can easily activate a tool such as Move and drag around an entire mesh without selecting all the polygons or vertices first. When you edit the mesh this way you are adding vertex map data to every vertex in the mesh. Think of Item based transforms

as editing the layer that contains the mesh data itself rather than the geometry within the layer. This is why Item transforms are limited to rigid moves, rotates and scales and do not allow any of the deformation available at the component level. Item level transforms are also wicked fast as the edit is handled at a higher level. One easy way to know if you are performing an Item or component level edit is to check your selection mode. If you are in Vertex, Edge or Polygon mode you are performing component edits. Only when you are in Item mode can you perform an Item level transform.

If you find that you have performed Item level transforms when you intended to do component transforms, you can use the Freeze transform option found on the Items Property form. This handy command makes it useful to perform Item level transforms for performance on heavy meshes and then simply push the transform information into the vertices by Freezing after the edit.

Each Mesh item layer can be considered its own *object* within a scene, with its own transforms, subdivision settings and its own UV Maps. Users can add additional Mesh items to a scene by pressing the 'Add Item' button in the upper right corner of the Items list and select the 'Mesh' option, additionally users can use the menu bar command "Geometry > New Mesh Item". Finally, users can also easily create an empty mesh layer by simply pressing the 'n' keyboard shortcut while in the modeling interface tabs. Once selected a number of attributes become available for editing within the Properties viewport-

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. By default, Position transforms originates from the Center position.

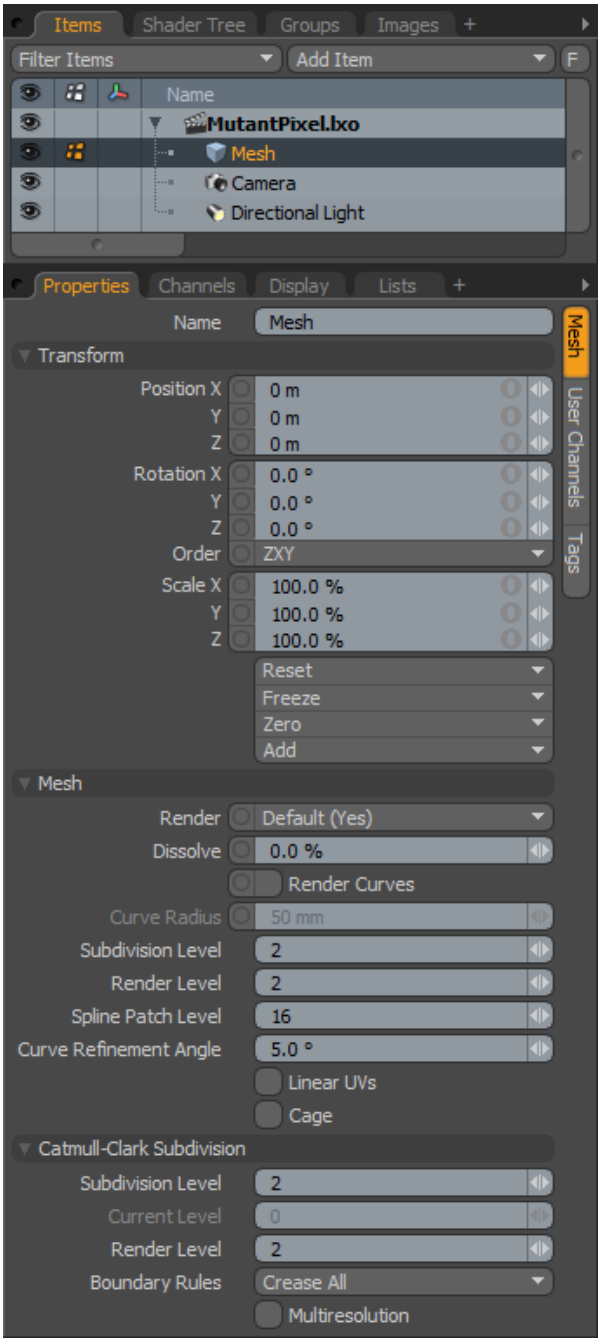
Rotation: An Item transform that allows the user to numerically set the rotation of the item. By default, Rotation transforms originate from the Center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. By default, Scale transforms originate from the Center position.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Freeze: Returns the items 'Center' position to the world space center (0,0,0) without changing the position of the mesh item itself.



Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Mesh position intact. This is done by adding a negative transform item to the Mesh items channels.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Mesh--

Render: This drop down menu allows the user to select from 3 choices, when set to 'Default', the user can enable/disable mesh item using the visibility column (eyeball icon) of the item list. When the mesh layer is visible, it contributes to the final rendered scene and when invisible, it will not. On some instances the user may prefer to fix this state, setting the mesh as 'On' (enabled) or 'Off' (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving the user from manually enabling mesh layers for test renders.

Dissolve: The 'Dissolve' function controls the visibility of an item layer, at 0% the item renders normally, utilizing the shader tree settings, values above 0% would fade the visibility of the item layer ramping toward 100% where the item would be completely invisible.

Render Curves: Generally, curves are non-rendering items in a layer meant to facilitate other functions -- hair guides, patching guides, duplication paths and so forth. However, with the 'Curve Radius' set to a value above 0, and the 'Render Curves' function enabled the previously invisible curves renderable. Material tags for surfacing can be applied to curves in the same way they are for regular polygons (by selecting the curve in 'Polygons' mode and assigning a 'Material' tag). Rendered curves essentially become tubes that have several benefits over extruded geometry, they reduce the amount of geometry in the scene since render curves are generated at render time making for a lighter file size and their size is editable *and* animateable as well. Additionally morph deformations applied to a curve will keep a consistent thickness. Note that because of their procedural nature, they cannot be displaced. Use the 'Curve Extrude' function to make an actual tube when using displacement.

Curve Radius: When 'Render Curves' is *enabled*, the 'Curve Radius' determines the size of the 'tube' that is generated. The radius is centered around the curve itself, so a setting of 12mm would generate a tube with a 24mm diameter.

Subdivision Level: Any selected geometry within a layer can be converted to a subdivision surface by utilizing the 'Tab' key shortcut. This produces a smoothing effect to the surface, essentially becoming a proxy for the smoothed SDS geometry which cannot be directly edited. The 'Subdivision Level' setting specifies to what degree a SDS model is divided. There are certain functions that override this setting at reder time, such as those for 'Adaptive Subdivision'. Otherwise, when using the Freeze command, the 'Subdivision Level' controls the number of polygons generated as does it control the number of polygons displayed in any 3D viewport. This value can be interactively adjusted with the +/- keys on the number pad when a Mesh item is selected.

Render Level: When 'Adaptive Subdivision' is disabled, users can set a independant 'Render Level' for subdividing their geometry only when any of the 'Render' commands are invoked. This can be helpful in increasing OpenGL viewport performance by setting a low Subdivision Level and a high Render Level, keeping a finely subdivided smooth surface only when the scene is rendered.

Spline Patch Level: Spline patching is a way of parametrically defining a surface with a number of curves. When Patching the curves to define the surface, the 'Spline Patch Level' controls the number of polygons used. These polygons are not directly editable unless one applies the 'Freeze' command found under the menu bar "Geometry > Freeze..."

Curve Refinement Angle: Curve Refinement Angle controls how curves are quantized for display and freezing. This is analogous to the patch resolution for subdivision surfaces.

Linear UVs: modo automatically switches between linear UVs for straight polygonal geometry and smoothed UVs for subdivision surface (SDS) geometry. There may be some instances where an image map created outside of modo utilized linear for an SDS model, in which case enabling 'Linear UVs' will correctly apply the image map to the geometry within modo.

Cage: The 'Cage' option toggles any Subdivision Surface geometry in the current mesh layer. Especially useful in temporarily disabling SDS in complex models without losing the SDS definitions for the geometry.

Catmull-Clark Subdivision--

Subdivision Level: Users may convert any surface into a Catmull-Clark Subdivision Surface by pressing the 'Shift+Tab' key combination. The Catmull-Clark Subdivision Surfaces are superior to regular Subdivision Surface geometry, especially in their handling of edge creasing, however they are more computationally intensive than regular Subdivision Surfaces geometry and require more memory to render. The Subdivision Level controls how finely a surface is subdivided.

Current Level: When the 'Multiresolution' option is enabled, users can use the 'Current Level' option to step up and down through the different levels of subdivisions (up to the maximum level defined by the 'Subdivision Level'). Useful when sculpting providing the ability to add different degrees of detail to a model at different levels.

Render Level: When 'Adaptive Subdivision' is disabled, users can set a independant 'Render Level' for subdividing their geometry only when any of the 'Render' commands are invoked. This can be helpful in increasing OpenGL viewport performance by setting a low Subdivision Level and a high Render Level, keeping a finely subdivided smooth surface only when the scene is rendered.

Boundary Rules: The boundary rules determine how to move the points on the boundaries of a Catmull-Clark Subdivison Surface mesh. Creasing will produce a sharper transition between surfaces, smoothing, quite obviously, produces smooth surface transitions. The options are-

Smooth All- All of positions on the boundary are moved by the *smooth rule* of Pixar Subdivision Surfaces.

Crease All- The points that have two incident edges (valence = 2) are moved by the *crease rule*, other positions on the boundary are smoothed by two boundary edges.

Crease Edges- The points that have two incident edges (valence = 2) are moved by the *smooth rule*, other positions on the boundary are smoothed by two boundary edges.

Multiresolution: Only available to a Pixar Subdivision Surfaces model, users can enable this option step up and down through the subdivision levels when sculpting a mesh, allowing users the ability to add varying degrees of detail without obliterating detain present in other levels. Users can adjust the current level using the 'Current Level' option above, or step through the levels using the Multi-res settings in the sculpting toolbox of the 'Paint' interface tab.

Bend Effector

Effectors are a means to affect a target item, specifically a Mesh layer, deforming it in specific ways. The 'Bend Effector' when assigned to an item will deform the object in a curving/arcing fashion. As the 'Angle' value is increased, the more pronounced the bending will be. The 'Bend Effectors' position relative to the mesh item it is deforming should be considered, as it's position provides the origin location of the Effectors influence.

Applying an Effector

While simply adding an Effectors with the 'Add Items' option is possible, placing the appropriate Effector in the Items list, it won't produce the expected results, as the Effector, being a separate item, need to be connected to the Mesh layer by way of a 'General Influence'. The General Influence acts upon the mesh, while the Effector itself produces the instructions on how the Influence should modify its target. This connection can be made automatically (*while in 'Items' mode*) by first selecting the target item(s) either in the Items list or any 3D viewport, and then RMB+clicking on the item, opening the contextual menu and selecting the menu option "Add Deformer > Effectors > Choose Type". Additionally, users can select the target item(s) and press the Effector button in 'Deformers' toolbox subtab of the 'Layout' and 'Animate' interfaces

and produce the same result. Once the Effector is created, users can position the Effector and modify its values to produce the deformed results. When applied to multiple items, each item will be deformed by the single Effector based on its relative position to the mesh.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. These values defines the origin of influence for the Effector.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. These values defines the direction of influence for the Effector.

Order: Allows the user to set the order that rotations are applied to the Effector item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item representation in the 3D viewport. To adjust the scale of the arc's influence, disable the 'Use Length' option and scale the 'Radius' value.

Deformation Mode:

Reset: Resets the selected transform values to (0,0,0) returning the item back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Mesh position intact. This is done by adding a negative transform item to the Mesh items channels.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Bend Effector--

Enable: Toggles the effect of the layer on or off. When un-checked (*disabled*), the layer has no influence on the scene. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Axis: Determines the general directional Axis for the Effectors deformation, without the need to directly rotate the Effector item.

Angle: The 'Angle' value determines the amount of Bending deformation applied the target mesh as an arc. For longer or shorter bend lengths, change the Effectors position relative to its target, or disable the 'Use Length' option and adjust the 'Radius' value.

Use Length: When 'Use Length' is *enabled*, the Radius value is determined automatically by the Length option value.

Length: The 'Length' value determines the maximum range of influence along the chosen 'Axis', where values will attenuate along the curve, but those beyond the defined 'Length', mesh will be affected at 100% strength.

Radius: The 'Radius' value determines the size of the bending arc, and consequently its scale of influence.

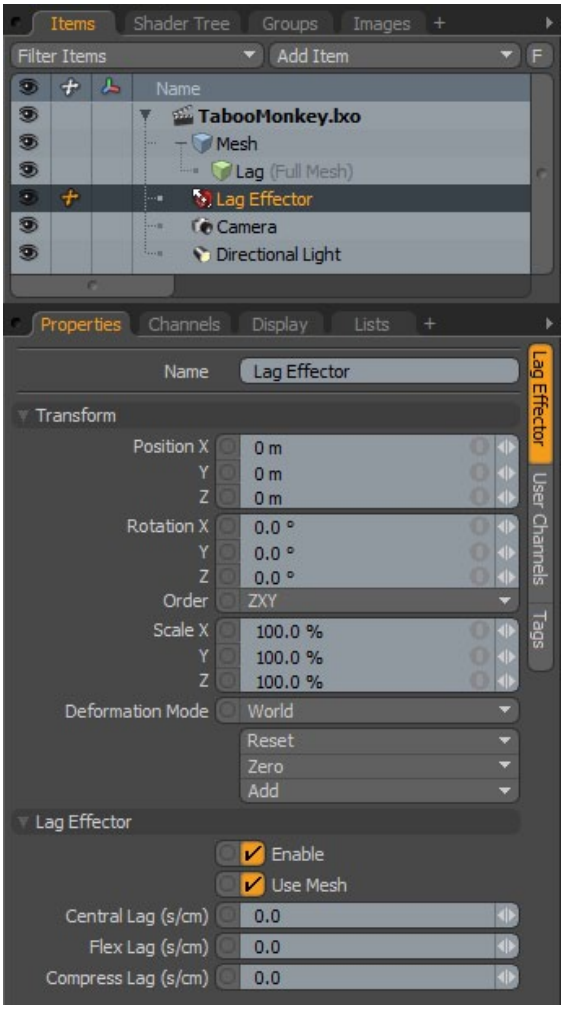
Lag Effector

Effectors are a means to affect a target item, specifically a Mesh layer, deforming it in specific ways. The 'Lag Effector' when assigned to an item will deform the object along a directional vector, determined by the target items animation (essentially it will be deformed along the direction it is traveling). If the target item isn't animated, no effect will be visible. As the Lag values are increased, the greater the deformation will be.

Applying an Effector

While simply adding an Effectors with the 'Add Items' option is possible, placing the appropriate Effector in the Items list, it won't produce the expected results, as the Effector, being a separate item, need to be connected to the Mesh layer by way of a

'General Influence'. The General Influence acts upon the mesh, while the Effector itself produces the instructions on how the Influence should modify its target. This connection can be made automatically (*while in 'Items' mode*) by first selecting the target item(s) either in the Items list or any 3D viewport, and then RMB+clicking on the item, opening the contextual menu and selecting the menu option "Add Deformer > Effectors > Choose Type". Additionally, users can select the target item(s) and press the Effector button in 'Deformers' toolbox subtab of the 'Layout' and 'Animate' interfaces and produce the same result. Once the Effector is created, users can position the Effector and modify its values to produce the deformed results. When applied to multiple items, each item will be deformed by the single Effector based on its relative position to the mesh.



Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. These values defines the origin of influence for the Effector.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. These values defines the direction of influence for the Effector.

Order: Allows the user to set the order that rotations are applied to the Effector item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item representation in the 3D viewport. To adjust the scale of the arc's influence, disable the 'Use Length' option and scale the 'Radius' value.

Deformation Mode:

Reset: Resets the selected transform values to (0,0,0) returning the item back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Mesh position intact. This is done by adding a negative transform item to the Mesh items channels.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Lag Effector--

Enable: Toggles the effect of the layer on or off. When un-checked (*disabled*), the layer has no influence on the scene. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Use Mesh: When *enabled*, this option sets the Lag effector to use the center of the mesh it is deforming, instead of the position of the Lag effector itself as the origin. It is not much different than parenting the effector to the mesh, but it is much easier.

Central Lag: This is the amount of time by which a point 1cm from the center will lag behind the center, so it is the amount of time lag per cm distance from the center. The distance is the absolute distance from the point to the origin location, be it Leg effector position or the Mesh's Center.

Flex Lag: Similar to the 'Central Lag', this is also the Lag amount per cm, but the distance is only the distance perpendicular to the line of motion (vector).

Compress Lag: This is the lag amount per cm, but the distance is measured only along the line of motion, giving a *stretchiness* just along the motion. The 'Flex' and 'Compress' lag amounts are components of the central lag, so by combining them in different levels, you can get good control over how motion deforms the mesh.

Magnet Effector

Effectors are a means to affect a target item, specifically a Mesh layer, deforming it in specific ways. The 'Lag Effector' when assigned to an item will deform the object along a directional vector, determined by the target items animation (eseentially it will be deformed along the direction it is traveling). If the target item isn't animated, no effect will be visible. As the Lag values are increased, the greater the deformation will be.

Applying an Effector

While simply adding an Effectors with the 'Add Items' option is possible, placing the appropriate Effector in the Items list, it won't produce the expected results, as the Effector, being a separate item, need to be connected to the Mesh layer by way of a 'General Influence'. The General Influence acts upon the mesh, while the Effector itself produces the instructions on how the Influence should modify its target. This connection can be made automatically (*while in 'Items' mode*) by first selecting the target item(s) either in the Items list or any 3D viewport, and then RMB+clicking on the item, opening the contextual menu and selecting the menu option "Add Deformer > Effectors > Choose Type". Additionally, users can select the target item(s) and press the Effector button in 'Deformers' toolbox subtab of the 'Layout' and 'Animate' interfaces and produce the same result. Once the Effector is created, users can position the Effector and modify its values to produce the deformed results. When applied to multiple items, each item will be deformed by the single Effector based on its relative position to the mesh.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. These values defines the origin of influence for the Effector.

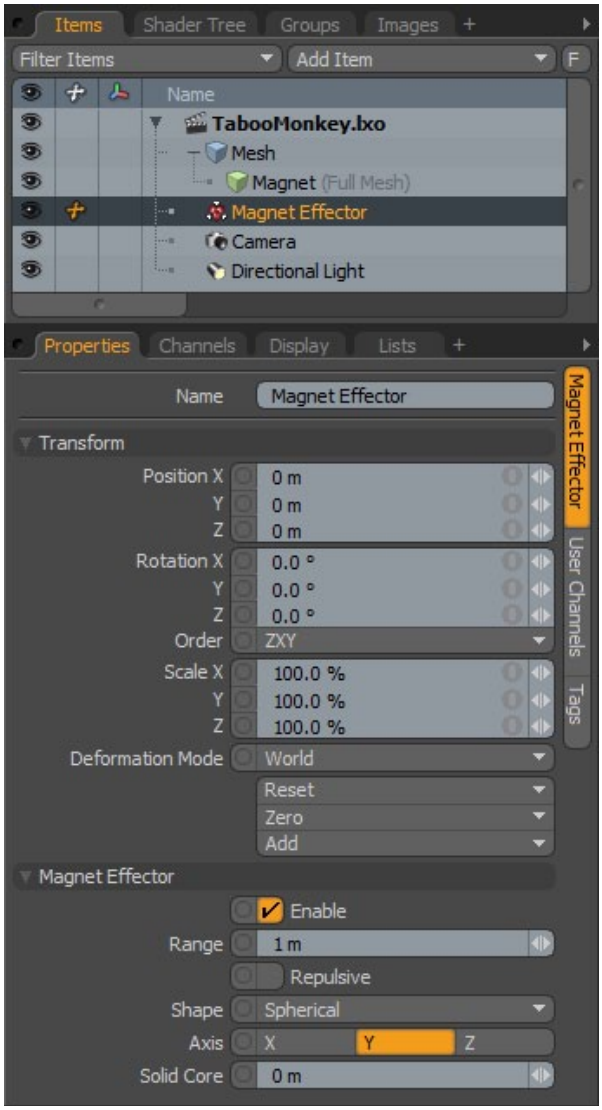
Rotation: An Item transform that allows the user to numerically set the rotation of the item. These values defines the direction of influence for the Effector.

Order: Allows the user to set the order that rotations are applied to the Effector item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item representation in the 3D viewport. To adjust the scale of the arc's influence, disable the 'Use Length' option and scale the 'Radius' value.

Deformation Mode:

Reset: Resets the selected transform values to (0,0,0) returning the item back to their default state.



Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Mesh position intact. This is done by adding a negative transform item to the Mesh items channels.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Magnet Effector--

Enable: Toggles the effect of the layer on or off. When unchecked (disabled), the layer has no influence on the scene. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Range: This value determines the range of influence, value within the range will be deformed, those outside the range will not.

Repulsive: The Magnet Effector pulls points (attracts) toward the locator position, when 'Repulsive' is enabled, points will be pushed from the locator position.

Shape: This is the shape of the effector's field, it determines how points are moved to or away from the effector.

Planar- Points are attracted linearly to a directional axis based on the 'Range' value.

Sperical- Points within the 'Range value are attracted to the center of a round area of influence.

Cylindrical- Points within the 'Range' value are attracted to the centerline column of a 'Cylinder' chape.

Axis: Determines the axis direction for the 'Shape' option. Only really affects the 'Planar' and 'Cylindrical' options (as Sperical produces the same results on any axis).

Solid Core: This specifies the distance from the center of the magnet that will be considered solid, thus causing any points inside that range to be moved to the surface of the shape.

Morph Influence

Morph maps offer users a means of saving alternate user-defined deformations for any mesh. Individual morph map deformations can be animated over time providing the ability to animate say for instance facial expression. In order to animate any morph map, a morph deformer must first be applied to the model.

Applying a Morph

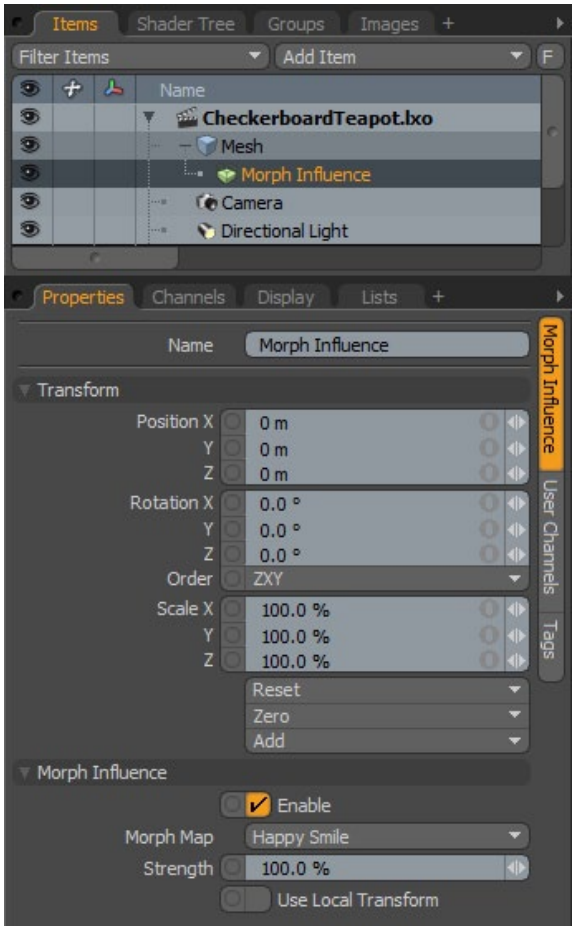
Before a morph can be applied, it needs to be created. New morph maps are added using the 'Vertex Map List' viewport. Any modifications made to the map while *selected* will be stored in that Morph Map. For more information on creating Morph Maps, please reference the 'Working with Vertex Maps' page of the documentation under the Morphs section.

To apply a morph to a mesh, for the purposes of animation, users can select the target Mesh item in the Items list and RMB+click on the item name to open the contextual menu. In the menu, select "Add Deformer > Influences > Morph Influence", this automatically add the Morph Influence item, connecting it to the mesh. Users can also select the 'Morph' option from the 'Animate' and 'Setup' toolboxes, under the 'Deformers' subtab. Once applied, users can access the attributes by selecting the Morph item, and viewing them within the Properties panel. Each Morph Influence will support a single morph map, so for multiple maps, additional Influences will need to be added. Combined morph maps are additive, so all will blend together appropriately providing predictable results. Additionally, the morph 'Strength' value can be driven beyond 100% (or into negative values) producing interesting results.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

(Transform values will change the position of the Influence items locator, but has no affect over the Morph itself)



Position: An Item transform that allows the user to numerically position the influence item in XYZ space.

Rotation: An Item transform that allows the user to numerically set the rotation of the influence item.

Order: Allows the user to set the order that rotations are applied to the Influence item.

Scale: An Item transform that allows the user to numerically set the size of the influence item.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Item position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Morph Influence--

Enable: Toggles the effect of the layer on or off. When unchecked (*disabled*), the layer has no influence on the scene. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Morph Map: This option is used to select the target Morph vertex map.

Strength: This control allows users to modulate the strength of the Morph. At 100%,

mesh will be deformed fully at the morph state.

Use Local Transforms: Adjusts the offsets of the Morph so that they deform in the local coordinates. This is useful in creating corrective Morphs when coupled with other deformers.

Vortex Effector

Effectors are a means to affect a target item, specifically a Mesh layer, deforming it in specific ways. The 'Lag Effector' when assigned to an item will deform the object along a directional vector, determined by the target items animation (eseentially it will be deformed along the direction it is traveling). If the target item isn't animated, no effect will be visible. As the Lag values are increased, the greater the deformation will be.

Applying an Effector

While simply adding an Effectors with the 'Add Items' option is possible, placing the appropriate Effector in the Items list, it won't produce the expected results, as the Effector, being a separate item, need to be connected to the Mesh layer by way of a 'General Influence'. The General Influence acts upon the mesh, while the Effector itself produces the instructions on how the Influence should modify its target. This connection can be made automatically (*while in 'Items' mode*) by first selecting the target item(s) either in the Items list or any 3D viewport, and then RMB+clicking on the item, opening the contextual menu and selecting the menu option "Add Deformer > Effectors > Choose Type". Additionally, users can select the target item(s) and press the Effector button in 'Deformers' toolbox subtab of the 'Layout' and 'Animate' interfaces and produce the same result. Once the Effector is created, users can position the Effector and modify its values to produce the deformed results. When applied to multiple items, each item will be deformed by the single Effector based on its relative position to the mesh.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

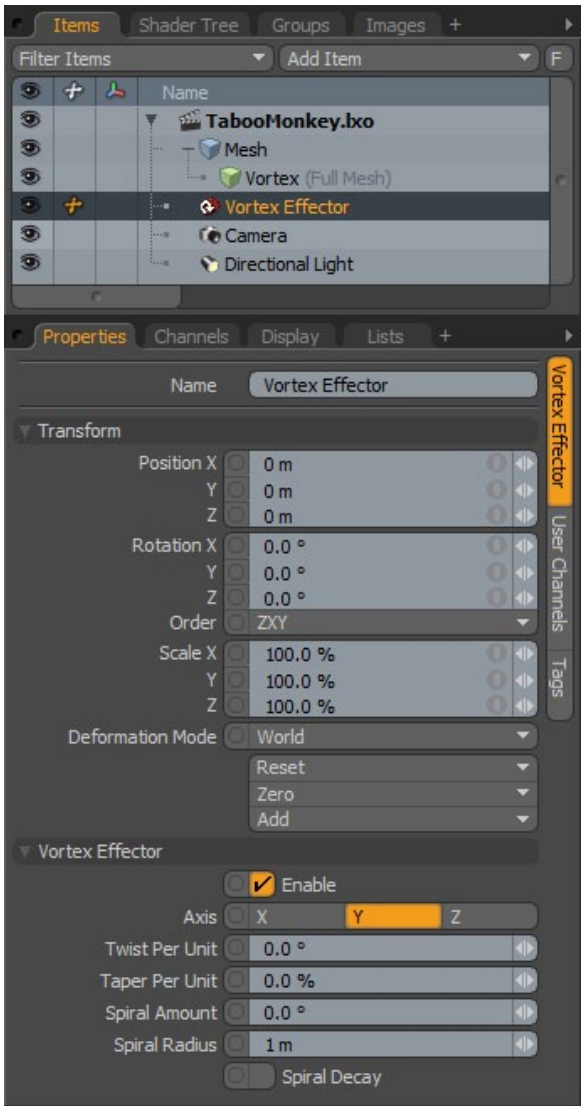
Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. These values defines the origin of influence for the Effector.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. These values defines the direction of influence for the Effector.

Order: Allows the user to set the order that rotations are applied to the Effector item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item representation in the 3D viewport. To adjust the scale of the arc's influence, disable the 'Use Length' option and scale the 'Radius' value.



Deformation Mode:

Reset: Resets the selected transform values to (0,0,0) returning the item back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Mesh position intact. This is done by adding a negative transform item to the Mesh items channels.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Vortex Effector--

Enable: Toggles the effect of the layer on or off. When unchecked (*disabled*), the layer has no influence on the scene. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Axis: This option determines the axis direction for the twisting motion of the Vortex.

Twist Per Unit: The amount of rotation added for every unit of distance from the base of the Vortex.

Taper Per Unit: The change in the radius for every unit of distance from the base of the Vortex.

Spiral Amount: This is the rotation amount added as we move away from the center axis of the cylinder. The angle will exactly equal the 'Spiral Amount' when the distance equals the 'Spiral Radius'.

Spiral Radius: This is the distance from the axis at which the spiral rotation added is equal to the Spiral Amount.

Spiral Decay: This toggle sets whether the spiral increases or decreases with distance along the axis.

Weight Container

Vertex Weight maps can be used to modulate a variety of function within modo. When used for animated deformations (essential for character animation), each joint is assigned a specific Weight Map that defines its area of influence upon the mesh. While being incredibly useful and tremendously powerful, the drawback of a traditional embedded Weight Map is exactly that, its part of the mesh and when the mesh isn't available, then the weight map isn't either. The Weight Container is exactly the same thing as a traditional Weight Map, storing weight values for vertices, but its been separated from the mesh, becoming an item of its own. This is useful when rigging, where the weight values can be abstracted from the geometry. This way a rig can be created and attached to the geometry, users can then simply apply weight to the model, storing them in the Container and it will instantly be affected by the rig. Additionally, since the Weight Container is its own item for storing values, it can store weights for more than a single item, further streamlining the rigging process.

Creating a Weight Container

Weight Containers can be easily added to a scene by using the 'Add Items ' function of the Items list -"Add Items > Deformers > Weight Container" Before any weights can be created they need to be linked to the target mesh(es) There are a number of commands that facilitate these connections. Located in the 'Setup' and 'Animation' interface tabs in the toolbox, under the 'Deformers' subtab. The commands are-

Add Weight Container

This command adds a weight container to the items list. If any items are selected when the command is invoked, they will automatically be added to the container. Additionally, if vertices are selected, they will be automatically assigned a weight value of 1.

Add Item

This command will add any selected Items to any selected Weight Container. Select any number of Items and a Weight Container prior to invoking. Additionally, if vertices are selected, they will be automatically assigned a weight value of 1.

Remove Item

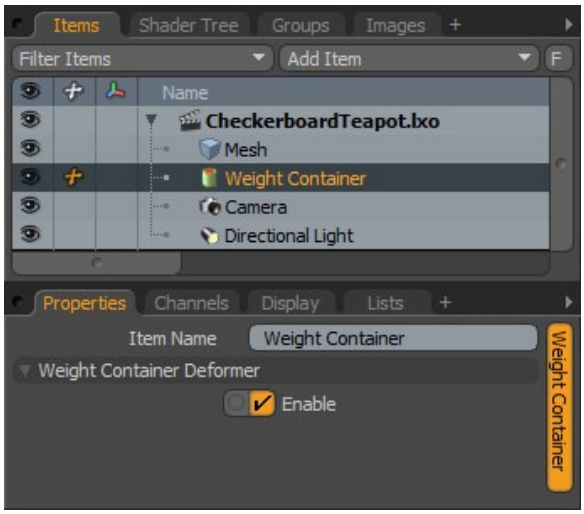
This command will remove any selected Items from any selected Weight Container. Select any number of Items and a Weight Container prior to invoking.

Add Points

This command will add any selected vertices to any selected Weight Container automatically assigning them a value of 1.

Remove Points

This command will add any selected Items to any selected Weight Container. Select any number of Items and a Weight Container prior to invoking. Additionally, if vertices are selected, they will be automatically assigned a weight value of 1.



Item Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Weight Container Deformer--

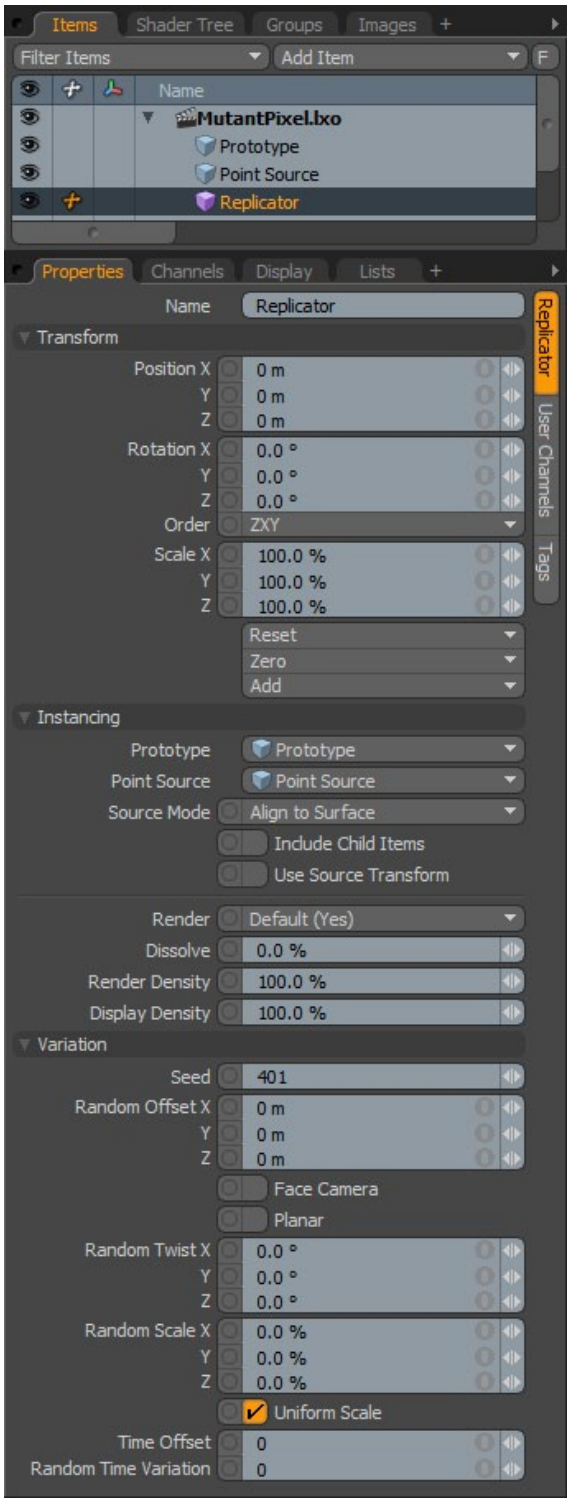
Enable: Toggles the effect of the Container item on or off. When unchecked (disabled), the layer has no influence on the scene. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Replicator Item

Replicators are a means of generating render time duplicates of geometry in a scene allowing users the ability to render literally billions of polygons with full support for all of modo's shading and rendering capabilities -global illumination, soft shadows, refractions, reflections, caustics ...well you get the idea. Replicators are similar in concept to modo's Instances, they derive all their shape and shading information from a base mesh, but unlike an instance, *replicates* can't be manipulated directly --i.e. users cannot position, rotate or scale individual replicas with the usual transform tools. Replicators generate positions using a mesh item as a 'Point Source' utilizing the vertex positions to place the individual replicas. Their positioning can also be further manipulated by adding a 'Surface Generator' shader tree layer allowing for a greater degree of randomization and overall control.

Replicators require two mesh items, the first being a 'Point Source' item, and the other the acting 'Prototype'. The Point Source object acts as the surface determining the position of the replicated objects, and the prototype is the object that will be duplicated. Consider the example of a dry creek bed littered with river stones, the creek bed is the point source determining the placement of the stones, and the stones themselves are the prototypes. Users can add variety to a replicator by adding multiple Mesh items to a 'Group' defined in the Groups palette and assigning that as the Prototype instead of the single mesh item. When specified as such, modo will randomly grab a member of the group to replicate, providing additional variety to the cloned objects.

The easiest way to add a replicator to a scene would be to select first the 'Point Source Item' in the 'Item List' by LMB+clicking on the desired layer (selection order is important), and second, hold down the Ctrl' key and LMB+click the Prototype item (so both items are highlighted). Next, invoke the menu bar command 'Item > Add Replicator' and that's it. You will now see a 'Replicator' item added to the item list and a series of bounding boxes positioned over the surface of your point source geometry. Each bounding box represents the individual replicas. If you'd prefer to set up your Replicator manually, or just wish to understand what the 'Add Replicator' function is doing, you can add a new item to the Item List by clicking the 'Add Item' button in the upper right corner of the viewport and choosing the option "Duplication > Replicator" from the popup list. Now in the instancing section of the replicators properties you can choose the appropriate geometry in the 'Point Source' and 'Prototype' selectors. There are also a number of settings allowing the ability to add some random variations to the replicas. Further randomization is possible through the use of a 'Surface Generator', users can RMB+click on the Replicator item and choose the 'Scatter Replicas' option from the popup menu to assign one to the Shader Tree. Selecting the Replicator item in the Item List will display the attributes in the Properties viewport-



Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. By default, Position transforms originates from the Center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. By default, Rotation transforms originate from the Center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. By default, Scale transforms originate from the Center position.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Item position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Instancing--

Prototype: The 'Prototype' object will be the mesh item that is duplicated over the point source. This can be something as simple as a single polygon or cube, or something as complex as a tree or hi-rise office building. If your object is made up of multiple mesh item layers, make sure to parent the items together in a hierarchy and select the 'Include Child Items' toggle (This also works for animated items).

Point Source: The "Point Source' geometry will be the surface that controls the positioning of 'Prototype' items. The default behavior is a replica generated for each vertex of the 'Point Source' geometry. Each 'Prototype' to be aligned with each 'Point Source' vertex, based on the point of origin for the prototype (not the pivot) and the positive-Y of the prototype will be aligned with the vertex normal direction.

Source Mode: There are 3 different source modes modo can use to generate replicas -- 'Point Data' will use the position of each vertex in a mesh, but the prototype will retain its original orientation. 'Align to Surface' will use the vertex position as well as the normal direction to align each replica, also taking into account subdivision surface smoothing. 'Use Polygons' will generate a replica for each polygon, positioned at the center of each poly and aligned with the polygons normal direction. If you need to generate replicas over a surface with a displacement map, a 'Surface Generator' will be necessary.

Include Child Items: This toggle, when enabled, will tell modo to replicate a parent/child hierarchy. When disabled, only the parent item is replicated. If your 'Prototype' geometry consists of only a single mesh item layer, users can leave this setting disabled.

Use Source Transform: By default, Replicated elements are not affected by item transforms of the source Prototype, their positions are strictly defined by the vertex positions of the Point Source relative to the sources Center position. What this means is that any Position, Rotation and Scale changes applied to the base item don't translate to the Replicated elements. When the 'Use Source Transforms' option is enabled, then the source item transforms will affect the replicated elements, each offset the same value from their initial Point Source positions.

Render: This drop down menu allows the user to select from 3 items, when set to 'Default', the user can enable/disable an item layer using the visibility column (eyeball icon) of the item window. When the item layer is visible, it contributes to the final rendered scene and when invisible, it will not. On some instances the user may prefer to fix this state, setting an item layer as 'On' (enabled) or 'Off' (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving the user from manually enabling item layer for test renders.

Dissolve: The 'Dissolve' function is a convenient way to fade all replicated items from

view. At 100%, the item will render normally, while values lower than 100% will fade all replicas ramping toward 0% where the replicated items would be come completely invisible.

Render Density: The "Render Density' function provides a degree of control over the number of generated replicas. Normally, an item is generated at each vertex position, so for a 10x10 grid, you would end up with 121 vertices, therefore 121 replicas. Setting the 'Render Density' to 50%, the replicator would randomly generate replicas reducing the overall number by the approximate Density value. Values below 100% will reduce the number or replicas, however, values above 100% will not generate additional replicas. You may wish to use a 'Surface Generator' to provide better control over the number of generated items.

Display Density: Only the bounding box of replicated items is displayed in the 3D viewport. If the number of instanced items becomes too great, you may wish to reduce the number of bounding boxes displayed by adjusting the 'Display Density' value. A setting of 0% would disable the bounding box's display without affecting the final rendered image.

Variation--

Seed: The 'Seed' value is an initial setting used when random numbers are generated. Different 'Seed' values will produce different random variations, however you will want to use the same 'Seed' value when it is necessary that items retain the same variations.

Random Offset: The 'Random Offset' value when set to any number above 0 for X, Y or Z, will provide a range from where random values can be generated. With an offset value of say 1m, the replicated item will be randomly offset up to 1 meter from its initial position. Values of 0 disable this functionality.

Face Camera: The 'Face Camera' setting will rotate replicated items automatically so that they always face toward the camera, regardless of its position, using the 'Prototype' objects origin point as the center of rotation.

Planar: Used in conjunction with 'Face Camera', when 'Planar' is enabled, replicated items only rotate around the 'UP' axis, especially useful for billboard style trees (flat polygons with a transparent tree image mapped on onto it).

Random Twist: The 'Random Twist' when set to any number above 0 degrees for X, Y or Z, will provide a range from where random values can be generated. For example, with an twist value of 90 degrees, the replicated items would be randomly rotate 90 degrees in either direction from initial position. Values of 0 disable this functionality.

Random Scale: The 'Random Scale' when set to any amount above 0% would produce

a random scaling amount anywhere within the range for the value specified. For example, with a setting of 50%, replicated items would vary anywhere between 50% to 150% of the items original size. Values of 0 disable this functionality.

Uniform Scale: When using 'Random Scale' values, all 3 axis are generated independently. When 'Uniform Scale' is enabled, random scaling of generated replicas would be uniform across all axes eliminating squashing or stretching.

Time Offset: The 'Time Offset' option will offset the playback of an animation applied to replicated objects. It can also be used to move the random span of time when combined with 'Random TIme Variation'. For example if this is set to 5 seconds and the variation is set to 1 second, then the replicas will get their motion from the animation of the prototype between 5 and 6 seconds into the future.

Random Time Variation: The 'Random Time Variation' option selects a span of time for replicas to vary their animation within from their current time. So if this is set to 1 (second) then the replicas will play their animations on a random start frame between the current time and one second in the future.

Falloffs Definition

Falloffs have a multitude of uses. They can be used as Deformers, where their area of influence can be assigned to deform the target mesh; think of this like having a magnet that is used to push and pull a surface. Falloffs can also be used to attenuate the effects of other Influences upon an Item or Mesh, similar to a modeling Falloffs that controls the amount of influence a tool has on a surface. Finally, Falloffs may also be used to control the degree to which a 'Force' influences a Dynamic item, such as a Rigid or Soft Body. How the Falloff affects other items is largely dependent on how it is added to the scene. The influence shape is related to its type. The 'Capsule' Falloff produces an attenuation area shaped as a cylinder with rounded ends, with user controls to adjust the size, shape and transition of the Falloff itself.

Adding a Falloff as a Deformer

A Falloff item can be added to a scene easily enough using the 'Add Item' function of the Items list, but the Falloff would have no direct effect upon any Items in the scene until it was linked to the target Item it is meant to act upon. This can be done manually using the Schematic view. modo offers a few ways to create these links automatically. While in 'Items' mode, Select the target mesh Item and in the 'Setup' or Animate' interface tabs, under the 'Deformers' subtab of the toolbox, select the "Add Falloff Deformer" and select the target Falloff type. This adds the Falloff and links it to the selected mesh. Moving the Falloff at this point will also move any points within the influence of the Falloff. Users can adjust the initial 'Rest' (undeformed) state by

entering 'Setup' mode (press 'Enter') and position the Falloff to where it should be in its undeformed state. Exiting 'Setup' now (press Enter again) will allow users to animate or adjust the position of the Falloff, deforming the target Mesh.

Adding a Falloff to an Influence

To have a Falloff attenuate an Influence is done using the 'Deformers' viewport, where users can automatically create and link a Falloff to the Influence in a single step. Within the viewport, select the target Influence item and RMB+click on it to open the contextual menu, from there select the "Add Override > Falloff > Falloff Type" adding it to the scene. Once added, the Positions Size and Range can all be animated, where the Influence in questions will only affect the target mesh within the bounds of the Falloff (with settings to Invert this behavior, if desired).

Adding a Falloff to a Force

A Falloff can also be automatically linked to a 'Force' item attenuating the effects of the Force on dynamic items. This can be done easily by simply selecting the 'Force' item (while in 'Items' mode) and using the 'Add Falloff' option of the 'recoil' subtab of the same toolbox. With the Falloff active, Dynamic items within the bounds of the Falloff will not be affected tot he full degree of the Force, attenuating to the core of the Falloff where Items wouldn't be affected at all.

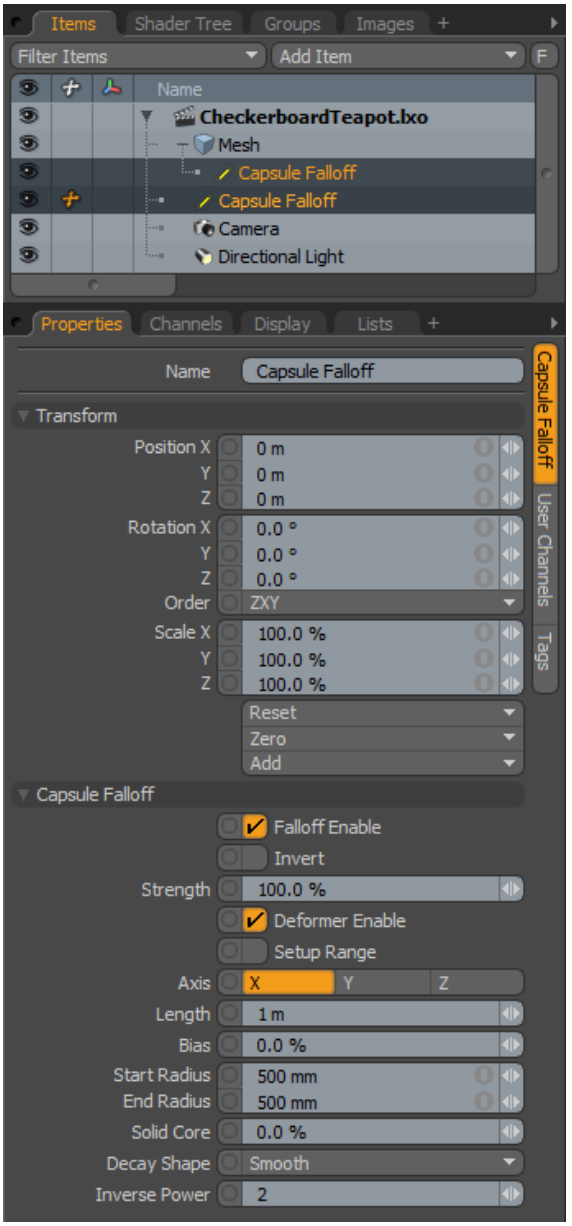
Capsule Falloff

Like with modeling, a Falloff is used to attenuate the affect any deformer has on a target item, providing additional control how deformers influence a mesh.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. Position transforms originate from the Center position.



Rotation: An Item transform that allows the user to numerically set the rotation of the item. Rotation transforms originate from the Center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. Scale transforms originate from the Center position.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Item position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Capsule Falloff--

Falloff Enable: Toggles the Falloff on or off. When unchecked (disabled), the Falloff has no influence. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Invert: When *enabled*, reverses the influence of the Falloff.

Strength: The 'Strength' option controls the maximum amount of influence the Falloff has on its target. At a value of 50%, even areas that are affected fully by the falloff,

would only be influenced half as much.

Deformer Enable:

Setup Range:

Axis: The 'Axis' option determines the initial axis direction of the Capsule shape.

Length: The 'Length' option determines the extent of the capsule shape from end to end.

Bias: The Center of influence is directly between the Start and End points of the Capsule. The 'Bias' value modifies the center toward the Start side of the capsule with positive values, and toward the End side with negative ones.

Start Radius: Determines the size of the origin side of the capsule as a radius; its center to its outside perimeter.

End Radius: Determines the size of the terminating side of the capsule as a radius; its center to its outside perimeter.

Solid Core: Generally, the Falloffs influence attenuates from its center line toward its outer edge. The 'Solid Core' option defines an inner range where the Falloffs influence is at maximum amount as a percentage of the total size of the Capsule shape.

Decay Shape: The Decay shape controls the rate of attenuation across the falloff with three possible choices-

Smooth-

Inverse Power-

Gaussian-

Inverse Power: When the 'Inverse Power' option is selected as the 'Decay Shape', the 'Inverse Power' value determines the

Linear Falloff

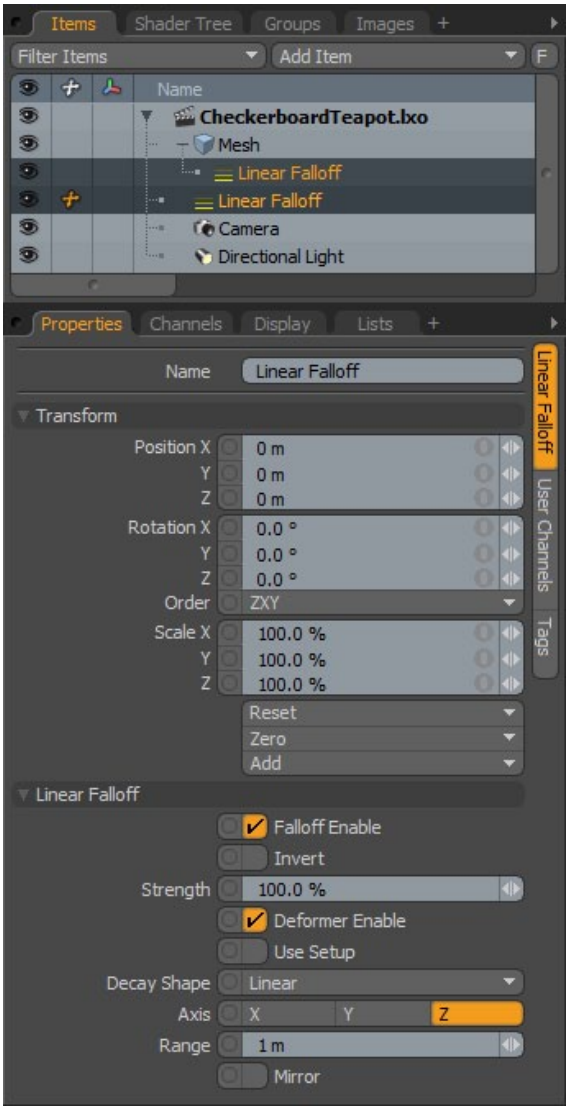
Like with modeling, a Falloff is used to attenuate the affect any deformer has on a target item, providing additional control how deformers influence a mesh.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ

space. Position transforms originates from the Center position.



Rotation: An Item transform that allows the user to numerically set the rotation of the item. Rotation transforms originate from the Center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. Scale transforms originate from the Center position.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Item position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Linear Falloff--

Falloff Enable: Toggles the Falloff on or off. When unchecked (*disabled*), the Falloff has no influence. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Invert: When *enabled*, reverses the influence of the Falloff.

Strength: The 'Strength' option controls the maximum amount of influence the Falloff

has on its target. At a value of 50%, even areas that are affected fully by the falloff, would only be influenced half as much.

Deformer Enable:

Use Setup:

Decay Shape: The Decay shape controls the rate of attenuation across the falloff with three possible choices-

Linear- An even attenuation of the falloffs influence across its total range.

Ease In- The influence attenuates more abruptly toward the origin of the the falloffs range.

Ease Out- The Influence attenuates more abruptly toward the termination of the the falloffs range.

Smooth- The Influence attenuates more abruptly toward the center of the the falloffs range.

Axis: The 'Axis' option determines the initial axis direction of the Linear Falloffs shape.

Range: The 'Range' option determines the overall extent of the Falloffs influence as a length originating from the Falloffs center.

Mirror: The 'Mirror' option will produce an inverted duplicate of the falloffs influence across its origin and axis.

Radial Falloff

Like with modeling, a Falloff is used to attenuate the affect any deformer has on a target item, providing additional control how deformers influence a mesh.

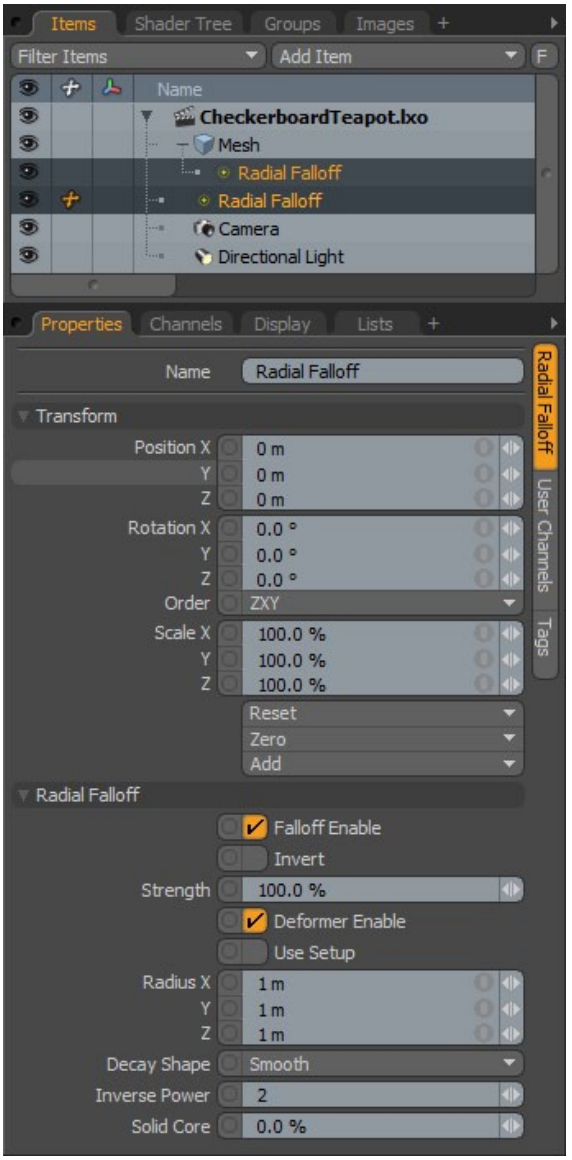
Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. Position transforms originates from the Center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. Rotation transforms originate from the Center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.



Scale: An Item transform that allows the user to numerically set the size of the item. Scale transforms originate from the Center position.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Item position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Radial Falloff--

Falloff Enable: Toggles the Falloff on or off. When unchecked (*disabled*), the Falloff has no influence. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Invert: When *enabled*, reverses the influence of the Falloff.

Strength: The 'Strength' option controls the maximum amount of influence the Falloff has on its target. At a value of 50%, even areas that are affected fully by the falloff, would only be influenced half as much.

Deformer Enable:

Use Setup:

Radius X/Y/Z: These three values determine the size of the sphere as a radius for each axis; from its center to its outside perimeter.

Decay Shape: The Decay shape controls the rate of attenuation across the falloff with three possible choices-

Smooth-

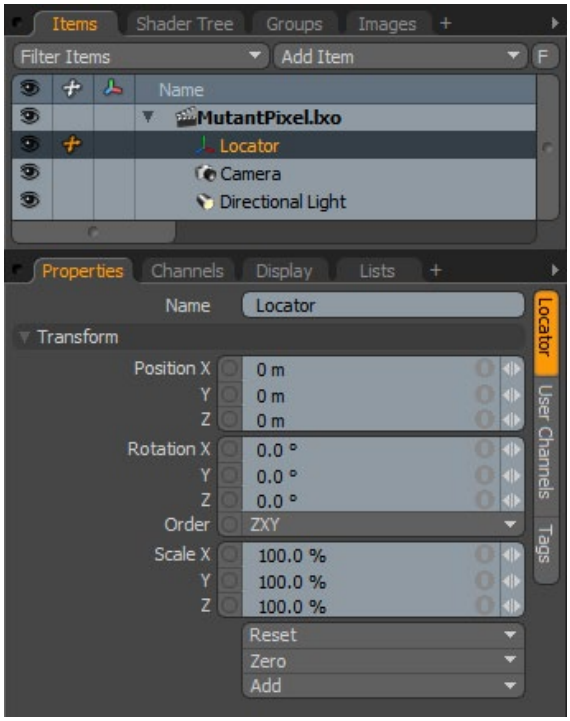
Inverse Power-

Gaussian-

Inverse Power: When the 'Inverse Power' option is selected as the 'Decay Shape', the 'Inverse Power' value determines the

Solid Core: Generally, the Falloffs influence attenuates from its center line toward its outer edge. The 'Solid Core' option defines an inner range where the Falloffs influence is at maximum amount as a percentage of the total size of the Spherical shape.

Locator Item



Locators are like the swiss army knife of 3D, playing a key role in animation serving many important functions -targets, controllers, parents, manipulators, selectors. Since locators are just *items*, they have their own set of properties, most importantly their own *transforms*, and as such, they can be positioned and animated independently of mesh items in a scene. This might not seem so exciting by itself, as locators don't even render, but when working with constraints and modifiers, one will quickly discover how useful they really are. As such, there are some special properties associated specifically with locators. Additional draw properties specific to Locators are available in the Display viewport. Pick Walking and Selection Redirect settings are available via the Assembly viewport.

Users can add a Locator to a scene using the 'Add Item' button in the top right corner of the Items list viewport. Additionally, users can use the menu bar command "Item > Create Locator". Once selected the Locators attributes become available in the Properties viewport-

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. By default, Position transforms originates from the Center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. By default, Rotation transforms originate from the Center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. By default, Scale transforms originate from the Center position.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Item position intact.

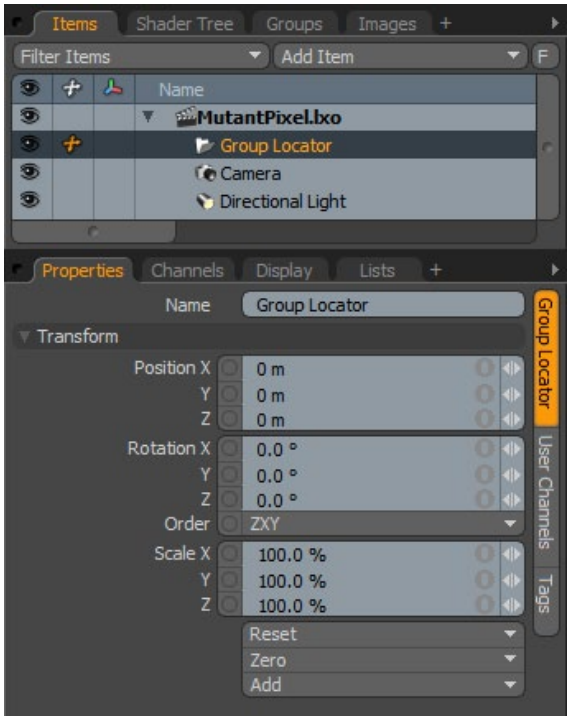
Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Group Locator Item

As a project becomes more complex, user may find that the flat listing of layers in the Items List viewport becomes unwieldy to work with. Parenting items to a 'Group Locator' will help organize the project. Different from an actual 'Group', Group Locators are simply a special kind of 'Locator' item that does not draw anything in the 3D viewports, eliminating additional scene clutter. Group Locators can be easily collapsed within the Item List which keeps the viewport clean and manageable. Group locators have an icon that looks like a folder in the Item List to distinguish them from regular Locator items. In all other aspects they act the same as a normal locator; they can be moved and rotated in the 3D views for example.

To create a Group Locator, users can choose "Add Item > Group Locator" from the button within the Item List. Once created, users can begin to drag and drop items into a hierarchy, organizing the list. Several items can be quickly parented to a Group Locator by selecting all the target layers in the Items List, holding down the 'Ctrl' between each LMB+click to select multiple items; once selected press 'Ctrl+G' to automatically

create a new Group Locator, with all selected items parented in one step. Users can then LMB+click the small arrow '' preceding the layer to toggle the visibility of the child items.



Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. By default, Position transforms originates from the Center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. By default, Rotation transforms originate from the Center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. By default, Scale transforms originate from the Center position.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Item position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Texture Locator Item

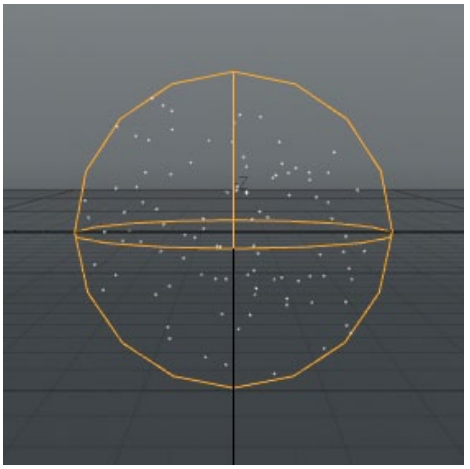
When any texture layer is created in the Shader Tree, image based or procedural, an

associated 'Texture Locator' item is automatically generated and added to the items list and linked by the texture layers 'Locator' property. By default, Texture Locators are generally added to a folder (really a group locator item) called 'Texture Group'. This makes it easy to toggle the visibility of the locators in the 3D viewports, as some users find them distracting. Users may select the texture locator directly in the items list, or by selecting the locator icon in any 3D viewport. Most often though, users will select them via the shader tree; when any texture layer is selected in the tree, the locator is available in the named subtab on the right side of the items associated 'Properties' panel. Additionally, the locator may be toggled by the small '+' sign preceding the texture layers name in the Shader Tree, LMB+clicking the '+' icon reveals the layer.

Texture locators provide a very important function as they control the application of the texture layer over the rendered surface. A Texture Locator is like a regular 'Locator' Item, in that it has Position, Rotation and Scale information. Additionally there are settings specific to controlling the images projection method, meaning how it is mapped or applied to the surface. For certain projection modes, such as 'Planar', users can interactively position textures in the 3D viewport by transforming the Texture Locator item itself. Other modes, UV in particular, completely ignore the positional and scale attributes as the mapping is defined by the UV map itself.

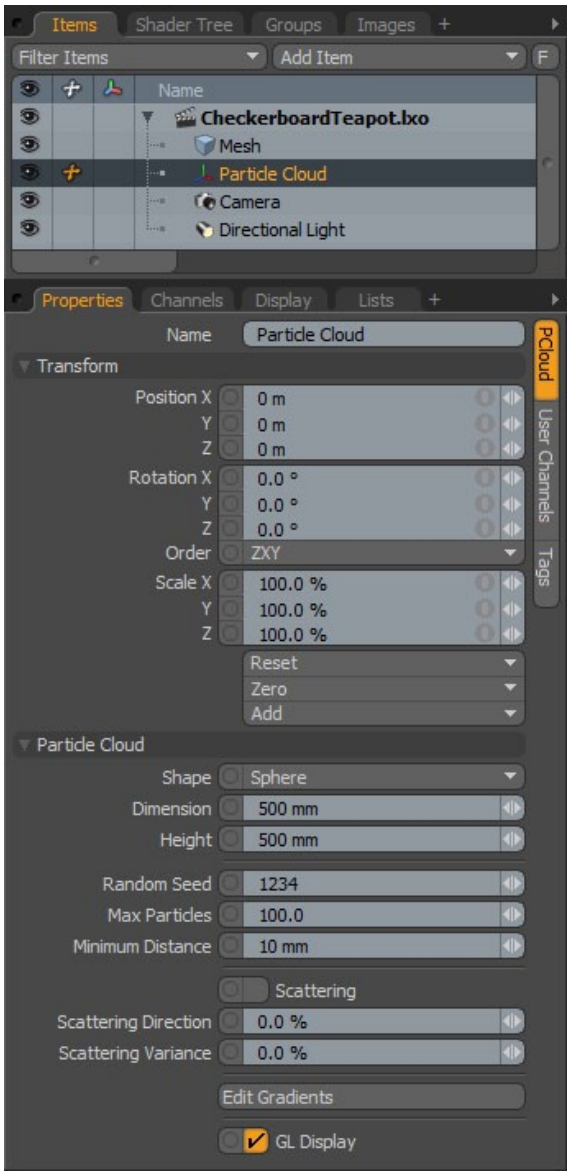
The attributes of the Texture Locator item found in the properties panel are documented on the Texture Locator page of the Shading Lighting and Rendering section of the documentation, please reference that page for additional information.

Particle Cloud



The Particle Cloud item generates a randomized array of single vertices. Points won't render by themselves, however, they can be used as a point source for a variety of

items, such as for Blobs, Sprites and Volumes, or when working with Replicators. Animation can be added to the particles within the cloud by adding gradient controls over times (keyframes) with the 'Scattering' function.



Adding a Particle Cloud

Users can add a Particle Cloud item simply by using the 'Add Item' function of the Items List. LMB+click on the button to open the menu, then select the "Particles > Particle Cloud" option. Once added, users can then specify the item from the 'Point Source', or 'Particle Source' options from within the appropriate properties panel of the related function.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. Rotation transforms originate from the Position value.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. Scale transforms originate from the Position value.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Mesh position intact. This is done by adding a negative transform item to the Mesh items channels.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Particle Cloud--

Shape: Determines the overall shape of the containing volume for the particles, either a 'Sphere', 'Cube' or 'Cylinder'.

Dimension: Determines the size of the volume as a radius, so a 500mm Dimension would produce a 1m wide container volume for the particles.

Height: When the 'Cylinder' option is selected, the 'Height' value determines the overall length of the cylinder shape.

Random Seed: The 'Random Seed' value is the initial number used when generating the random procedural values. Each Seed will produce different random results.

Max Particles: Determines the maximum number of total particles generated within the volume.

Minimum Distance: Determines the minimum amount of distance between two particles within the volume.

Scattering: The 'Scattering' option is provided to allow for simple animation of the particles. When *enabled* each individual particle is scattered in a random direction, based on the 'Scattering Direction' and 'Scattering Variance' values. In order to see any changes in the 3D viewports, it will be necessary to keyframe some values using the 'Edit Gradients' option.

Scattering Direction: The 'Scattering Direction' option provides some additional control of the overall random distribution of the particles.

Scattering Variance: The 'Scattering Variance' option controls how much variation there is in distance when the particles are scattered. Higher values will move some particles further from the volume than others, while smaller values will tend to scatter particles in more uniform distances.

Edit Gradients: Pressing the Edit Gradients button will open a Gradient Editor window, allowing users to set up gradient controllers for Particle density within the cloud as well as the Scattering Distance.

GL Display: Toggles the visibility of the 3D viewport proxy representation. *Enabled* (visible) by default.

Particle Generator

The 'Particle Generator' produces similar results to the 'Surface Generator', in that it is used to scatter vertices (particles) across a specific item or surface. However, the Particle Generator is evaluated at the scene level making it more flexible and faster. The resulting particles can then be used as a point source for Blobs, Sprites, Volumes, Replicators and when working with Texture Replicators. One thing to keep in mind though, that since it is scene evaluated, it wont take in to consideration rendering effects such as Displacements applied to the surface; in that case users should apply the Surface Generator.

Adding a Particle Generator

Users can add a Particle Cloud item simply by using the 'Add Item' function of the Items List. LMB+click on the button to open the menu, then select the "Particles > Particle Generator" option. Once added, users can then specify the item from the 'Point Source', or 'Particle Source' options from within the appropriate properties panel of the related function.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

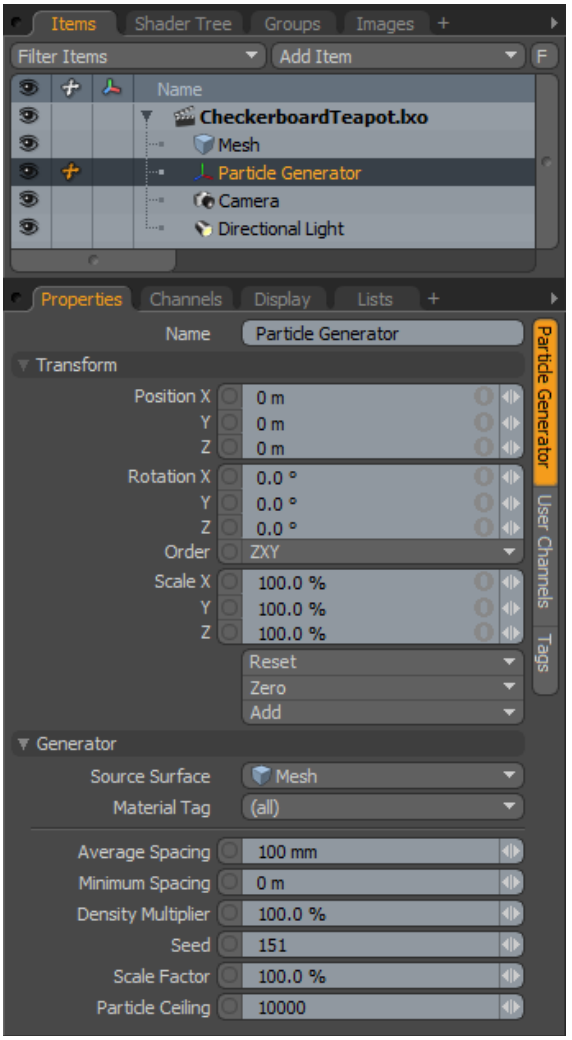
Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. Rotation transforms originate from the Position value.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. Scale transforms originate from the Position value.



Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Mesh position intact. This is done by adding a negative transform item to the Mesh items channels.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Generator--

Source Surface: This dropdown menu can be used to specify an item within the scene to apply the particles over its surface. This is the equivalent of an 'Item' mask in the Shader Tree.

Material Tag: The 'Material Tag' option allows users to further limit the application of the Particles by specifying a surface definition from the dropdown menu. This is the equivalent of an 'Material' mask in the Shader Tree.

Average Spacing: Precisely as the name implies, the 'Average Spacing' value specifies the average distance between particles.

Minimum Spacing: Where the 'Average Spacing value only specifies an average, the 'Minimum Spacing' value allows one to specify the minimum amount of space between replicated items, especially useful in eliminating overlaps for neighboring replicas when used as a Replicator source.

Density Multiplier: The 'Density Multiplier' offers simplified control over the overall density of replicated items. Lower values will reduce the overall number of replicas

while higher values will increase the overall number of replicas up to the 'Particle Ceiling' value.

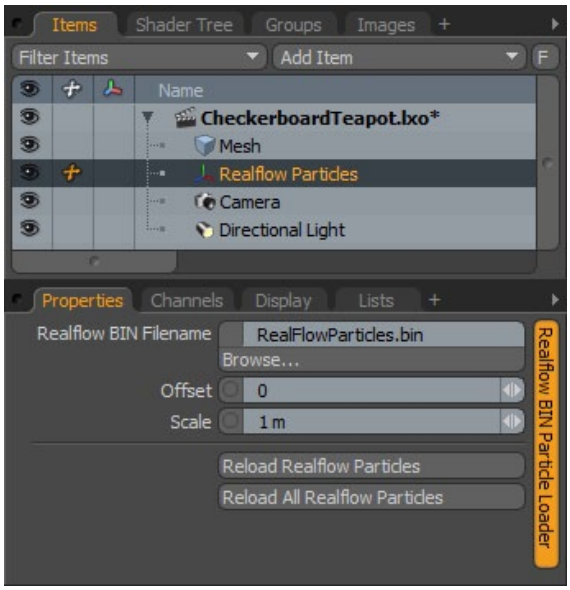
Seed: The 'Seed' value is an initial setting used when random numbers are generated. Different 'Seed' values will produce different random variations, in this case, initial particle positions. However you will want to use the same 'Seed' value when it is necessary that items retain the same variations.

Scale Factor: The 'Scale Factor' acts to control the scale of replicated objects, eliminating the need to manually scale the base 'Prototype' item. With a value of 100%, replicated items would be generated at the identical size of the prototype, smaller values would decrease each replicas size, while larger value would increase each replicas size.

Particle Ceiling: The 'Particle Ceiling' is a maximum for the total number of generated particles.

RealFlow Particles

RealFlow is a software tool that produces realistic liquid simulations, that include physical interactions with scene elements as well as soft and hard body collisions. The calculated simulations can be saved as binary files that store the motion information for each individual particle within the simulation with additional information related to speed, age, etc. When properly exported, these BIN files can be read into modo, producing animated particles that can be used as point sources for Blobs, Sprites and Volumes, or when working with Replicators.



Adding the RealFlow Particles

Users can add a RealFlow Particles item simply by using the 'Add Item' function of the Items List. LMB+click on the button to open the menu, then select the "Particles > RealFlow Particles" option. Once added, users can then specify the BIN files name and location using the 'Browse' option found in the items properties panel. Users should note that the RealFlow BIN file is not saved internally to the modo scene and the external file will need to remain in its same relative

relative location to when it was loaded in order to render properly.

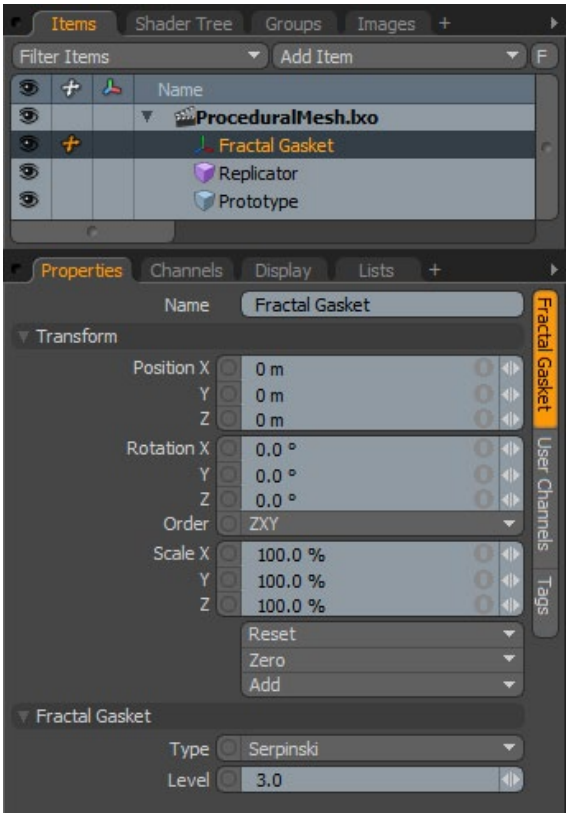
RealFlow BIN Filename: This option displays the Name of the file loaded via the 'Browse' button. Pressing the button will open an OS specific file open window, where users will specify the name and location of the saved RealFlow cache.

Offset: The 'Offset' option moves the simulation time forward or backward, depending on value, allowing users to compensate or adjust the resulting simulation.

Scale: The 'Scale' option allows users to re-size the simulation results.

Reload RealFlow Particles: Useful for when adjustments made in RealFlow are exported over the existing file, pressing this button will trigger a reload of the RealFlow simulation, updating the results in modo.

Reload All RealFlow Particles: Pressing this button will trigger a reload of all imported RealFlow particle simulations.



Fractal Gasket

Fractal Gasket item is a plug-in item type that generates various geometric arrays of objects. Used in combination with a Replicator, users would add the Fractal Gasket item and assign it as the point source within the Replicators properties. Users can then adjust the options of the Fractal Gasket item to change the array types and their level of complexity.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. By default, Position transforms originates from the Center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. By default, Rotation transforms originate from the Center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. By default, Scale transforms originate from the Center position.

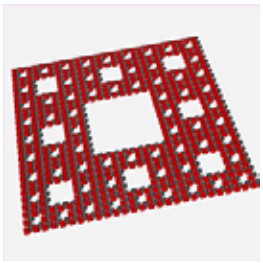
Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and item position intact.

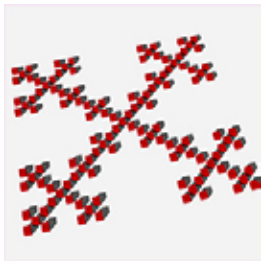
Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Fractal Gasket--

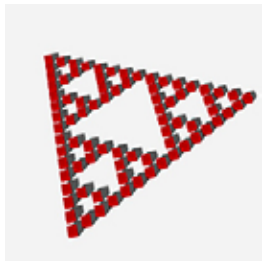
Type: This option determines the array type, there are four options to choose from. The first three, 'Square Hole', 'Square Flake' and 'Serpinski' generate flat two-dimensional arrays, while the 'Tetrahedron' option generates a three-dimensional array.



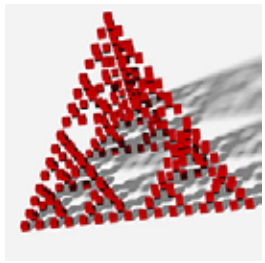
Square Hole



Square Flake



Serpinski

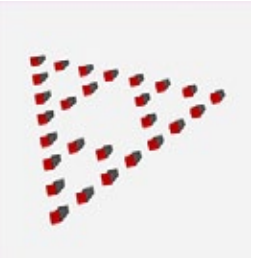


Tetrahedron

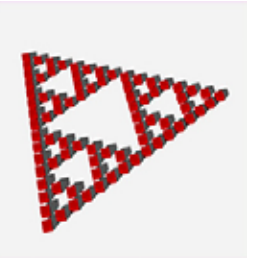
Level: This option determines the complexity of the array with each additional level adding an exponential amount of detail. Each new level increases the 'Square Hole' type by a factor of 8, 'Square Flake' by a factor of 5, 'Serpinski' by a factor of 3 and 'Tetrahedron' by a factor of 4. Therefore care should be given to not go overboard when increasing levels, especially with the more complex types.

Serpinski Level 2

Serpinski Level 3



Serpinski Level 4



Serpinski Level 5

TIP: To easily change the size of items in a Fractal Gasket Array, users can set the Replicator to 'Point Mode' and then scale the size of the prototype object in one of the component modes.

Gear Item

The Gear item is a plug-in item type that generates accurate gear and cog shaped geometry with ease. Users can then adjust the options of the Gear item to customize the look and level of complexity. If desired, users can convert the procedural Gear to a 'Mesh Item' by 'Freezing' the object for further editing. Users can add a 'Gear Item to the scene using the 'Add Item' button within the Items List. Selecting the item will display its associated attributes in the Properties panel.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

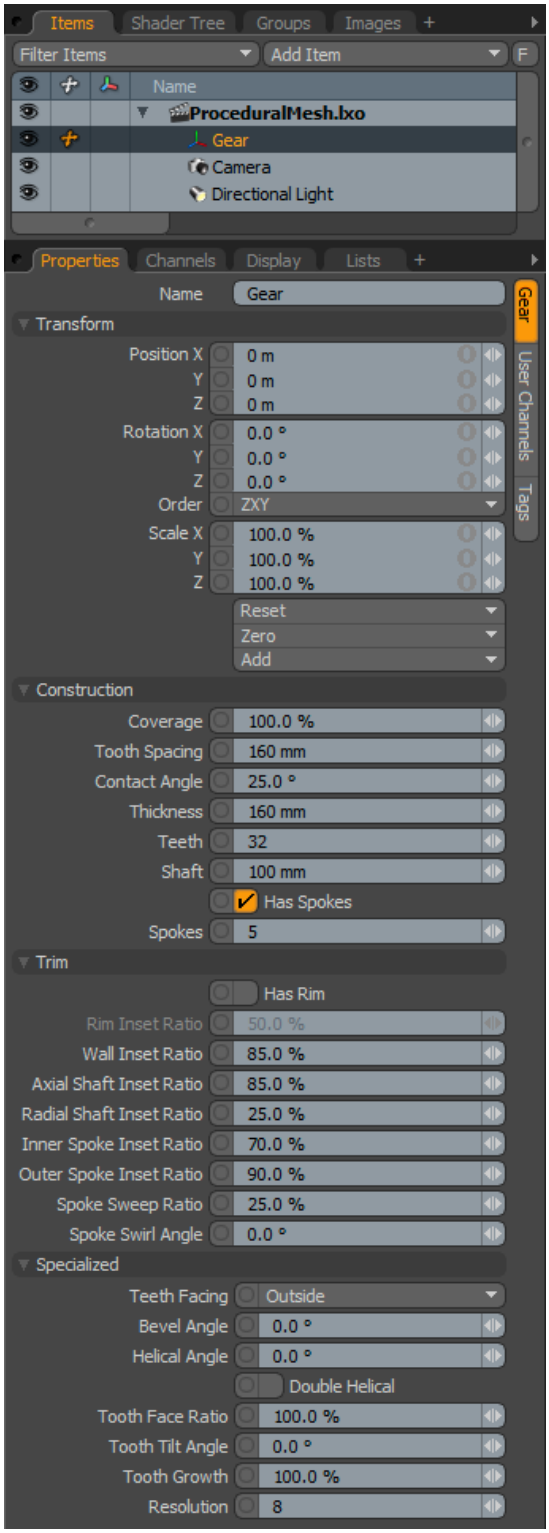
Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. By default, Position transforms originates from the Center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. By default, Rotation transforms originate from the Center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. By default, Scale transforms originate from the Center position.



Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Item position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Construction--

Coverage: Percentage of constructed shape

Tooth Spacing: Space between each tooth

Contact Angle: Angle of each tooth (up to 45 degrees)

Thickness: Thickness of the gear

Teeth: Number of teeth for the gear

Shaft: Size of the shaft

Has Spokes: If the interior of the gear has spokes

Spokes: Number of spokes in the interior of the gear

Trim--

Has Rim: If the interior of the gear has a rim

Rim Inset Ratio: Ratio of rim thickness to total thickness

Axial Shift Ratio: Ratio of gear wall to gear radius

Radial Shaft Inset Ratio: Ratio of shaft rim to gear radius

Inner Spoke Inset Ratio: Ratio of inner spoke thickness to total thickness

Outer Spoke Inset Ratio: Ratio of outer spoke thickness to total thickness

Spoke Sweep Ratio: Ratio of spoke sweep to spoke spacing

Spoke Swirl Angle: The angle by which the spokes should be swirled

Specialized--

Teeth Facing: The direction in which the teeth are facing

Bevel Angle: Angle of tooth bevel

Helical Angle: Angle of tooth helix

Double Helical: Double helical (herringbone) pattern

Tooth Face Ratio: Ratio of tooth face to thickness

Tooth Tilt Angle: Angle of tool tilt along the rotational axis

Tooth Growth: Amount of tooth growth

Resolution: Geometry resolution

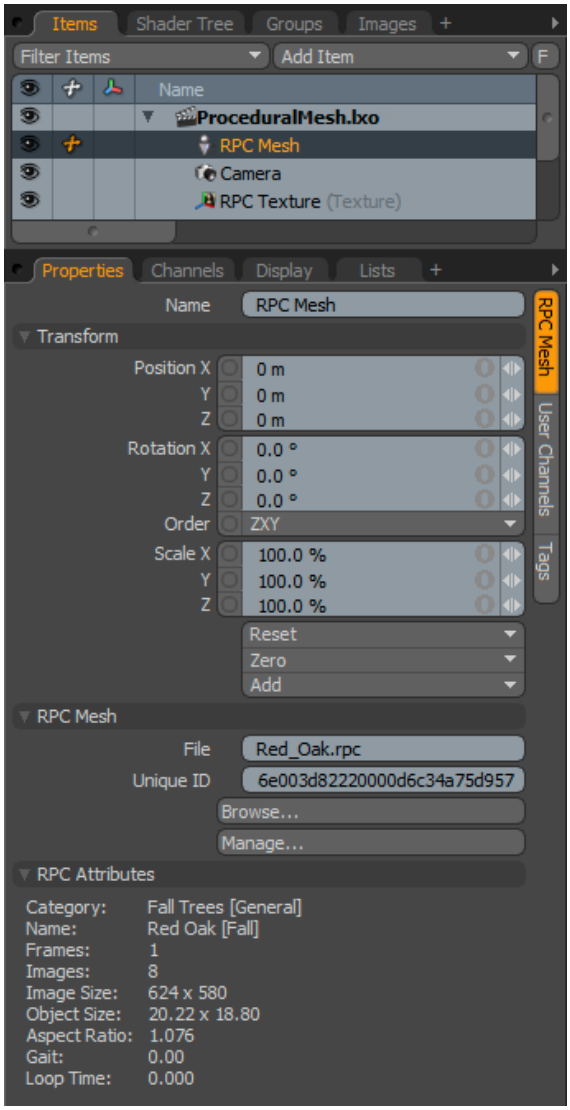
RPC Mesh Item

RPC Items allow users to utilize 'Rich Photoreal Content', specially processed photographic content that can be leveraged to increase the complexity and depth of a scene. Developed by Archvision, RPC technology uses high quality image data combined with minimal polygonal geometry to create the illusion of 3D. By synthesizing image data as the camera moves through the scene, images effectively reproduce the appearance of a 3-Dimensional object without relying on complex geometry. An easy to use interface makes the placement of RPC content a simple operation. A simplified representation is first used to orient RPC content within the scene, and at render time the RPC plug-in calculates the viewing angle of the RPC item relative to the camera. The appropriate corresponding images are then calculated, creating the illusion that a complete 3D object exists in the scene.

Using RPC Items in modo

To add an RPC item to a scene, utilize the 'Add Item' function of the Items List. Once

the item itself is added, user will need to browse a library to and specify an RPC element before anything will show in the 3D viewport. modo includes a selection of Preset content that users can browse through and add to their scenes, or users may also purchase additional libraries directly from Archvision's website "http://www.archvision.com/". Libraries include trees, automobiles, people, bushes, furniture and other useful objects. Whenever an RPC item is added, a corresponding Material Group and RPC Texture will be added to the Shader Tree as well, providing users a means to adjust the image's 'Gamma' for linear workflows and other situations.



Once an RPC Item has been added to the Item List, selecting the Item (in the Items List) and navigating to the Properties panel will present two buttons- 'Browse' and 'Manage'. The 'Browse' option provides users a means to locate RPC content archives locally on their systems. The Manage Button opens the 'RPC Thumbnail Browser'. Here users can select and configure content for use in their scenes. Once a specific item is selected, press 'OK' to utilize the selected content in the scene. By default, the item will appear at the origin location. Users can then utilize modo's item transform tools to position the element within the scene.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. By default, Position transforms originates from the Center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. By default, Rotation transforms originate from the Center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. By default, Scale transforms originate from the Center position.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Item position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

RPC Mesh--

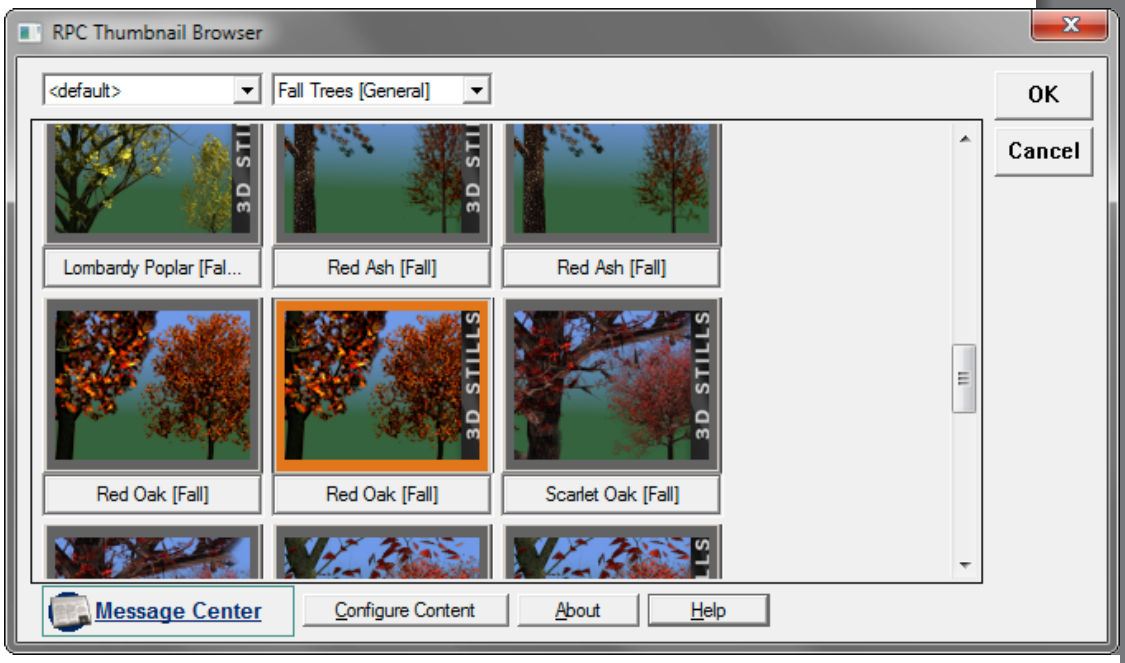
File: This data field displays the file location of the currently loaded RPC mesh.

Unique ID: An identifier related to the File.

Browse: The browse function is used to locate and select RPC archives on the local machine.

Manage: Opens the 'RPC Thumbnail Browser' (shown below). Here users can select and load elements visually within the RPC archive and configure the content.

RPC Browser Window



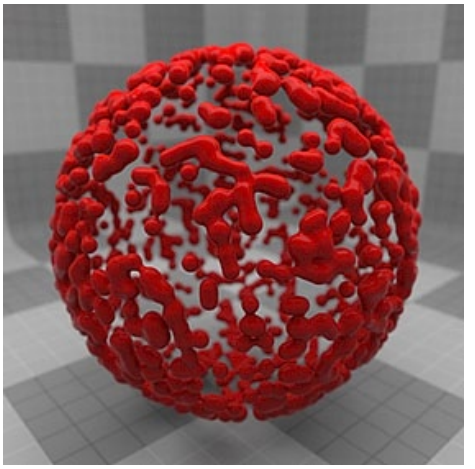
RPC Browser Window

RPC Attributes--

These options are merely information displays regarding the chosen asset, providing uses a quick reference to the technical

INFO: RPC objects are only compatible with the Windows version of modo.

Blob Item



The 'Blob' item provides a means for users to procedurally create blobby meta-ball like shapes; combinations of spherical forms with an interpolated smoothly curved surface covering them. The positions of individual blob shapes are determined by the vertices of a 'Particle Source' item, where each vertex in the Particle Source produces a single blob (The Particle Source can be a number of items described under the 'Particle Source' option below). The form itself is dependant on the radius of the Blob particles and the distance between adjoining vertices, as the surfaces get closer together, they will try to join their surfaces (think like water droplets when they get close to each other). The resulting surface can be further controlled with the addition of textures added via the 'Shader Tree' producing complex surfaces very easily.

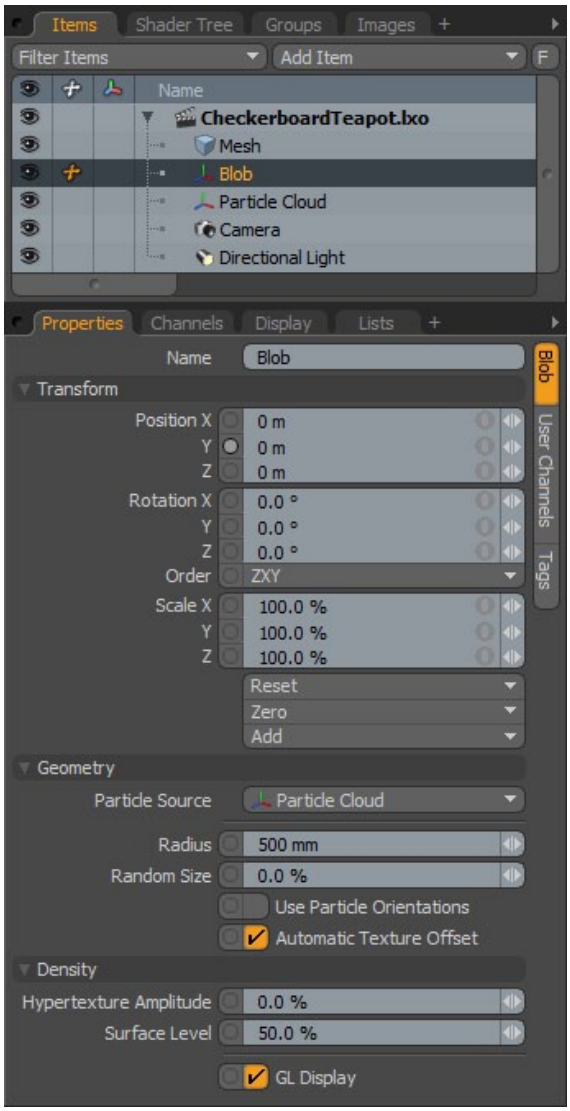
Adding Blobs

A Blob item can be added to a scene by utilizing the 'Add Item' function found in the 'Items List'. Users can LMB+click the button itself, opening the popup menu and select the "Volumes > Blob" option. Once added, the item will appear at the bottom of the Items List viewport. When selected, a number of attributes controlling the blobby shape appear in the 'Properties' viewport.

Applying Materials

Normally, surfaces are applied through the use of polygon tags that define masks in the Shader Tree controlling material placement, however, 'Blobs' are a procedural surface created at render time and have no user selectable polygons to apply a tag to, so surfacing is applied to the 'Blob' item through the use of an Item Mask. Users can quickly create an 'Item Mask' by RMB+clicking on the Blob item in the Items list and choosing the 'Create Item Mask' option from the contextual menu. Users can then add

'Materials' and other Shader Tree items, setting the appropriate 'Effect' for each layer building up surfacing attributes just as one would for any other surface. Users may also wish to control additional attributes of the Particle Source that are not surface specific (such as 'Particle Size' and 'Particle Density' for the Source), this can be done by creating an additional Item Mask and applying texture layers set to the appropriate 'Effect' under the 'FX Item' of the Shader Tree. For more on using FX, see that page of the documentation.



Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. By default, Position transforms originates from the Center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. By default, Rotation transforms originate from the Center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. By default, Scale transforms originate from the Center position.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Mesh position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Geometry--

Particle Source: When a 'Sprite' item is added to a scene, it is represented by a single locator type point at the world origin producing a single *sprite* image. To produce additional sprites, users will want to define a 'Particle Source', this can be any Mesh Item layer that contains vertices, where each vertex position is used to determine the position of the Sprite particle (any type of geometry can be used- single point polygons, vertices, two-point polygons and regular polygons, however, only the vertex positions of any component will be used). Source layers can be normal geometric forms, point clouds that are imported or created using the 'Particle Cloud' item or any of the 'Particle Tools'. They can be imported particle simulations controlled by way of a MDD deformer, RealFlow Point Cache or an Alembic format data file.

Radius: This option determines the base size of the Sprite image. From here, the size can be altered via the 'Random Size' option, the addition of a 'Particle Size' vertex map that allows users to *paint* the scale for individual particles interactively, or with the application of a texture layer, under the FX Item of the shader tree, set to control 'Particle Size'.

Random Size: This option controls the amount of randomness added to the scale of each individual particle blob. The amount is determined by a scale percentage- a setting of 25% would adjust the scale of the base radius size by an amount of 0%-25% randomly across all particles, a setting of 50% would adjust the size by 0%-50% randomly across all particles and so on.

Use Particle Orientations: Aligns the particle to the direction vector from the particle source. The direction vector is usually the surface normal for surface particles or the particle velocity vector for simulation particles (like from RealFlow). Note that in the case of blobs, this option is disabled by default as it adds a bit to render time (around 30%), so don't use it if you don't need it.

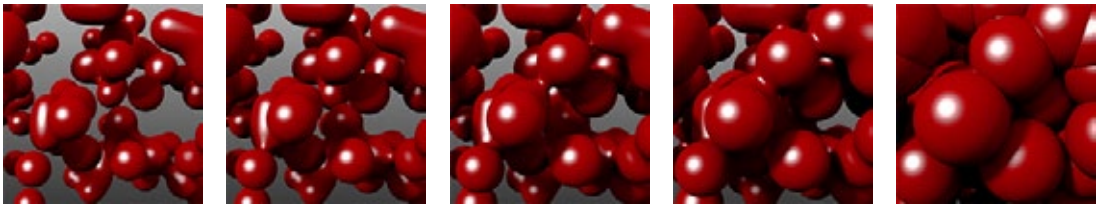
Automatic Texture Offset: Adds an offset to the hypertexture different for each particle so that each particle looks different.

Density--

Hypertexture Amplitude: Determines the amplitude (intensity) of a 'Volume Density' texture layer applied to the Blob volume. The higher the value the more you see the effect of the density texture. It's similar in concept to displacement amplitude but applies to volumes.

Amplitude 0% Amplitude 25% Amplitude 50% Amplitude 75% Amplitude 100%

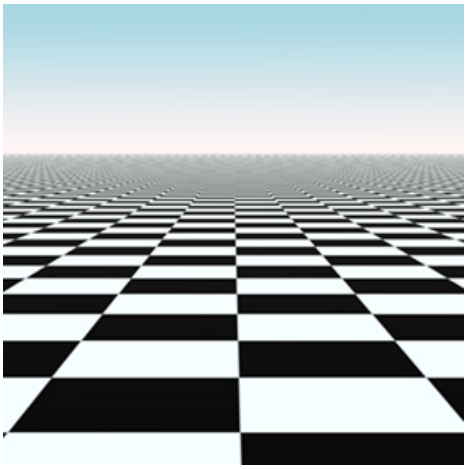
Surface Level: The surface level option determines the height of the limit surface, in-turn controls how smooth or abrupt the transitions are between neighboring Blobs. Low values produce smoother blobs that are slightly smaller, and higher value will produce more spherical shapes that are slightly larger. At 100%, individual blobs will no longer blend and simply look like a collection of spheres.



Surface Level 0% Surface Level 25% Surface Level 50% Surface Level 75% Surface Level 100%

GL Display: This option toggles the display of the 'Blob' proxy in the 3D OpenGL viewport, generally user will want this enabled to get an approximation of how the blob particles will render (the default state), but in heavy scenes it may be useful to disable this option to increase scene interactivity or reduce overall visual clutter.

Ground Plane

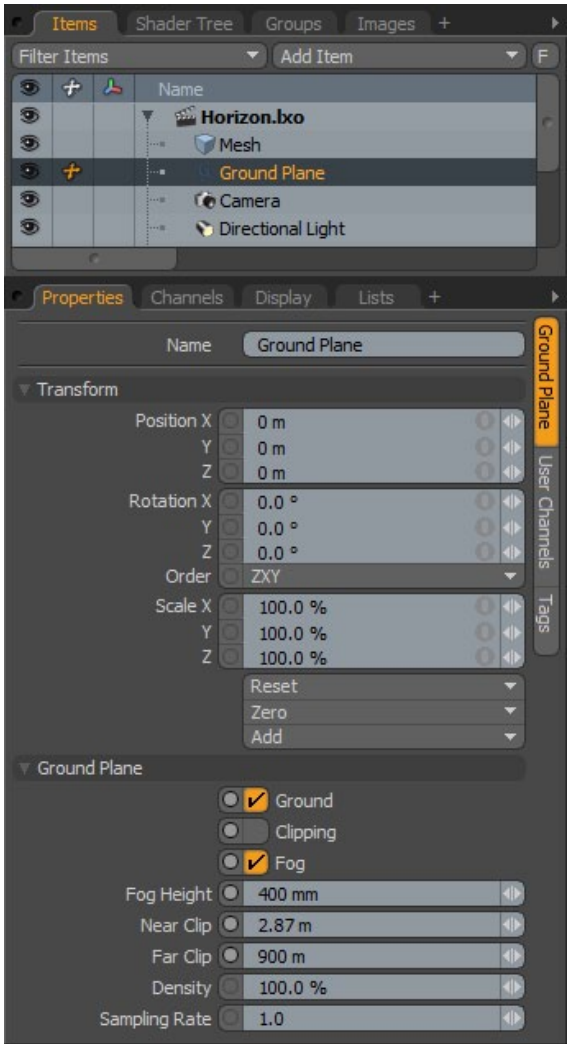


The Ground Plane is a procedurally created item that generates a flat infinite plane that extends in all directions to the horizon. It provides several benefits, first in that it reduces the overall polygon count in a scene, as there is no geometry directly related to the plane, and it also extends all the way to the horizon meaning that users don't need to scale their geometry to crazy high values to get the same result. However since there is no geometry that users can select and tag, users will need to use an 'Item' tag to apply any surfacing to the plane. Users should note that since the Ground Plane is a volumetric effect, it will not render as quickly as geometry would.

Adding a Ground Plane

Users can add a Ground Plane item simply by using the 'Add Item' function of the Items List. LMB+click on the button to open the menu, then select the "Volumes > Ground Plane" option. In order to apply any surfacing to the plane, it will be necessary to create an item mask. This can be done by RMB+clicking on the item layer in the Items List, opening the contextual menu, and then selecting the 'Create Item Mask' option. This creates a 'Material Group' item with the 'Ground Plane' item specified in its Grouping options. Now any textures added into this group will apply exclusively to the Ground Plane (*the sample above has a checkerboard texture applied*). For more information on adding and working with Shader Tree layers, please reference that page of the documentation.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.



Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. Since the plane is infinite, only changes in the 'Y' position will be noticeable.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. The rotation originates form the 'Position' value.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. Since the plane is infinite, scale values will not produce noticeable results.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the current position intact. This is done by adding a secondary transform item with negative values. Users can modify these values directly in the Channels list.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Ground Plane--

Ground: This option toggles the rendering of the Ground Plane itself, *enabled* draws the plane, *disabled* does not.

Clipping: When this option is *enabled*, any geometry above the plane will be clipped (won't render). The clipped areas will take on the surfacing of the plane itself.

Fog: The 'Fog' option renders a Ground Fog effect when enabled.

Fog Height: The 'Fog Height' option determines the maximum level for the Fog effect offset the height distance from the plane. The Fog will be thickest at the ground level, based on the 'Density' value, attenuating up to the 'Fog Height' value.

Near Clip: The 'Near Clip' option determines the closest point to the camera that the fog effect will render.

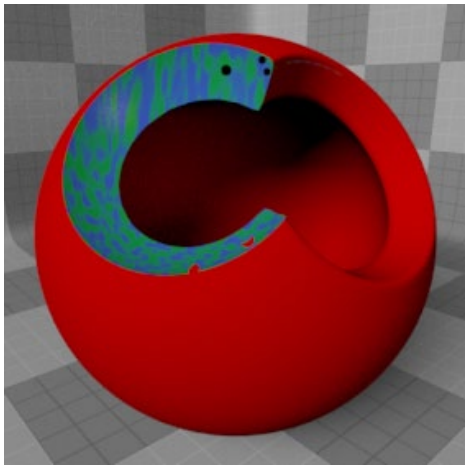
Far Clip: The 'Far Clip' option determines the furthest point from the camera that the fog effect will render.

Density: The 'Density' option determines the overall thickness or opacity of the Fog effect.

Sample Rate: The 'Sample Rate' option determines the quality of the sampling of the Fog effect, high values will produce cleaner results at the expense of longer render times.

TIP: Because of its procedural nature, neither Bump Maps or Displacement Maps will render properly on a Ground Plane. If either of those effects are required, then an actual geometry plane will need to be used.

Render Boolean



Booleans, as they relate to 3D computer graphics, are a technique of using geometry from one mesh item to modify the geometry of another; where they intersect produces a subtractive trimmed surface that would be difficult to create otherwise. The 'Render Boolean' item provides users a means to trim away areas of an object or surface at *render time* without needing to physically modify the mesh itself. The benefits are that the mesh remains fully intact, the results are fully animateable and it produces a texturable surface where the Render Boolean intersects with the trimmed mesh. Also, since the boolean is generated at render time on the final subdivided geometry, there is a less likely chance of producing the errors generally associated with boolean trimming of Subdivision Surfaces.

Adding Render Booleans

A Render Boolean item can be added to a scene by utilizing the 'Add Item' function found in the 'Items List'. Users can LMB+click the button itself, opening the popup menu and select the "Volumes > Render Boolean" option. Once added, the item will appear at the bottom of the Items list viewport. When selected, a number of attributes controlling the Booleans functions appear in the 'Properties' viewport.

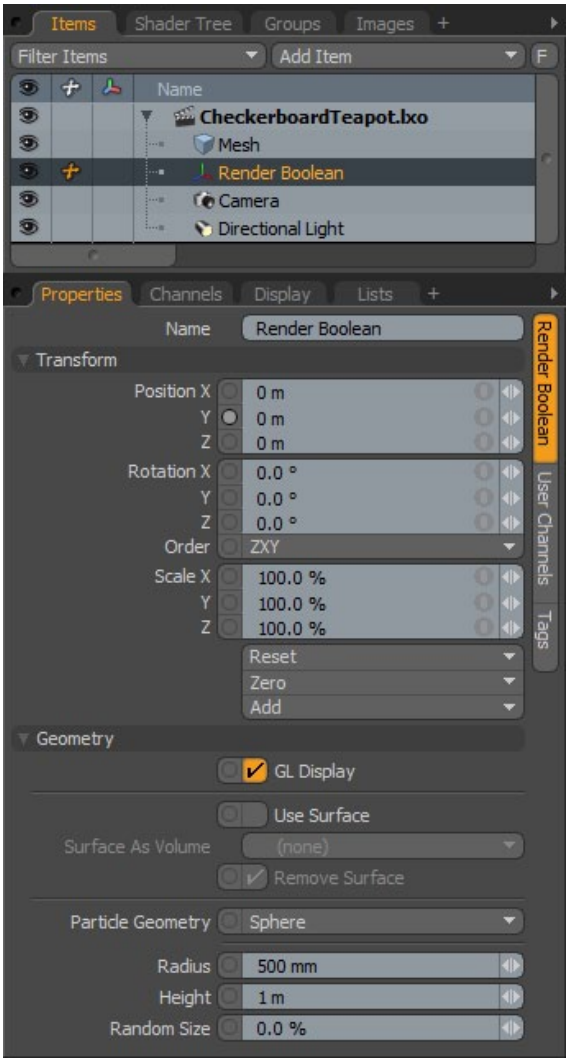
Applying Materials

When utilizing the procedural geometry options of the Render Boolean, users can texture the resulting boolean surface with the use of an item mask. By RMB+clicking on the 'Render Boolean' item in the 'Items List' and selection the 'Create Item Mask' option, a new Material Group is added to the Shader Tree. Users can use the 'Add Layer' option to add 'Materials' and 'Texture' layers that will then be confined only to the areas where the booleans intersect with the target mesh. For more on working

with layers in the 'Shader Tree', please reference that page of the documentation. The 'Render Boolean' item will trim all items/surfaces in the scene unless the user explicitly disables the 'Enable Surface Clipping' option of the Material Item on a per surface basis (This option is enabled by default for all surfaces).

Animated Booleans

Users can animate the Render Boolean operation by creating keyframes on the Render Boolean item when using the procedural shape option, or when a 'Surface as Volume' mesh layer is specified, animating the target mesh layer will produce the animated boolean operation. For more on setting keyframes and animating in modo, please reference the 'Rigging and Animation' section of the documentation.



Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. By default, Position transforms originates from the Center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. By default, Rotation transforms originate from the Center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. By default,

Scale transforms originate from the Center position.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Mesh position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Geometry--

GL Display: This option toggles the display of the Particle Geometry proxy in the 3D OpenGL viewport, generally user will want this *enabled* to get an approximation of how the boolean item will render (the default state), but in certain instances it may be useful to *disable* this option.

Use Surface: When this option is *enabled*, users can select any arbitrary mesh layer from the scene and use it as a boolean volume. While Render Booleans work for nearly any geometry type, for best results both should be fully closed *watertight* models, meaning they could virtually hold water without leaking.

Use Surface As Volume: This option determines the mesh layer that will be used for the boolean operation, only when 'Use Surface' is *enabled*.

Remove Surface: When this option is *enabled* (the default behavior), the specified mesh layer will be hidden at render time, and will only show the resulting intersection of the Boolean volume and the trimmed geometry.

Particle Geometry: The Render Boolean item also generates procedural geometry, users can choose between a 'Cube', a 'Sphere' and a 'Cylinder'.

Radius/Height: Defines the outside dimensions of the Particle Geometry object. 'Radius' refers to the overall dimensions of the 'Sphere' and 'Cube' shapes, and 'Radius' and 'Height' work together to define the dimensions of a 'Cylinder'.

Random Size: Adds random size variations to the particle geometry.

TIP: Render Booleans aren't strictly for trimming geometry, they can also affect volume items such as 'Blobs', 'Sprites' and 'Volumes'.

Sprite

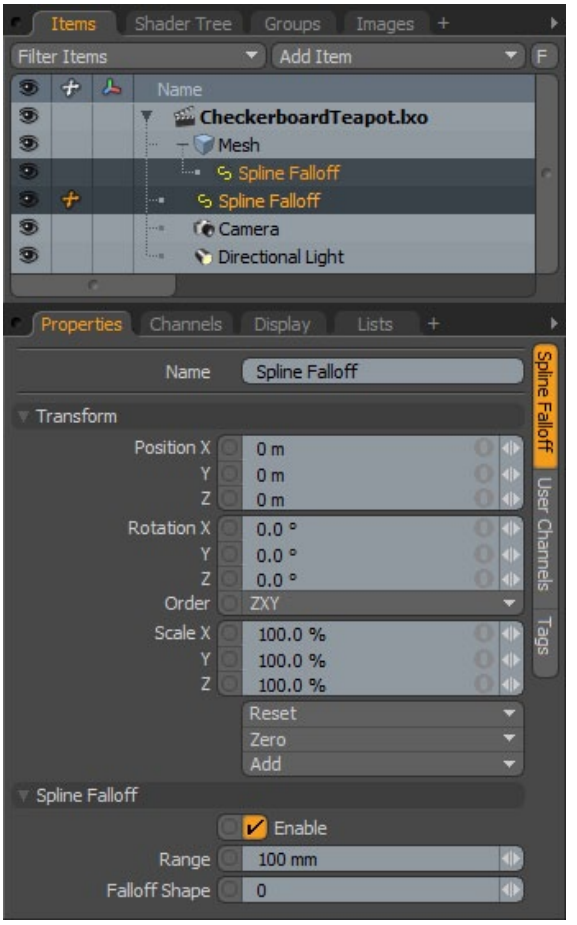
Please refer to the on-line documentation.

Volume

Please refer to the on-line documentation.

Spline Falloff

Like with modeling, a Falloff is used to attenuate the affect any deformer has on a target item, providing additional control how deformers influence a mesh.



Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. Position transforms originates from the Center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. Rotation transforms originate from the Center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. Scale transforms originate from the Center position.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Item position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Spline Falloff--

Enable: Toggles the Falloff on or off. When unchecked (*disabled*), the Falloff has no influence. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Range: Defines the distance away from the Spline that the falloff will influnece its target.

Falloff Shape:

Static Mesh Item

Static Mesh items can streamline workflows where extremely heavy/dense meshes are used. Static meshes, as their name suggests are non editable, essentially frozen objects.

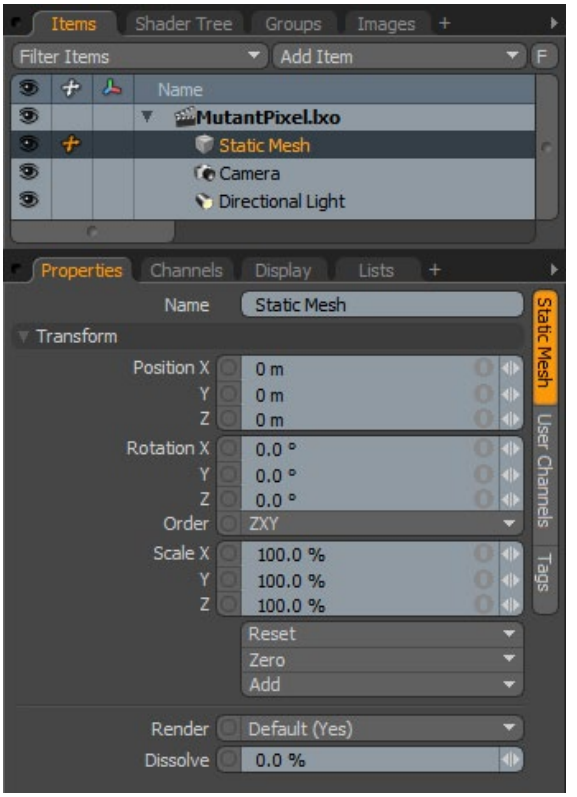
A good amount of memory and processing power is used in modo so that objects can be edited; each and every vertex ready to accept any type of transform or modification. What happens when a mesh gets really dense, this overhead can bog down a scene file; adding a few items like this to a scene will in many cases substantially reduce viewport interactivity. A Static Mesh is a means to solve the issue by removing the overhead associated with editing a mesh, increasing overall performance and minimizing file size. Static meshes render just as any regular mesh would and can be positioned by regular item level transformations and can be used with replicators as well.

Static mesh's are made by converting regular mesh items. To convert, RMB-click over the mesh item layer containing the geometry you wish to convert in the item list, and select 'Change Type > Static Mesh' from the popup menu. Once converted, all component level mesh editing is disabled including morph maps, and deformers (which are removed). Subdivisions are frozen as well, converting any subdivided geometry into triangulated polygons. Basically, when converting to a static mesh, modo is using the freeze command and then triangulating the model. Static mesh items were implemented mainly to facilitate the rendering of high-poly CAD data and dense organic geometry such as trees. For special cases where a mesh uses displacement maps, deformers and such, users may wish to pre-freeze the item prior to converting

to static mesh by using the command on the menu bar 'Geometry > Freeze...' and selecting Freeze Displacement and Freeze Deformation in the popup dialogue.

For further editing, a Static Mesh can be converted back into a editable mesh layer, however, due to the freezing of the mesh's, some information will be lost.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.



Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. By default, Position transforms originates from the Center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. By default, Rotation transforms originate from the Center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. By default, Scale transforms originate from the Center position.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

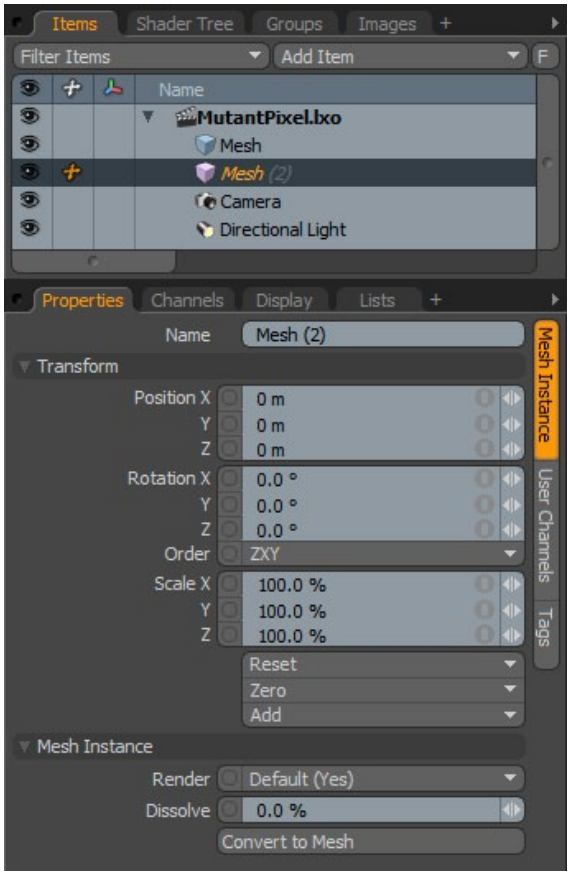
Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Item position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Render: This drop down menu allows the user to select from 3 choices, when set to 'Default', the user can enable/disable mesh item using the visibility column (eyeball icon) of the item list. When the mesh layer is visible, it contributes to the final rendered scene and when invisible, it will not. On some instances the user may prefer to fix this state, setting the mesh as 'On' (enabled) or 'Off' (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving the user from manually enabling mesh layers for test renders.

Dissolve: The 'Dissolve' function controls the visibility of an item layer, at 0% the item renders normally, utilizing the shader tree settings, values above 0% would fade the visibility of the item layer ramping toward 100% where the item would be completely invisible.

Instance Item



Instances are virtual duplicates of a mesh item in modo, meaning they do not add any additional geometry to the scene. Instances are merely containers that reference the original item at render time. As such, Instance Meshes can only be edited at the Item level. Component level edits must be applied to the original item, where changes will propagate to all instances.

Instances can be created in several ways. When in 'Items' mode, many of the duplication tools allow for instance creation. The simplest method is to RMB+click on a Mesh Item in the Item list and choose 'Instance' from the drop down menu, creating a new Instance Item at the bottom of the items list. The link between Instances and the original always remains live. If you wish to convert an Instance Mesh into a standard Mesh Item (removing the link between the Instance and source) you can RMB+click on the

Instance in Item List and choose Change Item "Type> Mesh" or simply use the 'Convert to Mesh' button in the items properties.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. By default, Position transforms originates from the Center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. By default, Rotation transforms originate from the Center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. By default, Scale transforms originate from the Center position.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Item position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Mesh Instance--

Render: This drop down menu allows the user to select from 3 choices, when set to 'Default', the user can enable/disable mesh item using the visibility column (eyeball icon) of the item list. When the mesh layer is visible, it contributes to the final rendered scene and when invisible, it will not. On some instances the user may prefer to fix this state, setting the mesh as 'On' (enabled) or 'Off' (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving the user from manually enabling mesh layers for test renders.

Dissolve: The 'Dissolve' function controls the visibility of an item layer, at 0% the item renders normally, utilizing the shader tree settings, values above 0% would fade the visibility of the item layer ramping toward 100% where the item would be completely invisible.

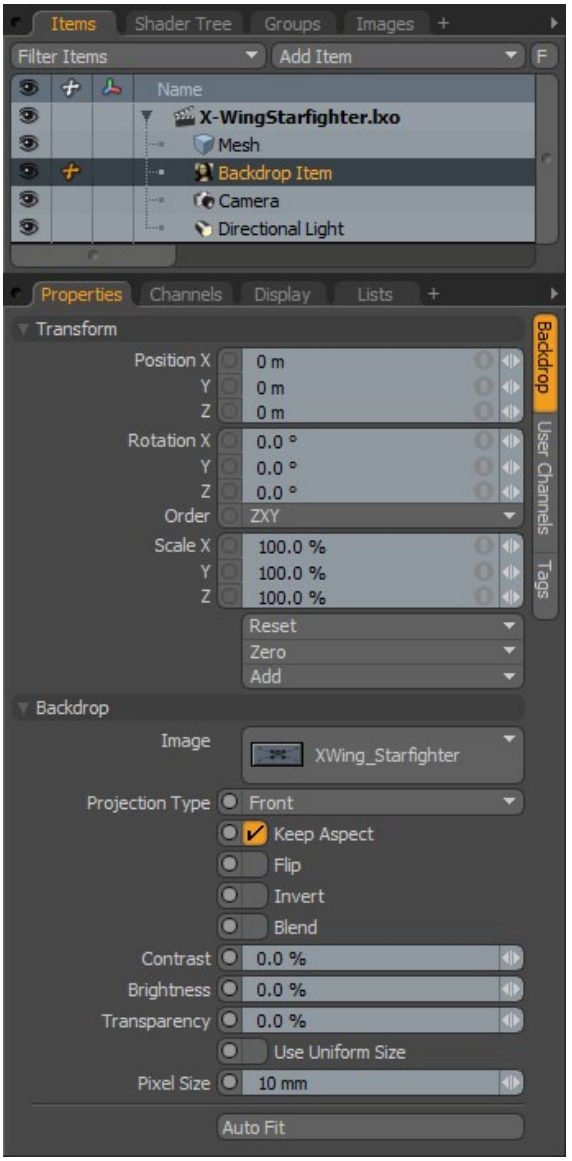
Convert to Mesh: By pressing this button, users can convert an mesh Instance into a fully editable Mesh, removing the link between the instance and the source. Keep in mind that doing so will also increase file size.

TIP: While Instance Items derive all mesh information from the original, users can create custom material and texture settings for an instanced item utilizing 'item masks', which can in turn be specified in the material group item.

Backdrop Item

Backdrop items are terrific modeling aides allowing users to add images into a background plane that acts as reference for geometry creation. This can be a photograph, a line drawing or even blueprints. As Backdrop Items are only meant for reference they will not contribute to rendered output. Best of all, since they are items in the item list, they are saved with the scene and can be manipulated with the standard transform tools.

To add a background image users can use the menu bar command "Item > Create Backdrop" which creates a new Background item layer in the Item List. Users can then select the newly created layer and further define the backdrop settings within its associated properties panel, where users set the image, orientation and adjust its image properties (detailed below). Once users have defined a number of different views, it is handy to group them together using the 'Ctrl+G' keyboard shortcut providing users the ability to toggle the all of the Backdrop images visibility with a single action.



Backdrop Item

Transform--

Position: The 'Position' setting allows the user to specify a numeric XYZ values for the placement of the backdrop image within the viewport. All transforms originate from the planes center point.

Rotation: With 'Rotation' you can adjust a backdrop items rotation in the viewport, however in most instances, it would be best to simply adjust the 'Projection Type' to position the image toward a specific viewport.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: The 'Scale' function allows the user to adjust the size of the backdrop items.

Reset: Resets the selected transform values to (0,0,0) returning the items back to the world space center position.

Zero: Returns the items Center position to the world space center (0,0,0) without changing the position of the backdrop item itself.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms. Since Backdrops don't render, this setting is unnecessary.

Image: Click in the image drop down to specify an image or use the load command to load an image from disc.

Projection Type: The 'Projection Type' setting allows the user to specify which viewport the backdrop image will face toward as well as a 'Camera' projection mode that appears automatically in the background of any Viewport set to 'Camera' as the type. For multiple viewports, such as front, side and back, three different backdrop items would need to be created with the appropriate image and projection type specified for each.

Keep Aspect: The 'Keep Aspect' function will retain the aspect ratio of an image when it is scaled.

Flip: The 'Flip' command will mirror the images pixels across the vertical axis, useful for using the same image for front and back pairs or left and right pairs.

Invert: The 'Invert' toggle will invert the RGB values for the backdrop image, useful when inverting a black and white line drawing to make the lines more visible.

Blend: The 'Blend' function will produce a small amount of smoothing between neighboring pixels, reducing the jagged appearance of low resolution images.

Contrast: Contrast is the apparent visual different between light and dark values in an image. Values greater than 0% will increase the contrast of the backdrop image, while values lower than 0% will decrease it.

Brightness: Brightness is the degree or amount of light or luminosity in an image. Values greater than 0% will increase the brightness of the backdrop image, while values lower than 0% will decrease it.

Transparency: Transparency determines the opacity of the backdrop image. Values greater than 0% will increase the transparency, ramping toward 100% where the image would become completely invisible.

Auto Fit: The 'Auto Fit' command will automatically scale the Backdrop item to match the height of the currently active mesh items, or the height and width when 'Keep Aspect' is disabled. However, this will only work when 'Display Backdrop Items in Uniform Size' is enabled in the preferences file. To find it, click in the menu bar "System > Preferences' and then in the preferences window, select the 'Display > OpenGL' section, backdrop items are listed toward the bottom.

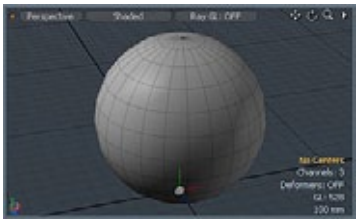
TIP: Within modo's 'Preferences', there are a few options related to Backdrop items and how modo scales images when placing them. For more information, please reference the appropriate Preferences page of the documentation.

Centers and Pivots

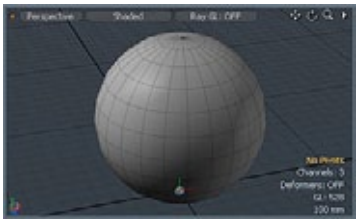
The modo universe, that is the virtual representation of three-dimensional space within modo is fixed. Users can modify the units that represent scale, and change what axis represents the 'Up' direction, but otherwise the world is a set known quantity. Everything that happens in modo, modeling, animation or otherwise happens within these fixed coordinates. More importantly, every coordinate values is calculated as an offset from the world origin; the position where all axes values are 0. When working with any item in modo, such as a Mesh layer, a Camera or a Light, modo provides two elements for determining the origin of transformation operations. This occasionally causes some confusion. In order to help users understand their use, definitions are in order-

Center: This is the origin of the mesh *item* layer, regardless of the position of any geometry within the layer. It is the location that sits at the world origin when all the item transforms are at their default values, i.e. when the item has not been moved, rotated or scaled. It is also the default origin from where all item transformations occur, especially Rotate and Scale.

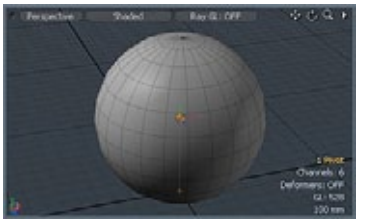
Pivot: This is an offset from the item's Center, when a pivot is specified the rotation and scaling originate from this location instead of the item's Center. Pivots are usually used when users need to animate the location around which an item rotates etc (a rolling cube for example) or when users need the rotation point to be in a different location from the item's center for any reason (such as when using Weight Deformers).



Center element (at origin)



Pivot element



Pivot Offset (note the line to the 'Center')

Usage--

In 'Center' or 'Pivot' selection modes, users first select the respective element to work with. Then applying any of the normal transform tools to edit their location. Since they use the regular transforms, they will also automatically support all the 'Action

Centers', 'Axis' and 'Snapping' functionality. It should also be noted that editing Center positions is only applicable when working with Mesh layers, moving the Center for any other element, such as a Camera or a Light actually moves the element itself, so the Pivot would need to be edited instead to change the point of origin. Constraints default to using an Items Center, so if the Pivot is offset, make sure to set the Constraint to explicitly use the Pivot.

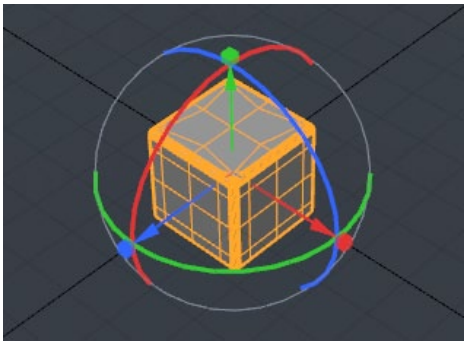
TIP: Within modo's 'Preferences', there are a few options related to Backdrop items and how modo scales images when placing them. For more information, please reference the appropriate Preferences page of the documentation.

Transform Item Tools

The text here needs to be updated to reflect the tools presented here. For information on using a specific tool, click on any of the link options below to be directed to the proper page.

Transform		Clone	Array
Transform	Axis Rotate	Spline Falloff	Array
Move	Scale	Scatter Clone	Instance Item
Rotate	Uniform Scale	Curve Clone	

Transform



The Transform tool combines Move, Rotate and Scale transformations into a single tool. By using the various handles the user can quickly Position, Rotate and Scale from one convenient tool. Holding the 'Ctrl' key will constrain the position transform to a single axis when hauling, and the rotation transform to 15° increments. After applying a transform amount, users may press the MMB to re-apply the previous offset amount, making it easy to apply an edit multiple times. The command can be activated in the 'Basic' subtab of the Modeling viewports, or by pressing the 'Y' keyboard shortcut.

In the component modes, 'Vertices', 'Edges', and 'Polygons', the tool affects the selected geometry directly; when nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected (*highlighted*) in the 'Item List'. In 'Items' mode the tool affects all the currently selected items in the Item List, if nothing is selected, then the tools will appear to do nothing. In material mode, it works the same way as in any of the component modes. Lastly, in the 'Center and 'Pivot' modes it modifies the Center or Pivots respectively. Note that users will need to physically *select* either the Center or the Pivot itself, depending on the mode, prior to activating the tool, as in Items mode, with no selection the tool will appear to do nothing.

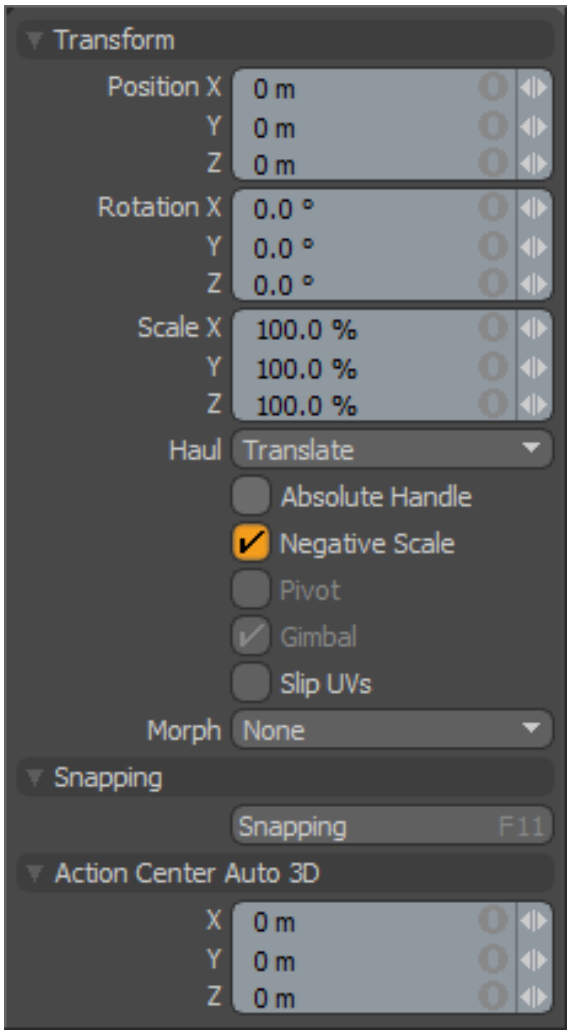
When users define any of the various Falloffs modo provides, application of the tool can be attenuated across a user defined area. For more on using Falloffs, please reference the Selection Falloffs page of the documentation.

Once the tool becomes active, modo draws the relevant tool handles within the viewport- arrows for moving, boxes for scaling and circles for *rotating*. At this point, users may simply click and haul anywhere inside the desired viewport to interactively transform the geometry. Additionally, users may prefer to utilize the various tool handles, constraining the application of the function to one or two axes. The various tool handles are color coded to match the coloring of the different axes, red for X, green for Y and blue for Z. Hover the mouse over a specific handle until it changes to

a yellow color and then LMB-click and drag to modify. To interactively scale the size of the handles themselves, users may use the ' - ' and ' = ' keys on the keyboard to adjust them; minus makes them smaller, while equals makes them larger.

The Transform tool also respect the settings for 'Action Center', and will originate the appropriate action from the defined position. For more information on understanding and using Actions Centers, please reference the Action Center page of the documentation

Transform--



Position: Users may use the X,Y and Z input fields to apply specific offset distance values (calculated from position the object was in when the tool was activated). When adjusting object interactively in the viewport, the 'Position' input fields will display the current offset distance applied, providing useful user feedback.

Rotation: Users may use the X,Y and Z input fields to apply specific rotation values. When adjusting object interactively in the viewport, the 'Rotation' input fields will display the current offset distance applied, providing useful user feedback.

Scale: Users may used the X,Y and Z input fields to apply specific scaling values. When adjusting object interactively in the viewport, the 'Scale' input fields will display the current scale factor applied, providing useful user feedback.

Haul: Users have three options available when hauling in the viewport (i.e. dragging the mouse when not using the tool handles), 'Translate' will allow the users to move the selection, 'Rotate' allows the user to rotate, and 'Scale' would scale the selection.

Absolute Handle: The 'Absolute Handle' toggle applies an additional function to the tool handles allowing users to scale the handle itself without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.

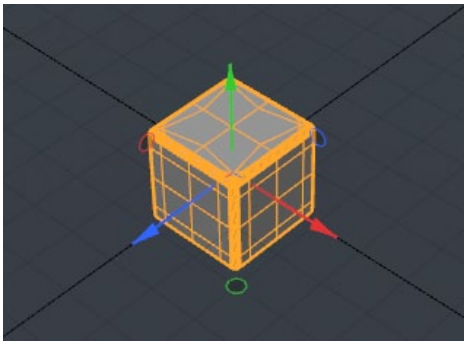
Negative Scale: The 'Negative Scale' toggle enabled/disabled modo's ability to interactively apply negative scale values when manipulating the tool handles in the viewport. When disabled, users can always input negative values in any of the input fields directly.

Pivot: The 'Pivot' toggle allows users to easily modify an items pivot position without the need to go into Pivots mode.

Gimbal: With the 'Gimbal' toggle enabled, users can assign rotation values numerically, but may arrive at a situation that produces gimbal lock, the loss of one rotation axis when another's value is too close. Users can avoid this by disabling this setting, but are then unable to apply specific values within the input fields.

Child Compensate: The 'Child Compensate' toggle when disabled will pass along the transform value to any child items of the current selection, when enabled, the child items will ignore any transform values.

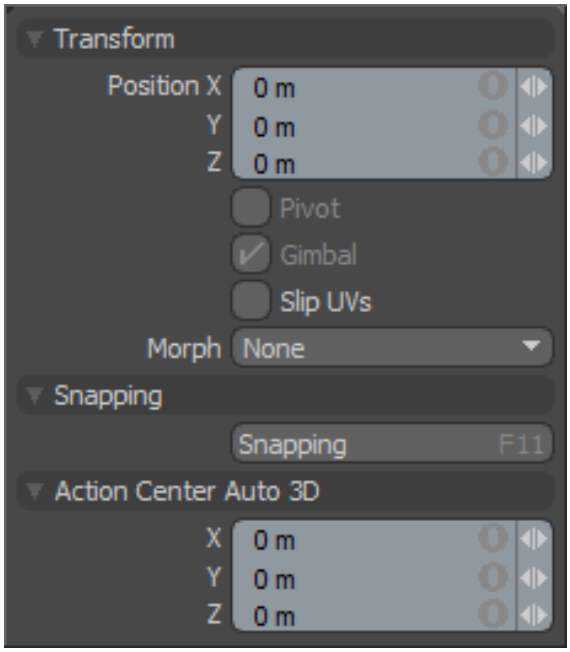
Move



The Move tool obviously allows users to move, or *translate* a selection across a specific distance. By using the various handles the user can easily constrain the movement to one or two axes. Holding the 'Ctrl' key will constrain to a single axis when hauling. After applying a transform amount, users may press the MMB to re-apply the previous offset amount, making it easy to apply an edit multiple times. The command can be activated in the 'Basic' subtab of the Modeling viewports, in the menu bar under 'Edit >Transform > Move' or by pressing the 'W' keyboard shortcut.

In the component modes, 'Vertices', 'Edges', and 'Polygons', the tool affects the selected geometry directly; when nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected (*highlighted*) in the 'Item List'. In 'Items' mode the tool affects all the currently selected items in the Item List, if nothing is selected, then the tools will appear to do nothing. In material mode, it works the same way as in any of the component modes. Lastly, in the 'Center and 'Pivot' modes it modifies the Center or Pivots respectively. Note that users will need to physically *select* either the Center or the Pivot itself, depending on the mode, prior to activating the tool, as in Items mode, with no selection the tool will appear to do nothing.

When users define any of the various Falloffs modo provides, application of the tool can be attenuated across a user defined area. For more on using Falloffs, please reference the Selection Falloffs page of the documentation.



Once the tool becomes active, modo draws the relevant tool handles within the viewport. At this point, users may simply click and haul anywhere inside the desired viewport to interactively position the geometry. Additionally, users may prefer to utilize the various tool handles, constraining the application of the tool, the arrows to a single axis, and the circles to to two axes. The various tool handles are color coded to match the coloring of the different axes, red for X, green for Y and blue for Z. Hover the mouse over a specific handle until it changes to a yellow color and then LMB-click and drag to modify. To interactively scale the size of the handles themselves, users may use the ' - ' and ' = ' keys on the keyboard to adjust them; minus makes them smaller, while equals makes them larger.

The Transform tool also respect the settings for 'Action Center', and will originate the appropriate action from the defined position. For more information on understanding and using Actions Centers, please reference the Action Center page of the documentation

Transform

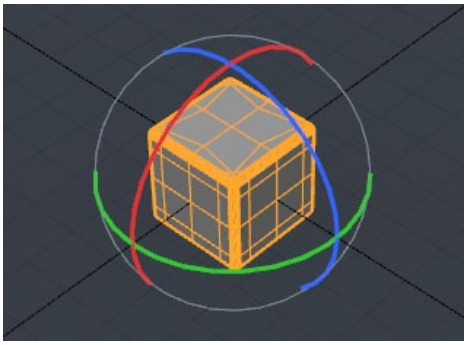
Position: Users may use the X,Y and Z input fields to apply specific offset distance values (calculated from position the object was in when the tool was activated). When adjusting object interactively in the viewport, the 'Position' input fields will display the current offset distance applied, providing useful user feedback.

Pivot: The 'Pivot' toggle allows users to easily modify an items pivot position without the need to go into Pivots mode.

Gimbal: The Gimbal setting is grayed out in move mode.

Child Compensate: The 'Child Compensate' toggle when disabled will pass along the transform value to any child items of the current selection, when enabled, the child items will ignore any transform values.

Rotate



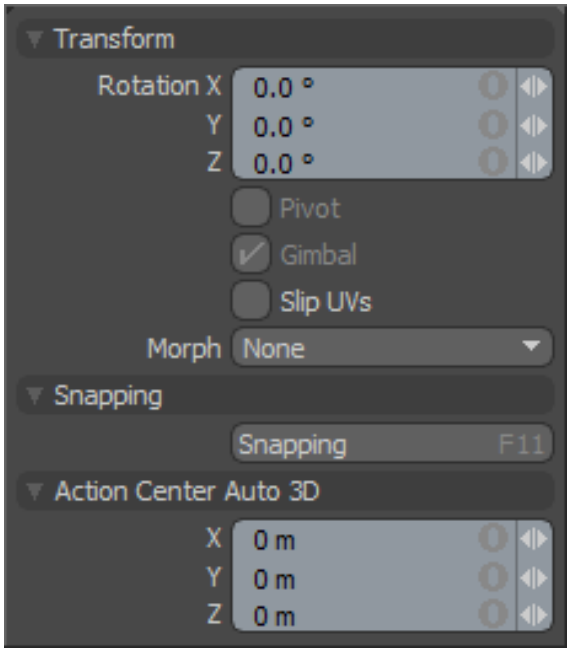
The Rotate tool obviously allows users to rotate a selection by a specific number of degrees. By using the various handles the user can easily constrain the movement to a single axis. Holding the 'Ctrl' key will constrain the the rotation to fixed 15° intervals. After applying a transform amount, users may press the MMB to re-apply the previous offset amount, making it easy to apply an edit multiple times. The command can be activated in the 'Basic' subtab of the Modeling viewports, in the menu bar under 'Edit >Transform > Rotate' or by pressing the 'E' keyboard shortcut.

In the component modes, 'Vetices', 'Edges', and 'Polygons', the tool affects the selected geometry directly; when nothing is selected, it affects all active forground geometry. Active foreground items are those that are visible and selected (*highlighted*) in the 'Item List'. In 'Items' mode the tool affects all the currently selected items in the Item List, if nothing is selected, then the tools will apppear to do nothing. In material mode, it works the same way as in any of the component modes. Lastly, in the 'Center and

'Pivot' modes it modifies the Center or Pivots respectively. Note that users will need to physically *select* either the Center or the Pivot itself, depending on the mode, prior to activating the tool, as in Items mode, with no selection the tool will appear to do nothing.

When users define any of the various Falloffs modo provides, application of the tool can be attenuated across a user defined area. For more on using Falloffs, please reference the Selection Falloffs page of the doumentation.

Once the tool becomes active, modo draws the relevant tool handles within the viewport. At this point, users may simply click and haul anywhere inside the desired viewport to interactively rotate the geometry. Additionally, users may prefer to utilize the various tool handles, constraining the application of the tool to a single axis. The tool handles are color coded to match the coloring of the different axes, red for X, green for Y and blue for Z. In some action center modes, there is an additional gray *screen space* handle around the outside of the tool handles, which rotates the selection based on the viewport cameras postion. Hover the mouse over a specific handle until it changes to a yellow color and then LMB-click and drag to modify. To interactively scale the size of the handles themselves, users may use the ' - ' and ' = ' keys on the keyboard to adjust them; minus makes them smaller, while equals makes them larger.



The Rotate tool also respect the settings for 'Action Center', and will originate theaction from the defined position. For more information on understanding and using Actions Centers, please reference the Action Center page of the documentation.

Transform

Rotation: Users may use the X,Y and Z input fields to apply specific rotation values. When adjusting object interactively in the viewport, the 'Rotation' input fields will display the current offset distance applied, providing useful user feedback.

Pivot: The 'Pivot' toggle allows users to easily modify an items pivot position without the need to go into Pivots mode.

Gimbal: With the 'Gimbal' toggle enabled, users can assign rotation values numerically, but may arrive at a situation that produces gimbal lock, the loss of one rotation axis when another's value is too close. Users can avoid this by disabling this setting, but are then unable to apply specific values within the input fields.

Child Compensate: The 'Child Compensate' toggle when disabled will pass along the transform value to any child items of the current selection, when enabled, the child items will ignore any transform values.

Axis Rotate

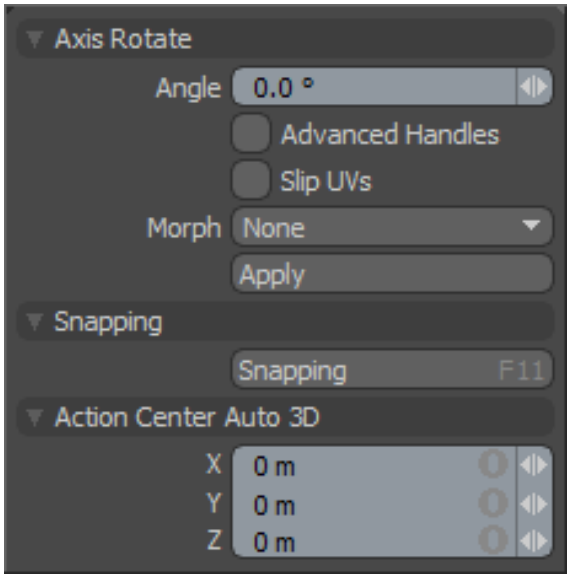
The Axis Rotate tool obviously allows users to rotate a selection by a specific number of degrees. The Axis Rotation tool is different from the standard Rotate tool in that it is limited to editing a single axis at a time, determined by the Work Plane position. It also has a different on-screen control display than the other transform tools providing a greater degree of control over the amount of rotation depending on how long or short the user drags out the handle length. Holding the 'Ctrl' key will constrain the the rotation to fixed 15° intervals. After applying a transform amount, users may press the MMB to re-apply the previous offset amount, making it easy to apply an edit multiple times. The command can be activated in the 'Basic' subtab of the Modeling viewports, users may need to press the 'Shift' key to show alternate tool. User may also activate the tool from the menu bar under 'Edit >Transform > Axis Rotate' or by pressing the 'Alt+Shift+E' keyboard shortcut.

In the component modes, 'Vertices', 'Edges', and 'Polygons', the tool affects the selected geometry directly; when nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected (*highlighted*) in the 'Item List'. In 'Items' mode the tool affects all the currently selected items in the Item List, if nothing is selected, then the tools will appear to do nothing. In material mode, it works the same way as in any of the component modes. Lastly, in the 'Center and 'Pivot' modes it modifies the Center or Pivots respectively. Note that users will need to physically *select* either the Center or the Pivot itself, depending on the mode,

prior to activating the tool, as in Items mode, with no selection the tool will appear to do nothing.

When users define any of the various Falloffs modo provides, application of the tool can be attenuated across a user defined area. For more on using Falloffs, please reference the Selection Falloffs page of the documentation.

Once the tool becomes active, modo draws the relevant tool handle within the viewport. At this point, users may simply click and haul anywhere inside the desired viewport to interactively rotate the geometry, depending on the action center, this may also set the rotation center based on the mouse click position intersecting the Work Plane. Additionally, users may prefer to utilize the on screen display, either clicking on the wheel to rotate to a specific amount, or by dragging the handles on the value bar across the top of the viewport (when 'Advanced Handles' is enabled). When clicking on the wheel, a handle is drawn displaying the number of degrees. When using this handle the rotation sensitivity is adjusted based on the distance the mouse is from the handles as you drag. This can be thought of as extending a lever. The longer the lever, the more mouse movement is required to rotate the element and thus the granularity of control is increased. To interactively scale the size of the handles themselves, users may use the ' - ' and ' = ' keys on the keyboard to adjust them; minus makes them smaller, while equals makes them larger.



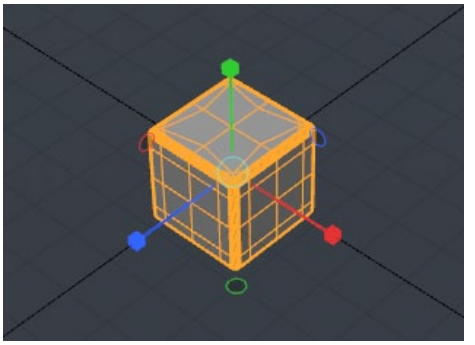
The Rotate tool also respect the settings for 'Action Center', and will originate the action from the defined position. For more information on understanding and using Actions Centers, please reference the Action Center page of the documentation.

Axis Rotate

Angle: Users may use the 'Angle' input field to apply specific rotational value, calculated in degrees. When adjusting object interactively in the viewport, the 'Angle' input fields will display the current rotation amount, providing useful user feedback.

Advanced Handles: Determines whether or not modo displays the additional information in the viewport.

Scale



The Scale tool allows users to make any selection larger or smaller by changing the scale percentage factor. By using the various handles the user can quickly scale in a single axis dimension. After applying a transform amount, users may press the MMB to re-apply the previous offset amount, making it easy to apply an edit multiple times. The command can be activated in the 'Basic' subtab of the Modeling viewports, in the menu bar under 'Edit >Transform > Scale' or by pressing the 'R' keyboard shortcut.

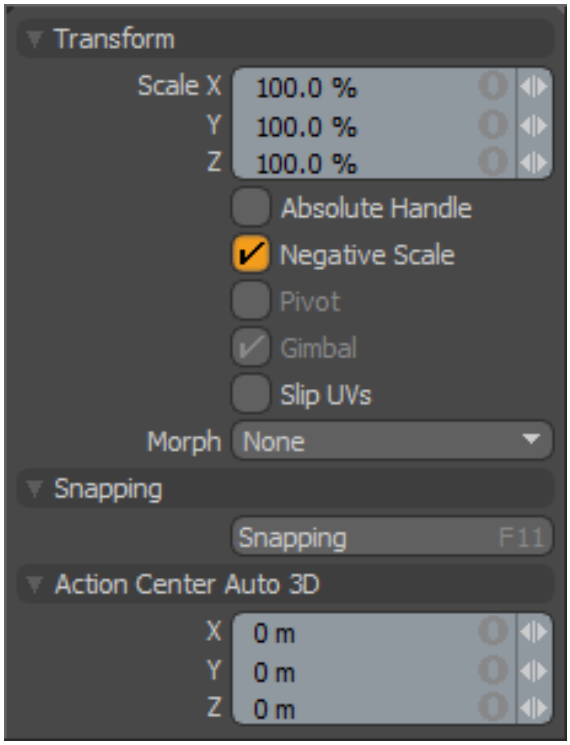
In the component modes, 'Vertices', 'Edges', and 'Polygons', the tool affects the selected geometry directly; when nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected (*highlighted*) in the 'Item List'. In 'Items' mode the tool affects all the currently selected items in the Item List, if nothing is selected, then the tools will appear to do nothing. In material mode, it works the same way as in any of the component modes. Lastly, in the 'Center and 'Pivot' modes it modifies the Center or Pivots respectively. Note that users will need to physically *select* either the Center or the Pivot itself, depending on the mode, prior to activating the tool, as in Items mode, with no selection the tool will appear to do nothing.

When users define any of the various Falloffs modo provides, application of the tool can be attenuated across a user defined area. For more on using Falloffs, please reference the Selection Falloffs page of the documentation.

Once the tool becomes active, modo draws the relevant tool handles within the viewport. At this point, users may simply click and haul anywhere inside the desired viewport to interactively transform the geometry. Additionally, users may prefer to utilize the various tool handles, constraining the application of the function to one or all three axes. The various tool handles are color coded to match the coloring of the different axes, red for X, green for Y and blue for Z. Hover the mouse over a specific handle until it changes to a yellow color and then LMB-click and drag to modify. To

interactively scale the size of the handles themselves, users may use the ' - ' and ' = ' keys on the keyboard to adjust them; minus makes them smaller, while equals makes them larger. Additionally for the Scale tool, there is a small circle at the intersection of all three handles that users can use to uniformly scale the selection interactively.

The Transform tool also respect the settings for 'Action Center', and will originate the action from the defined position. For more information on understanding and using Actions Centers, please reference the Action Center page of the documentation.



Transform

Scale: Users may used the X,Y and Z input fields to apply specific scaling values. When adjusting object interactively in the viewport, the 'Scale' input fields will display the current scale factor applied, providing useful user feedback.

Absolute Handle: The 'Absolute Handle' toggle applies an additional function to the tool handles allowing users to scale the handle itself without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.

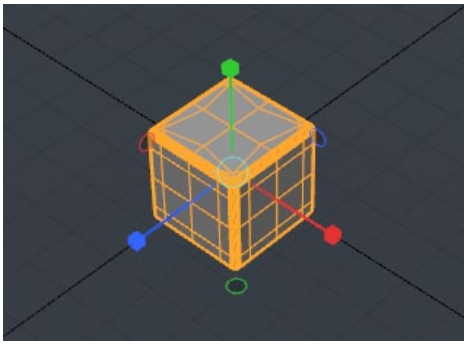
Negative Scale: The 'Negative Scale' toggle enabled/disabled modo's ability to interactively apply negative scale values when manipulating the tool handles in the viewport. When disabled, users can always input negative values in any of the input fields directly.

Pivot: The 'Pivot' toggle allows users to easily modify an items pivot position without the need to go into Pivots mode.

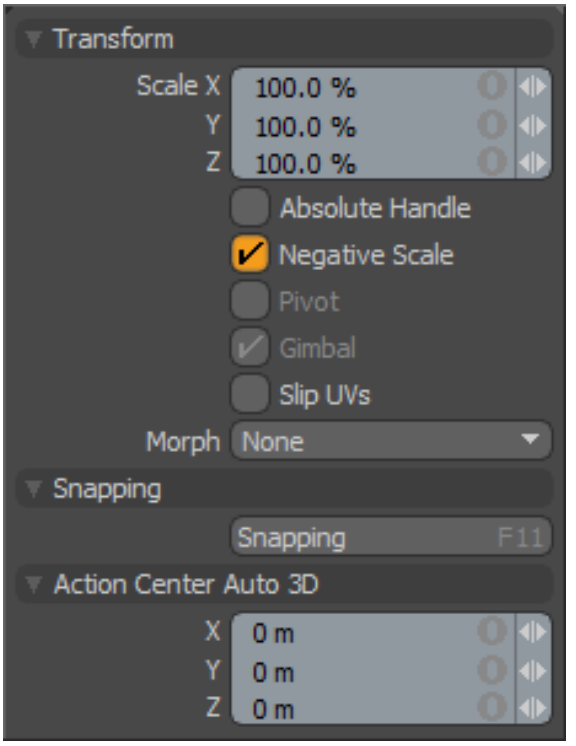
Gimbal: The Gimbal setting is grayed out in scale mode.

Child Compensate: The 'Child Compensate' toggle when disabled will pass along the transform value to any child items of the current selection, when enabled, the child items will ignore any transform values.

Uniform Scale



The Uniform Scale tool is identical to the normal Scale tool (above), except that users may haul anywhere in the viewport and all three axes are constrained forcing all scaling operating to be uniform. This ensures the mesh will not become distorted when scaling.



Transform

Scale: Users may used the X,Y and Z input fields to apply specific scaling values. When adjusting object interactively in the viewport, the 'Scale' input fields will display the current scale factor applied, providing useful user feedback.

Absolute Handle: The 'Absolute Handle' toggle applies an additional function to the tool handles allowing users to scale the handle itself without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.

Negative Scale: The 'Negative Scale' toggle enabled/disabled modo's ability to interactively apply negative scale values when manipulating the tool handles in the viewport. When disabled, users can always input negative values in any of the input fields directly.

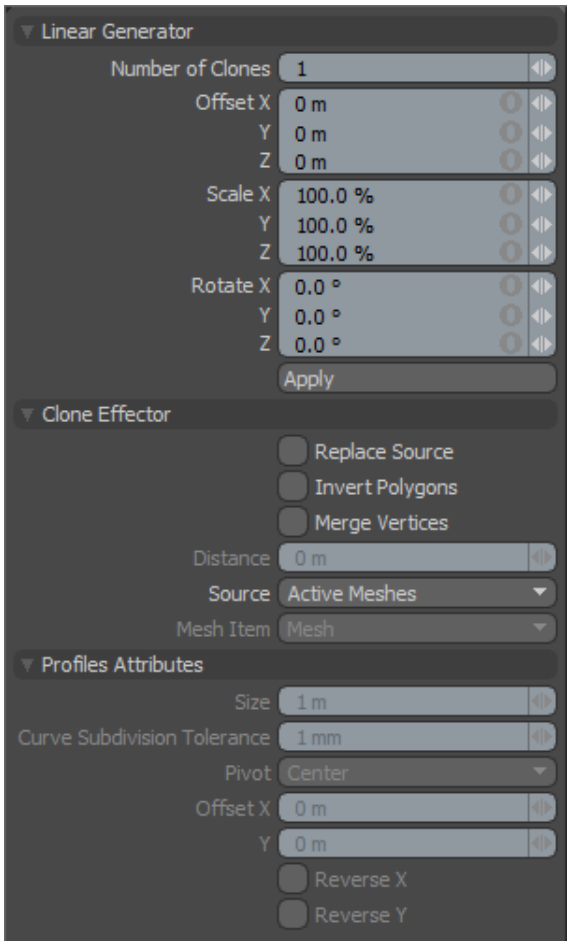
Pivot: The 'Pivot' toggle allows users to easily modify an items pivot position without the need to go into Pivots mode.

Gimbal: The Gimbal setting is grayed out in *scale* mode.

Child Compensate: The 'Child Compensate' toggle when disabled will pass along the transform value to any child items of the current selection, when enabled, the child items will ignore any transform values.

Clone Items

The Clone tool will create duplicates of the currently selected geometry in a linear offset determined by user input. Clone can be accessed on the Duplicate tab of the modo Tools toolbar, or by choosing Geometry > Create > Clone. The tool is controlled by the following settings.



Clone--

Number of Clones: This numeric value determines the number of copies of the original geometry to be created.

Offset XYZ: These distance values determine how far each clone is from the previous. Setting this value to 1 meter would place each successive cloned object 1 meter away from the previous clone or the original geometry in the case of the first clone.

Scale XYZ: These three percentage settings control the relative size of each clone to the one before it. Setting the value to 50% would force each clone to be half the size of the previous.

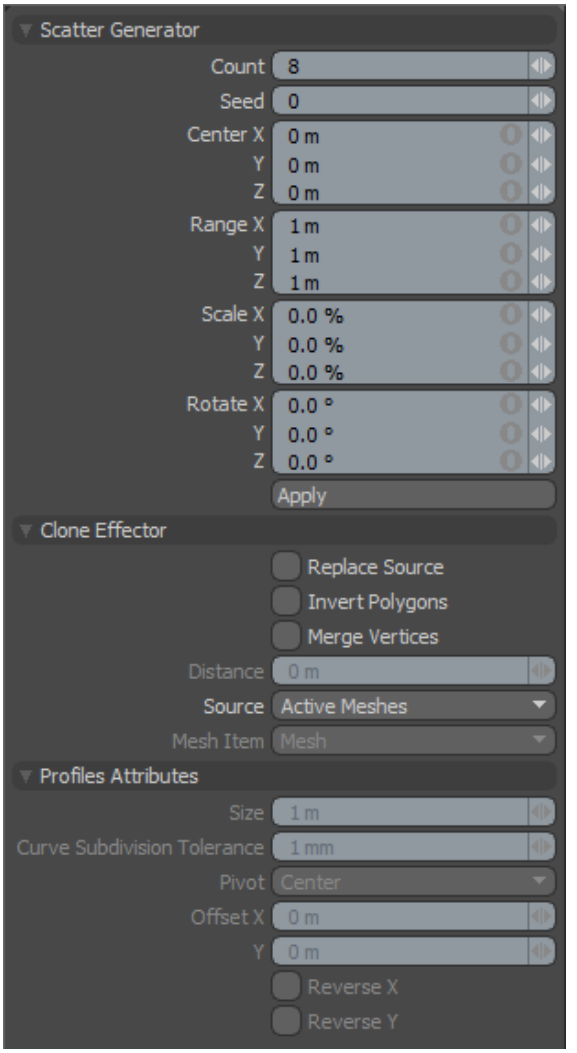
Rotate XYZ: The rotate control allows you to set a rotational value in degrees for each of the three axes. This rotation amount will

be applied to each clone. This can be very useful for creating objects such as spiral staircases etc.

Replace Source: This toggle option will cause the tool to delete and replace the original source geometry. This can be very useful when using scale and rotation values.

Scatter Clone Items

The Scatter Clone tool generates a random point cloud onto which clones of the original geometry are placed.



Scatter Clone--

Count: This numeric value determines the total number of clones (including the source).

Center XYZ: This triplet determines the center position of the scatter clone effect.

Range XYZ: This triplet distance value determines the maximum distance on each axis the clones can be from the center of the effect.

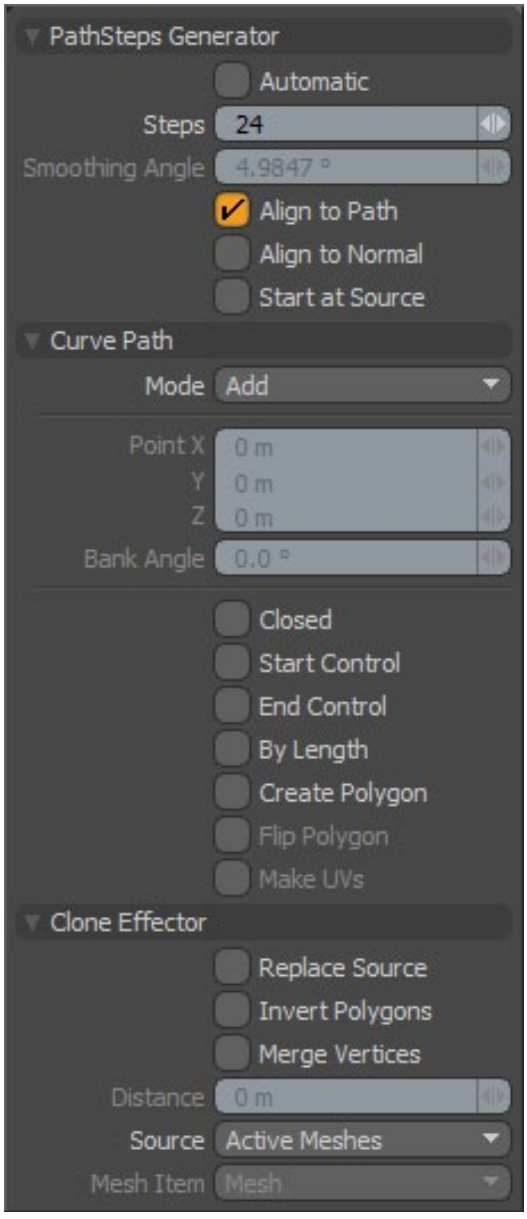
Scale XYZ: This triple percentage value determines the maximum amount of randomized fluctuation for any given clone.

Rotate XYZ: This triple degree value determines the maximum amount of randomized rotation for any given clone.

Replace Source: When activated, this control will remove the original geometry and create a replacement clone for it at the center of the scatter clone effect.

Curve Clone Items

The Curve Clone tool allows you to quickly clone geometry by clicking down control points for a curve. If you wish to begin with a pre-created curve, simply put the curve in visible but deselected mesh item prior to activating the tool. Once the curve has been created with the tool active the clone effector will create geometry along it using the various controls.



Path Steps Generator--

Automatic: This toggle control will force a computed number of clones based on the Smoothing Angle value such that a "particle" is placed along the path to maintain a level of smoothness at or above the user set level.

Steps: Active only when Automatic is off this is the number of particles generated along the curve path to feed into the Effector.

Align: When enabled the generated particles will include rotation data so the newly created mesh components can align to the surrounding geometry.

Start at Source: Often a curve used for cloning will have its initial control knot imperfectly aligned with the geometry to be cloned. WIth Start at Source active the curve is used as a relative guide using the original geometry as the true starting location. Disabling this option may cause the newly created geometry to move away from the intended original location.

Curve Path--

Mode: By default the mode is set to Add. In the Add mode clicking on a curve point and dragging will edit the

curves position. Clicking anywhere in the 3D viewport other than the handles will create a new curve control point. To avoid creating new control points when dragging existing points around you can set this mode to Edit. Delete mode allows quick removal of control points by simply clicking on them.

Closed: This toggle option will cause the curve to become a closed loop.

Start Control: Enabling this toggle will cause the first knot in the curve to become a control point but not an actual segment of the curve. This provides a simple method for changing the curvature at the start of the curve.

End Control: Enabling this toggle will cause the last knot in the curve to become a control point but not an actual segment of the curve. This provides a simple method for changing the curvature at the end of the curve.

By Length: This option forces the newly created geometry to be evenly distributed across the entire length of the curve rather than spacing based on the control knots.

Create Polygon: This toggle option will generate a curve type polygon that remains after the tool has been dropped. This is useful if you want to re-use the curve you create with the current tool with additional curve based tools later in your project.

Flip Polygon: This toggle will invert the faces of the generated polygons.

Make UVs: When this toggle is active the tool will populate the current UV map with data for the generated polygons.

Clone Effector--

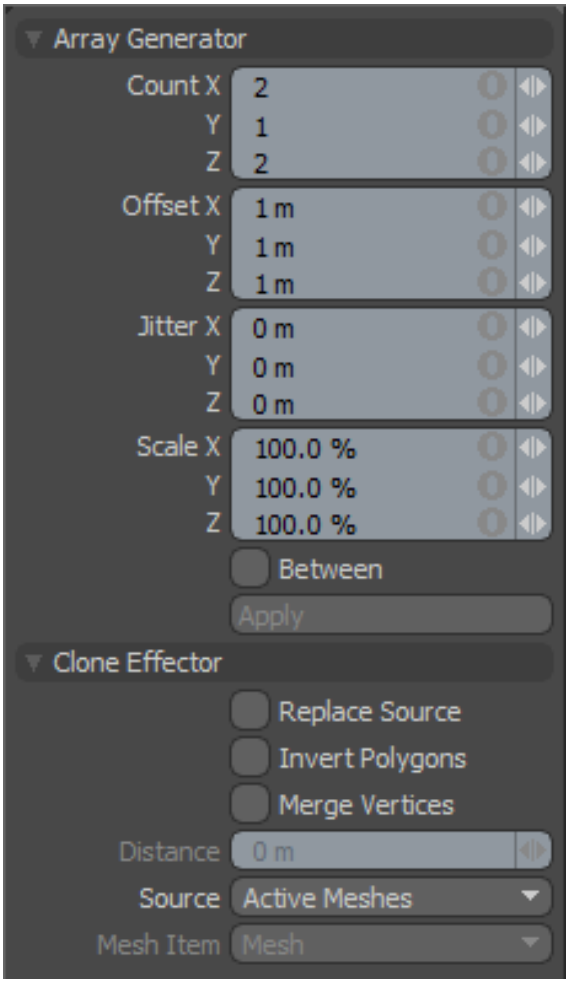
Replace Source: This control will replace the original mesh with a generated mesh that is perfectly aligned with the initial control point. This is very useful when the Start at Source option is off and the initial curve point is located far from the original geometry.

Array Items

Found on the Duplicate Tab of the modo Tools toobar, the Array tool creates copies of the currently selected geometry in an ordered pattern.

Holding the Alt key before clicking the tool active will convert the tool into an Instance Array generator which will create instances of the mesh in the array rather than creating new geometry in the existing layer.

To set the number of copies in the array, adjust the Count X,Y, and Z values in the Tool Properties > Array section.



The Offset settings allow you to control the distance from the original geometry to the copies created by the tool. This value can be controlled directly on the property form or by using the handles in the 3D viewport.

The Jitter controls allow some random variation to be added to any of the three axes (X,Y or Z). Each copied mesh will be positioned with the offset and a random variation using the Jitter distance as the maximum possible offset variation.

The Scale XYZ values allow you to have the cloned geometry be scaled differently than the original.

The 'Between' option, when enabled will allow users to define a first position and a last position and the rest of the copies will be generated between the exisiting copies.

Replace Source will delete the original geometry and replace it with a copy of the geometry made by the tool. This is very useful if you are using any Jitter or Scale values other than the defaults.

Radial Array Items

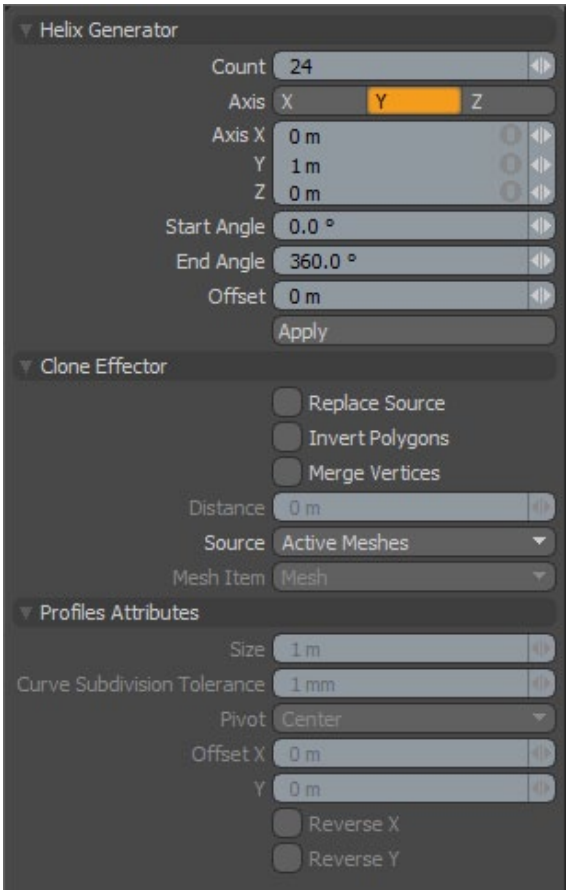
The Radial Array tool copies the selected geometry into a circular or helix pattern. There are a variety of controls that allow precise control over the count, placement, angles and spiral.

Count: Controls the number of clones created.

Axis: Determines along which axis to create the helix.

Axis XYZ: These three values determine the location of the tools Axis control handles. If

the values are set to 1 meter on all axes the handles ends will have one end at -1 X, -1 Y, -1 Z and the other at 1 X, 1 Y, 1 Z creating an skewed off axis plane for the array.



By default the Start and End Angles go from 0 to 360 degrees. This creates an array that completes a circle. Changing these two values can create an array that revolves many times or one that leaves one side of the circle agape. Setting the end value to greater than 360 degrees is very useful when combined with some non-zero offset value.

Start Angle: This degree setting determines the rotation placement for the first cloned geometry.

End Angle: This degree value determines the rotation placement for the last coned geometry.

Offset: Moves each successive clone a user defined distance from the previous along the tools axis. This value is very useful when combined with an End Angle greater than 360 degrees so that the array creates a spiral.

Replace Source: When this toggle is active the original mesh elements cloned by the tool will be replaced with an additional clone.

Transform Modifiers

Transform modifiers are any function that changes the way transforms are applied to items in modo. Falloffs are the primary modifier, but modo's Snapping options also affects Transforms when applied in Items mode. As Falloffs are generally utilized to deform geometry components directly, like a soft selection, that function is documented in the modo toolbox section of the documentation. This is also where the Snapping functionality is covered.

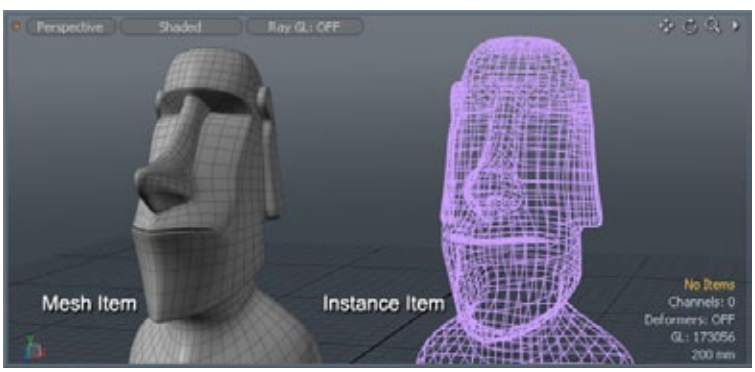
Duplicating Items

Instances

Instances are a special item type in modo that add little additional data overhead to a scene, as Instances contain no actual geometry. They instead references a source object at render time, allowing users to generate a huge number of duplicates for an item with little additional memory usage or performance hit. As an added benefit, users can modify the source geometry at any time and the changes will automatically propagate to each Instance. For example, consider a model of a computer case, where each bolt is an instance of a single mesh. When the client decides the bolt needs to be a hex head rather than a standard phillips head, all instances of the bolt can be easily updated by simply modifying or replacing the geometry in the source item.

Instances are as easy to create as a normal duplicate, and many cloning and array tools offer additional functionality to generate Instances instead of direct geometric clones. Instances themselves can be treated like normal mesh layers, each having its own set of transforms and render options. Users can select them like any other item, and position, scale and rotate them. This allows users a great deal of control over placement of the copies. While Instances reference their shading and material settings from the source object, users can assign an 'Item Mask' through a 'Material Group' item in the shader tree and target custom surface settings to a specific Instance.

Mesh Instances are differentiated in the Items list by their pink cube shader tree icon (as opposed to the standard blue-gray cube icon for Mesh items). Instances are also distinct in that the layer name is displayed *italicized* and their unselected wireframe representation in the viewport is always by default pink in color. Instances are not limited to mesh items, instances are useful whenever users have a large number of like items. If a series of streetlights in a scene all shared identical settings, users could instance the first light and then later on, just as with mesh items, changes made to the source light will propagate to all the instances.



Creating Instances

The simplest method of creating an Instance is to simply select a mesh item (while in 'Items' mode of course) and press 'Shift+D' on the keyboard. Users can also RMB+click directly on a item layer in the Items list (while in any mode) and select the 'Instance' (or 'Instance Hierarchy') option from the context menu. A menu bar command is provided as well. Select an item layer in the item list or directly in the 3D viewport and invoke the command "Item > Instance".

When invoked while in 'Items' selection mode, many of the duplication tools, such as 'Clone', 'Curve Clone' and 'Radial Array' provide an option to generate Instances; with the tool active, simply *enable* the 'Instance' checkbox and the clones will be instances instead of generating full geometry. Alternate versions of the tools are generally available as alternate buttons (alternate buttons are denoted by the small arrow in the bottom right corner, LMB+click and hold to open the flyout menu to select the alternate tools)

Selecting Instances

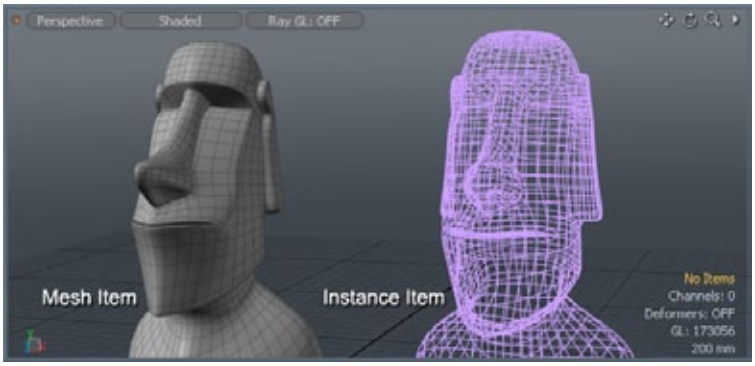
Once an instance has been created, users can work with it like any other item in the scene. Users can select them and transform then directly in the 3D viewport or use the numerical settings found in the Properties panel when the item is selected. While it is possible to select and modify the 'Center' and 'Pivot' position of an Instance, this actually moves the Instance itself. To modify a Center, users would nee to edit the source item. However, since Instance placement is based on the items Center, modifying the Center point of the source would move all the instanced items.

If a complex scene makes it difficult to locate the source item of an Instance, users can select the Instance and then invoke the menu bar command "Select > Source of Instance" This command is also available via the RMB context menu within the Items list viewport. RMB+click in the instance and choose the 'Select Source of Instance' option from the popup menu.

Users can convert an Instance to a normal mesh item for further editing, if desired, by using the 'Convert to Mesh' button available in the 'Properties' panel whenever an Instance is selected. Additionally, users can RMB+click on the item layer itself, opening the context menu, and choosing the 'Change Type > Mesh' option. Finally, users may access the same command from the menu bar under "Item > Change Type > Mesh".

TIP: The difference between cloning (duplicating) geometry in a component mode versus cloning while in Items mode is that each clone becomes its own Item layer in the Items list. Cloning while in a component mode merely duplicates the geometry in the same item layer. If you feel the items list becomes too cluttered with Instances, users can simply LMB+click on the top instance layer, and then pressing and holding down shift and LMB+clicking on the bottom layer to gang select all layers, then pressing 'Ctrl+G' to automatically parent all the selected items to a Group locator in the Items list. Users can then toggle the visibility of all the child items by LMB+clicking the arrow icon ' ▶ ' preceding the Group layers name. Likewise, users can also toggle visibility of the Instances themselves by enabling/disabling the eye ' 👁 ' of the parent group (assuming that the Preference to do so isn't disabled).

Replicators



A Replicator is more like a *controller* than an *item*, unlike all the other items which have visible representations in the scene (OK, except for the Group Locator), Replicators are only visible once their 'Prototype' and 'Point Source' option have been populated. Once defined, the Replicator then *controls* how duplicates of the item are copied onto the surface. Replicators are also unique in that they *require* these additional elements to function. The 'Prototype' is simply any mesh layer within the given scene. The 'Point Source' defines the positioning of the clones; by default the center position of the initial mesh layer will be positioned precisely at the location of each vertex in the point source item, so while users don't have direct control over the positions of Replicated meshes (like with Instances) users can indirectly control their position with the point source.

Creating Point Sources

Point Sources can be any mesh layer, so any of the ways that users can create geometry can be used to create point sources. This could even be something as simple as a flat plane created using the Box tool. If someone were to apply a replicator to a single poly plane, they could easily see that there were four replicas at each corner (vertex position) of the polygon. In order to get additional replicas, the polygon would

require additional subdivisions. This can be achieved by increasing the 'Segments' option found in many of the primitive geometry tools, or after the fact by simply applying the 'Subdivide' command found in the "Mesh Edit" sub tab of the modeling toolbox. If someone were to sculpt this now subdivided plane, adding hills and valleys, the replicas would still be positioned at each vertex, but instead of facing upward normally, they would now each align themselves to the *normal* direction of the background vertex (which is an average of the *normal* direction of all the surrounding polygons) following the contours of the surface. This indirectly allows users to control position and to a degree rotation by simply using the shape of the point source geometry. Virtually any kind of geometry modo creates can be used as a point source, including the various Curves. Once users have good understanding of how Point Sources work with Prototypes to clone replicas, then generating them becomes an easy task.

TIP: Replicators align the Prototype mesh's 'Center' to the vertex position of the Point Source. Therefore users need to make sure the Prototype geometry is positioned appropriately in relation to its Center. By default this is the World 0,0,0 position for each layer. Users can easily align the Center in relation to the geometries bounding box using the 'Center to Bounding Box' commands found in the 'Edit' menu. User may also manually transform the Center while in Center Selection mode. For more on working with Centers, see the documentation page on Selection Modes.

Creating Replicators

Replicators can be created like any item in the scene by using the 'Add Item' function of the items list. Users can LMB+click to open the 'Add Item' menu and select the "Duplication > Replicator" option. This adds a blank Replicator item to the item list. User can then manually populate the 'Point Source' and 'Prototype' fields with the appropriate items. Users can also automatically populate these options using selection order. Select the item that will be the 'Point Source' first in the 'Items' list or 3D viewport, and the 'Prototype' item second. Then use the menu bar command "Item > Replicators > Add Replicator". Once invoked, users will nearly instantly see their geometry cloned onto the source item.

Adding Randomness

Generally, most geometry, especially that created using Primitives, when utilized as a point source produces replicas that users may find too *regular*. In some cases this geometric uniformity to the positions can be desired, but in most cases, more natural random placements will be the goal. Users can use the 'Variation' settings of the Replicator itself to randomize the positions as an offset from the vertex position. Adding in some Scale and Twist (rotation) randomness will help to reduce the uniformity of the replicas as well.

Users can introduce randomness by starting at the Point Source level. When fine control over replicas isn't necessary, users can utilize what is known as a *point cloud*. Point Clouds are collections of vertices, with no edges and no polygons. These could be useful Point Sources for splashes of water droplets, or exploding shrapnel. Before users can create a point cloud they'll need to enable their view as they are disabled by default. This can be done by opening the viewport options panel, press 'O' while hovering the mouse over the target 3D viewport. Then pressing the 'Visibility' subtab and then *enabling* the 'Show Vertices' option. Users can then draw individual vertices using the 'Vertex Tool', or better still, create some geometry using a primitive tool, such as the Sphere, and the utilize the 'Remove' command to eliminate all the edges and polygons leaving behind only vertices. Then users can use the deform tools to push and pull vertices into position. The 'Jitter' function is also a great way to further randomize the point source. Point clouds are assigned as Point Sources in the same way as any other mesh layer in the Replicator item.

Surface Generators

Sometimes randomizing the Point Source isn't desirable or viable so modo has another option to help randomize placement called a 'Surface Generator'. Surface Generators work as shaders in the Shader Tree generating controlled random placements of points over a surface and then utilizing the normal direction of the surface itself to align the replicas. Surface Replicators have several benefits outside of randomizing placements. First, they are shader tree 'aware' meaning they will take in to account Shader Tree settings such as Displacement layers where normal Replicators do not. Secondly, since they are shader tree layers themselves, they can be affected by the other layers, including being masked by the Group item. When used as such, their application can be limited to a specific surface or area of a mesh and can also be controlled by image layers for options like Density, Alignment, Size and Color.

For custom created *point clouds*, that provide all sorts of interesting options,users can add a Surface Generator manually to the Shader Tree by using the 'Add Layer' function. LMB+clicking on the menu and selecting the "Special > Surface Generator" option. This can be created in an existing Material Group item. Users should remember that in order to work properly, the point source mesh will need to be properly tagged with the appropriate polygon tag defined in the Material Group. Users will also need to make sure the Surface Generator is defined as the new Point Source in the Replicators properties. A much easier way to create a Surface Generator would be to simply RMB+click on the Replicator item itself opening the context menu and choosing the 'Scatter Replicas' command. This automatically creates an Item mask in the Shader Tree for the Point Source item and adds the Surface Generator to the Group, appropriately specifying it as the Point Source.

To apply Image Maps or Textures to control the Surface Generators, users only need to apply the texture to the same Material Group as the Surface Generator itself, and then set the layers 'Effect' to the appropriate option. The various 'Effects are documented on the 'Effect - Texture Item' page of the documentation. For more on adding a working with Shader Tree layers, please see that page of the documentation.

Working with Groups

Another way users can add a great amount of believability and realism to a scene when using Replicators is to replicate more than a single mesh. This can be done easily by collecting multiple elements into a Group and the Replicator will then randomly grab members of the group when placing individual clones. Not to be confused with items parented to a Group Locator, to create a Group, users need to first open the Groups viewport. Within the viewport, LMB+click the 'New Group' button and define a new Group. Then, while in 'Items' selection mode, users can either select and drag-and-drop elements from the 3D viewport onto the newly created Group, or select the items in the 'Items List' and then RMB+click on the named group in the Groups viewport to open the context menu and then select the 'Add Items' option. After the group is created, users need to define the named Group as the 'Prototype' source in the Replicator itself. modo will then randomly grab grab one of the items specified in the group for each replica. This can be a great option for applying rocks and plants to an outdoor scene, or trees to a forest scene.

TIP: Users can control the percentage that a group member is used by cloning the elements. For example, if the Group contains 9 red spheres and 1 blue cube, the blue cube only appear approximately 10% of the time across the surface.

Animating with Replicators

Many users may not know that is is also possible to replicate animated items as well. These animations can be either through keyframes, or via deformers. To specify an animated item as a Prototype, users will need to ensure the item layers are properly set up as a hierarchy, and then will need to specify the top most parent item as the prototype in the Replicator. Then enabling the 'Include Child Items' option will ensure all items in the hierarchy will be replicated as well. To add additional randomization, there are 'Time Offset' and 'Random Time Variation' options.

In order to animate the replicator itself, users can animate the settings of the Replicator and/or Surface Generator, or animate the Point Source directly. Users can

apply normal keyframed position and rotation transforms to the point source, but will find that scaling will also scale the replicas themselves. In order to scale the point source properly, users will need to add a morph to the item and then animate the morph over time. Please see the Morph Deformer for more on applying and animating a morph.

Freezing Replicators

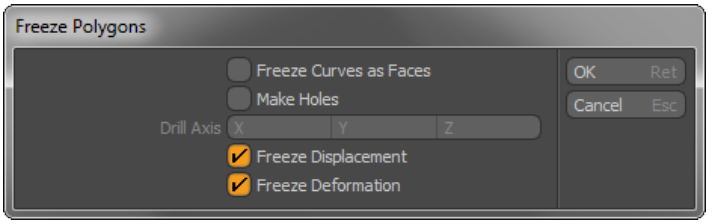
Replicators are all well and good if you intend to keep your geoemtry within modo, however, there are times when you need to export your geoemtry out to an external application, or possilby you just need control of over the placement of a few individual clones, so modo offers the ability to convert Replicators to 'Instances' with the command 'Convert Replicators', found in the menu bar, under the "Item > Replicator > Freeze Replicator" menu. Users merely need to select the Replicator item in the Items list and invoke the command. Once finished, the Replicator will have been removed from the Items list and replaced with a Group Locator that contains all the instances of the original Replicated object. Users may now select specific Instances and affect them individually.

The 'Freeze Surface Generator' provides similar functionality, converting a 'Surface Generator' item in the Shader Tree to regular Mesh Item of the same name with individual vertices for each replicator position. Invoking the command from the menu bar "Item > Replicators > Freeze Surface Generator" converts the Item layer to a same named Mesh Item of single point polygons. Users may need to enable th 'Vertices' display in the 3D viewport in order to see the indvidiual point when unselected. This allows users the ability to affect individual points without resorting to Image Map masks and other workarounds.

Freezing

Freezing Geometry

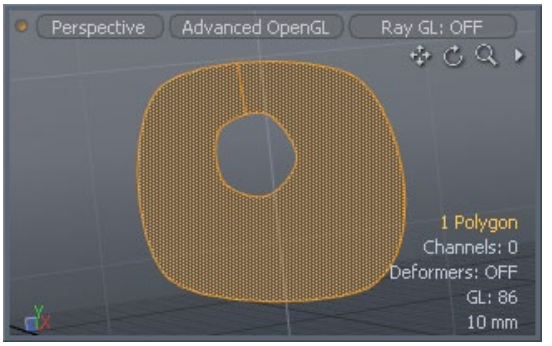
The Freezing of an object is the act of fixing the geometry into a specific state, such as when a Displacement Map is applied, users can *freeze* the displaced polygons, allowing for the removal of the Image Map. This can be useful when users wish to further edit the model, or export it out of modo into another application. Freezing can be applied in a number of situation, to a number of items- Curves, Beziers, Patch Surfaces, SDS Subdivide and Catmull-Clark Subdivision Surfaces geometry, deformed geometry and Multi-Res sculpted models. The 'Freeze' command can be found in the menu bar under "Geometry > Freeze". The command will affect any selected geometry when invoked.



Freeze Curves as Faces: When either of the modo Curves types are selected for freezing (Spline or Bezier), the 'Freeze Curves as Faces' option, when *enabled* will convert the curve to a flat polygon whose edges match that of the original curve. For open curves, modo will draw a single edge between the last and first vertices producing a closed continuous loop. When *disabled*, the curve will convert into a continuous strip of two-point polygons. The number of edges in the frozen polygon is determined by the Item 'Curve Refinement Angle' setting. Low values will produce polygons with many edges that match those of the original curve, while higher angle values will produce polygons with less edges; a rougher facsimile of the original curve.

Make Holes: When Freezing curves as Faces with overlapping shapes, especially those created by the importing of Encapsulated Postscript curves from a vector drawing program, enabling the 'Make Holes' option will tell modo to attempt to drill the overlapping shapes automatically.

Two overlapping curves



'Make Holes' enabled

Drill Axis: Depending on the facing direction of the initial curves, the 'Drill Axis' option is used in conjunction with the 'Make Holes' command, determining the axis direction of the automatic Drill operation.

Freeze Displacement: When the 'Freeze Displacement' option is *enabled*, modo will recognize any Image Map layers assigned to to the geometries surface defined as

'Displacement' or Vector Displacement' when freezing and apply them accordingly retaining the mage based displacements. Once frozen, users can remove the displacement effects. When *disabled*, modo will ignore displacement layers and only freeze the raw Subdivision cage (also called the Base Mesh). The number of polygons of the final frozen model is determined by the 'Subdivision Level' option in the Item properties, or the 'Current Level' when freezing Multiresolution meshes.

Freeze Deformations: The 'Freeze Deformation' option, when *enabled* will consider Vertex Map Deformers or MDD Deformers applied to the item for the purposes of animating when freezing the object. For objects with Morph Deformers, modo will freeze the mesh in its current visible state in the viewport (so Morph Deformers simply need to be *enabled* when freezing to be considered).

TIP: Displacements applied by way of procedural textures layers will not be considered when freezing, regardless of the 'Freeze Displacements' option. Users will need to convert the procedurals to Image Maps by way of baking prior to freezing the mesh.

Rigging & Animation

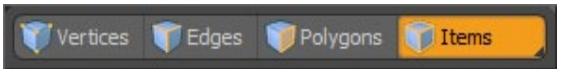
Animation is the introduction of movement to static objects, for many this is truly what brings 3D images to life and is often considered the ultimate goal for many 3D projects. modo offers an array of options and functions that ease in the creation of animated scenes.

Animation Overview

Model Setup for Animation

So your object is modeled, you've surfaced it and set it up in a scene and added lights. The next step is to bring it to life with animation. In order to ensure the smoothest animation workflow in modo, there are a number of basic concepts one should clearly understand before proceeding. modo has a number of *modes* for editing mesh items -'Vertices', 'Edges', 'Polygons' and 'Items'. In most default layouts, these modes are displayed across the top of the screen, right under the menu bar.

modo modes



Hitting the space bar on your keyboard will toggle through the first three modes -'Vertices', 'Edges' or 'Polygons'. When in these modes, applying any type of transform -position, rotation, or scale, you are editing the raw geometry that makes up your model, in modo, these are called component level transforms. Now, you may have noticed that regardless of where you position the geometry in a mesh item layer, the center position for that item in the properties panel will always rest at 0,0,0.

Now if you select the 'Items' mode, and apply some transforms, you are now applying an item level transform. You can see that as you move the object in 3D space using position, rotation or scale, the values in the items properties panel are now changed from something other than 0,0,0. You can no longer affect individual vertices in 'Items' mode, as you are manipulating the layer as a whole. This is how objects are animated (and positioned*) in modo, by applying item level transforms. (For more on the Mesh Items properties panel, please reference that page of the documentation) It can become confusing as the same tools for move, scale and rotate are used for transforming items in component *and* Item mode, but if you keep an eye out for which mode you are in, it will soon become second nature.

Mesh Items and Layers

When modeling objects, it is convenient to think of mesh items as layers, especially when using tools like booleans that use background layers for cutting and utilizing background constraints for modeling and retopologizing. You might be inclined, though, to keep all the elements of an object, say for instance a car, in the same layer. For still

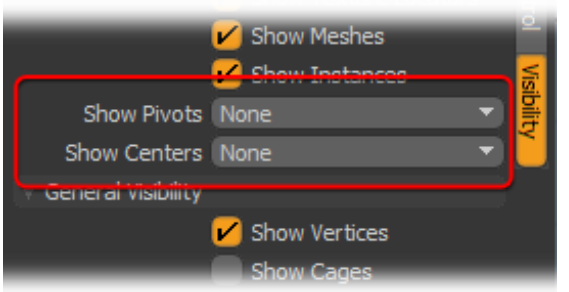
images, this is fine and works well. But each *layer* only has one center point, so it would not be possible to animate that cars tires rolling as it moved without affecting the entire layer. It should quickly become obvious that each tire will need its own separate layer, as each would need its own position and rotation to animate properly. Really, any part of the car that you want to move independently from the cars body itself would need its own layer -doors, hood, trunk. Once you've placed all the appropriate geometry in its own layer, each has now become its own mesh item.

Hierarchies

Now that the various individual mesh items have been successfully created for the make-believe car, how does one move the whole thing around in the scene? It's far too cumbersome to select each individual item in the viewport or item list, it's also prone to errors. To ensure all the elements remain together when moving the cars, we need to create a hierarchy. Hierarchies are groups of items that define how each element relates to other elements. Creating hierarchies are covered more fully in both the Item List and Animation Setup documentation sections. For simplicities sake here, we can just create the most basic modo hierarchy - a group. By selecting all the elements of the car and using the keyboard shortcut 'Ctrl-G', we have placed all the mesh items into a group (folder icon in the item list). If one wants to easily move the entire car and have its individual parts follow along, one just needs to select the top most group item (in this case, the group locator) and apply a transform to it, all the related items in the group will follow along. Another benefit of creating a hierarchy is that it nicely organizes the Items list, especially important when building up complex scenes..

Centers and Pivots

modo provides two methods for setting the point around which an item rotates and scales, 'Centers' and 'Pivots'. You may be familiar with pivots from other applications; modo's pivots can be used in the same manner. However, modo also provides 'Centers' which in the many situations are used instead of pivots. Before proceeding though, we need to be able to see the Pivots and Centers, as they are not visible for all viewports (this is a '*per viewport*' setting). To do this, open the 'Drawing and Visibility' options for viewports by hovering the mouse over the target viewport and pressing 'O'. In the popup, select the 'Visibility' subtab and see the options for 'Show Centers' and 'Show Pivots'.



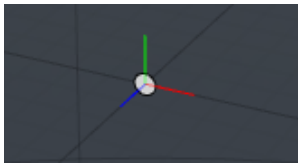
Setting them to 'Selected' will show a layers associated Center or Pivot whenever it is selected in the Items list, 'All' will always display them regardless of selection, but for scene with many items, their display may become overwhelming. So, now that we can see them, lets look at their definitions-

Center: This is the local origin of the model and the default location around which item transforms originate. It's the location that sits at the world origin (0,0,0) when all the item transforms are at their defaults, i.e. when the item has not been moved, rotated or scaled in 'Items' mode. Centers cannot be animated relative to the element they represent, but adjusting their location allows you to set where the item will rotate and scale from.

Pivot: Pivots also provide the ability to specify the location around which an item will rotate and scale, determined as an offset from the Center (When adjusted you'll see a small dotted line draw between the Center and the Pivot as a reminder). They can be used when one needs to animate this location (say a cube rolling across the floor for example, where the rotation point jumps to each edge as it rolls), as Centers cannot be animated. Once a Pivot is moved (defining that offset from the Center), the rotation and scaling occurs around this position instead of the Center.

So, the only real difference between them is that Pivots can be animated and Centers cannot. This is because the pivots are an offset from the Center (Pivots animate relative to the Center). In most cases Centers are perfectly fine for animating *items*, this is why they are the default method. Now consider the car again from earlier, having modeled the entire object in a single layer the individual mesh items were then created by copy/paste functions from layer to new layer. While each item now has its *own* 'Center', the default location for them is at the world origin. If you were to try to animate the cars tires when created this way, you would find they don't rotate as one might expect, they probably rotate from a point originating somewhere under the center of the vehicle, sending the tire out into giant crazy circles; centers don't *default* to the center of the geometry bounding box. The alternative could have been to model the tire separate

from the car at the origin, and use an Item level transform to then move the tires to the proper position in the wheel well but luckily modo doesn't care how the item was created and can easily reposition the Center point without affecting the geometry itself.



If one RMB+clicks over the 'Items' mode button, a popup appears allowing for some alternate mode selections. By clicking on 'Center' you are now in a mode that allows you to edit the location of center positions without affecting the location of the original geometry in any way. So, by making sure the tire mesh item is selected in the item list, we can now see a small circle with the three lines coming out from it, this icon represents the center of our items layer. By clicking on the center icon itself (the white circle), it will turn orange, showing that it has been selected, now the 'Move' (W) and 'Rotate' (E) transform tools will affect the Center point. You could move it and eyeball it into the center of the wheel, or you could use a trick with the Work Plane that is much more precise.

Switching back for a moment to 'Polygons' selection mode, there should be some type of selection that could be made that would represent the position from where the wheel should rotate from, make this selection and use the 'Align Work Plane' command from the menu bar "Edit > Work Plane > Align to Selection". Now switch back into 'Center' mode and select the center again. In the 'Animate' and 'Setup' interface tabs there is a command called 'Set to Work Plane', click it, and that will move the Center to the origin of the workplane which was precisely aligned to the center of your previous polygon selection. Reset the Work plane by pressing the 'End' key and everything is back to normal, but the Center was precisely positioned with no guessing or eyeballing required (For more information on the Work Plane, please reference that page of the documentation). Now when one animates the tire, it will rotate around the proper location. Just like in modeling there is a process to getting the best results, animation requires a bit of setup, but once one understands the basic setup techniques, it eventually becomes quick and easy.

Animating Workflow

Once the model is set-up (also referred to as rigging for more complex setups), you are ready to start *animating*. Making anything occur over time in modo requires a keyframe. A keyframe simply sets a specific value for an elements in time, be it a position, or a surface setting or whatever. Nearly any setting in modo can be keyframed to change

over time. In traditional 2D cartoon animation, artists draw the important poses for an animated sequence, and then another artist, called the in-betweener, draws all the subsequent frames that transition from one pose to the next. Animating in 3D is very similar, where users set up the key poses (or positions, or settings) but unlike traditional animation, modo takes care of all the in-between frames for you.

In modo the general workflow is to select the type of element you want to work with and then run a tool or fire a command that will affect your selection. Select a vertex, activate the move tool, move it, that kind of thing. Keyframing is similar in modo, in this case the type of 'things' you are working with are channels. Channels are simply the animateable properties of any item.

When you press the 'Key' button ('s' shortcut key in Items mode) modo needs to know which channels you want to create keys on. It does this by using channel selections, it runs through each channel you have selected and creates a keyframe on that channel at the current time. Selecting channels can be done in a couple of ways, by clicking on them in a 'Channels' viewport or by clicking on the control label on a properties viewport, (the small gray circle that precedes an attribute in the properties panel), when users click on the dot, it will change from gray to red signifying a keyframe is set at that point in time for that property.

Constantly selecting channels to keyframe would quickly become tedious so modo provides some 'Auto Key' behaviors to help with this. These are set with the 'Auto Key' menu at the bottom of the "Animation' interface tab. The default setting is 'Animated', with this on modo will automatically create keyframes on channels that are already animated (i.e. at least one key has been made on the channel previously) whenever the value on the channel changes. This can be by using the transform tool, using the mini-slider or by typing in a value on a forms control. If changes are made to a channel that has not been keyed the changes are still applied but keyframes will not be made. The next setting is 'All', and as it's name suggests this will create keyframes on any channel at the current time whenever it's value is changed. You have to be careful when using this setting and be aware of whether you're in animation mode or not as it's obviously very easy to create animated values when they may not be wanted. The last setting is 'None', this will never create keyframes automatically for you, to make keys you must use the 'Key' button whenever you've made changes.

Which setting you use is entirely up to the user of course, depending on how hey like to animate. Some people like auto keying, some people loathe it! The recommended setting is the default 'Animated' one, it provides a workflow somewhere between the two extremes of on and off and only requires the additional initial step of creating keys on the channels of the item you want to animate in advance.

The other area that modo tries to help with keyframing is automatic channel selection. Auto keying only works when channel values have changed, when you need to create a set of keys for channels without changing the current values you need to use 'Key' button (which requires the channels to be selected) or individually click on the channel buttons on the properties forms. To help with this modo will automatically select channels for you when using the transform tools, activating the move tool for example will select the position channels.

A general Auto Key workflow--

1.) Select the Camera and go to its Properties viewport.

2.) Click on the channel keying buttons for the six position and rotation channels. They should turn red.

3.) Activate the transform Move tool (W key) and move the camera.

4.) Activate the transform Rotate (E key) and rotate the Camera

5.) Move the time slider to a new time

6.) Move\rotate the Camera again

7.) Go to a new time...etc

Once those initial keys have been made you should not have to do any further manual keying unless you want to key the camera in place. This will also work when using viewport navigation to move the camera. This general overview should be enough information to get you up and animating. So, what are you waiting for? Get to it, and have fun.

** When working with Instances and Replicators, it is best to model the base mesh at the origin as all duplicated items will use that position as the transform origin when propagating duplicates. Of course, adjusting the Center position after the fact will remedy any peculiarities as well.*

Working with the Timeline

The timeline is the chief function for controlling *time* in modo. Users can modify the 'current' time position, represented by the white highlighting line and frame number, by LMB-click dragging left and right within the numbered area. There is only one *current time* in modo, modifying it in one viewport, changes it in all viewports, so there can be no confusion. To animate any item over time, one need only to set a specific value at a point in time, known as a *keyframe*, move to a different point in time, set a new keyframe and modo will smoothly interpolate the in-between values.

Keyframes can be set many ways. Within a properties viewport, each item property is

preceded by a dot that users can LMB+click, setting a keyframe. When set, the dot will turn red, signifying the presence of the keyframe. Users can also set keyframes by clicking any of the various keyframing buttons under the timeline. Users may also enable the 'Auto Key' feature setting keys automatically whenever an attribute is modified. Additionally, users can set keyframes by clicking in the far right column of the 'Channels' viewport or, finally, by selecting one or more channels in the 'Channels' viewport and simply pressing 'S' on the keyboard. Keyframe positions are indicated as white dots on the timeline, with in-between interpolated areas indicated by a green line connecting the various keys.

The timeline itself represents a specified length of time, which is a product of the number of frames divided by the frame rate. By default*, modo displays 120 frames; divided by the default frame rate of 24 'Frames Per Second' (FPS) this represents a total of 5 seconds of time. For longer or shorter sequences, one can adjust the length several ways. A 12 second sequence would require 288 frames (12 sec. x 24 fps = 288 frames). This value can now be entered into the 'Scene' items properties panel under 'Scene Range', or in the timeline 'Options' popup, centered below the timeline, and also directly in the interface below the timeline, to the far left and right sides. The two left most values, left to right, are the the scene start frame and current start frame, and on the right it is, left to right, the current end frame followed by the scene end frame. The scenes frame range represents the total sequence length. A fractions of this length is specified using the 'Current' frame range, allowing users the ability to concentrate on a specific interval without needing to preview the entire sequence. Directly below the timeline itself, is a gray bar with two handles, one on each end. Dragging either of those handles will adjust the current frame range interactively.

Further below the timeline is a series of buttons. First are the time control buttons.

> Sets the current time position to the first frame of the 'Current' frame range. (Keyboard shortcut - 'Shift+Up Arrow')

> Sets the current time position to the first keyframe of the currently selected item. For multiple items, first item selected will be used. ('Ctrl+Up')

> Moves the current time position to the nearest previous keyframe. (Keyboard shortcut - 'Ctrl+Left Arrow')

> Moves the Current time position backwards one frame. (Keyboard shortcut - 'Shift+Left Arrow')

> Plays the animation sequence backwards based on the 'Current' frame range.

> Plays the animation sequence forward based on the 'Current' frame range. (Keyboard Shortcut - '/')

> Stops sequence playback. (Keyboard Shortcut - '/')

> Moves the Current time position forwards one frame. (Keyboard shortcut - 'Shift+Right Arrow')

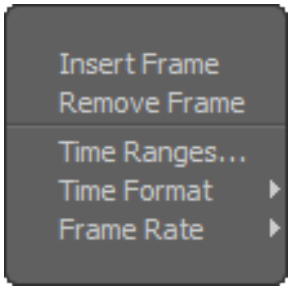
> Moves the current time position to the nearest next keyframe. (Keyboard shortcut - 'Ctrl+Right Arrow')

> Sets the current time position to the last keyframe of the currently selected item. For multiple items, first item selected will be used. ('Ctrl+Down')

> Sets the current time position to the last frame of the 'Current' frame range. (Keyboard shortcut - 'Shift+Down Arrow')

IMPORTANT! These keyboard shortcuts are part of the "Item mode" keymap. When modo is in one of the component selection modes such as Vertex, Edge or Polygon many of these keys will perform important modeling tasks. This allows modo to provide an efficient workflow for both processes.

> Displays current time position. Users can manually specify the current time by editing this field.



Timeline Contextual Menu

When RMB+clicking on the timeline itself, a menu opens with a number of useful contextual options-

Insert Frame: This command inserts a single additional frame for all selected items, at the current time positions, pushing further keyframes out. Useful in adjusting keyframe positions for selected elements.

Remove Frame: This command removes a single frame for all selected items, at the current time positions, pulling further keyframes toward it (It does not delete

keyframes). This is useful in adjusting keyframe positions for selected elements.

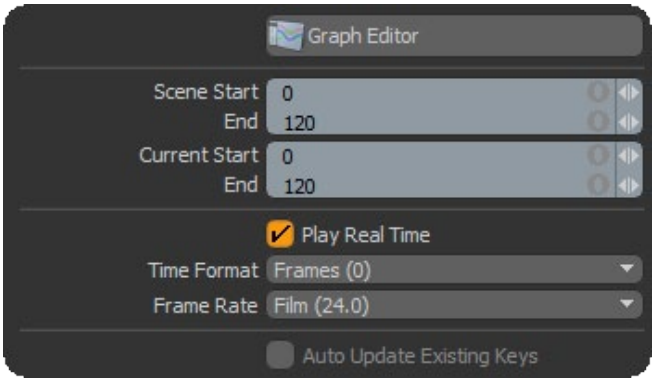
Time Ranges: This command opens a popup dialogue allowing users to numerically define either the 'Scene Range' or the 'Current Range'.

Time Format: Allows users a quick way to define the timeline display format (described on the following pages).

Frame Rate: Allows users a quick way to define the playback frame rate (described on the following pages).

Options

Opens the Options popup window with setting similar to those found in the Scene item properties.



Graph Editor: User can access the Graph Editor window by using the 'Graph Editor' button. The graph editor allows the user to adjust how modo interpolates values between keyframes.

Scene Start/End: Sets the sequence length using the Start and End frame numbers.

Current Start/End: Sets the current length using the Start and End frame. When animating, users may wish to concentrate on a single area of a scene for fine tuning, the current range setting allows the users to essentially temporarily crop out the rest of the scene so they can concentrate on the one section.

Play Real Time: When enabled, the sequence will play back an OpenGL viewport using the 'Frame Rate' setting to the best of the host systems abilities, skipping frames if necessary. When disabled, modo will play every frame in the sequence, making its best attempt to maintain the specified frame rate.

Time Format: Allows the user to specify which standard is used when displaying individual frame numbers

Frames- Indicates frames as sequential numbers, using standard numbering format.

Seconds- Indicates frames as a passage of time, using the "0.0s" format.

SMPTE- SMPTE time code specifies frames utilizing a shorthand method "00:00:00:00"; representing hours:minutes:seconds:frames

Film Code- Specifies frames by the length of the film in feet (linear measurement) using "0.0 + 0" format.

Frame Rate: Users can specify a frame rate by standard definitions, such as Film (24 fps), PAL (25 FPS), and NTSC (29.97 fps). Frames Per Second, usually referred to as FPS, is the number of still images that playback in sequence in one second of time.

Auto Update Existing Keys:

> Opens modo's 'Graph Editor' as a floating palette.

> Drops the current Channel selection.

> (Press Shift) Drops the current Item Selection.

> Sets a keyframe for all currently selected channels.

> (Press Shift) Sets a keyframe for all currently selected Items.

> Sets a keyframe for an items position channels.

> (Press Alt) Sets a keyframe for an items pivot position channels.

> (Press Shift) Assigns the current World Position value to the local item.*

> Sets a keyframe for an items rotational channels.

> (Press Shift) Assigns the current World Rotation value to the local item.*

> Sets a keyframe for an items scale channels.

> (Press Shift) Assigns the World Scale transform value to the local item.*

* Useful when retaining the position of driven items when removing a Constraint or Modifier.

Auto Key: As an items position, rotation and scale are changed, using modo's 'Auto Key' function users can avoid setting keyframes manually. When enabled, modo will generate a keyframe for any modification made to an item at the currently selected time. It can be a great time saver, though some users dislike auto keyframing as it can make it possible to inadvertently modify an item, overwriting previous values. The three settings are:

All- Sets a keyframe for any and all changes made to an item. In this mode, users should be careful as it can be very easy to introduce keyframes for settings that were not intended to be animated.

Animated- Sets a keyframe only for channels that have existing keyframes values. Example- If there is a keyframe for the position channel only on an item, moving the time slider and changing the position and rotation will only set a position keyframe. Users will need to first explicitly set a key for a channel to activate auto keying.

Off- Turns off the Automatic keyframing function.

Keyframing Shortcuts

To aide users in setting keyframes while working with the time line in the Animation view, modo provides a number of keyboard shortcuts, in addition to the interface buttons. This group allows you to quickly set keyframes on an entire item or specifically on the various transform elements or selected channels. When using the Key Selected Items command, keys will be created on any channels for that item that already contain at least one keyframe. Other channels for that item will not be keyed.

Shift-Y = Key transforms

Shift-E = Key Rotations

Shift-W = Key Positions

Shift-R = Key Scale

S = Key Selected Channels

Shift-S =Key Selected Items

These two commands quickly "slide" any keyframes forward or backwards that are *after* the current time. This is incredibly useful for quickly retiming portions of your animation.

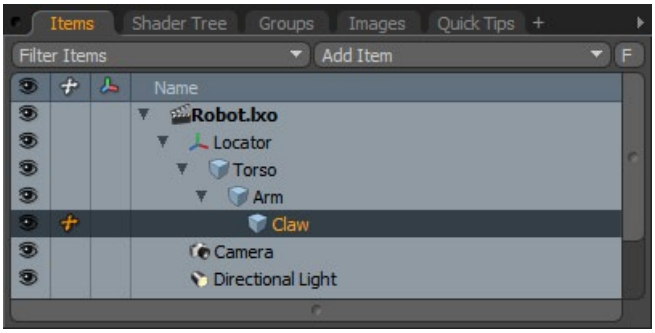
, (comma)= Time Offset Forward

. (period)= Time Offset Backward

Hierarchy

Hierarchy is defined as a series of ordered groupings. In 3D animation these hierarchies are also known as 'Parenting' or 'Parent/Child' relationships, modo uses these grouping hierarchies to define how multicomponent objects relate to each other, such as a car with spinning wheels. Moving the car around is much easier when the wheels move *with* the car automatically, but they still need to be separate objects from the car as they move differently than the body itself, this is where hierarchies come in really handy. Every hierarchy starts with is base item, called the 'Root', this is the main *parent* item, each subsequent item in the chain is considered a *child*, and the item directly above a child is always considered its parent.

Hierarchies are essential to animation as they define the orders of inheritance for transformations in a chain of linked items. Consider an arm, if you move your shoulder (or torso), your arm follows along, as does your wrist and hand. So the shoulder bending is passing its *transforms* on to the elbow, which can also bend, both of those *transforms* are passed on to the hand so on. modo visualizes these hierarchy chains as cascading items within the Item List.

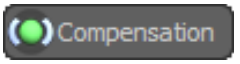


Defining Hierarchies

In the image above, note the indentations of each subsequent item in the list. The

top layer called 'Locator' is the parent item, the next its 'child', followed by siblings in lower orders. Proper ordering of items in a hierarchy is imperative as it defines how the transforms relate to items further down the chain. Creating hierarchies in the item list is a simple process that utilizes drag-and-drop functionality. By simply selecting any layer and then LMB+click and dragging and then *dropping* the layer onto another layer (target layers are highlighted) it will automatically parent the items *in place*, meaning the items will intelligently inherit the parent items transformation, if it exists (i.e. it won't move from where it is when parented). By pressing the 'Ctrl' key before releasing the mouse button, it will Parent the items directly, so if the target has any transformations, this will cause the child items being parented to move as if it were connected to the item all along (i.e. if it were parented before it was moved from 0,0,0).

The animation tab setup provides some additional tools for creating and modifying hierarchies.



Compensation: When parenting items together that have prior transforms applied, you might find your items popping out of position, what is happening is the child is *inheriting* the transforms of the parent, instantly sending it off to a position you likely don't want. Enabling the 'Compensate' button which is positioned in the top area of the animation layout, next to the mode buttons, changes the Parent/Unparent function to Parent in Place/Unparent in Place. What this does is ignores prior parental transforms, so items retain their initial positions in relation to their parent.

Parent/Unparent: When in item mode, the 'Parent' and 'Unparent' commands produce the same results as the item list, but work with selections in the 3D viewport. Selection order is important when using this function as it sets the relationship properly. The last item selected becomes the parent while everything else becomes the child of the parent. Multiple items selections won't create an entire chain, to do that, you would need to apply the parenting command multiple times. To use the parent command first make sure you are in 'Item' mode. Select an item in the 3D viewport by clicking on it, this will be the child item, hold down the shift key and select the second item, with both items selected, click the parent button. And that's it. You can see that the first item selected, has now become the child of the second item. Unparent works the same way, select any item or items and click the unparent button to remove them from the hierarchy. Keep in mind when setting up hierarchies that any item can be parented to any other item, cameras, lights, and locators.

Insert: The 'Insert' command is a useful function for adding items into an existing hierarchy without breaking or upsetting the existing chain. 'Insert' will add the first selection as a child of the second selection, inheriting the latter selections transforms and siblings.

Insert at Parent: The 'Insert at Parent' command reverses the insert command, inserting an item as a parent instead of a child, while still inheriting transforms and siblings. Useful for inserting an item into the top of a hierarchy.

Extract: The Extract command will remove an item from within a hierarchy, without disrupting the rest of the hierarchy chain, by automatically parenting children of the extracted item, to its parent, once the item is removed.

3D Viewport Parenting Actions

While in 'Items' selection mode users can use keyboard shortcuts to quickly set up basic hierarchies simply using selection. The last item selected always represents the parent item. While in any 3D viewport, users can select two or more items and use the 'P' keyboard shortcut to parent in place, 'Ctrl+P' performs normal parenting, 'Shift+P' unparents in place, and 'Ctrl+Shift+P' unparents the selected items. For more complex actions users can utilize the 'Drop Action' function.

Drop Action

Users can also utilize drag and drop actions directly in the 3D viewport to define more complex hierarchies, the 'Drop Action' menu in the 'Animate' and 'Layout' interface tabs define what action happens when an item is dropped onto another. Users must be in 'Items' mode with no active tools for the actions to work. Simply LMB+click on an element and drag it toward the target item. The actual element in the interface won't move until the target item is reached and the mouse button is released, producing the desired action.

None- No action is performed on the elements.

Parent- Source item is made a child of the Destination item.

Parent in Place- Source item is parented as above, but with 'Compensation' applied.

Match- Source item is modified to match position scale and rotation of destination.

Match Position- Source item is modified to match the position only of the destination item.

Match Rotation- Dropped item is modified to match the rotation only of the destination item.

Match Scale- Dropped item is modified to match the scale only of the destination item.

Insert- Inserts the source item into the hierarchy of the destination item. The item will be positioned and orientated to match the destination item and will become the parent of the destination item.

Insert at Parent- Similar to 'Insert' but the source item will match it's position and orientation to the destination item's parent (if a parent item is present).

Insert in Place- The source item will be inserted into the destination item's hierarchy as it's parent but will retain it's current position and orientation.

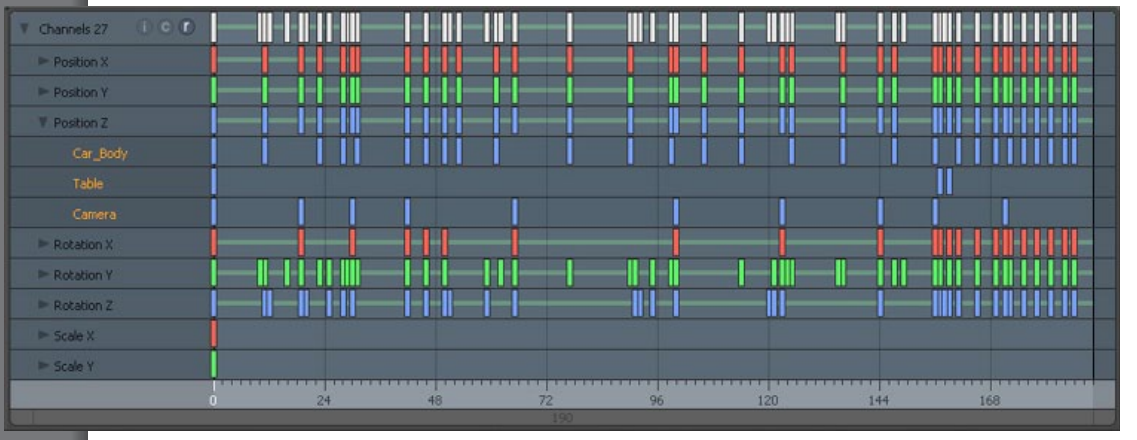
Place- Positions the selected item at the intersection of the mouse pointer and surface.

Place and Align- Positions the selected item at the intersection of the mouse pointer and surface rotating the item to match the surfaces normal direction.

Animation Viewports

Track View

The Track View allows you to easily view and edit all the keyframes in an animation and can be accessed by expanding the timeline up when in the Animate layout.



Keyframes can be viewed inside of Track View in three different ways. Selecting an Item will display it's keyframes in a summary view. This allows you to easily edit all the keyframes of a specific Item or group of Items. To access individual channels you need to expand the Item group. Alternatively selecting the round "C" radio button in the Track View will change the display so it is based on channel type, and accessing the channel for a specific Item requires you to expand the channel group. Finally selecting

a specific animation channel or group of channels will display those channels directly in the Track View. Which method you choose depends on the type of edit you wish to make.

To select keys simply click on them with the LMB or drag out a shaded rectangular selection area with the LMB. Selected keys are by default, shown with a yellow highlight. The key on the topmost row is a "master" key that provides a handy way to affect all keys at a specific frame, so if you move the master key all keyed information below it will be affected. Similarly, if you move the key for say Position X, the "master" key is automatically affected and will be moved along with your Position X key. To add to a selection hold down Shift as you select, and conversely, to delete from a selection hold down the Ctrl key. With a selection made you can then move that selection along the timeline to the left or right. The number of frames that you have moved is shown in real time in the upper left of the Track View (positive numbers above the zero start frame indicate you are moving to a later time). Using the RMB when doing this avoids having to LMB click on a specific key. If you want all of the keyframes after the selection to move in relation to the selection then activate ripple editing by clicking the "r" radio button at the top of the Track View. Please note that if you are using the RMB in ripple mode, you will need to RMB down inside the active key zone (indicated by horizontal green bars) in order for the ripple mode to be in effect.

Scaling keyframes requires you to have at least two keys selected and there are two ways of doing it. Holding down Shift and the RMB will allow you to scale (spread out) the selection from the cursor location. You will see the amount of the scaling operation in the upper left of the Track View as you move your cursor. Holding down Ctrl and the RMB allows you to scale a set of keys from the position of the last or the first selected keys depending on which side of the selection your curser is located. For example, if you want to keep the first key (in time) in a fixed position and scale the other keys in the selection relative to it, then you would hold down CTRL and RMB down to the right of the key selection and scale them by dragging them to the right or left. Conversly if you want to scale a selection relative to the last selected key hold down CTRL and RMB down to the left of the key selection and drag to the right or left.

Right clicking while in the Track View gives you access to a contextual menu. In this menu there are tools that allow you to cut, copy, paste and delete keyframes and also scale and shift keyframes by specific amounts. There are also options for setting the slope of keyframes.

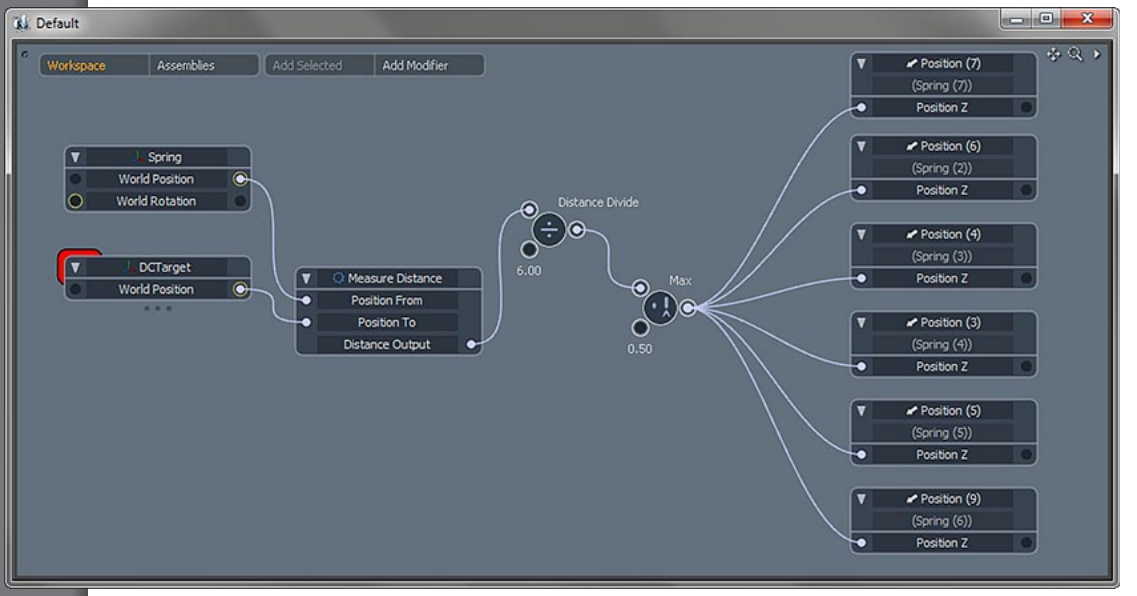
Clicking the MMB in the Track View will create a new keyframe at that position on the timeline for all selected channels or Items.

If you open the Graph editor you will see that it now includes a new track at the

bottom that allows you to select and edit keyframes in the same way as you do in the Track View. Right clicking on this track will bring up the same context menu as in the Trackview.

Schematic Viewport

modo offers the powerful ability to create relationships between items channels when animating, allowing users to procedurally control elements without needing to add specific keyframes. The addition of constraints and modifiers extends that functionality allowing users to, for instance, force certain items to always face the camera, or to keep an element centered between two others. The Schematic viewport brings all of that functionality together into a unified environment, so users can *visually* link elements, including constraints and modifiers, providing a more intuitive and powerful way to build and rig animated scenes. The Schematic viewport is used in conjunction with the Channels viewport, and works by linking channels together via drag and drop.



Navigating the Viewport

The default Schematic view's use is similar to that of the UV view, where users are viewing and working with a flat 2D representation of the information. Users can navigate the view of complex assemblies by way of the navigation widgets in the upper right corner. LMB+click and drag on the arrow widget pans the view, while clicking the magnifying glass widget zooms the view. Users can also use keyboard familiar shortcuts to navigate, Ctrl+LMB+click and drag to pan, and Ctrl+Alt+LMB+click and drag to zoom.

Adding and Arranging Elements

When first opening the Schematic view, users are presented with a blank workspace, modo needs users to first tell it what elements they wish to work with and link. To add the necessary elements to the schematic view, users can use drag and drop functionality, dragging any item from the 3D viewport, Item List, etc. and dropping it directly into the schematic view adding the dropped element to the schematic. Users may also simply select the item(s) with the 3D viewport (or the Item List) and LMB+click the 'Add Selected' button at the top of the viewport. Once items have been added Node representations will appear in the viewport for each element, but only the elements themselves are represented. Users also need to add the appropriate channels for those particular items to actually create links. Users can do this by selecting the appropriate channels in the Channels viewport and drag and drop them onto the schematic view. They don't need to be dropped onto any particular item, anywhere over the window is fine to add them. Items and channels can be added at any time. TO remove a node, use the context menu, by RMB+clicking on the node header and choosing the remove node option or by selecting the node and pressing the 'Backspace' key.

User will find it easier to work with elements by organizing them and arranging them in the view. This can be accomplished by LMB+click on the node header and drag to move, then releasing the mouse button to position the element. modo will try to *snap* positions to surrounding nodes to make it easier to place and align nodes for an orderly schematic.

TIP: When adding items to the schematic view, if channel links or custom channels are already present, these will be added under the 'Header' node for their associated item.

Defining Links

To create a link between channels simply click and drag on an output connector (the round dots to the *right* of the channel name) with the left mouse button. A link will be drawn as the mouse is moved. Move the mouse over a channel's input connector (the round dots to the *left* of the channel name), it will turn green if the link is allowed, release the mouse button to create the link. Any incompatible channels are drawn *disabled* to indicate that a link cannot be made when the user drags over them. To delete links, LMB+click on the linking curve itself to select and press the 'delete' key. Users can simply move node links once created by LMB+click and drag the yellow connector dot and dragging it to a new position.

If there's a link to or from an item or channel that's not present in the view then a yellow circle is drawn around the channel connection dot. To load MMB+click on the channel dot to reveal. To quickly see where the link is from, or going to, position the mouse pointer over the connection dot and a tooltip will appear listing the connected channel(s). Channels that are not present in the view are enclosed in brackets. Note that only a maximum of five channels are shown for readability. If there's more than five links the number of channels not shown is appended at the end of the list. To add the channel to the view MMB+click the connector. This will add all the channels that are linked to it.

Workspace and Assemblies

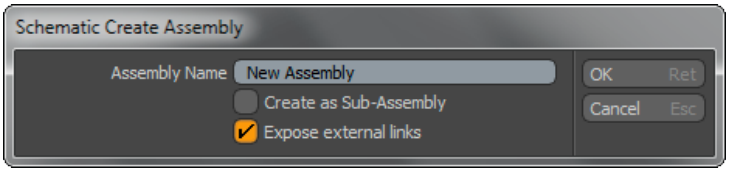
There are two main areas of the Schematic viewport 'Workspace' and 'Assemblies' defined by the mode buttons in the upper left corner. The 'Workspace' is an area in which users can experiment with setups and examine existing scenes. Any rigs, or portions of rigs, created in the Workspace may be extracted into individual assemblies, covered further below.

When the 'Assemblies' button in the toolbar is clicked on the view changes to show all the root level assemblies created in the scene, i.e. those that have no parent assemblies. The assemblies can be re-arranged by drag and drop same as nodes, drag an assembly and drop it in the gap between another assembly. Dropping an assembly onto another will cause it to become a child of the assembly it was dropped on. Assemblies that have a hierarchy beneath them have a small 'badge' drawn in the lower right corner of the assembly graphic which when clicked switches the view to show the complete hierarchy, allowing quick navigation between the child groups. Alternatively the 'hierarchy' button in the toolbar can be used.

In the Groups viewport users can see an option for 'Assembly' Groups, these are the groups that are displayed in the Assembly mode of the schematic view, new assemblies can be created via the 'New Group' button. This will create a new blank display in the schematic. The schematic always shows the most recently selected assembly group.

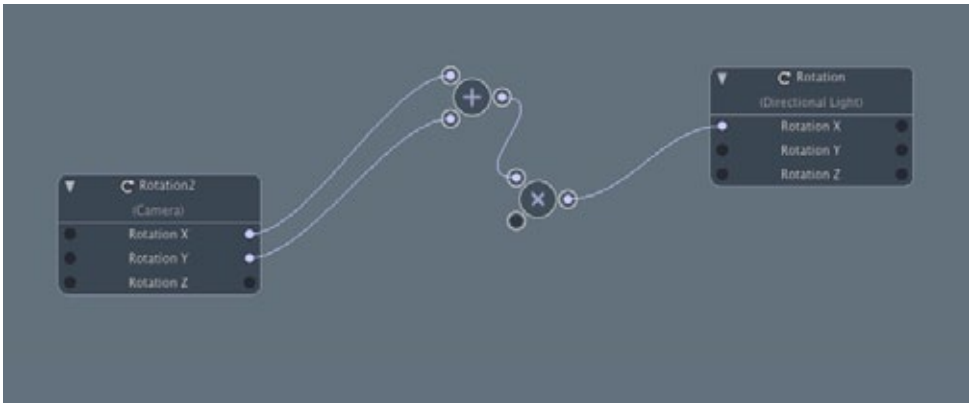
The contents of an assembly, or sub-assembly, can be viewed by double clicking on them. This shows the contents of the assembly and can be identified by the border that's drawn around the nodes. To each side of the display there are sidebars drawn in a darker shade. These contain any exposed input channels (to the left) or output channels (to the right). Clicking on the 'Add Channel' button in these areas will create a channel on the assembly which can be used as inputs or outputs to the assembly contents. If the assembly being viewed is part of a hierarchy, or a sub-assembly, the button in the toolbar with an arrow pointing upwards will allow you to view the immediate parent assembly. The toolbar button with the downwards pointing arrow will cause the view to change to the contents of any selected sub-assembly.

Creating Assemblies

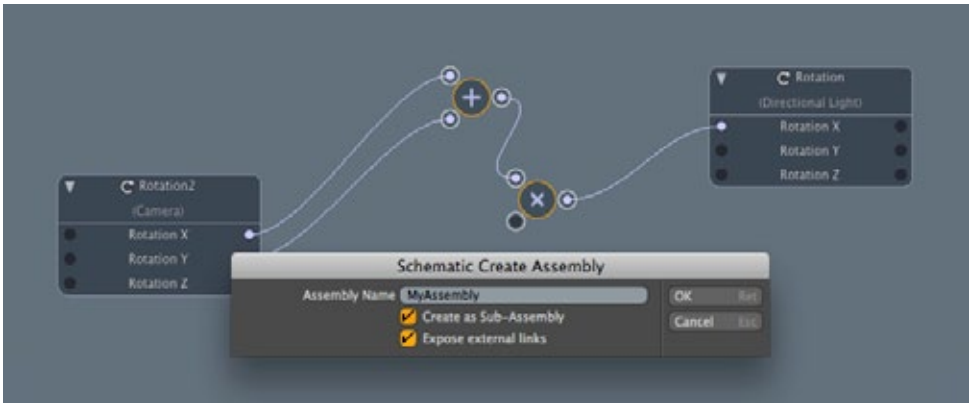


Once the user has a rig or even a portion of a rig setup, an assembly can be created from it. This is a good way of separating out parts of a rig for organization, allowing users the ability to work on each area independently, or to create a useful asset that can be saved out and re-used later. To create an assembly select each of the nodes you want to include and then right click on the background and choose "Create Assembly..." from the context menu. Provide a name in the dialog that's displayed. The new assembly can be created in two ways, as a normal assembly or as a sub-assembly, determined by the "Create as Sub-Assembly" option. When enabled this option will create an assembly that becomes part of the current group, otherwise a standalone one will be created. In both cases the selected nodes are removed and added to the new assembly. The last option, "Expose External Channels" is only applicable to sub-groups, this will create channels on the new group for each channel link to or from a channel within the group. This is very useful for creating assemblies that can be re-used elsewhere, a kind of 'black box' that can be connected up without knowing about the internals. When you click 'OK' the new assembly will be created, if it was a standalone assembly the view will change into the 'Assembly' view to show it's contents.

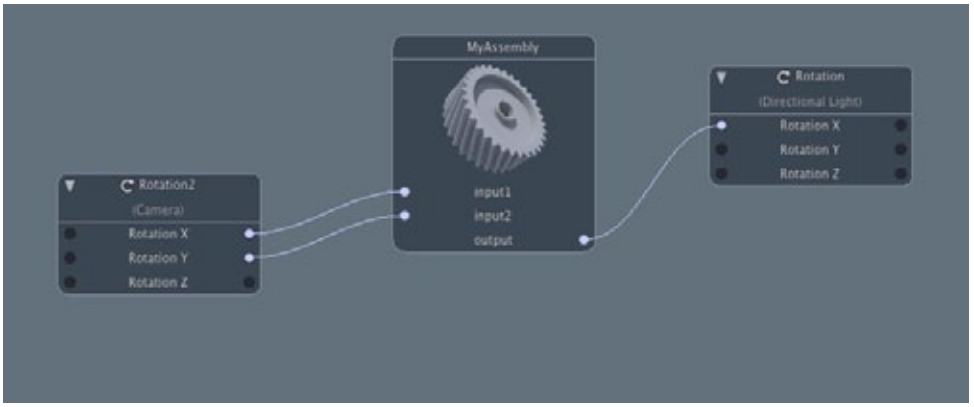
Creating Sub-Assembly Walk-through



Select the elements that make up the sub assembly, here it is the 'add' and 'multiply' nodes.



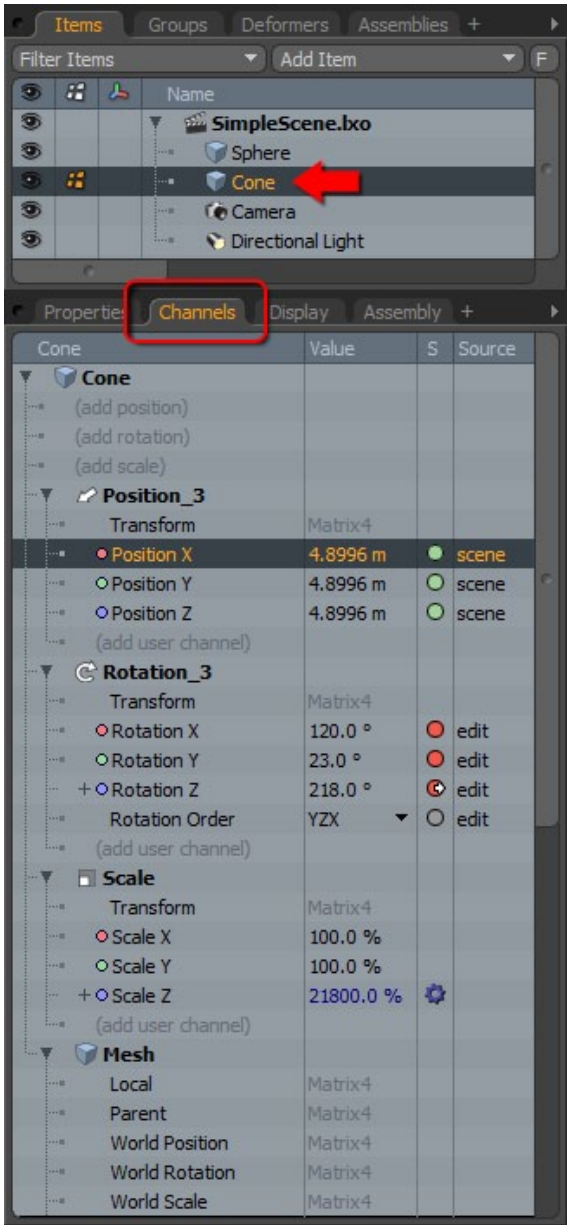
RMB+click on an open area of the workspace and choose 'Create Assembly' from the context menu.



Clicking 'OK' changes the selected elements into the Sub-Assembly.

The contents of any Sub-Assembly can be viewed by LMB+double clicking on the node.

Channels Viewport



The Channel viewport provides an exhaustive window into the soul of the modo animation system. All Channels of any item can be viewed in the Channel Viewport. Channels are simply *any* of the animateable attributes of an item (the identical attributes found in the Properties viewport!). The Channels viewport provides a direct method for selecting channels as well as creating, navigating and managing any keyframes on said channels. There are four columns on the Channel viewport; Channels, Value, Status and Source. Obviously the first column displays the channel names for the currently selected Item, the second column displays the value of the channel at the current scene time. The Status (S) column shows the state of the channel be it animated, static or actually on a keyframe. This is communicated via the same dot method used on the Channel keyframing control. The last column displays the 'Source' of the value, which means where the current value is stored, internally, these will be important to users when defining Render Passes and when using the skeleton tools to define rigs for animation.

Interacting with Channels--

Same as the 'Properties' viewport, the Channels viewport updates to reflect the current item selection. This can be items within the item list, the shader tree, or even the Groups palette. Once selected, all the animateable attributes appear in the Channels

viewport. To affect an attributes value, the user simple needs to LMB+click directly over the current value in the 'Value' column. For toggles (boolean values), this will change the value field between true/false. For attributes with various properties, a menu will open presenting all the possible options. For numerical values, users can use the click+pause action to activate the field to enter a new numerical value, pressing enter to set it, or simply clicking away form the field.

Setting Keyframes--

Users can set animation keyframes in all the usual ways using the keyframe buttons, the 'S' keyboard shortcut, in the 'Properties' viewport, or within the 'Graph Editor'. To set a keyframe for a value directly in the Channels viewport, users can RMB+click on the target value and choose the 'Add Key' option from the context menu. Users may also remove keyframes using this methodology. When any attribute has a keyframe, colored circles will appear next to the value in the 'S' (Status) column referencing the attributes current state. The possible states are--

- None- Channels is at the default value
- Constant- Channel has a constant value
- Animated- No keyframe at the current time position
- Animated- Keyframe at the current time position
- Mixed- Channels have differing values or state
- Driven- Channel is driven by an external source
- Undefined- Channel is not defined

An arrow over the dot represents that channel as driving another for a channel link.

- Driver- Channel is linked to another

A gear icon represent a channel that is being controlled by another for a channel link

- Driven- Channel is linked to another

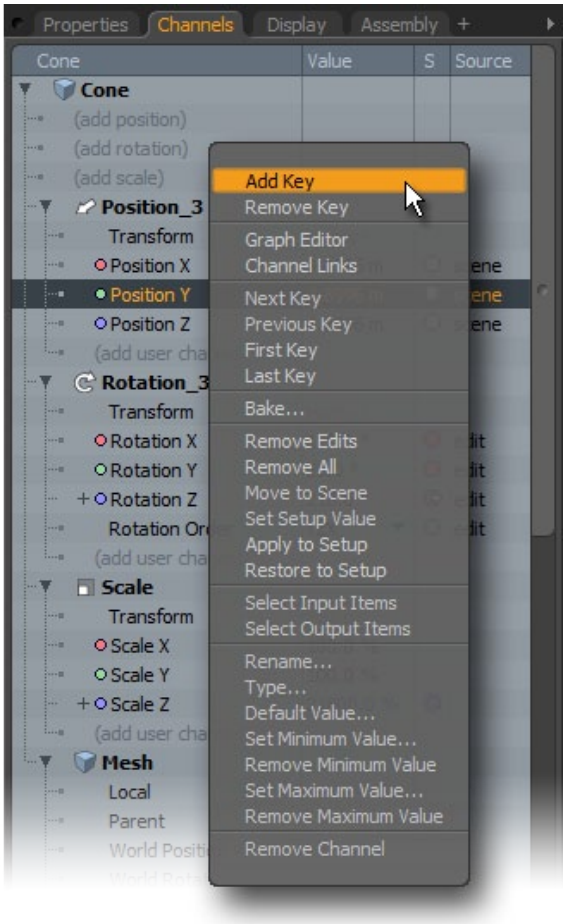
Users can automatically select input/output items by selecting the appropriate channels and pressing ' i ' to select *input*, and 'o' to select *output*.

Contextual Menu--

RMB+clicking in the Channels viewport opens up the contextual menu with options for modifying the keyframes of the various channels.

Add Key: Sets a keyframe at the current time with the current value.

Remove Key: Removes the keyframe at the current time.



Graph Editor: Opens the Graph Editor window. The Graph Editor represent the interpolations of keyframes, basically what happens in-between.

Channel Links: Opens the Channels Links palette, useful in having one channel control another automatically, but the functionality has been replaced by the superior Schematic View.

Next/Previous/First/Last Key: Moves the timeline to the appropriate time.

Bake: Bakes, or *freezes* the current animation to individual keys at every frame. Useful when exporting animations of driven items.

Remove Edits: Removes all keyframes, retaining the value of the initial key.

Remove All: Removes all keyframes and returns to the default value.

Move to Scene: Takes the current channel value and moves it to the 'Scene' level, the default base level value used when working with Render Passes.

Set Setup Value: Used only in Setup mode to tell modo that the values for this channel should be kept separate between 'Setup' and 'Scene' actions. So when the user is editing a channel out of 'Setup' mode it won't affect the setup value (so it has a value in Setup and a different one in the Scene action). This can be used for example to hide certain rig elements when animating but have them visible in setup mode without having to toggle visibility all the time.

Apply to Setup: Takes the current channel value and set it as the default 'Setup' value, essentially the resting value of an item prior to being deformed.

Restore to Setup: Removes the existing value for the channel reverting it back the the 'Setup' state.

Select Input Items: For driven channels, this option selects the driver item or channel.

Select Output Items: For driven channels, this option selects any items or channels being driven by the current channel.

The following options are relevant only to User Created Channels.--

Rename: Opens a dialog box to Rename the selected user channel.

Type: Opens a dialog box allowing users to modify the type of user channel.

Default Value: Opens a dialog box allowing users to sp ecify a default value for the user channel.

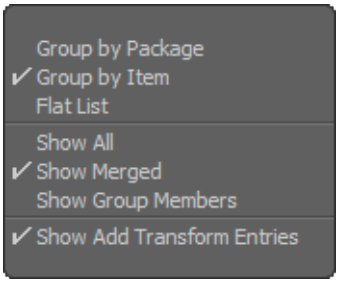
Set Minimum Value: Opens a dialog box allowing users to set a minimum value for the user channel.

Remove Minimum Value: Removes any minimum value, if specified.

Set Maximum Value: Opens a dialog box allowing users to set a maximum value for the user channel.

Remove Maximum Value: Removes any maximum value, if specified.

Remove Channel: Deletes the selected user channel.



Context Menu--

RMB+clicking on the 'Channels' tab will open the viewport menu. Select the 'Viewport Settings' to open the Channels viewport options menu.

Grouping: The first three options define how channels are grouped in the channel list

Group By Package- Organizes channels by *package*, sub-groupings of related elements by individual item.

Group By Item- Organizes channels strictly by item.

Flat List- Channels listed as straight list with no grouping.

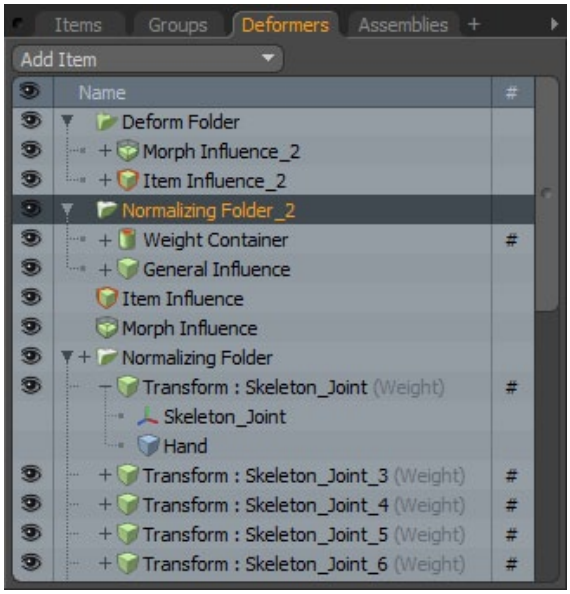
Show All: When *enabled* shows all channels on an item, when disabled certain less used channels are hidden such as Matrix channels.

Show Merged: When *enabled*, combines all the channel values for all selected items, instead of listing each item's channel one after another. Useful when gang editing multiple items at once.

Show Group Members: When enabled, show the members of any selected Group.

Show Add Transform Entries: Displays the 'Add Position/Rotation/Scale' entries to the channel list. These options provide a shortcut to users for adding additional transform channels, but for smaller monitors removing them can save some screen real-estate.

Deformers Viewport



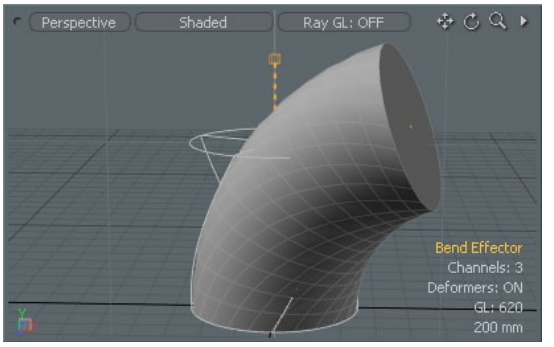
The Deformers viewport provides a view of all the Deformers in a scene. More accurately, it provides a view of the Deformers and Influence items, allowing user to select, reorder, rename and remove them. Deformations in modo require two items to work properly, the deformer itself and an influence, the Deformer describes how something is modified, and the Influence defines *what* gets modified. The Deformers viewport is found in the 'Setup' and 'Animate' interface tabs of the default interface in the upper right side panel bank. Selecting any of the items in the list will display their associated attributes in the 'Properties viewport. For organizational purposes, users can also group together multiple elements into a folder by RMB+clicking on one or more selected elements and selecting the 'Place in Folder' option from the contextual menu.

To select any of the items in the list, simply LMB+click on the item layer name, selected layers will be denoted by the dark bar highlighting the layer, subsequent layers can be added or removed from any selection by pressing the 'Ctrl' key and LMB+clicking on the layers. Users can click+hold and drag layers to reposition them within the list. Using the standard modo click, pause click method layer names can be edited inline, and removed by pressing the 'Delete' key on the keyboard.

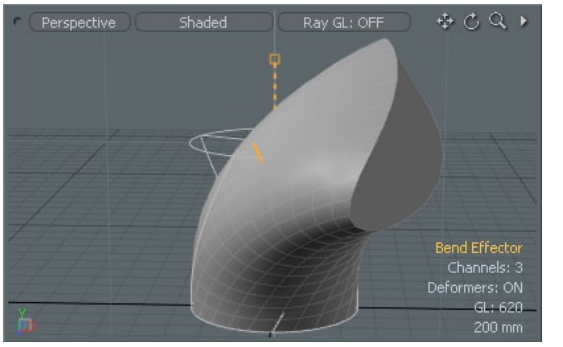
Order of Operations

Being able to see the order that Influence items fire (calculate) is important as the order of operations can have a dramatic effect on the final deformed outcome when multiple deformers are applied to a single mesh. Take for instance a twist and a bend

applied to a cylinder, if the cylinder is twisted and then bent, the results would be very different than if the item were bent and then twisted. The modo order of operations is determined by layer order starting at the bottom of the list moving up. Lower deformers are calculated and then the values are passed on to the next deformer in the list, and so on.



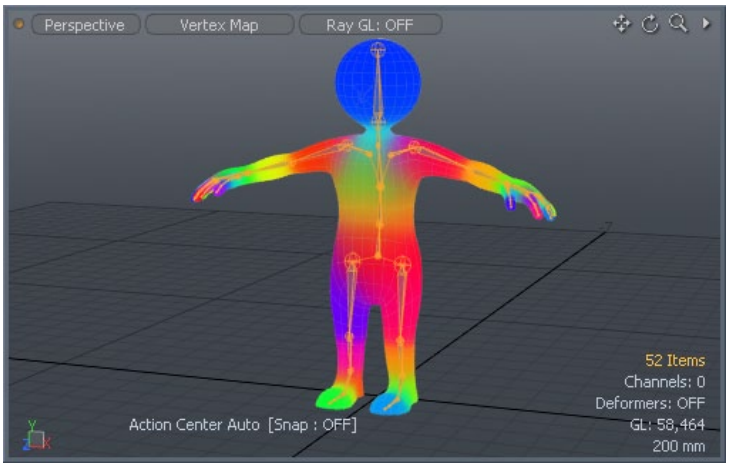
The twist applied first and then the bend

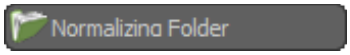


The bend applied first and then the twist

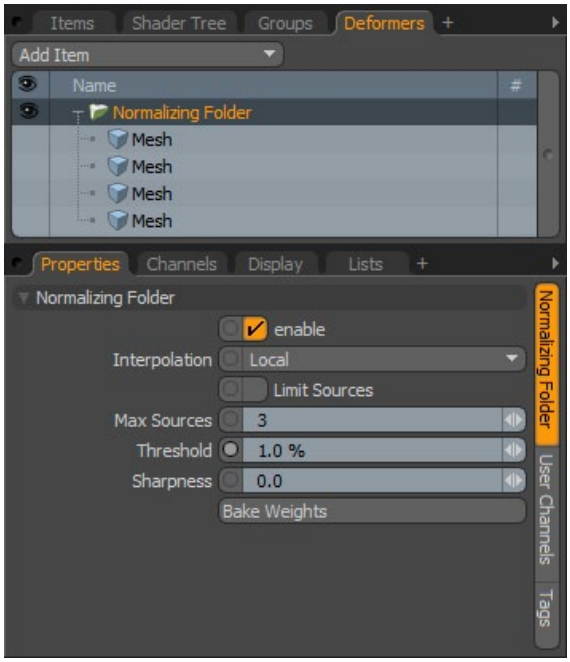
Normalization

Be selecting the 'Normalization' folder directly, a special weight drawing mode is enabled, allowing users to more easily see how overlapping weights affect each other. User mush have the 'Vertex Map' display style applied to the viewport in order to see the weights.





Sometimes you don't want an order of operation but instead you want everything to calculate at once, such as with joints applied to a character. Mathematically, if a vertex between two bones was weighted 50% for each ones influence, one fired right after the other wouldn't add together to produce 100% of the deformation. To resolve this, users can add a 'Normalization' folder and place any Deformers and Influences into it, any items within the folder will have their weight values *normalized*. What this means is that when multiple weight map values are influencing a single vertex, all the values are adjusted so that their combined values add up to a 1.0 weight value(100%), this way no vertex will get more than 100% deformation, no matter how many items are influencing it, additionally the normalization folder makes sure that deformers are calculated concurrently, so that when those two 50% weighted influences will actually produce a 100% deformation. Normalization is typically applied to deformations generated using skeleton joint, where multiple weight mapped deformers can influence a single vertex. Luckily, most users won't ever need to think about this as the Normalization is added automatically during the 'Bind' operation. For more information on the Bind & Setup process, please reference that page of the documentation.



Normalizing Folder--

Enable: Toggles the affect of the Normalization Folder over its contents On or Off. When un-checked (disabled), the layer has no affect over the Influences. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Interpolation: The 'Interpolation' option allow users to determine how the Normalization interpolates each Influence-

Local- Each influence does it's own interpolation, whether that's 'Parametric' or 'Linear'.

Linear- Overrides the Influences and does 'Linear' for all. Same as if influences were all individually set to 'Linear'.

Quaternion- Implements dual quaternion skinning as the method of interpolation. Only works if all the deformers are Transform Effectors (Bend, Lag etc. are not supported).

Limit Sources: When enabled, the 'Max Sources' attribute sets the upper bound on the number of influences that can affect a single vertex. It picks the 'X' amount of the highest weights and ignores the rest (Good for Heat binding which creates lots of very small weights, and for Unity which can handle a max of four joints per vertex).

Threshold: Any Weight values below this defined threshold are ignored. Also good for cleaning up Heat binding.

Sharpness: This can fine-tune the strength of the binding. Positive values make vertices bind more strongly to nearby bones, and negative values make them more permissive and bind to weaker bones.

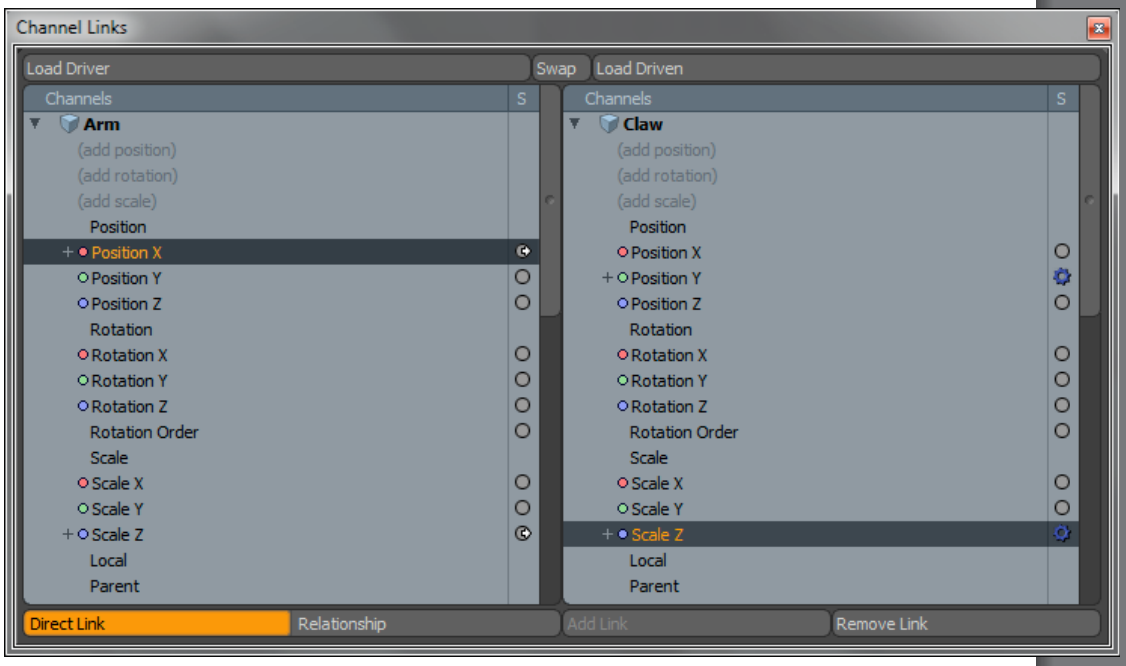
Bake Weights: The previous three options are computed on the weights on the fly during normalization and can be changed on the fly. Bake Weights pushes those computed weights down to weight maps.

Channels

Channel Links

The 'Channel Links' viewport allows users to define and manage linked channels. 'Channel Links' are a means of allowing the property of one item to control the property of another.

If one selects 2 items prior to opening the viewport, the Driver/Driven channels will automatically populate. Selection order is important as the first item selected becomes the driven channel while the second item selection becomes the driver. One can also load an item into a particular column by selecting the appropriate item in any 3D viewport, or the item list and clicking the 'Load Driver/Load Driven' buttons at the top of the Channel Links viewport. The 'Swap' button will switch the driven item with the driver.



Defining a link between two channels is simple. Select any two channels, one from each column, choose the type of link you wish to define, 'Direct Link' or 'Relationship' and press the 'Add link' button. The 'S' column (Status) will update to reflect the link. The 'Driving' channel will display a tiny arrow over the keyframe indicator dot indicating its controlling state, while the 'Driven' channel will display a small gear indicating it is being controlled by another channel. To remove a link, simply select any 'Driven' or 'Driver' channel and click the 'Remove Link' button.

Direct Link

When a direct link is defined, the value of the driving channel directly controls the value of the driven channel. Take for instance the example above, the 'Position X' of the locator is driving the 'Scale Z' of the mesh item. Positioning the locator item at 1m on the X axis scales the Z axis on the mesh item to 100%, at this position, it appears to do nothing, but move the X position of the locator to 2 meters, and the mesh item will scale to 200%. Once the link has been specified, only the driving item will need to be keyframed.

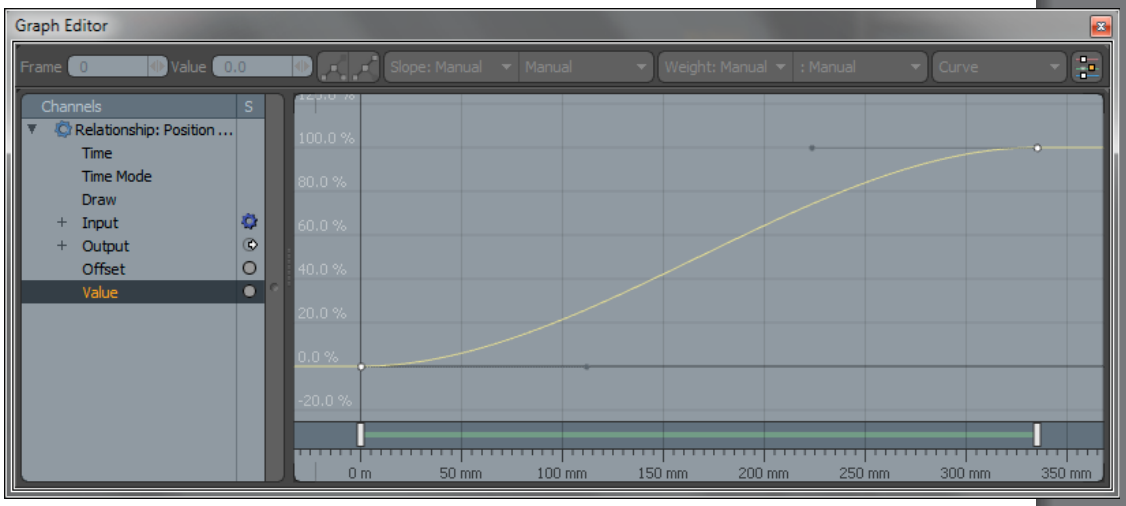
Relationship

A 'Relationship' is very similar to a 'Direct Link', however, the link itself can be non-linear and is defined by a graph. Unlike a 'Direct Link', where once applied, modifying

the driver instantly affects the driven item, once a 'Relationship' is linked, the relationship itself must be defined numerically in the relationship interface window that appears in the 3D viewport. if you don't see the window right away, you might need to select the relationship modifier item in the item list.

The window provides all the necessary information regarding the channel relationship, the 'Driver' channel is listed on the top, while the 'Driven' channel is on the bottom. The linked channel names appear in the darker area, LMB-clicking and dragging over the channel name will adjust the value interactively, and LMB-clicking on the value itself will allow the user to numerically input the value using the keyboard. The four round buttons in the middle allow the user to define the value relationship with set values called keyframes. The red X deletes a keyframe, the white square creates a keyframe. The two small arrows allow the user to easily jump between keyframes.

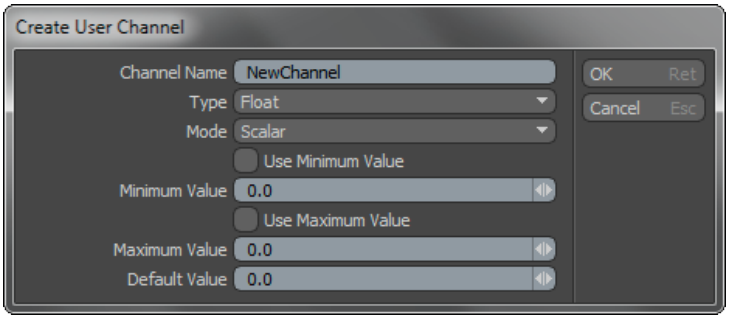
In the above window, the X Position channel is linked to the Scale Z channel. If you set a value keyframe at 0 meters for position and 0% scale, that means that when the driver is positioned at 0 meters in the X direction, there will be no scale change on the Z axis for the mesh item. Next, using the mouse to click on the scale value, enter 100%, and on Position X enter 10 meters. Once the values are entered, set another keyframe with the white square. That effectively sets the relationship, now when the first item moves to 10 meters in the X direction, the Z scale of the mesh item will automatically scale to 100%. If you'd like to visually see the relationship curve, you can use the 'Graph' button to open the Graph Editor.



You can see right off that the keyframes generated were placed into the 'Value' channel of the relationship modifier item. The driver values appear along the length, with the driven values the height. The yellow line is the graph of the relationship between the two channels; like any other curve, it smoothly interpolates between values. Modifying the slope and weight of the keyframes will modify the in-between interpolation values. For more on modifying graph editor keyframes, see the 'Graph Editor' section of the documentation.

Add User Channels

There are a couple ways users can add custom channels to an item for the purposes of animating. Users can use the menu bar command under "Animate > Channels > Add User Channel", or users may also click the button in the 'Modifiers' subtab of the Animate interface tabs default toolbox. Additionally, modo provides options to add user channels in the 'Channels' viewport, both in the viewport itself (as an option) and and from within the context menu. Invoking the 'Add User Channel' command pops open the 'Create User Channel' dialog. These are additional channels that can be added to items, often used to add settings or controls to assemblies\rigs and\or to drive multiple other channels from a single channel via channel linking. Select the item you wish to add the channel to before entering the dialog box.



Channel Name: Defines a name that will be used to identify the new channel.

Type: Choose the 'Type' of channel you wish to create from several options

Float- Channel produces a floating point value e.g. generally a decimal and typically between 0 & 1

Distance- Channel produces a distance output value.

Angle- Channel produces an angle output value

Percent- Channel produces a percentage value e.g. 0% - 100%

Integer- Channel produces a whole number value (not a fraction) e.g. 1,2,3,4,5...

Boolean- Channel produces only a 'True' or 'False' type value

Mode: By default a single channel will be created with the given name, but multiple channels can also be created by using the options of the Mode popup, they are as follows:

Scalar- Creates a single channel, e.g. MyChannel

Vector XY- creates two channels; e.g. MyChannel.X and MyChannel.Y

Vector XYZ- creates 3 channels; e.g. MyChannel.X, MyChannel.Y and MyChannel.Z

Vector RGB- creates 3 channels; e.g. MyChannel.R, MyChannel.G and MyChannel.B

Vector RGBA- creates 4 channels; e.g. MyChannel.R, MyChannel.G, MyChannel.B and MyChannel.A

Use Minimum Value: Allows the user to specify and define a minimum value

Use Maximum Value: Allows the user to specify and define a maximum value

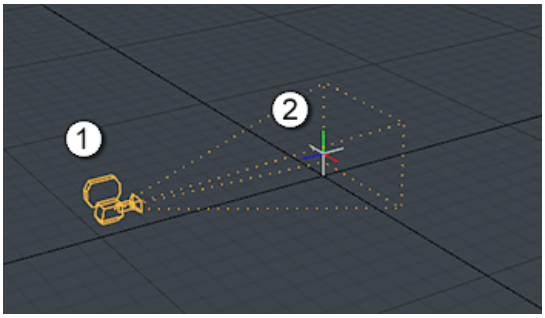
Default Value: Allows the user to specify the default value of the channel.

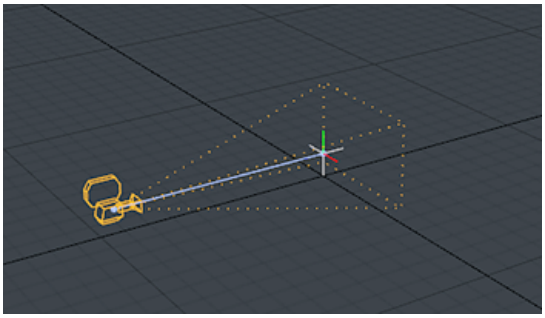
Applying Constraints

Constraints at their most basic are controllers, controlling how one item relates to another. Constraints can serve many purposes, and when used in combination with each other can create very complex controllers and motions. modo's constraints are weighted, meaning that a weight value can determine the influence a constrain has over the item it's controlling. Constraints, once applied, can be selected by toggling (LMB clicking) the small '+' sign expansion widget that appears next to the constrained item in the Item List. Once selected, its properties can be edited via the 'Properties' viewport or within the 'Channels' viewport. The 'Animate' tab of the default interface, under the modifiers sub tab offers a simple means of applying constraints to an item.

Applying Constraints--

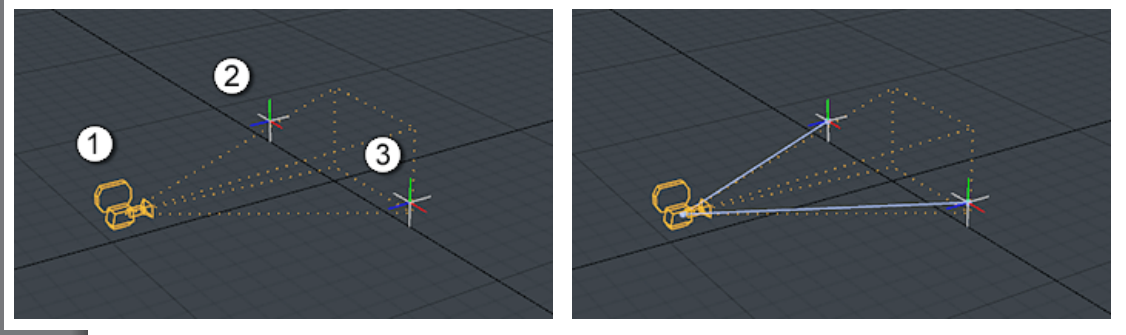
There are three basic methods for applying constraints, the default method creates the appropriate constraint item(s) from item selections. The simplest constraint is between two items. For this example, below, the Camera has been selected first, followed by the Locator item (once an item has been selected, hold down the shift key to select subsequent items). Now clicking on the 'Direction' constraint button applies the constraint. Once applied, the locator will control the direction of the camera; wherever the locator moves to, the camera will point directly at it.



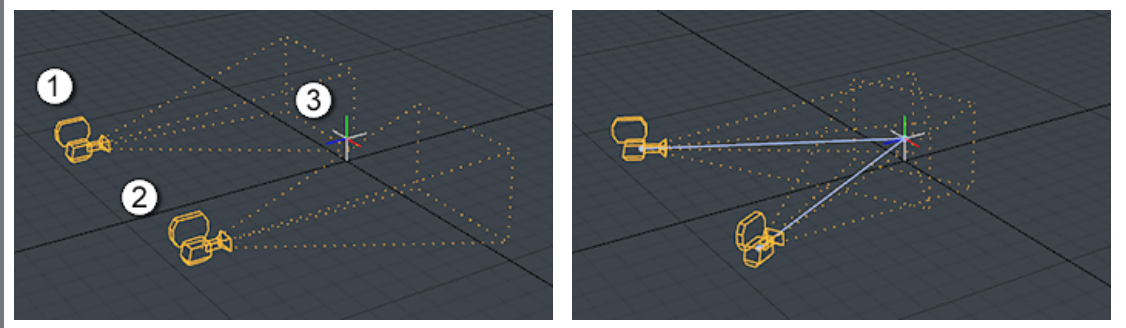


The second method provides a means for multiple items to control another. For this example, the camera was selected first, followed by the two locators. Clicking on the Direction constraint button applies the constraint so the position of both locators is used to determine the orientation of the Camera. In this case aiming towards a point half way between the two locators, and average of the two positions. The influence of one item over the other can be controlled by adjusting the weight amount, tending

the camera to face more toward the item with higher weight (influence). This can be adjusted in the constraints properties viewport or channel list.



The third method provides a means for multiple items to be controlled by a single item. For this example, the two camera items were selected in sequence, followed by the locator item. The difference this time, is the constraint is applied by holding down the 'Ctrl' key, the button will change to say 'Direction (multiple)' (signifying that *multiple* items will be affected by the one controller). Click the button to apply the constraint. Once applied, you can see that each camera now independently orients toward the one locator.



Removing Constraints--

Constraints can be easily removed from any item, by first selecting the constrained item, the holding down the 'Shift' key, the Constrain buttons will then change, for example the direction constrain button will now read 'Remove Direction'. Click the button to remove the constraint from the selected item. Additionally, the constraint can be selected below the constrained item, (click the small '+' sign expansion widget if you can't see it) and by RMB+clicking on the constraint itself, one can select 'Delete' from the drop down menu.

Constraint Weights--

When an item is constrained to multiple items, a weight channel is added to the constrained item, one for each of the constraining items. The weights control how much influence a particular item has on the item being constrained. Weights can be adjusted and animated so the effect of the constraint can be biased towards (or away from) a particular item's input. Setting a weight channel to 100% means that the constraining item will exert full influence on the constrained item, a value of 0% means the constraining item will have no influence on the constrained item. To access the weight channels for a constraint, select the constraint item in the Item List. The weight channels are listed at the end of the constraint item's channels in the Channel List viewport. Values can be manually entered here or the Channel Haul tool provides a means to interactively adjust the values in the 3D views (select the weight channels and press the 'C' shortcut key or the button in the Modifiers tab).

Constraints

Constraints at their most basic are controllers, controlling how one item relates to another. Constraints can serve many purposes, and when used in combination with each other can create very complex controllers and motions. modo's constraints are weighted, meaning that a weight value can determine the influence a constrain has over the item it's controlling.

Transform Constraints	Geometry Constraint	Path Constraint	Intersection
Position	Vertex Position	Path	Line 2 Items
Rotation	Vertex Normal	Path Position	Linear Curve
Scale	Edge Position	Path Normal	Plane
Direction	Edge Normal		Surface
Distance	Polygon Position		
	Polygon Normal		

Position

A 'Position' constraint allows an item to follow the world position of another item or a weighted average of the positions of multiple items.

Rotation

A 'Rotation' constraint allows an item to follow the rotation of another item or a weighted average of the rotations of multiple items.

Scale

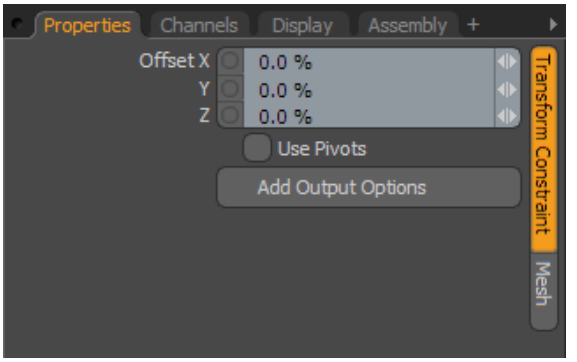
A 'Scale' constraint allows an item to follow the scale of another item or a weighted average of the scale of multiple items.

Direction

A 'Direction' constraint forces an item so that is always points toward the world position of another item.

Distance

A 'Distance' constraint sets a fixed, 'Minimum' or 'Maximum' distance away from another object that an item can travel.



Transform Constraints

Offset: These channels define an offset amount that is applied after the constraint, expressed as an XYZ value for position, a number of degrees for rotation and a percentage for scale. When the constraint is applied with 'Compensation' enabled, these channels are set to values that will maintain the item's original transform values, but they can also be used to adjust the constrained item's transform if required.

Use Pivots: Directs modo to use an items 'Pivot' position, instead of the default 'Center' position.

Add Output Options: This button will add a Channel Effect modifier to each of the constraints outputs, described in more detail in the *Modifiers* section, but in short

these allow you to blend the effect of the constraint with the item's underlying Forward Kinematics and\or restrict the constraint to one or more axes. Note: If the item being constrained is parented, then the Channel Effect modifier requires a channel link to be added from the constrained item's parent world position or rotation channel (depending on the type of the constraint) into the Channel Effect's 'Local' channel.

Vertex Position

The 'Vertex Position' constraint attaches an items center to a deformed vertex position of a mesh item.

Vertex Normal

The 'Vertex Normal' constraint matches an items orientation to that of a specified vertex's normal direction. A vertex derives its normal direction as an average of all adjoining polygons.

Edge Position

The 'Edge Position' constraint attaches an items center to a deformed edge of a mesh item.

Edge Normal

The 'Edge Normal' constraint matches an items orientation to that of a specified edge's normal direction. An edge derives its normal direction from an average of its two bordering polygons.

Polygon Position

The 'Polygon Position' constraint attaches an items center to the center of a deformed polygon of a mesh item.

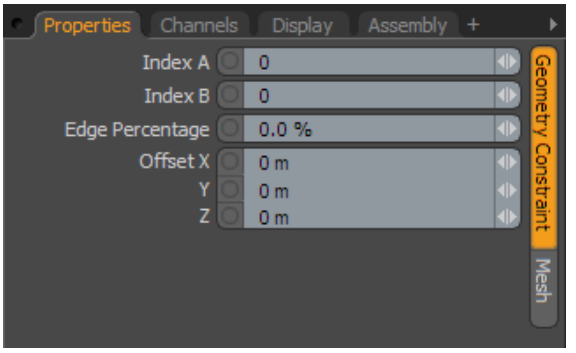
Polygon Normal

The 'Polygon Normal' constraint matches an items orientation to that of a specified Polygon's normal direction.

Geometry Constraints

Index A: This is the component index for vertices and polygons that the constraint will use to get position and normal information from. For edges this is the index of the first vertex that defines the edge. By default modo will pick the component index that is closest to the center of the item being constrained. To find the component index value

for a particular component (vertex, edge or polygon), make the appropriate selection of the item you wish to constrain to. The value is now revealed in the 'info' form within the 'Info and Statistics' panel.



Index B: This is only used for edges and specifies the second vertex to use for the edge. Note that this can actually be any vertex in the same mesh, they are not required to be connected.

Edge Percentage: As the name suggests this is only used for Edge constraints, it allows the constraint to use an interpolated position or normal between the first and second vertices that define the edge. This can be used to slide the constrained item along the edge for example, 0% being the start of the edge and 100% being the end of the edge.

Offset: This is a world offset usually set when the constraint is applied with compensation on but can be used for further adjustment if required.

Path

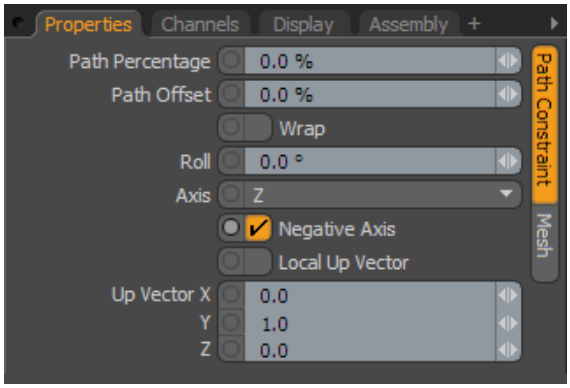
The 'Path' constraint will bind an item to a user generated curve or the weighted average of multiple curves affecting both its position and rotation values orienting it to face along the curves normal direction. Once constrained, the 'Percent' amount will control the items position and rotation along the length of the curve. This setting can be animated for items moving along the path.

Path Position

The 'Path Position' constraint will bind an item to a user generated curve or the weighted average of multiple curves affecting its position values only. Once constrained, the 'Percent' amount will control the items position along the length of the curve. This setting can be animated for items moving along the path.

Path Normal

The 'Path Normal' constraint will match an items rotation to the normal direction of a particular position along the curve. Once constrained, the 'Percent' amount will define the items rotation based on that positions normal direction.



Path Constraints

Path Percentage: Defines a position along the path to which the item will be constrained, 0% being the start of the curve (represented by the small circle around the first curve vertex) and 100% the end of the curve. Paths can be open or closed.

Path Offset: This control can be used to offset the point along the curve from the position specified by the Path Percentage control above. This can be used to specify a starting offset along the curve or to animate some variation into the constraint position for example.

Wrap: By default, constrained items will stop when they reaches either end of the curve, when enabled, 'Wrap' allows the item to continue it's motion from the start of the curve. For example, if the Percentage channel is set to animate from 0% to 200% then the constrained item will travel along the curve twice.

Note: *The following controls are not used by the Path Position constraint.*

Roll: This control specifies a rotation around the axis that's aligned to the curve.

Axis: This is the axis of the constrained item that will be orientated along the curve.

Negative Axis: Inverts the defined 'Axis' from the default positive direction, toward the negative direction.

Local Up Vector: By default the Path and Path Normal constraints will use the values in the Up Vector controls as World coordinates. However this checkbox allows you to specify that these should be interpreted as Local coordinates relative to the transforms of the Curve mesh item. This is useful if the Curve mesh is animated or transformed in any way.

Up Vector: When 'Local Up Vector' is enabled, this XYZ value will be used to specify the Up Vector direction as used in the Path and Path Normal constraints.

Intersection Constraints Types

The Intersect constraints provide the means to determine closest point and ray intersections with lines, planes and surfaces. They are slightly different from other

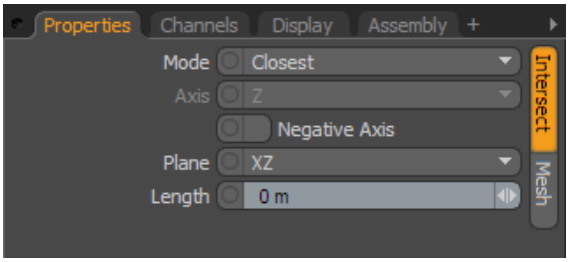
constraints in that a new constrained item is created when the constraint is applied, rather than being specified in advance from the selected items.

Intersect Line-Two Items: This creates a new locator at the closest point along an infinite line, defined as running through the world center positions of two selected items.

Intersect Line-Linear Curve: This creates a new locator at the closest position along a user generated *linear* curve, meaning a curve defined by only two vertices.

Intersect Plane: This constraint creates a new locator at the closest point (by default) from the first selected item to an infinite plane derived from the world axes of the second selected item. The default axis plane can be changed by adjusting the 'Plane' setting.

Intersect Surface: This constraint can be used to project a closest point or ray intersection from an item to a mesh surface. Select the item that will be the source of the intersection followed by a mesh item. A new locator will be created at the closest point on the surface from the driver item.



Intersection Constraint

Mode: Specifies whether a 'Closest' point or 'Ray' intersection is performed to determine the items position along the line. Closest point is exactly that, the closest point along the line to the constrained item. A ray will intersect the line along the chosen axis, taking in to account the items position and rotation transform values.

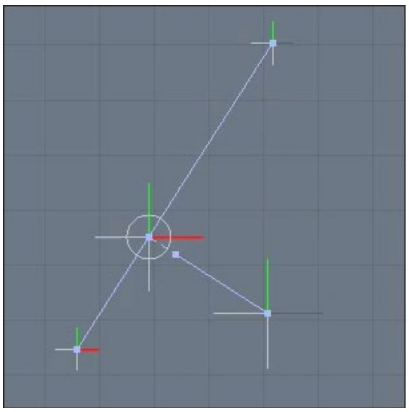
Axis: When the 'Mode' is set to 'Ray', this control defines which Axis channel on the driver item is used to project the ray.

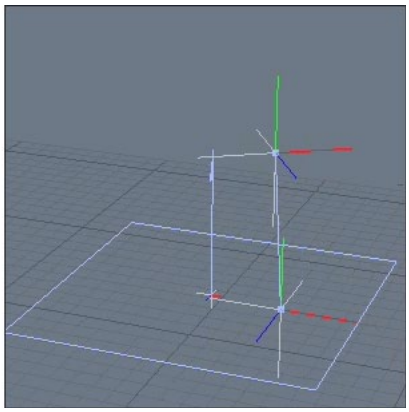
Negative Axis: Inverts the defined 'Axis' from the default positive direction, toward the negative direction.

Plane: For a Plane intersection, this value defines the plane.

Length: The length value is slightly different when used with a line intersection or plane intersection. When used with a line intersection, the 'Length' value can be used to specify a *minimum distance* between the driving item and the intersection point. If the driving item is located further away from the line than the distance set in this channel, then the closest distance will continue to be used. However, if the distance is less than

the length value, then the intersection will be 'pushed' along the line, maintaining the specified length between the driver and the intersection point. When used with a Plane intersection, the direction in which the intersection point is pushed is determined from the positive Z axis of the driving item. An example of each is shown below.





Animation Modifiers

Animation modifiers provide users a way to procedurally control elements within a scene without requiring direct keyframe values . Useful functions such as keeping elements centered between others, or forcing an element to always face toward the camera are options possible with modifiers. Most modifiers apply as one-shot commands, meaning users make a selection to determine the elements to affect, and then LMB+click the command button to apply the modifier producing the result.

Channel Modifier	Conditional	Math	Matrix
Color HSV	Logic	Basic	Matrix Blend
Color Kelvin		Math Multiple	Matrix Compose
Channel Noise		Trig	Matrix Vector
Channel Oscillator			

Measure	Vector	Other	
Measure Angle	Vector	Dual Joint Planar IK	Revolve

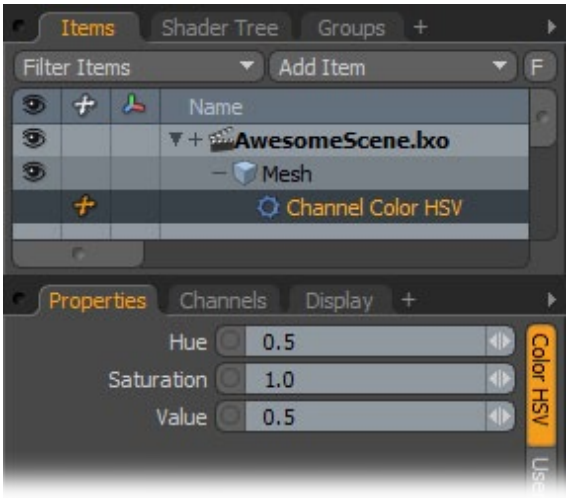
Measure Distance

Vector Maths

Dynamic Parent

Simple Kinematics

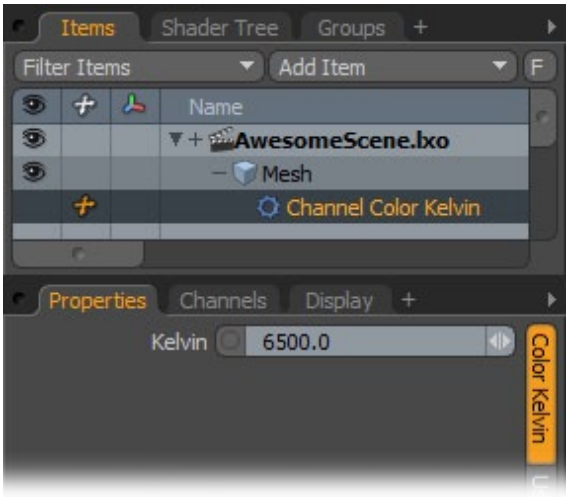
Linear Blend



Channel Color HSV

This modifier produces an RGB color (3 channels) output, from an input color defined by hue, saturation and value. This allows the user to smoothly animate the hue, saturation or value of a color with correct interpolation, rather than making excessive keyframes for red, green, and blue channels individually.

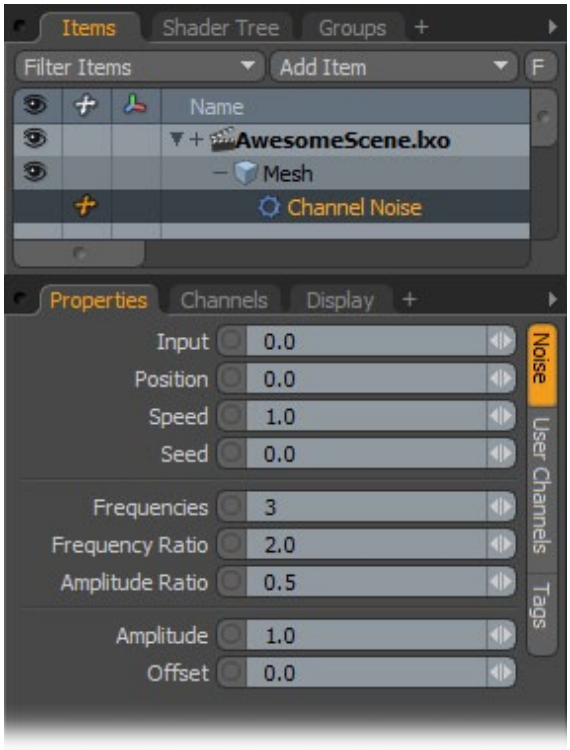
Hue/Saturation/Value: These options determine the color value using the HSV color model. Their RGB output values can be linked to a materials RGB value input channels to control the materials colors with the modifier.



Channel Color Kelvin

This modifier produces an RGB color (3 channels) output, from an input color defined by Kelvin color temperature scale based on thermal dynamics. This allows the user to smoothly animate the color of the sky at sunset, for instance, with correct interpolation, rather than making excessive keyframes for red, green, and blue channels individually.

Kelvin: These options determine the color value using the Kelvin color model. Their RGB output values can be linked to a materials RGB value input channels to control the materials colors with the modifier.



Channel Noise

This motion generator creates smoothly changing random value by taking a 1D slice through a 3D fractal noise field.

Input: x-coordinate for slice

Position: y-coordinate for slice

Speed: scale time to make z-coordinate for slice

Seed: random start time to move z-coordinate

Frequencies: number of 'octaves' of noise to add together

Frequency Ratio: ratio of frequencies of successive octaves

Amplitude Ratio: ratio of amplitudes of successive octaves

Amplitude: Scaling factor for the output

Offset: Offset added to the output

Channel Oscillator

This modifier can be considered more of a motion generator, since it simply generates the standard sine wave so common in nature. When applied to a channel, this modifier creates smoothly undulating values. Use a 'Channel Noise' generator for a more random effect.

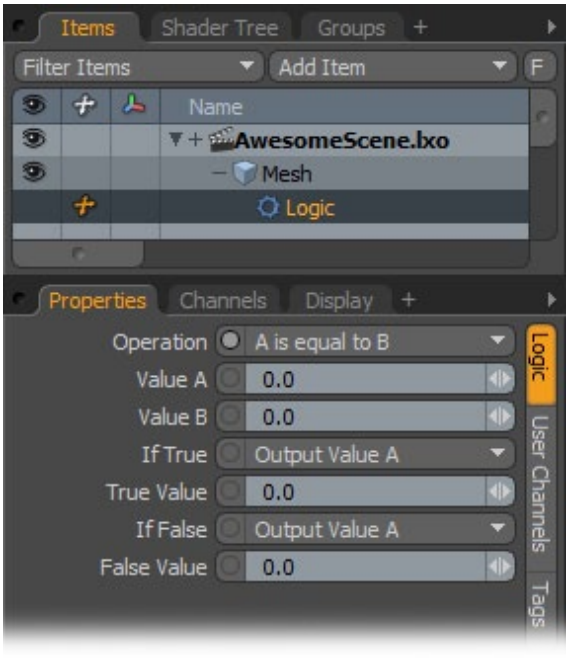
Amplitude: Size of base waveform without damping. The channel will range from -amplitude to +amplitude.

Frequency: Frequency of wave in cycles per second (Hz)

Phase: Phase shift for wave in form of a percent of a wavelength/cycle (i.e. 90 deg shift = 90/360 = 25%)

Damping: Damping amount for wave expressed as fraction of amplitude lost per cycle

Offset: A 'DC' offset added to the wave. The value will range from 'offset - amplitude' to 'offset + amplitude'.



Logic

The Logic modifier allows users to apply logic to channel operations.

Operation: Users can choose from the various logic operation options available-

A is equal to B

A is greater then B

A is greater than or equal to B

A is less than B

A is less than or equal to B

A is not equal to B

And

Or

Value A/B: These fields determine the values of 'A' or 'B'. Typically users will connect the output of another items channel to these channels for the logic operation.

If True: Users can decide what is output when the logic is 'True'

Output Value A

Output Value B

Output True Value

True Value: Value for when 'Output True Value' is selected.

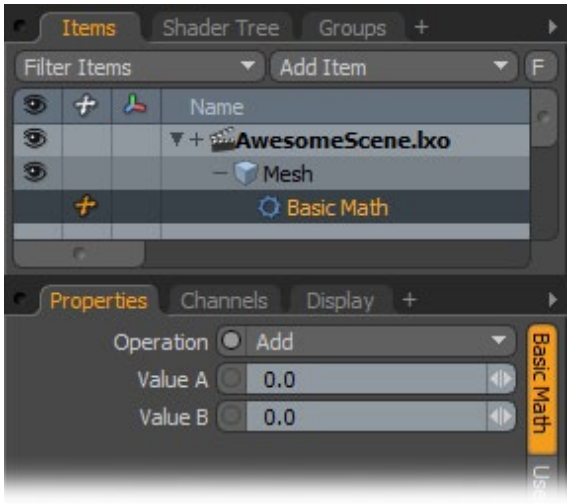
If False: Users can decide what is output when the logic is 'False'

Output Value A

Output Value B

Output False Value

False Value: Value for when 'Output True Value' is selected.



Basic Math

This Channel Modifier allows basic math operations to be performed and the result output to a channel that can then be read by other Channel Modifiers, or linked to a compatible (i.e. numeric) channel on any item. The inputs can be constant or animated values on the Value A and\or Value B channels of the modifier itself or values from other channels can be linked in. In the descriptions below the channel values for the inputs are called A and B, and the output value is C.

The math operation available are:

Operations: Users can choose form the various nasic math calculations-

Add- Value A + Value B = Output C

Subtract- Value A - Value B = Output C

Multiply- Value A * Value B = Output C

Divide- Value A / Value B = Output C

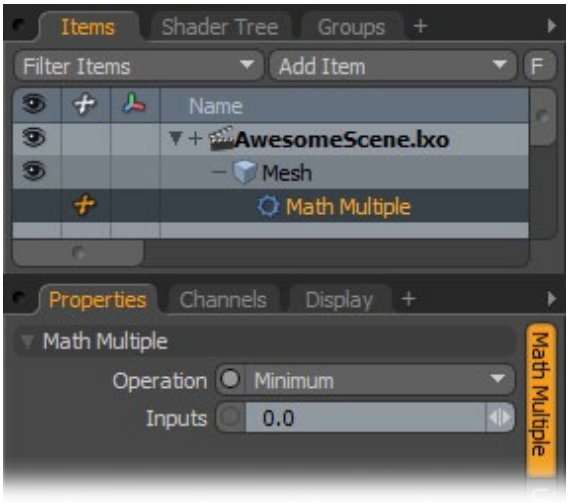
Modulo- The remainder of Value A / Value B = Output C

Average- The average of Value A and Value B = Output C

Minimum- The lesser of Value A and Value B = Output C

Maximum- The greater of Value A and Value B = Output C

Value A/B: User can connect the input of others channels into this modifier to calculate the equation and link the output channel to to the target channel.



Math Multiple

Users can define multiple inputs for the math operation and then output the result using the Result (Output C) output channel.

Operations: Users can choose form the various nasic math calculations-

Add- Value A + Value B = Output C

Subtract- Value A - Value B = Output C

Multiply- Value A * Value B = Output C

Divide- Value A / Value B = Output C

Modulo- The remainder of Value A / Value B = Output C

Average- The average of Value A and Value B = Output C

Minimum- The lesser of Value A and Value B = Output C

Maximum- The greater of Value A and Value B = Output C

Inputs: This is the input value for the numeric calculation, typically this value is linked from another channel.

Math

Additional Math operators are available for calculating different values as necessary. Values can be connected to the 'Input' channel, with the calculations results fed out of the 'Output' channel.

Square Root- Calculates the Square Root of the input value.

Exp- Calculates the Exponent of the input value, based on the Exponent value (which can also be used as an input).

Log- Calculates the natural logarithmic value.

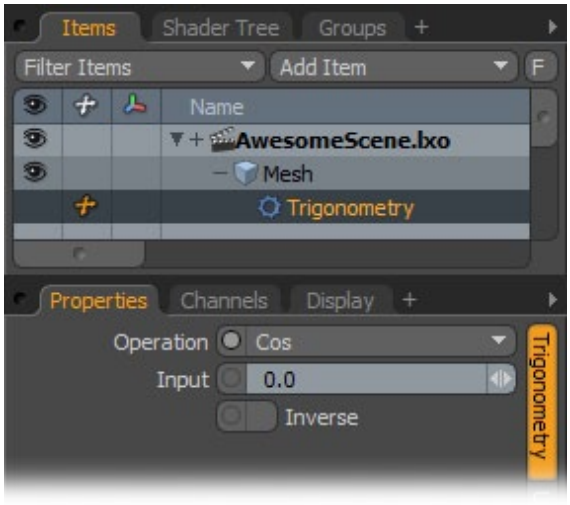
Log10- Calculates the common logarithmic value.

Absolute- Calculates the Absolute value for a number regardless of its sign, for example 3=3 and -3=3.

Power- Calculates the exponential value for a number based on the input Power.

Ceiling- Calculates a maximum integer (whole number) value.

Floor- Calculates a minimum integer (whole number) value.



Trigonometry

Users can apply triginometric operations to a channel. They are used to relate the angles of a triangle to the lengths of the sides of a triangle.

Operation:

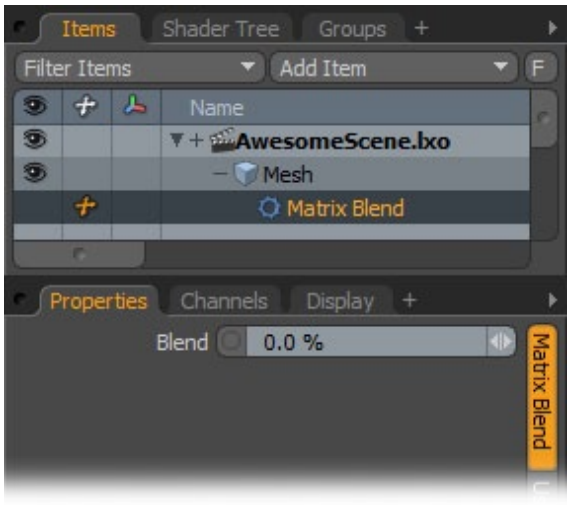
Cosine- ratio of the adjacent side to the hypotenuse of a right-angled triangle

Sine- ratio of the length of the side opposite the given angle to the length of the hypotenuse of a right-angled triangle

Tangent- ratio of the opposite to the adjacent side of a right-angled triangle

Input: This is the input value for the numeric calculation, typically this value is linked from another channel.

Inverse: When this option is enabled, the output value will be inverted.



Matrix Blend

This Channel Modifier can be used to blend between a pair of rotation or translation input matrices.

Matrix A/Matrix B: Link the two matrices to be blended into these channels, these can be world or local transform matrices from 3D items or the matrix output from other Channel Modifier items. For best results always link channels containing matrices of the same type.

Blend: This channel specifies the blend amount that will be applied to the input matrices. A value of 0% here will output the Matrix A matrix unaltered and 100% will output the Matrix B matrix unaltered. At values between 0% and 100% the matrices will be interpolated between the two inputs. Rotation matrices use a quaternion interpolation and Translation matrices are linearly interpolated.

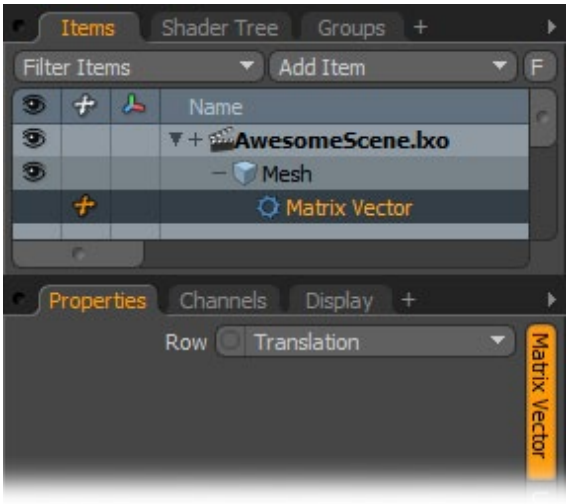
Matrix Output: The result of the interpolation is output to this channel and can be linked into a compatible (matrix) channel on a 3D item or another Channel Modifier.

Matrix Compose

The Matrix Compose Channel Modifier multiplies together two or more input matrices and outputs the resulting transform matrix.

Matrix Inputs: This channel accepts multiple input links from the matrix channels of 3D items or other Channel Modifier outputs. The matrices are multiplied together in the order they are linked to this channel.

Matrix Output: The composed matrix is output to this channel which can be linked into a compatible (matrix) channel on a 3D item or another Channel Modifier's input.



Matrix Vector

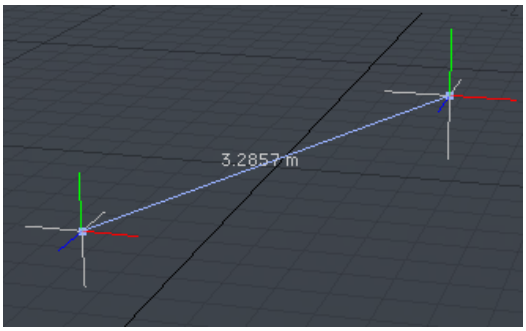
This Channel Modifier is used to extract a normalized direction vector from a rotation matrix or the translation row from a translation matrix input. Direction vectors in a rotation matrix are the axes formed after the individual X, Y and Z rotations have been performed. For example, given the world rotation matrix of a 3D item the Z Direction vector will pointing in the direction the item is facing and the Y axis towards the top of the item.

Matrix Input: Link the matrix channel from which the vector is to be extracted into this channel. This can be a matrix channel from a 3D item or the output matrix from another Channel Modifier.

Row: This channel specifies whether to extract the X, Y, Z direction vector from a rotation matrix or the translation from a translation matrix.

Output X/Y/Z: The vector extracted from the matrix is output to these three channels.

Measure Distance



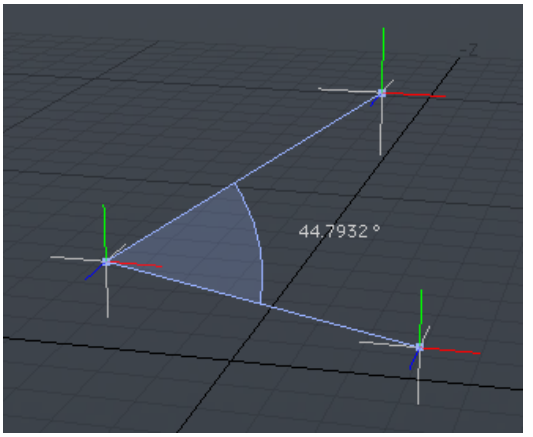
The 'Measure Distance' Channel Modifier can be used to measure the distance between any two items in the scene.

Position From: Link the World Position transform channel of a 3D item or the Position transform output of another Channel Modifier item into this channel.

Position To: Link the World Position transform channel of a second 3D item or the Position transform output of another Channel Modifier item into this channel.

Note: Leaving either of these input channels un-linked will output the distance from the world origin.

Measure Angle



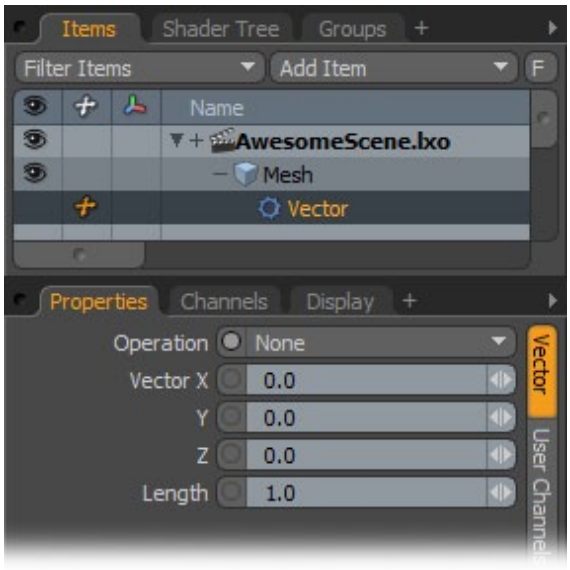
This Channel Modifier type can be used to measure the angle formed between any three 3D items in the scene.

Position From: Link the World Position transform channel of a 3D item or the Position transform output of another Channel Modifier item into this channel.

Position To A: Link the World Position transform channel of a second 3D item or the Position transform output of another Channel Modifier item into this channel.

Position To B: Link the World Position transform channel of a third 3D item or the Position transform output of another Channel Modifier item into this channel.

Angle Output: The floating point angle between the three items is output to this channel.



Vector

'Vector' modifiers can be used in the creation of vectors, setting their length and normalizing them (same as setting their length to one). A vector is simply a position with both direction and intensity.

Operation:

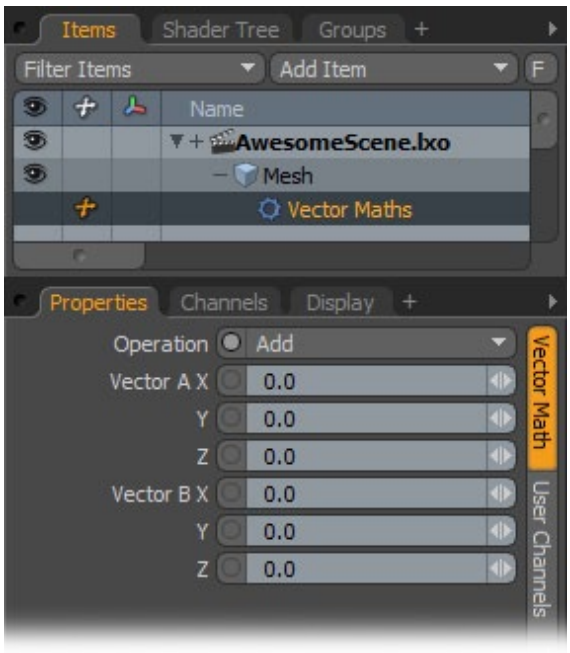
None- This simply allows the user to create a vector to connect into other Channel Modifiers or Item channels.

Normalize- Sets the length of the input vector to 1.0.

Set Length- Sets the length of the input vector to the value set in the "Length" channel.

Vector: The XYZ coordinates of the input vector.

Length: Specifies the length of the vector for the "Length" operation.



Vector Maths

Provides some basic math operations on two input vectors. Note: in the case of the dot product which is a single floating point value the result is written to the first output channel only, i.e. Output X.

Operation: Defines the math operation applied-

Add- Adds the two input vectors.

Subtract- Subtracts the input vectors.

Cross Product- Outputs a vector that is at right angles to the two input vectors on the plane containing them.

Dot Product- Outputs the cosine of the angle between the two input vectors.

Vector A X/Y/Z: XYZ coordinates for vector A

Vector B X/Y/Z: XYZ coordinates for vector B

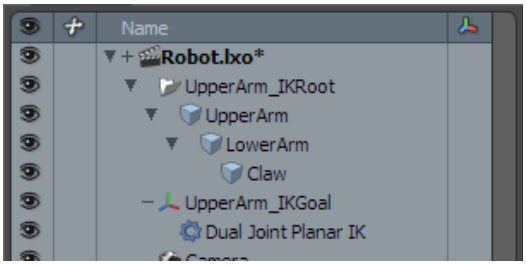
Dual Joint Planar IK

Inverse Kinematics or 'IK' as it is called is a method of animating items where the computer calculates the bending of a joint based on the position of a goal object, which in modo's case, is usually a locator. Once IK has been set up for a series of items, by simply moving the goal item around, the IK hierarchy will continually reach toward the goal, modifying joint rotations as necessary. The benefit of using a system of this type, is that it is much easier to animate the position of one item, the goal, than to animate the rotations of several separate items. Consider the scenario of a hand on a doorknob, with IK the goal can be positioned at the door knob and the hand will be in the proper place, but with a standard hierarchy, the shoulder and the elbow both must be rotated independently to try and find the correct position for the hand. It is called 'Dual Joint' as it affects two joints, so for an arm, this could be the shoulder and the elbow. Planar meaning all 3 rotation axis must share a single axis plane value. (While 'Dual Joint Planar IK' has its place, you may also consider the more powerful IK system - Full Body IK).

Applying IK

To apply IK to a chain of items, first the hierarchy must be created. You'll need 3 items parented to each other in a descending chain. Also make sure the center positions are in the proper location as well. In order for modo to calculate the IK properly, all center positions must share at least one axis value, so they all appear on the same plane. Also note that the centers for chain of items cannot be positioned as a straight line, there needs to be a bend to the chain so that when the IK is applied, a preferred angle for bending can be set. Once the items are in order, one simply needs to select the items in descending order and then press the 'Apply IK button.

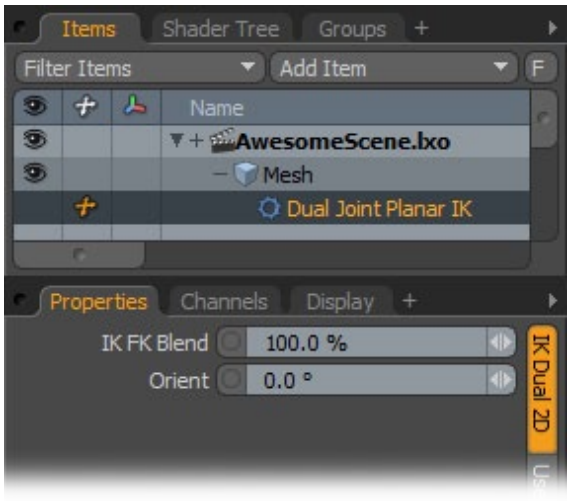
Once the IK modifier has been applied to the chain, a group locator will have been added as the overall parent for the IK chain, and a goal item will have been created. Moving this goal item will modify the angle of the joints in the IK chain.



Selecting the modifier item in the item list will display the properties related to the IK chain.

IK FK Blend: A setting of 100% will fully use the IK solution, while values below 100% reduce the effect of IK on the chain, ramping toward 0% where the IK would be fully disabled.

Orient: Modifying this value will allow the user to adjust the planar orientation angle of the IK chain. For example, for an arm this would be like adjusting the angle the elbow points toward.



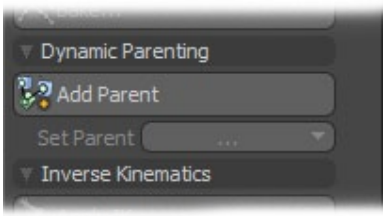
Create IK Up Vector: In IK Up vector defines the up direction for an IK constraint and can help to keep the chain oriented during animation. The 'Create IK Up Vector' button is similar to the Assign Up Vector command, except the user doesn't need to create and position a locator, as that is generated automatically based on the preferred bending angle. Once created, the new 'Up Vector' locator can be animated to adjust the orientation of the IK rig, similar to adjusting the 'orient' setting in the IK properties panel.

Dynamic Parenting

When an item is parented to another it will inherit that item's position, rotation and scale. Any transforms of it's own are added to the those of it's parent. Normally this parent child hierarchy is a fixed link. Parenting defined in this way cannot be animated and remains in effect for the duration of the scene.

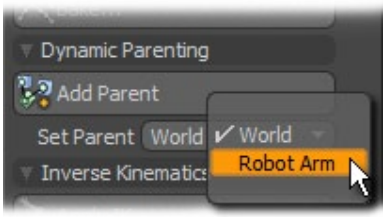
There are many situations where the ability to change the parent of an item is very useful. Consider, for example, a robot arm on an assembly line, perhaps a component is picked up and attached to a product being assembled on a conveyor, once done the product moves off for the next stage of assembly. This could be animated by having multiple versions of the component, one in the parts bin, one parented to the robot arm and another parented to the product on the conveyor. By animating the visibility of each of these component copies you could give the impression that it's being picked up and attached to the item being assembled but this can quickly become tedious to implement, and difficult to manage.

Ideally you'd want to specify that, at this point in time, this item is parented to that item and, at another point in time, it's parented to another and so on. 'Dynamic Parenting' allows you to do just that and has many applications for animation beyond this example.

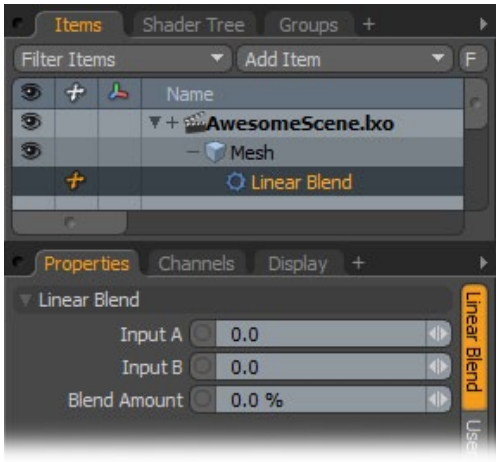


To define a Dynamic Parent relationship first select the item that will be the child (in the example above, this would be the *component*) and then the acting parent item (for example; the *Robot Arm*). Once both items are selected make sure that 'Compensation' is on (so the item remains in it's current location) and click on the 'Add Parent' button to define the link.

Underneath the 'Add Parent' button is a popup menu that allows you to choose the current parent for the item. If you open the menu now you'll see it contains two entries, one for the parent that you just assigned and one for the 'World', which effectively means 'no parent'. If you select and move the new parent item, the constrained item will follow along just as if it was parented by conventional means. To add further parents, you simply select the constrained item and the item that is to become the parent and press the 'Add Parent' button again. You can add as many items as you wish and they can be added at any time.



To change the current parent, just select the constrained item and choose the new parent from the popup menu, either in the tool bar or in the properties viewport for the 'Dynamic Parent' Channel Modifier item (click the '+' sign expansion widget that appears next to the constrained item in the Item List). Once all the parent items are defined, they can be changed over time with keyframes. If Auto Key is enabled, then by just moving the timeline to a new position and specifying a new parent, the child item will dynamically change to the new parent. You can also manually apply keyframes on the 'Parent' Channel under the 'Dynamic Parent' modifiers channels. They are specified numerically, but represent the items in the list top down. So, for our example 0 would equal *World*, 1 equals *Robot Arm*, 2 *Component* and so on. Note: It will likely be necessary to enable 'Compensation' when dynamically parenting items to ensure items don't jump around when parented.



Linear Blend

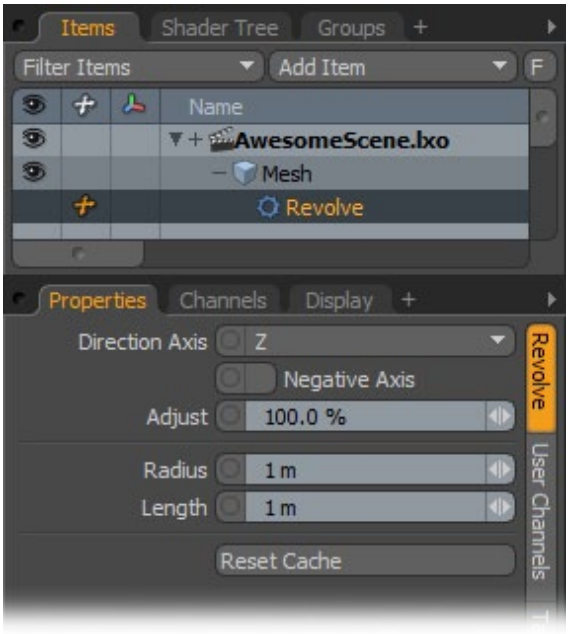
Performs a linear blend between the two input values, the blend amount controls the, well, blending between the two values. At zero percent 'Input A' will be output and at 100% 'Input B' will be output. Values in between will be interpolated linearly.

Input A: Input value determined from the source input.

Input B: Input value determined from the secondary input.

Blend Amount: Determines the amount of *blending* between the two values.

Revolve



The primary output of the Revolve Channel Modifier is a world space distance traveled along a specified vector, as well as outputting a distance traveled the modifier also has channels to compute a rotation value from a given radius and a percentage traveled along a specified length. The former is useful for driving wheels on a vehicle for example.

Input: Position: Link the World Position transform of the item whose movement is being measured.

Input: Rotation: Link the World Rotation transform of the item whose movement is being measured.

Input: Radius: Taking a radius set in this channel the modifier will calculate the rotation required to roll a cylinder over the distance traveled without slipping and output it to the 'Angle Output' channel.

Input: Length: Given a length set in this channel the modifier will output the percentage of this length traveled to the 'Percentage Output' channel.

Input: Adjust: This channel can be used to alter the outputs of the modifier, by default it is set to 100% meaning that the output is not affected. By setting this to 0% the output is static and at 200% the outputs are doubled. This can be used for things like braking, wheel slip etc.

Input: Axis: This channel sets which of the direction vectors defined by the Position and Rotation transform inputs is used to work out the distance traveled.

Input: Negative Axis: Enabling this setting will cause the modifier to use the negative axis of that chosen in the above channel.

Output: Angle Output: This is the angle of rotation calculated from the 'Radius' channel. Connect this to one of the rotation channels on the Rotation transform of the driven item.

Output: Percentage Output: This is the percentage of the 'Length' input channel traveled. For example, if the distance traveled is twice as long as the input length then the output will be 200%.

Output: Distance Output: This is the total distance traveled of the driver item along the vector specified by the 'Axis' channels.

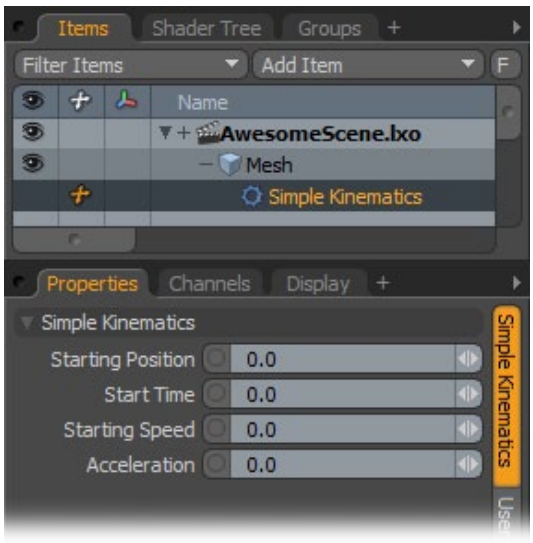
Notes on using Revolve--

Cache: This modifier uses a cache to store the total distance traveled, it may become necessary to reset this cache periodically when working in the scene, especially after a lot of scrubbing or jumping to different points in time. This is achieved by pressing the 'Reset' button found in the properties form for the modifier.

Network Rendering: Scenes using this modifier will require the output to be baked and the modifier removed or it's outputs unlinked in order to render over a network. This is not necessary when rendering locally provided the frames being rendered are in sequence. Resetting the caches on each modifier before baking or rendering locally is advisable. The caches are not stored in the scene file so will be reset automatically on scene reload.

Dependency Loops: To avoid a dependency loop, do not use the world position and rotation channels of the item being driven as inputs to this modifier. Create a locator at the same position as the driven item and parent the driven item to it. The world position and rotation channels of the locator can then be used as inputs for the modifier.

Simple Kinematics



The 'Simple Kinematics' modifier applies a constant acceleration to a channel. The obvious use for this is to produce an physically based trajectory for something flying/falling through the air in gravity (much like a gravity modifier, but without bouncing). To create a trajectory for a thrown ball, apply the modifier to to the ball, link the modifier's 'Output' to the ball's Y-position channel, and set the proper options.

Example usage- Set the 'Starting Position' to the Y-value the ball should

start moving from, leave the 'Starting Time' at 0.0. For acceleration (*meters per second*), enter 9.8, for Earth. Set the Starting Speed to 22 (*meters per second*), which would be a 50mph baseball throw. For something more interesting that a vertical throw, add a constant velocity along the ground by either setting a linear enveloped motion on the X-channel, or by adding another Simple Kinematics modifier with just a constant speed and no acceleration on the X channel.

Less obvious uses for this common case of constant velocity/constant acceleration would include buoyant objects accelerating upwards, cars and vehicles sometimes, and any other process where the resulting channel shape should be linear or parabolic.

Setup & Tools

Animation Setup Tools

To aide users in the setup of of their elements for rigging and animation, modo provides a number of tools and function to make it easier for users to do numerous tasks when simply positioning elements and defining hierarchies.

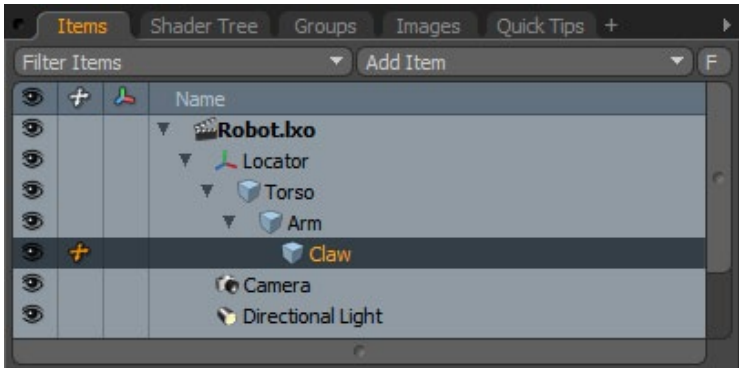
Hierarchy	Match and Align		Set
Parent	Item Position	Pivot Rotation	To Workplane
Unparent	Center Position	Match Scale	Position
Insert	Pivot Position	Match	Rotation
Insert at Parent	Item Rotation	Align	Center to Pivot
Extract	Center Rotation		Mirror

Hierarchy

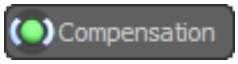
Hierarchy is defined as a series of ordered groupings. In 3D animation these hierarchies are also known as 'Parenting' or 'Parent/Child' relationships, modo uses these grouping hierarchies to define how multi tiered objects relate to each other. Without them, it would be far too cumbersome to animate anything beyond simple single item transforms.

Hierarchies are essential to animation as they define the orders of inheritance for transformations in a chain of items. Consider an arm, if you move your shoulder (or

torso), the arm follows along, as does the wrist and hand. So the shoulder bending is passing its transforms on to the elbow, which can also bend, both of those transforms are passed on to the hand so on. modo visualizes these hierarchy chains as cascading items within the item list.



Note the indentations of each subsequent item in the list. The top locator item is the 'parent' item, the next its 'child', followed by siblings in lower orders. Proper ordering of items in a hierarchy is imperative as it defines how the transforms relate to items further down the chain. Creating hierarchies in the item list is a simple process covered in the 'Item List' section of the documentation. The animation tab setup provides some additional tools for creating and modifying hierarchies.



Compensation: When parenting items together that have prior transforms applied, you might find your items popping out of position, what is happening is the child is inheriting the transforms of the parent, instantly sending it off to a position you likely don't want. Enabling the 'Compensate' button which is positioned in the top area of the animation layout, next to the mode buttons, changes the Parent/Unparent function to Parent in Place/Unparent in Place. What this does is ignores prior parental transforms, so items retain their initial positions in relation to their parent.

Parent/Unparent

When in item mode, the 'Parent' and 'Unparent' commands produce the same results as the item list, but work with selections in the 3D viewport. Selection order is important when using this function as it sets the relationship properly. The last item selected becomes the parent while everything else becomes the child of the parent. Multiple items selections won't create an entire chain, to do that, you would need to apply the parenting command multiple times. To use the parent command first make sure you are in 'Item' mode. Select an item in the 3D viewport by clicking on it, this will

be the child item, hold down the shift key and select the second item, with both items selected, click the parent button. And that's it. You can see that the first item selected, has now become the child of the second item. Unparent works the same way, select any item or items and click the unparent button to remove them from the hierarchy. Keep in mind when setting up hierarchies that any item can be parented to any other item, cameras, lights, and locators.

Insert

The 'Insert' command is a useful function for adding items into an existing hierarchy without breaking or upsetting the existing chain. 'Insert' will add the first selection as a child of the second selection, inheriting the latter selections transforms and siblings.

Insert at Parent

The 'Insert at Parent' command reverses the insert command, inserting an item as a parent instead of a child, while still inheriting transforms and siblings. Useful for inserting an item into the top of a hierarchy.

Extract

The Extract command will remove an item from within a hierarchy, without disrupting the rest of the hierarchy chain, by automatically parenting children of the extracted item, to its parent, once the item is removed.

Match and Align

The Match and Align functions provide a simple means of positioning multiple objects in relation to each other in 3D space for animation, or simply for layout purposes as well. Most option are simple one-shot commands, where users make a selection, and then press the appropriate command button to get the desired result. For finer control, there are also the 'Match' and 'Align' dialogues that provide users a way to choose specific options.

Item Position

Exactly as the name states, 'Match Position' command will match the position of two items based on their center points. Selection order matters, as the first item selected is moved to match the position of the second item selected. Subsequent selection will be ignored, as will scale and rotation of the items.

Center Position

The 'Center Position' command will match the center position of an item, to the center position of another without actually moving the item itself, it only affect the center position and ignores rotations and scale. To use, select the first item whose center you wish to edit, then shift select the second item whose center position you wish to match and click the 'Center Position' command. If you can't find it, it is a popup command under 'Match Position', if you LMB click and hold, the popup will allow you to select the alternate tool.

Pivot Position

The 'Pivot Position' command will match the pivot position of an item, to the pivot position of another without actually moving the item itself, it only affect the center position and ignores rotations and scale. To use, select the first item whose pivot you wish to edit, then shift select the second item whose pivot position you wish to match and click the 'Pivot Position' command. If you can't find it, it is a popup command under 'Match Position', if you LMB click and hold, the popup will allow you to select the alternate tool.

Match Rotation

The 'Match Rotation' command will match the rotation of two items based on their center points. Selection order matters, as the first item selected is rotated to match the rotation of the second item selected. Subsequent selection will be ignored, as will position and scale of the items.

Center Rotation

The 'Center Rotation' command will match an items center rotation to that of another item, without actually rotating the item itself. To use, select the first item whose center you wish to edit, then shift select the second item whose center rotation you wish to match and click the 'Center Rotation' command. If you can't find it, it is a popup command under 'Match Rotation', if you LMB click and hold, the popup will allow you to select the alternate tool.

Pivot Rotation

The 'Pivot Rotation' command will match an items pivot rotation to that of another item, without actually rotating the item itself. To use, select the first item whose pivot you wish to edit, then shift select the second item whose pivot rotation you wish to match and click the 'Pivot Rotation' command. If you can't find it, it is a popup

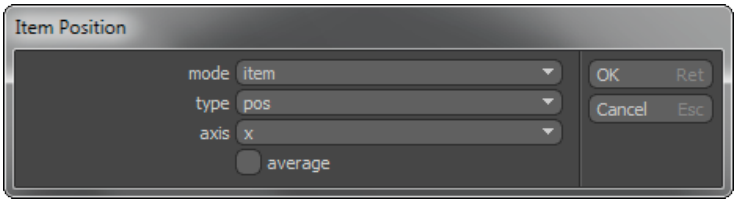
command under 'Match Rotation', if you LMB click and hold, the popup will allow you to select the alternate tool.

Match Scale

The 'Item Scale' command will match an items scale value to that of another's. To use, select the items whose scale you wish to modify, shift-select a second item whose scale you wish to match and click the 'Item Scale' command. The scale value will now match for both items.

Match

The match tool will allow the user to select precisely how two or more selected items are matched. Before, invoking the command, you'll want to make your item selections. Selection order is important, the first item selected, is the item whose properties will be edited to match the properties of subsequent item selections. Use shift key and LMB-click to select additional items in the 3D viewport. Once you have the items selected, you can click the 'Match...' tool to open the popup requestor.



Mode: Mode selects what is being affected, 'Item' modifies the item itself, 'Center' will modify only the items center point and 'Pivot' will modify only the items 'Pivot' point.

Type: 'Type' specifies what value will be matched from one item to another. Option to choose either 'Position', 'Rotation' or 'Scale'.

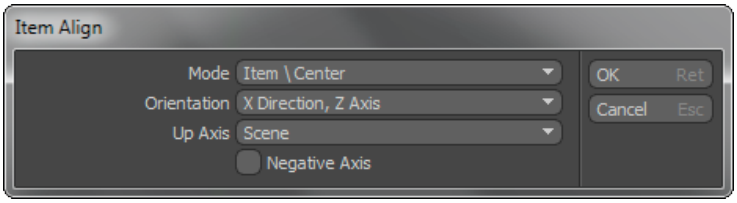
Axis: 'Axis' specifies what axis will be matched from one item to another. Option to choose between 'X', 'Y', 'Z' or 'All'.

Average: Enabling the "Average' function will tell modo to average the values of subsequent item selections. The first selection is the item whose values are modified, if the average checkbox is enabled, and two more items were selected whose scale was 200% and 300% respectively, then the first item would be scaled to 250%, the average of the last two items selected. This will work for any type of value. If you wish to center an item between several others, you could simply select the item you wish to center and then shift-select the other items, set match to Mode:Item, Type:Position, Axis:All

and enable average, the first item selected would now be centered between all the the other items that were selected.

Align

When setting up item hierarchies for Inverse Kinematics, a parents center position must point toward its child's center position. If the items in the hierarchy were created off-axis, meaning they are rotated away from any of the axes planes, the 'Align' command will rotate the center position of the item to align directly toward its child's center. To use, select the target item layer in the hierarchy you wish to affect, and run the command. The dialog box offers some options-



Mode: Choose to modify either the items 'Center or 'Pivot' rotations.

Orientation: Options to select the orientation of the center item when rotated, 'Direction' defines the axis that will point toward the child, while 'Axis' the actual orientation

Up Axis: Users can choose to use the 'Scene' or one of the three world axes for rotation.

Negative Axis: Orients the 'Up Axis' in the negative direction when enabled.

Set

To Work Plane Position

The 'Set To Work Plane Position' command is a powerful option allowing users to position and align 'Items', 'Centers' or 'Pivots' relative to the current Work Plane. With the Work Plane being a dynamic function, this might seem less than useful, however, nothing could be further from the truth. The true power of the Work Plane comes in being able to fix it to a specific location by way of a component selection and then applying the 'Align Work Plane' command. This essentially rotates the entire *modo universe* to align with the selected element. With the Work Plane origin now centered on the selection, it makes aligning objects to specific positions very easy.

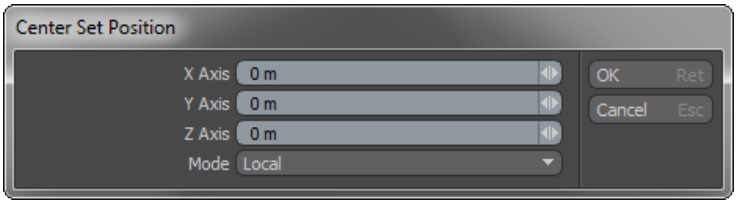
To best use this command, select one or more geometry components you wish to align to, such as vertices, edges or polygons, and use the 'Align Work Plane to Selection' command found in the Work Plane menu. This will fix the Work Plane in place, centered around the current selection, aligning it to match the normal direction of the selected elements. Next, select the Item you wish to move. This can be the item itself, or just the items Center or Pivot in their respective selection modes. Next, run the 'Set to Work Plane' command, this will move the selected element to the center (0,0,0) position of the current Work Plane. The Work Plane can then be reset to its normal operation using the 'Reset Work Plane' command or by pressing the 'End' key.

To Work Plane Rotation

The 'Set to Work Plane Rotation' command works in the same fashion as the 'Set to Work Plane Position' (detailed above) but will only affect the rotation of the selected Item, Center or Pivot, allowing users to match the rotation of an element without changing its position.

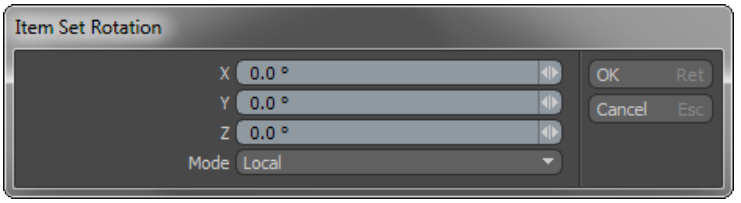
Position

The 'Position' popup requestor will allow the user to numerically set the position of an Item, Pivot, or Center in 3D space. Use the 'X', 'Y' and 'Z' values to define a specific location. Use the 'Mode' option to determine if the transform is 'Local' - offset from the items current position, or if it is 'World' - a specific position within modo's universe.



Rotation

The 'Rotation' popup requestor will allow the user to numerically set the rotation an an Item, Pivot or Center in 3D space. Use the 'X', 'Y' and 'Z' values to define a specific rotation. Use the 'Mode' option to determine if the transform is 'Local' - offset from the items current rotation, or if it is 'World' - a specific rotation within modo's universe.



Center to Pivot

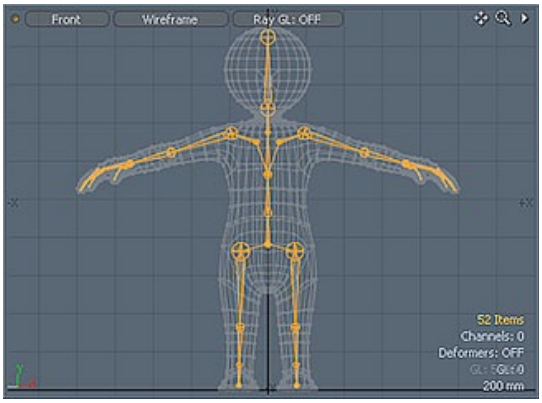
All of modo's animation transform values, by default, originate from the center position, if your model is set up to use pivots instead, the 'Center to Pivot' command will match the center position to the pivot position. Simply select the item in question (in 'Items' mode) and LMB+click the 'Center to Pivot' button.

Mirror

The 'Mirror' command *reflects* the selected items over the chosen axis, either 'X', 'Y' or 'Z' producing a mirrored opposite item.

Skeleton Tool

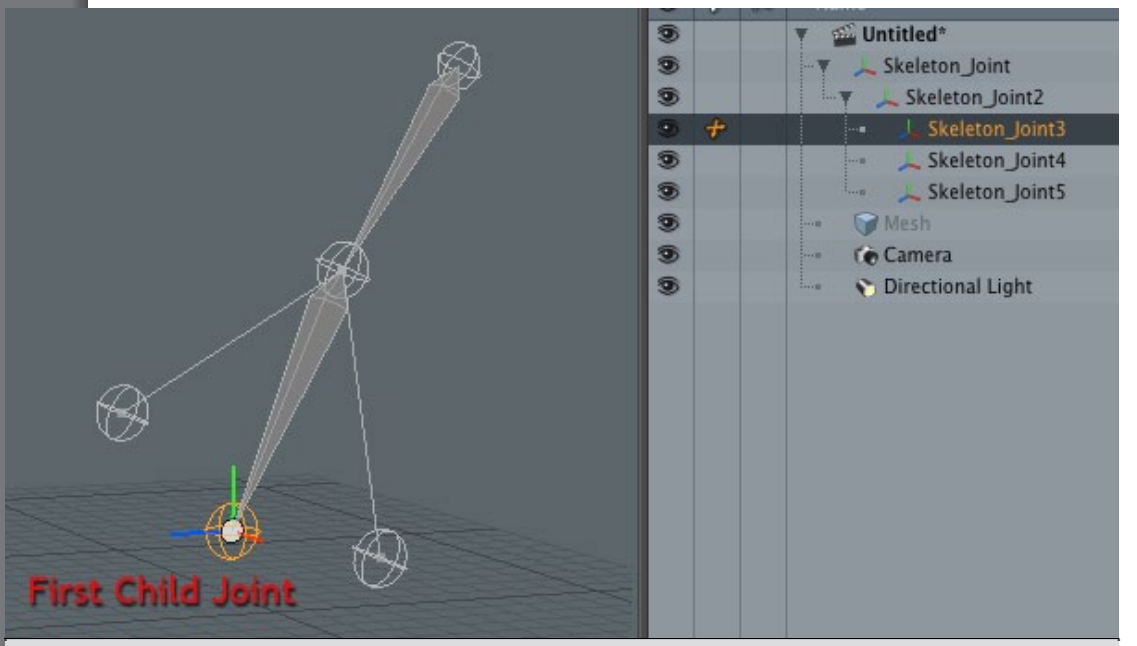
The skeleton tool is used primarily for creating a joint hierarchy for character posing and animations using deformations. Once the skeleton is created, it will still be necessary to Bind the joint hierarchy to the mesh using the 'Bind' command while in the 'Setup' action state. For more on binding, please reference that page of the documentation.



Drawing the Skeleton

Prior to drawing the skeleton, users should give some thought to the position of the root element, this is the base joint for the entire chain, every other joint created is a parent/child cascading hierarchy from this joint. Often times for bipedal human characters, this is placed at the hips, with the legs and spine branching outward. To set the Root, users can simply LMB+click in the viewport over the target geometry (no need to drag out the joint). By default, the 'Intersection' option will be *enabled*, when it is, modo will attempt to find the halfway position between the forward and backward facing polygons intersecting with the mouse click position and place the joint segment

appropriately. A great time saver for placing joints in hands and feet. If the are no back facing polygons, or 'Intersection' is *disabled*, the joint will be created at the intersection of the mouse click and the geometry surface. Clicking away from the model creates the joint at the intersection of the mouse click and the current Work Plane. After setting the initial point that represents the base of the joint (its rotations origin), users can LMB+click a second time to set the actual joint, by LMB+click and dragging, users can interactively scale and position that joint. Subsequent mouse clicks will continue to create additional joints in the chain, each a child of its predecessor. Pressing 'Shift' when creating a joint will start a new chain. To add a branch to an existing chain, LMB+click on a joint to select it and the continue to click, drawing the new branch. The position of joints that have been drawn already can be modified by LMB+clicking and dragging on their base, or by entering the 'Edit' action state.



The first child of a joint is important as it defines the alignment of it's parent joint and is also used by other item alignment commands in modo. To make the first joint easier to identify modo only draws the rhombus link shape to the first child, links to other child joints are drawn as lines. Changing which joint is first is easily accomplished in the Item List by dragging it directly under it's parent.

Draw Hierarchy--

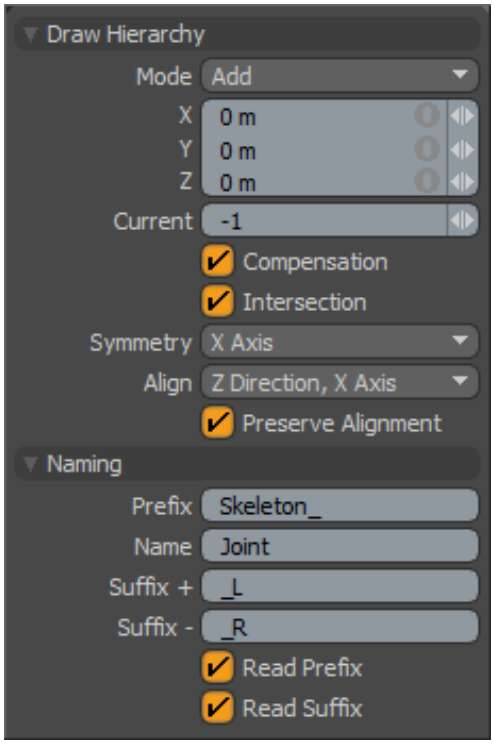
Mode: Various action modes of the skeleton tool-

Add- This is the default state, where the joints of the skeleton are created, each LMB+click creates an additional joint as the child of the previous joint, producing a cascading hierarchy.

Edit- The 'Edit' action allow users to modify the position of individual joints without the possibility of accidentally creating a new joint.

Insert- In the 'Insert' action, users can LMB+click on the base of any joint splitting it into two separate joints.

Delete- In the 'Delete' action, users can LMB+click on any joint to remove it from the hierarchy.



X,Y,Z: Displays the joints position in 3D world space, like any other

Current: Displays the joint number (in sequence from the Root), determining the current joint for editing, or the most recently created joint.

Compensation: When 'Compensation' is disabled, child joints will move with the parent, when enabled the child joint will change length with the rest of the chain remaining in position.

Intersection: By default, when drawing out joints, modo will draw the joint at the intersection of the mouse click and the Work Plane. When 'Intersection' is enabled, modo will attempt to find the mid-point within a volume, finding forward and backward facing polygons under the mouse pointer and center the joint between them.

Symmetry: When one of the 'Symmetry' options is selected, mirrored joints will automatically be created across the chosen axis. duplicated joints will be automatically named according the 'Naming' options. The 'Off' setting disables symmetry.

Align: The 'Align' option determines how the local axes of joints will be aligned towards its first child joint. 'Direction' specifies which axis will point towards the child joint, 'Axis' specifies the axis running horizontally through the joint. These can be thought of as Roll and Pitch respectively.

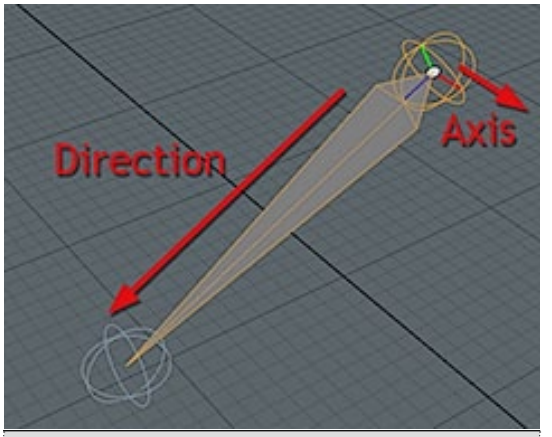


Image shows 'Z Direction, X axis' 'Align' default.

Preserve Alignment: The 'Preserve Alignment' option when *enabled*, maintains the current alignment of joints when selected. When *disabled*, the joint will be re-orientated to the alignment chosen in the 'Align' menu. This is a quick way to re-orient a number of joints in a hierarchy, by choosing the desired alignment and then simply LMB+click on on the target joint to be changed.

Naming--

Prefix/Name: Determines the beginning and second parts of the auto-naming string. 'Prefix' and 'Name' will be followed by a number (based on the number of items in the scene with the identical name). When more than one joint is created with an identical string. Values can be changed at any time during the joint creation process to denote different areas of the skeleton.

Suffix +/-: When the Symmetry option is *enabled*, the Suffix options are appended to the end of the string, based on its position on the X plane, relative to the global YZ plane as either + (positive) or - (negative).

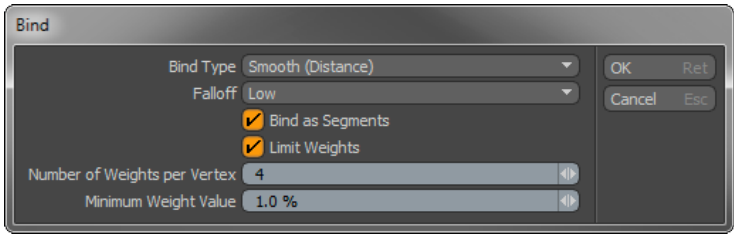
Read Prefix/Suffix: With these options *enabled*, the prefix and suffix will be read from the joint's current name when it's selected and the edit fields updated. When *disabled* the appropriate part of the joint's name will be changed to the text in the Prefix and Suffix edit fields when a joint is clicked on.

Binding and Setup Mode

Binding

Binding is the process of connecting a skeleton made with the 'Skeleton' tool, and attaching it to the target geometry so that it deforms when the skeleton is manipulated. The process is simply enough, with just a couple requirements that need to be met prior to running the command. Once geometry is ready to be animated, and a reasonable Skeleton is created with an appropriate hierarchy (created automatically when drawing out the *skeleton - reference the* Skeleton *tool documentation for more information*). The first step is to be in 'Items' mode where users select the target Mesh item(s) and the Root of the Skeleton hierarchy. Next enter the 'Setup' action state by pressing the button above the 3D viewport in either the 'Setup' and 'Animation' interface tabs or by simply pressing 'Enter'. 'Setup' action state is designated by the yellow outline of the viewport when enabled. From within the Setup action state the 'Bind' tool command found in the toolbox, under the 'Deformers' subtab, becomes accessible. LMB+clicking the button activates the Bind process opening a dialogue box presenting several methods of Binding the skeleton, described further below.

When clicking 'OK' in the dialogue, a series of actions then take place, all automatically. The first is the weighting of the skeleton to the geometry. Each joint will have an associated weight map created based on the 'Bind Type' and its settings. Next, a series of Deform Influences are assigned to each joint of the skeleton and each has an associated weight map generated and assigned to it. Finally, each Influence is automatically linked to its originating Joint and the entire hierarchy of Deformers is now placed into a Normalizing folder, where all the influences will be evaluated as a single unified deformer. After running the 'Bind' command, the user need only to exit 'Setup' mode in order to begin to deform and animate the character. As with all things modo, further refinement is of course possible, if users wish to fine tune the automatically created weights using the Weighting tools found in the appropriately named toolbox subtab 'Weighting'.



Bind--

Bind Type: By default the binding process treats the hierarchy as individual segments

between the items. Weights are assigned to vertices along the line between an item and it's child (similar to a bone). When an item has multiple children they are treated as a single segment.

Rigid- This is the simplest of all binding methods, a vertex will be assigned a 100% weight to the closest segment in the hierarchy, or to the closest joint when 'Bind as Segments' is disabled.

Smooth (Distance)- Produces a smooth graduated transition between segments based on its distance from the segment.

Smooth (Visible)- Produces a smooth graduated transition similar to the 'Distance' option, but weighting is only assigned to vertices that are visible to the segment. This can be useful to reduce or eliminate cross talk between neighboring segments, such as with legs or with fingers. However, it does require that the Skeleton is contained entirely within the volume of the geometry.

Heat- This method takes into account the topology of the mesh and the relative positions of the joints. It often produces superior results to the other methods, but the target mesh must conform to a number of requirements-

--Mesh must form a volume, though holes are supported (such as eye sockets).
--Target mesh must be *only* polygonal, no single vertices, floating edges or curves can be present.
--No shared vertices, edges or polygons (non-manifold surfaces) allowed between multiple components.
--All joints must be contained within the volume of the mesh.

When applying 'Heat' weighting, if any errors are encountered, the bind command will report the error with a message popup. Most geometry issues can be resolved by applying the 'Mesh Cleanup' command. Illegal geometry will be highlighted upon exiting the command f encountered allowing users to target the area to repair. Another error that might be displayed is a "Solver Error", this can occur when there are multiple mesh volumes in the same 'Mesh' item layer, especially if they overlap. A simple example would be two primitive spheres in the same layer. Often these solver errors can be fixed by only weighting the main mesh volume, e.g. the skin of a character, and placing any other elements into another layer (such as eyes, hair, clothing etc) before weighting.

Falloff: When using either of the the 'Smooth' Bind types, this value determines the attenuation amount between neighboring weight maps. 'Low' produces a softer more gradual transition between joint weights, where the 'High' option produces a more abrupt transition, with 'Medium' in the middle of the two.

Bind as Segments: When 'Bind as Segments' is enabled, the weighting is calculated as a radius around each joint rather than along the segment between a joint and it's child

Limit Weights: When 'Limit Weights' is *enabled*, this option can be used to limit the number of joints that can influence a weight value during the 'Bind' process (Useful for 'Unity' which can handle a maximum of four joints per vertex).

Number of Weights Per Vertex: When the 'Limit Weights' option is *enable*, this value determines the maximum number of influencing joints. This value is passed on to the 'Normalizing Folder' as the 'Max Sources' attribute where it can be adjusted later on.

Minimum Weight Value: Any Weight values below this defined threshold will be ignored. Useful in cleaning up 'Heat' Binding which has a tendency to generate a lot of small Weight values.

Setup

Users unfamiliar with animation and rigging may be curious to know why one needs to enter a 'Setup' mode in order to 'Bind' a skeleton to the geometry or set up 'Inverse Kinematics'. Basically, it is because when the Binding takes place, items in the 'Setup' action state are understood by modo to be in their resting or undeformed position. By binding in the 'Setup' action state, the initial position of the individual joints is defined and is zeroed so all joints will have the same starting value. Same for IK, as it defines the base position for segments along the chain, so when a Task or Goal is added, the IK solver know from where to begin its calculations. Also by having a specific 'Setup' mode provides a lot of flexibility allowing users an easy way to get an animated or deformed model back into its resting positions for further refinement without having to remove or disable any animation already applied to the geometry.

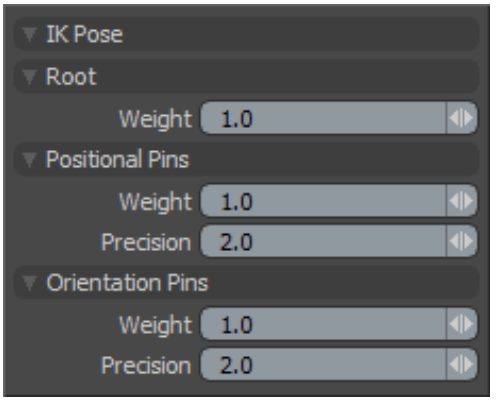
Additionally, the 'Setup' action state can be used to define Resting positions for 'Falloffs' when used as Deformers, so modo will know what areas of the mesh should be under the influence of the Falloff.

Pose Tool

The Pose tool allows users a simple means to pose and animate item hierarchies, manipulated by an intuitive pin and drag workflow. While primarily targeted at character skeletons, the Pose tool can be used to modify and animate any hierarchy of locator type items (this includes just about anything that appears in the Items list). The main benefit of the Pose tool outside of its simple to use operation is that it is not necessary to apply any IK to the hierarchy first; the Pose tools takes care of IK automatically, though any rig that does contains IK will enhance the use of the tool.

Using the Pose Tool

The Pose tool is found in the 'Setup' and 'Animate' interface tabs in the upper left toolbox. Prior to activating the tool, make sure to select the Root item of the target hierarchy. For more information on defining hierarchies, please reference that page of the documentation. Once selected, click the icon button to activate the tool. When active, small white circles draw in the 3D viewport at each items 'Center' point position representing its center of rotation. Hovering the mouse pointer over any of these circles will highlight the element yellow. A LMB+click and drag will move the highlighted item. If the root element was clicked and dragged, the entire hierarchy will move rigidly, but if subsequent child items are moved, they will move as if IK has been enabled, where one element will drag along another element earlier in the chain. In order to keep an element from moving it can be pinned, hover the mouse pointer over a joint and RMB+click; now when any other elements are moved the pinned elements will remain in place. Users can pin as many elements as required, and pins can be enabled and disabled freely while positioning the character. The Pose tools always uses the pins to localize the effect of dragging joints, working from the joint being dragging up the hierarchy chain until it finds a pinned joint or gets to the root. The tool also looks down the hierarchy for any pinned joints and will try to find the best fit it can for the skeleton based on the currently pinned joints and their initial orientations; some movement around the pins should be expected when dragging points.



Root--

Weight: This option determines how freely the root of the hierarchy will move. A 'Weight' of '0' will keep the root in place.

Positional Pins--

Weight: This option specifies the strength of the pins determining how much affect they have holding their position.

Precision: This option specifies how closely the tool will try to keep the pinned joints to their original location.

Orientation Pins--

Weight: This option specifies the strength of the pins determining how much affect they have holding their rotation.

Precision: This option specifies how closely the tool will try to keep the pinned joints to their original rotation.

Posing Controls

LMB+click drag to move any joint in Screen (viewport) space.

Ctrl+LMB click drag will move the joint in World space.

RMB+click a joint to pin/unpin, pinned joint will display as red circles.

Ctrl+RMB click to fix pin/unpin on subsequent keyframes/poses (requires IK assignment).

RMB+click on background to release all pins.

Shift+LMB click and drag to twist a joint.

MMB+click drag for traditional FK (forward kinematics - only the dragged joint will be affected)

LMB+click drag on background to move the Root item.

Posing IK Setups

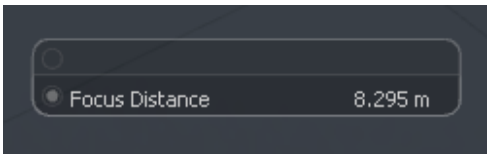
The Pose tool will work with any hierarchy of items, they don't need any special alignment or setup, but they can benefit from some hints on which way joints should rotate and on which axes. This is done by assigning full body IK to the hierarchy and setting the joint parameters, for example one can tell the Pose tool to only bend a knee joint on a single axis by setting that up in the IK properties for the knee. The Pose tool will read any IK settings from the joints, taking into account States, Axes, Weights and Limits. For more information on setting up IK, please reference that page of the documentation.

Simple Animation

The Pose tool can be used to provide a simple and quick means of animating characters with very little setup using modo's auto keying function. When set to 'Animated' any modified channels will automatically get a new keyframe. Prior to applying any animation, each item in the hierarchy need to have an initial keyframe. The Root item will need a 'Position' and 'Rotation' keyframe, and all Children will only need rotation keyframes. This should all be done on a default, base pose. A quick way to apply keys to many items is to LMB+double click the Root item (selecting the entire chain) and press the 'Key Rotation' button under the timeline. Don't forget to set the position key for the Root as well. Now users can move the timeline around and adjust poses with the appropriate keyframes being set automatically for joints as the time is changed. Keyframes can also be applied to entire Characters at once using the Actor & Pose and options. Please reference that page of the documentation for more information.

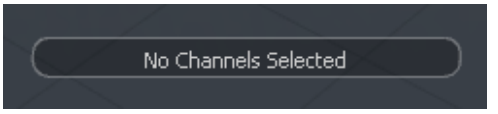
Tools and Commands

Channel Haul: The Channel Haul tool allows the adjustment of any selected channel by dragging with the mouse in the active viewport.

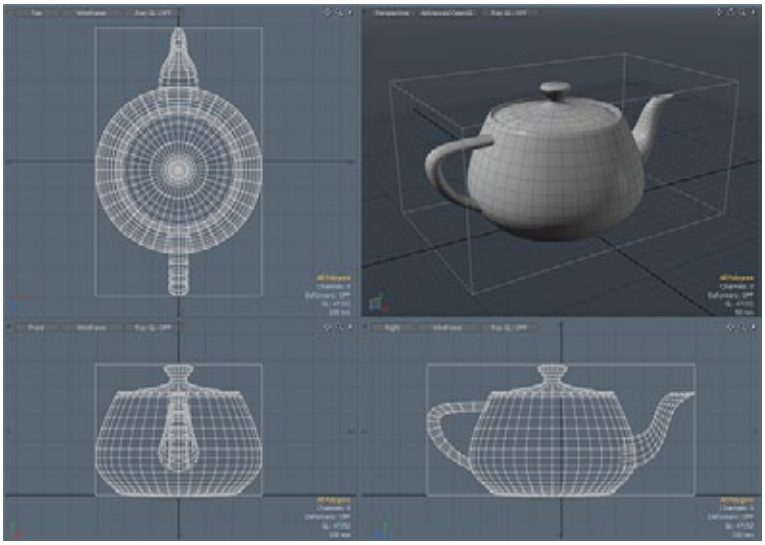


To activate Channel Haul, choose a channel within the 'Channels' viewport, and invoke the 'Channel Haul' tool button or use the 'C' keyboard shortcut. This brings up the 'Channel Haul' tool interface within the 3D viewport. By clicking on the channel name and dragging the mouse, users can interactively adjust the value within the viewport, or by clicking the value itself, users can numerically enter a new value. The channel haul tool can modify up to 6 chosen channels.

When no channel is selected, the display reflects this reminder that the tool is still active.



Center to Bounding Box: Each mesh item in modo, regardless of shape, has a virtual rectangular bounding box which is defined by an items overall size (height, width and depth). The 'Center to Bounding Box' command allows the user to easily position an items center to a specific position of that bounding box.



Found in the menu bar under "Edit > Center to Bounding Box", users can invoke the various commands to precisely position the 'Center' element in relation to the geometry contained in the layer.

Center: Positions the items Center element at the very center of the bounding box itself (as opposed to the world origin)

Top/Bottom/Left/Right/Front /Back: Positions the items Center element at the center of the plane that defines that particular surface of the bounding box volume.

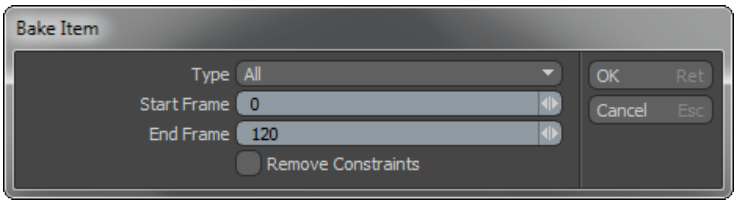
Pivot to Bounding Box: Found in the menu bar under "Edit > Pivot to Bounding Box", users can invoke the various commands to precisely position the 'Pivot' element in relation to the geometry contained in the layer. Working in identical fashion to the 'Center to Bounding Box' commands, however only the Pivot is moved, leaving the 'Center' element unchanged. For information ragarding the use of 'Centers' and 'Pivots, please referecne the Animation Overview page of the documentation.

Center: Positions the items Pivot element at the very center of the bounding box itself (as opposed to the world origin)

Top/Bottom/Left/Right/Front /Back: Positions the items Pivot element at the center of the plane that defines that particular surface of the bounding box volume.

Bake

Baking is a mean of 'freezing' the affects of a constraint on an items over time so that, when removed, the item retains the modifications. This can be useful for exporting animation out of modo, or in cases such as network rendering, where pre-evaluated transforms would reduce the possibility of errors. To use, select one or more items that are affected by a constraint/channel modifier and then select the 'Bake...' option found under the 'Animate' menu of the menu bar. When selected, the following dialogue will open. Once the desired values are assigned, click 'OK' to actually 'Bake' the target keyframes out.



Type: This specifies the transform channels to 'Bake'. By default it's set to 'All' but individual transform types can also be chosen from this menu allowing you to bake an individual constraint without affecting any others that may be driving the item. For example if you're baking an item that's direction constrained to a target then the item's rotation is driven by the constraint and you would choose 'Rotation' from this menu. If it's position constrained you'd choose 'Position" etc.

Start/End Frame: Defines the frame range to apply the bake command to.

Remove Constraints: When 'Remove Constraints' is enabled, the constraints driving the item will be removed, at the completion of the bake process and the item will solely use the baked keyframes for it's motion.

recoil Dynamics

recoil is a dynamics simulation tool that fully integrates into modo 601, and can be used to automatically simulate rigid body interactions between objects, as well as Soft Body deformations of non-rigid surfaces. recoil including robust collision detection, with the addition of constraints, including hinges, motors and springs, and dynamic forces a wide variety of actions can be simulated and rendered directly in modo. Based on the open source Bullet Physics Rigid Body Dynamics engine, recoil is simple to use and fast, able to simulate simple to complex setups in record time.

recoil Dynamics

Usage Overview

Dynamic Item

Constraints

Baking Sims

recoil Solver

Soft Body Item

Forces

Wakeup Triggers

recoil Usage Overview

What is recoil?

recoil at its essence is a collision detection tool, that realistically calculates how different dynamic elements interact with each other in a scene. Starting with collision detection and Soft Body deformations, additional constraints and forces can be applied to these dynamic items to simulate a wide variety of phenomena realistically; from very simple objects dropping, bouncing and jiggling in a scene, flags waving in the breeze, to the destruction of large structured arrays. Even basic machines can be simulated realistically, calculating interactions between numerous different elements in a scene, all automatically.

These collision interactions are calculated during what is called 'Running a Simulation'. In order to *run a simulation*, users first need to define the target objects as being *dynamic*. This is accomplished by selecting the geometric element in a scene they wish to *include* in the recoil calculations and tagging them appropriately. There are several options, depending on the desired outcome; items can be tagged as 'Active' or 'Passive' Rigid Bodies (*rigid meaning they do not bend or warp*), or 'Soft Bodies' for deforming surfaces. When set as such, recoil will then consider these elements in all simulations; all other elements not tagged as dynamic will be ignored during the simulation calculations. Next, the user simply need to play the animation timeline from frame 0 triggering recoil to begin its calculations. This is where users can see the settings of recoil in action. While highly dependant on the complexity of the scene and

the system recoil is running on, the resulting simulation will likely process in near real-time, displaying its results in the 3D viewport, providing a quick and easy way to see the simulation and make any tweaks to settings for further refinement.

NOTE: Motion and Deformation information for recoil dynamic items is stored in a local *cache*, created by recoil each time an item is tagged as dynamic, or when a scene is opened that contains dynamic elements. It is not saved as part of the scene, nor can it be saved separately. The data in the cache is generated each time a simulation is triggered, either by playing the timeline (using its standard transport controls), or by using either of the recoil simulation controls- 'Play' or 'Run'. Each time an attribute of any dynamic element is modified, it will invalidate the data stored in the cache requiring an new simulation to be calculated. Users shouldn't really need to concern themselves with the cache, but understanding how it works can help to avoid some frustration with its use. Its current state is always displayed toward the bottom middle of the 3D viewport.

Active vs. Passive

When working with rigid bodies, users must define target items as either active or passive in order for recoil to consider them during a simulation. The difference between an 'Active' rigid body' and a 'Passive' rigid body is that Active bodies will have their positions and rotation channels controlled by recoil during a simulation. Beyond the rest frame, users have no control over the Active object directly. Passive bodies are only considered for collision detection, any movement of the object itself is controlled by the user applying keyframes. Passive bodies are not considered by gravity, forces or falloffs, allowing users full control over the object during the course of the simulation. Active bodies are represented in the 3D viewport by a white bounding box, and Passive bodies are represented by their green bounding boxes. For simplicities sake, users can easily consider active bodies as those that dynamically move during a simulation, and passive bodies as just collision shapes (such as a ground plane or a wall).

Soft Bodies

Like its rigid body cousin, flexible, deforming elements must be tagged as a 'Soft Body' in order for recoil to consider it in a simulation. For Soft Bodies though, tagging is just the start. The Soft Body deformer can be used to simulate a wide variety of mediums (arguably, far more than the Rigid Body solver is expected to do), so in order to get an appropriate effect, users will need to make adjustments to the attributes of the item in the 'Soft Body' submenu of the 'Property' panel. A wide variety of mediums can be created, from billowing fabric that fall onto a furniture item of choice, to squishy rubber-like elements bouncing into cups, to stiffer materials such as paper or metal that flex only a little. Understanding what each setting does will be helpful in getting the most out of Soft Bodies, even so, some amount of experimentation will be necessary to zero in on the exact combination of setting to reach the desired effect.

There are two options for Soft Bodies in the recoil toolbox menu, each a variation on the Soft Body deformer itself. The 'Make Soft Body' option applies the 'Soft Body Influence' item to the target layer. The 'Make Ropes' option again applies the 'Soft Body Influence' to the item, presumably applied only to a curve, automatically disabling the 'Shear' and 'Bend' options, creating a dynamic curve that will dangle and swing as if made of string. Additionally, users should note that in order for any Soft Body element to collide with another mesh item, it will need to first be tagged as a some type of dynamic body, rigid or soft, otherwise, it will be ignored.

Collision Detection

Collision in recoil is the calculations that restricts the intersection between surfaces of dynamic object. It is, for example, what allows a ball to bounce and roll down a stair case. Without it, solid bodies easily pass through one another with no regard or knowledge of the opposing item. Once an object is defined as an Active or Passive body, an additional 'Dynamic' subtab is added to the items

properties panel providing settings that further control the object. The most important setting in this tab is in regards to collision as it defines how recoil sees the object. It also determines how soft body elements see the geometry. By default modo assigns a detection type that uses the actual geometry in the item layer, but often faster (and better) results can be obtained by choosing from the additional options available, for most applications the best results are obtained using the simplest collision shape appropriate to the geometry. For example, using the cylinder collision for a wheel will produce much better results more quickly than using the hi-res geometry with tire treads and all. Collisions relating to Soft Bodies are only required when there are issues with the surface intersecting itself, this can be fixed by *enabling* the 'Self Collision' option.

Constraints

If recoil were strictly limited to only calculating collisions between objects, it would very quickly become daunting to produce even modestly complex simulations. Depending on collisions exclusively to generate the mechanics of say a rotating drive shaft for a vehicle would no doubt introduce errors and be quite costly to calculate. Constraints are essentially joints that connect two dynamic bodies together, producing much more accurate results than collision detection alone, with very little calculation overhead. Constraints come in a variety of types that cover most any type of joint users would need. Clever users will also be able to stack more than one constraint to produce even more complex effect, such as combining a Point Constraint with a Slide Constraint to produce a swinging *and* sliding motion, much like riding suspended from a zip-line -a ride where the participant slides down a long, taut cable suspended between two posts.

Forces

In the real world, object are constantly subject to numerous forces that affect their motion in subtle and not-so-subtle ways; wind, gravity, friction among others. recoil offers a variety of forces all to produce ever more realistic physical simulations. Gravity is the chief force and is part of the recoil Solver item, affecting all items globally. It's default value is meant to simulate the same downward pull at full size (1:1 scale) as gravity in the real world. If a simulations isn't reacting as one might expect, it's a good bet that the scene isn't modeled to real world size, as very large items will react much differently under the same circumstances than very small items. Users can of course scale their objects, or scale the simulation to match more closely the expected result.

modo also offers six additional force types allowing users further control of the simulation. Linear, Drag, Radial, Turbulence, Vortex and Newton. These are added as locator *type* item layers in the 'Items' list and, by default, affect dynamic items globally, just as the Gravity value does. Each has its particular function which is covered fully

on the Forces page of the documentation. While Forces are applied globally, their application in a scene can be masked in a way using the 'Falloff' function (below) and the 'Force Shadow' option explained further in the Dynamic Items documentation.

Falloffs

Falloffs can be incredibly useful to provide additional control over Forces. Imagine flower petals drifting on the breeze, animated as Soft Body items that land inside of a box, the falloff would be used eliminate the wind blowing inside the box, increasing the realism if the simulation. recoil uses the same Falloffs used for other deformers in modo. Deformation Falloffs are meant to attenuate (*reduce or increase*) the effects of forces on dynamic items within the area of influence of the Falloff itself. Not to be confused with modo modeling falloffs; used for geometry creation and layout. Conceptually they are very similar, but instead of being applied as a function, recoil falloffs are added as a locator type item to the 'Items' list where users can select them and modify their attributes and even animated them within the scene. In the items properties, users can define an attenuation range that modulates the force it is applied to.

Unlike Forces that affect items globally, Falloffs must first be *attached* to the Force item it is attenuating in order to produce a result. This can be done several ways. One, by by selecting the target force item(s) prior to adding the Falloff, when using the 'Add Falloff' button in the recoil toolbox. When applied to a Soft Body item, users can use the 'Deformers' viewport, and select the 'Soft Body' Influence from the list, by RMB+clicking on the item to open the context menu, users can select the 'Add Override' option and select one of the Falloffs, automatically assigning it to the Soft Body. Additionally, users can use the 'Schematic' viewport to connect the two; dragging both items into the viewport (via drag and drop) and connecting the output of the Falloff item to the input of the Force item. For more information on using the Schematic Editor, please reference that page of the documentation.

Make Anchors

The 'Make Anchors' options is used to connect Soft Body items to Rigid Body items (or vice-versa). For simulations sake, it doesn't matter if the 'Rigid Body' item is *Active or Passive*. This could be useful to add some geometry to the end of a rope, for instance, or connect a flag to a flagpole. In order for the Anchor option to work, user must follow this specific workflow- First in 'Items' mode, users need to select the target Soft Body item AND the target Rigid Body item. With the items selected, change into the 'Vertices' mode (press '1' on the keyboard) and select at least one vertex on the Soft Body item, and at least one more on the Rigid Body (selection order isn't important). With the vertices selected, press the 'Make Anchors' option in the recoil toolbox. The two items

are now rigidly connected together via an additional 'Anchor' item added to the 'Soft Body'. To break the connection, simply delete the resulting 'Anchor' item, to make adjustments, remove the Anchor and recreate it, following the above instructions.

Play/Run

A recoil simulation can be calculated and cached simply by playing the timeline using the standard transport controls. For the best results, users should disable the 'Play Real Time' option. Additionally, recoil's 'Play' and 'Run' commands are provided offering some useful alternatives. The 'Play' command will process the entire 'Current' timeline, calculating the simulation for every frame, displaying the results in the 3D viewport as it is updated regardless of the 'Play Real Time' setting. The 'Run' option will do basically the same thing, but reduces overhead by not updating the 3D viewport until the entire timeline is calculated, providing a nice time savings for long or complex simulations. Users should note that neither process can be interrupted until complete. If you wish to stop the simulations before it is complete, it is best to use the standard transport controls.

Additional recoil Considerations

Items vs. Components

Animation in modo applies only to 'Items'. Keyframes are always set while in 'Items' mode with the modeling of objects done while at the component level, in the 'Vertices', 'Edges' and 'Polygons' modes. It is helpful for users to have an understanding of the differences between these modes. Layers in the items list should be thought of as containers. In 'Items' mode, users are editing the entire container, but while in the component modes, users are editing the contents of the container. recoil acts only at the item level, where each layer in the 'Items' list is its own dynamic recoil object. Therefore, if users wish to break apart a wall, each brick will need to be its own individual item layer. Users can easily accomplish stacks of bricks by simply creating the first brick item, and then assigning the 'Active Dynamic Body' tag to make the brick dynamic. Next, applying the 'Array' function in items mode will stack subsequent bricks, each inheriting the dynamics of the master object. With the wall created, all it takes is an animated wrecking ball to topple the bricks to the ground, complements of recoil.

Centers and Mass

The weight of Rigid Body objects is dependant on the position of the items 'Center'. The center point for an object doesn't only represent the center of rotation for the object, when using recoil it also represents the center of mass for the object. recoil will assume the weight of an object is equally distributed around the center point, so object

will move based on this assumption. If the center point is placed too far away from the items actual center, or worse outside the volume of the object, odd behavior is sure to occur. Users can see an object Center position by hovering over the 3D viewport in question and pressing the 'O' key. In the 'Visibility' subtab, there is an option to show Centers for selected items.

There are several ways to deal with a center that's been moved, the easiest is to select the item in 'Items' mode and invoke the menu bar command "Edit > Center to Bounding Box > Center"; this will automatically move the Center point to the objects actual bounding box center (a bounding box is based on a models overall extents).

Users cannot manually (by keyframes) animate active dynamic objects, however, sleeping objects can be animated (and are handed off to recoil once they are awakened). If the animated object requires rotations from a position other than the objects true center, users should modify the 'Pivot' position and animate from there to maintain the proper center of mass.

Accuracy

The number one issue users run into when calculating a simulation is objects passing through one another (like dropping chain links together to form a chain and they fall right through each other). While it can be caused by a multitude of issues, it is most often caused by the speed at which objects move. At one frame they were apart from each other, the next they had passed right through one another. This means the default subframe calculations didn't have ample time to calculate the collisions of the objects. Even though we only see the result at the whole frame values, and lot can happen between frames. Users can increase this sub-frame accuracy with the 'Steps' values found in the recoil solver (properly calculating subframes for all keyframed items as well). Doing so makes recoil more accurate, however the additional calculations take longer to compute; more accurate = slower. Users should experiment with different values to find one that produces acceptable results with the least number of steps.

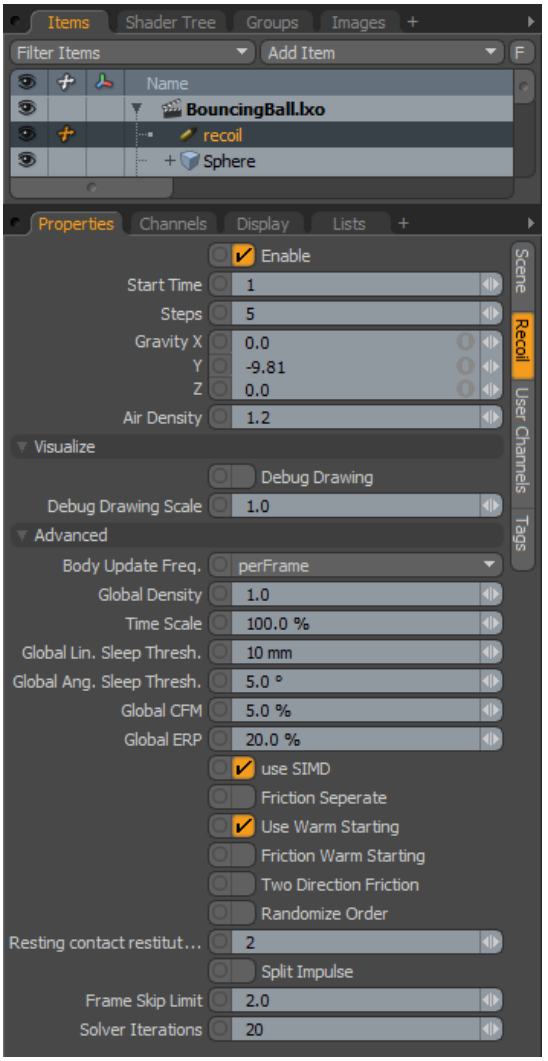
Bullet Physics

recoil is based on Bullet Physics, an open source collision detection and rigid body dynamics library that is free for commercial use, under the ZLib license. Thank you to Erwin Coumans and the Bullet developers for their efforts and generosity.

recoil Solver

When any item in the scene is defined as either an 'Active Rigid Body' or a 'Passive Rigid Body', a recoil Solver item is added automatically to the scene in the 'Items' list.

Only a single instance of the Solver item is ever necessary in a scene, as it controls all the global parameters of the simulation, such as time 'Steps', 'Gravity', 'Start Frame', etc. If the recoil Solver item is ever accidentally deleted from the scene, users may add another with the menu bar command "recoil > Add Solver" or by using the 'Add Items' function of the 'Items' list, LMB+click to open the menu and select the "Other > recoil" option. With the item selected in the 'Items' list viewport, a series of attributes can be modified in the 'Properties' panel using the standard modo control conventions.



Enable: This value can be toggled like an on/off switch. *Enabled* by default, users can *disable* this option to temporarily suspend the recoil dynamics simulation calculations.

Start Time: Expressed as a whole frame value, the dynamics simulation will begin to calculate from this point forward. Users may change this value to offset the calculations of a dynamic simulation to a later point in an animated sequence.

Steps: The 'Steps' value represents the subframes recoil calculates between frames. Even though users only see the results of a simulation at each whole frame step, there can still be a lot of movement happening between frames, especially with fast moving objects. If objects appear to intersect or are moving through each other, increasing this value will help to eliminate these errors, however, keep in mind as the Steps value increases, so do the number of calculations. Higher numbers offer more precise simulations at the expense of greater time to calculate.

Gravity X/Y/Z: The Gravity X/Y/Z settings control the direction and magnitude of *gravity*. Gravity is a world force value that affects everything in the scene. The default value of

-9.81 simulates the same downward pull of gravity on earth at a 1:1 scale (meters per second).

Air Density: The 'Air Density' option defines a global *drag* for 'Soft Body' items. The greater the value, the greater the drag.

Visualize--

Debug Drawing: Draws the internal representation of what the recoil simulator actually sees. It can be overwhelming to see in complex scenes especially, but can be useful for troubleshooting simulations that don't act as expected.

Debug Drawing Scale: Simply scales some of the debug drawing elements. Normally this will be the locators.

Advanced--

Body Update Frequency: This option determines how often recoil collects Channel and Deformations data to update its internal representations each dynamic item in the scene-

Begin- Internal representation is updated only once at the simulation start.

Per Frame- Internal representation is updated at each whole frame value.

Per Tick- Internal representation is updated at each 'Step', producing subframe accurate movements, but creating a large amount of additional overhead which may greatly slow down the simulation of the scene.

Global Density: Density is the measure of the amount of matter in any given volume. In modo, geometric objects are technically hollow, and therefore have no density, but if objects were simulated this way, they would just blow away like paper shells. Providing a density value for dynamic objects increases the realism of a simulation. Like *metal* versus *wood*, different materials have different densities, affecting their weight, and therefore the way they move when simulated. The 'Global Density' option determines the density of dynamic objects globally in the scene. Users can adjust this value, affecting all dynamic objects scene wide. Users can also override the 'Global Density' individually per item by setting the 'Mass Source' option in the items Dynamic properties to one of the 'Local' options and defining a value individually.

Time Scale: Generally a simulation moves items in time based on real world values of simulated forces. Users can use the 'Time Scale' option to globally adjust the speed at which a simulation runs simulating slow-motion or super-speed effects when the value is lowered or raised respectively.

Global Lin. Sleep Threshold: When objects in a recoil simulation stop moving, they are put into a sleeping state so that motion values don't have to be calculated for every object at every frame. The 'Global Linear Sleep Threshold' defines the lower threshold for putting these objects to sleep. Users may also override this value individually with the 'Linear' option found in the 'Dynamic' items properties.

Global Ang. Sleep Threshold: When object in a recoil simulation stop rotating, they are put into a sleeping state so that motion values don't have to be calculated for every object at every frame. This option defines the lower global angular (*rotational*) threshold for putting these objects to sleep. Users may also override this value individually with the 'Angular' option found in the 'Dynamic' items properties.

Global CFM: CFM stands for 'Constraint Force Mixing' and is a tweaking value for Constraints. CFM adds some small value to the main diagonal of the constraint matrix, to avoid degenerative matrices. Basically it helps to resolve instability in a Constraints calculations, in effect *softening* the constraint solution.

Global ERP: ERP stands for 'Error Reduction Parameter'. When applying CFM, ERP defines the rate at which the errors are reduced.

Use SIMD: The 'SIMD' option is a parallel computing optimization used when calculating recoil dynamics, *enabled* by default.

Friction Separate: When two objects slide against each other friction between the surfaces occurs. When *enabled*, this option forces recoil to calculate *friction* between two object separately. This can produce a more accurate simulation, but with a hit to performance.

Use Warm Starting: recoil utilizes an iterative algorithm where each iteration is based on the solution of previous iteration. If 'Use Warm Starting' is *disabled* the initial solution for the solver algorithm is set to zero for each frame. When 'Use Warm Starting' is *enabled*, each new iteration uses the last solution of the previous frame. This improves convergence towards a better solution, especially regarding recoil's stacking stability.

Friction Warm Starting: The 'Friction Warm Starting' option is basically the same concept as the 'Use Warm Starting' option above, but affects the calculations of friction between objects.

Two Direction Friction: If two objects surfaces are sliding parallel against each other for more than a few frames, errors can be introduced, enabling 'Two Direction Friction' can help to reduce or eliminate these errors.

Randomize Order: Normally constraints and collisions are calculated in order.

Enabling the 'Randomize Order' option causes the order objects are calculated to be randomized. When enabled, this option can aid in the stacking stability of recoil simulations. It has a very small performance cost which is why it is *disabled* by default.

Resting Contact Restitution Threshold: By default, recoil implements a mechanism for resting contacts to improve stability, by setting the restitution to zero. It compares the contact lifetime using a threshold to determine 'resting' contact.

Split Impulse: By default, recoil solves positional constraints and velocity constraints coupled together. This works well in many cases, but the error reduction of position coupled to velocity introduces extra energy (noticeable as *bounce*). Instead of coupled positional and velocity constraint solving, the two can be solved separately using the 'Split Impulse' option. This means that recovering from deep penetrations doesn't add any velocity.

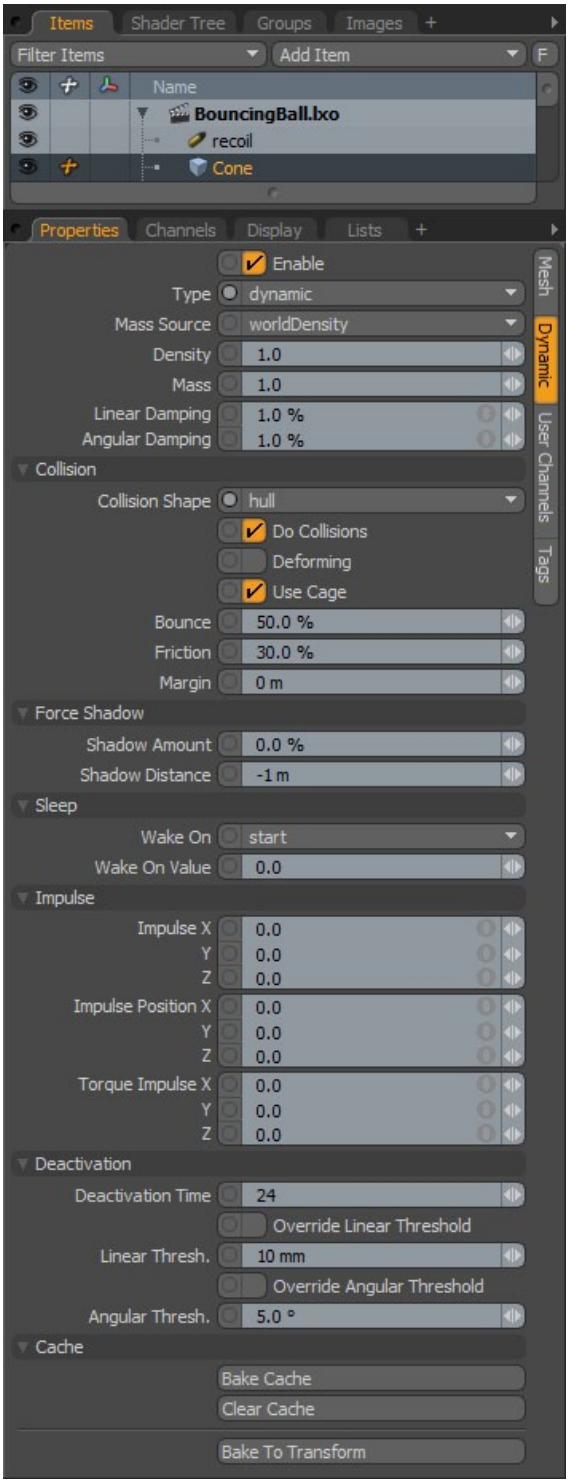
Frame Skip Limit: When 'Play Real Time' is enabled, modo maintains the playback framerate by skipping frames if necessary. The 'Frame Skip Limit' determines the biggest jump that there can be before recoil will stop trying to calculate the simulation. If this number is really large, modo may appear to hang while recoil tried to catch up.

Solver Iterations: This value determines the number of iterations between frames that the constraint solver uses to resolve collisions. Increasing this value will produce more accurate Constraint simulations, but at the expense of longer simulation calculation times.

recoil Dynamic Item

In order for any item in modo to be considered in a recoil dynamics simulation, it must first be converted to a dynamic item. This is done by tagging it as either a 'Passive Rigid Body' or an 'Active Rigid Body'. To apply the tag, simply select the item while in 'Items' mode and invoke either the 'Make Active Rigid Body' or Make Passive Rigid Body' commands. These can be found in the menu bar under the 'recoil' menu, or within the recoil toolbox found, by default, in the 'Animation' interface in the recoil sub-tab of its toolbox. Once the appropriate tag is defined, an additional property sub form called 'Dynamic' is added to the items properties panel with a series of attributes relating to the settings of the object and its dynamic simulation. While the default values will work in many cases, users will likely wish to modify the default options to settings more appropriate for their intended purpose.

To make an object un-dynamic, removing it completely from the recoil calculations, users simply need to select the target item and invoke the menu bar command "recoil > Make Un-dynamic" eliminating any dynamics information from the object. To temporarily disable the object from the dynamics simulation, toggle the 'Enable' option.



Enable: Toggles dynamics on and off for the item. When enabled, modo will calculate dynamics for the item when the timeline is played (or the 'Run Simulation' command is invoked). When disabled item will be ignored for dynamics and collision calculations.

Type: Dynamic items can be one of two types-

Dynamic- A collision object with a density greater than 0, that is controlled exclusively by recoil dynamics. It can also be affected by Constraints, Forces, and Falloffs. This is the option automatically designated when an object is defined as an 'Active Rigid Body'. Items tagged as Dynamic will be represented in the viewport with a white bounding box.

Kinematic- A collision object that is either static or controlled by user created keyframes. Kinematic objects are unaffected by Forces, such as gravity or Falloffs. This is the option automatically designated when an object is defined as a 'Passive Rigid Body'. Items tagged as Kinematic will be represented in the viewport with a green bounding box.

Mass Source: By default, with 'Mass Source' set as 'World Density', the mass of an object is calculated by multiplying the calculated volume of the dynamic item by the 'Global Density' value defined on the

recoil Solver item. Users can override that value by setting the 'Mass Source' to 'Local Density' or 'Local Mass' and expressly setting a value. The calculated volume is shown in the objects 'Volume' channel in the Channels viewport.

Density: Density is the measure of the amount of matter in any given volume. If the 'Mass Source' option is set to 'Local Density', this is the density value that will be used. The mass of the object will be determined by multiplying the computed volume by this value.

Mass: Mass is the properties of an object that give it bulk and weight. It is essentially a different way of determining *density.* If 'Mass Source' is set to 'Local Mass', this is the value that will be used for simulations.

Linear Damping: This is a force that acts upon the motion of objects, acting to slow them down over time. A small value here can benefit object stability since it helps objects come to rest.

Angular Damping: Same as 'Linear Damping' but applies to the rotation of the object.

Collision--

Collision Shape: This setting defines the shape that will be used to calculate collisions between dynamic objects. Generally users will want to choose the simplest shape that produces satisfactory results. The default collision shape is set to 'Hull' which generally balances accuracy with calculation speed.

BBox- A rectangular box defined by the bounding box of the geometry's maximum extents.

Sphere- A sphere that is sized to encompass all the geometry of the object.

Hull- The default collision option 'Hull' creates the smallest convex volume that encompasses all the points in this mesh. To visualize it, imagine the mesh enveloped in shrink-wrap. The 'Hull' collision type does not support holes or indentations in the object. While only approximating the items shape, it will provide very fast collision detection on complex shaped objects.

Mesh- Uses the actual mesh of the object for calculating collisions. For Subdivision Surfaces models, the actual frozen SubD cage is used. This should be used only if you need the total accuracy as performance can suffer with its use.

Plane- A flat plane that extends infinitely in the X and Z directions. This option should be applied when a ground plane that cannot be penetrated is desired. This option can only be applied to non-moving (static) items.

Convex Decomposition- The 'Convex Decomposition' collision shape tries to

approximate the shape of the original geometry with a greatly reduced polygon count, applying a reduction algorithm similar to modo's internal 'Polygon Reduce' function.

Static Mesh- An optimized mesh collider for non-moving geometry. This is ideal for displaced ground geometry.

Capsule X/Y/Z- A rounded end cylinder defined by the geometry of the object. Users may choose the appropriate axis for the object based on the objects un-transformed position.

Cylinder X/Y/Z- A cylinder defined by the geometry of the object. Users may choose the appropriate axis for the object based on the objects un-transformed position.

Do Collisions: Users can *disable* this option so that the object is not taken in to account for collision calculations (enabled by default).

Deforming: When the 'Deforming' options *enabled*, the mesh state will be updated per frame to reflect any deformations applied to it, so that when Deformers, such as MDDs and Morphs are applied to the object, they can properly collide.

Use Cage: When determining the collision shape of the item, *enabling* the 'Use Cage' option causes recoil to use the items un-subdivided cage if it is a Subdivision Surfaces shape. Otherwise it will use the actual subdivision surfaces geometry.

Bounce: This is the collision response when this object hits another dynamic object. A value of 0.0 means the object will not bounce or impart energy onto objects it hits. A value if 1.0 would conserve all of the objects energy.

Friction: The amount an object will slide against another object. A value of 0.0 would offer no resistance, while a value of 1.0 would keep the object from sliding entirely.

Margin: The 'Margin' value helps bullet to determine collisions with moving items. If the value is negative, recoil move the collision vertices of the item inward along their averaged normal and then extends the margin by the same amount which should result in a fairly tight fitting shape. If it is positive, recoil simply extends the collision vertices of the geometry outward by that amount.

Force Shadow--

The Force Shadow options block the affects of forces on objects and surfaces, in much the same way that objects block light from hitting surfaces. This can be useful to hide dynamic items from forces when located in 'protected' areas.

Shadow Amount: The 'Shadow Amount option controls to what degree the object

blocks the force value. 0% disables force shadows, attenuating toward 100% where the force would be completely blocked.

Shadow Distance: Determines the distance of the Force Shadowing effect, a setting of 1m would block forces up to 1 meter away form the surface, a value of -1m (negative one meter) sets the shadowing to infinity.

Sleep--

Wake On: Normally objects are *awake* (active) when any dynamic simulation starts and the object begins to react immediately. Users can change this behavior delaying when individual objects *wake up* and become active. Prior to the 'Wake On' event, items can be keyframe animated and the object will be handed off to recoil for simulation when the certain 'Wake On Value' threshold is achieved. The values defined in the 'Impulse' channels are applied when the object wakes up.

Start- Object begins the dynamic simulation awake and fully active at the 'Start Time' value specified in the recoil solver.

Collision Velocity- Object wakes up when either another passive or active object collides with it. Users can use the 'Wake On Value' to set a *collision velocity* threshold necessary to wake up the object.

Collision Force- Object wakes up when either another passive or dynamic object collides with it. Users can use the 'Wake On Value' to set a *collision force* threshold necessary to wake up the object.

Velocity- Object wakes up when a certain velocity is reached. Different from 'Collision Velocity', this velocity is determined by keyframe animated motion. Users can use the 'Wake On Value' to set the velocity threshold necessary to wake up the object.

Trigger- Object wakes up when the keyframeable channel, 'reWakeOnValue' is non zero (*any number but zero*). This has good use with conditional nodes in rigging.

Frame- Object wakes up when a specific frame number is reached.

Time- Object Wakes Up when a specific time has been reached, regardless of frame-rate.

Wake On Value: Depending on the 'Wake On' option selected, users can set a lower threshold that must be reached in order to wake up the object. It may be helpful to reference the recoil specific channels (below) in helping to determine values appropriate to the users intended result.

Impulse--

Impulse X/Y/Z: A directional force that is applied to the object when it wakes up. Useful when a keyframed object is handed off to recoil, imparting direction and velocity to the object, which would otherwise just drop without it as recoil has no awareness of the objects velocity prior to waking up.

Impulse Position X/Y/Z: Position the above impulse is applied from affecting the objects trajectory.

Torque Impulse X/Y/Z: A rotational that is force applied to the object when it wakes up. Useful when a keyframed object is handed off to recoil, imparting direction and velocity to the object, which would otherwise just drop without it as recoil has no awareness of the objects motion prior to waking up.

Deactivation--

Deactivation Time: The 'Deactivation Time' defines how long a dynamic object must be below the Linear or Angular threshold before being put to sleep. The default value is one second, so depending on the frame rate of the scene, the default value will change; generally it is 24 frames.

Override Linear Threshold: Users can enable this function to put dynamic objects to sleep once they reach a defined upper threshold value, overriding the Global value found in the recoil Solver. Essentially the opposite of the 'Wake On' settings, once an object has been below the threshold for the defined length of time, recoil puts it to sleep.

Linear Threshold: This is the motion threshold value when 'Override Linear Threshold' is enabled.

Override Angular Threshold: Users can enable this function to put dynamic objects to sleep once they reach a defined upper threshold value, overriding the Global value found in the recoil Solver. Essentially the opposite of the 'Wake On' settings, once an object has been below the threshold for the defined length of time, recoil puts it to sleep.

Angular Threshold: This is the rotation threshold value when 'Override Angular Threshold' is enabled.

Cache--

Bake Cache: Selecting the 'Bake Cache' option bakes the currently cached dynamic simulation to individual keys at every frame for both the item's position and rotation

channels, and then disables the dynamic item. This is done in local space by default. For more information on baking, please reference the 'Baking and Rendering' page of the recoil documentation.

Clear Cache: Selecting the 'Clear Cache' option clears the items cache, essentially removing the items simulation results.

Bake to Transform: Applying the 'Bake Cache' option (above) converts the dynamic simulation to a series of animation keyframes, for still scene renders, the animation keyframes are an unnecessary overhead, so the 'Bake to Transform' option provides users a means to simply freeze the position of the object at the current time in the simulation removing any dynamics or animation from the object.

Rigging and Dynamic Items--

When an item is converted to a dynamic item, there are a number of feedback channels only available in the 'Channels List' when the item is selected (recoil specific channels are preceded by an 're' value). These extra channels are useful for the rigging of dynamic items. The use of Schematic editor for rigging and channels in general is covered in modo's in-line help system. These additional channels are only valid during the recoil simulation.

reCenterOfMass: The calculated center of mass.

rePosition: The items current position in world space.

reRotation: The items current rotation in world space.

reVolume: The items calculated volume.

reVelocity: The items current velocity.

reAngularVelocity: The items current angular velocity.

reHitCount: The number of times this item has collided with other bodies.

reJustHit: True if this item was involved in a collision the last time step.

reLastTime: The last time this item was involved in a collision.

reLastHitForce: The force this item was hit with for the last collision.

reLastHitPosition: The world space position of this items last collision.

reLastHitNormal: The world space normal of this items last collision.

recoil Soft Body Item

recoil Soft Body items allow users to deform geometry as if it were made of a soft, pliable material. Unlike Rigid bodies that do not deform at all, soft body items are calculated as if springs were attached between adjoining vertices, providing a flex and stretch type warping of the surface. With it, it is possible to simulate a wide variety of materials by adjusting the various properties, from soft billowy fabric, to wiggly gel-like surfaces, to stiff cardboard-like surfaces. With the addition of Pin Maps, Soft Bodies that react to animation are also possible, such as the thin flexible membrane surface on a bat wings as it flies, the Deformation of the wings flapping is applied through a Skeleton and Joint animation, with the secondary membrane motion applied as a Soft Body, utilizing a Pin map.

Apply Soft Body Deformer

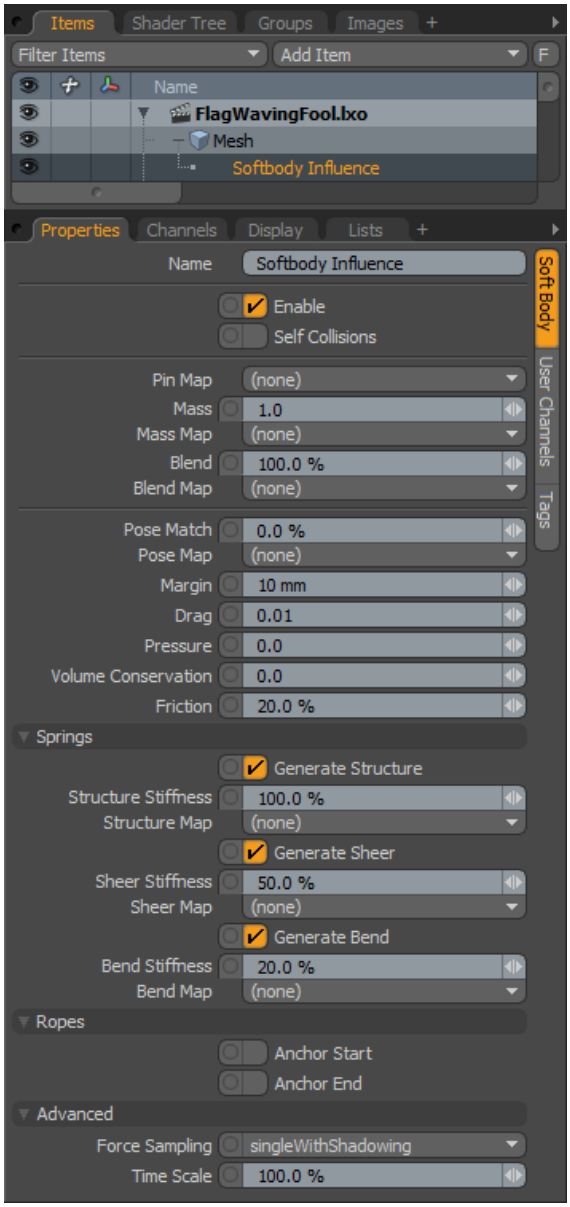
Applying a Soft Body deformer only requires that the target mesh item be selected, then simply LMB+clicking the 'Make Soft Body' button in the 'recoil' submenu, of the toolbox in the 'Setup' and 'Animate' interface tabs. This applies the 'Soft Body Influence' item, linking it to the mesh. From there, users can access the attributes of the item within the 'Properties' panel. In order to see the results of the Soft Body deformations users will need to 'Run a Simulation', this action actually calculates the deformations based on the settings of the Deformer. For more information on working with dynamic items, please reference the recoil overview page of the documentation.

Make Particle Body

The 'Make Particle Body' option defines any grouping of vertices, such as a point cloud, as Soft Body item, but automatically disables the 'Spring' options that are normally used to give shape to deforming geometry. This allows the particles to collide and interact with other Soft and Rigid body elements in the scene, as well as with Forces and Falloffs. When combined with 'Blobs', 'Sprites' or 'Volumes', a number of interesting particle effects can be created.

Make Ropes

The 'Make Ropes' option applies the Soft Body Deformer to any of modo's curve path types, created using either the Curve or Bezier tools. With only a Structure spring applied, the curve will dangle and flex as if made of rope, retaining its overall length, but with gravity, eventually resolving to a straight line. Users can assign the 'Anchor Start' and/or 'Anchor End' options to automatically pin the start and end points, fixing them to their start of simulation position. Additionally, using the Anchor option, users can fix any of the points along the curve to another Soft or Rigid Body item, such as to make a wrecking ball. Users will need to *enable* the 'Render Curve' option, in order for the resulting curve to be visible in the final rendered image.



Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Enable: Toggles dynamics on and off for the item. When *enabled*, modo will calculate dynamics for the item when the timeline is played (or the 'Run Simulation' command is invoked). When *disabled* item will be ignored for dynamics and collision calculations.

Self Collisions: When enabled, items will be able to collide with themselves (such as when one end of a sheet is *passing through* another). This option is only necessary for surfaces that inter-penetrate, as it will slow down the simulation.

Pin Map: A 'Pin Map' is simply a Weight Map (or Weight Container) that controls what areas of a surface are affected by the Soft Body Deformations. Limiting them to specific areas can be useful to create a wide variety of organic effects, such as adding secondary motions to characters. Vertices are either fully pinned or not pinned, as every non zero value is treated as a 1.0 value (values aren't attenuated).

Mass: The 'Mass' option controls the overall mass or weight applied to the item. The amount entered is divided by the number of vertices and the same amount is applied each one.

Mass Map: The 'Mass Map' is simply a Weight Map (or Weight Container) that controls how the mass value is applied to the mesh per vertex. The Vertex Map values are normalized by the Mass value. Example: if the 'Mass' is 5.0, and in the mass map there

are three vertices with weight values of 1.0, 0.5, 0.5, those vertices would masses of 2.5, 1.25, 1.25

Blend: The 'Blend' option controls how much of the mesh moves toward the simulated positions. A value of 100% means that the points in the mesh exactly match the simulated positions attenuating towards 0% where item would not deform at all (and act as if pinned).

Blend Map: The 'Mass Map' is simply a Weight Map (or Weight Container) that controls how the 'Blend' value is applied to the mesh per vertex.

Pose: The 'Pose' option controls to what degree a surface tries to maintain its original, undeformed shape (this differs from 'Blend' in that Blend maintains the original vertex positions, while 'Pose' maintains the original shape).

Pose Map: The 'Pose Map' is simply a Weight Map (or Weight Container) that controls how the 'Pose' value is applied to the mesh per vertex.

Margin: The 'Margin' determines a distance away for the surface that is added to the size of the object, useful for controlling collision detection.

Drag: The 'Drag' option determines the loss of energy on a surface that is in motion (say for jiggling type of effects).

Pressure: The 'Pressure' value exerts a force upon the overall volume of an item, similar to how pressure would react on an object in the real world, increasing values would add overall volume, inflating it and decreasing (negative) values would reduce its volume, deflating it. This option requires that the target mesh be completely watertight (no seams).

Volume Conservation: The 'Pressure' option tries to maintain an overall internal volume during the deformation simulation. So, if one side of an object is squashed, the other should balloon out. This option requires that the target mesh be completely watertight (no seams).

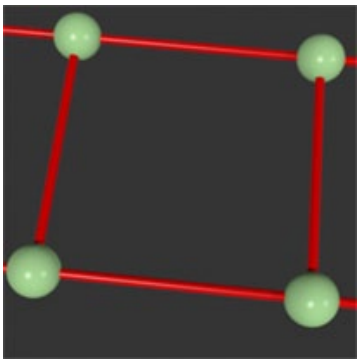
Friction: The amount an object will slide against another object. A value of 0.0 would offer no resistance, while a value of 1.0 would keep the object from sliding entirely.

Springs--

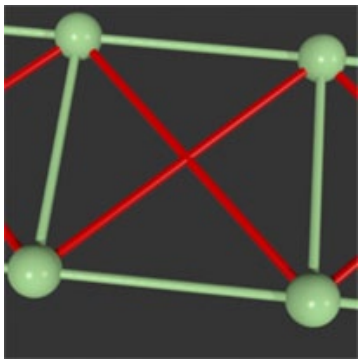
The Springs controls the rigidity of the object, and determine wether of not it is stiff or squishy and to what degree the surface maintains its original, undeformed shape. There are three types of springs that all provide different types of structure to the surface. All spring values are defined by the base value multiplied by the value in any associated Weight Map, if it has one.

Springs--

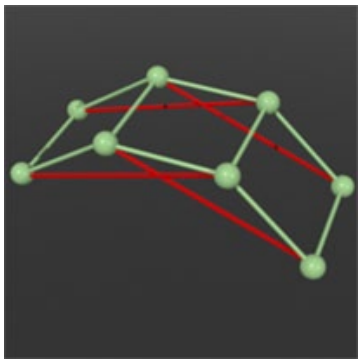
The Springs controls the rigidity of the object, and determine wether of not it is stiff or squishy and to what degree the surface maintains its original, undeformed shape. There are three types of springs that all provide different types of structure to the surface. All spring values are defined by the base value multiplied by the value in any associated Weight Map, if it has one.



Structure Springs



Shear Springs



Bend Springs

Generate Structure: When 'Generate Structure' is *enabled*, springs are generated that follow along the lines of the mesh, connecting the vertices along their polygon *edges*.

Structure Stiffness: This value determines how resistant the Structure springs are when deforming, depending on the value, they can be very rigid or very pliable.

Structure Map: With the 'Structure Map' option, users can assign a Weight Map that modulates Structure Stiffness as a multiplier. If 'Structure Stiffness' is set to 50%; with a Weigh Map value of 0.6 (60%), the final calculated Stiffness value would be 30%.

Generate Shear: When 'Generate Shear' is enabled, springs are generated that add extra connections to the inside of a face, on a quad polygon, the springs will form an 'X' shape. In a triangle, no additional springs are created. In an Ngon, a spring connected to each non-neighboring vertex. These springs help prevent the polygon from collapsing in on itself.

Shear Stiffness: This value determines how resistant the Shear springs are when deforming, depending on the value, they can be very rigid or very pliable.

Shear Map: With the 'Shear Map' option, users can assign a Weight Map that modulates the overall strength of the Shear as a multiplier. If 'Shear Stiffness' is set to 50%; with a Weigh Map value of 0.6 (60%), the final calculated Stiffness value would be 30%.

Generate Bend: When 'Generate Bend' is enabled, springs are generated that help an object maintain it's shape, following along the Structure Springs. These springs work to maintain the shape of the object by applying rotational force to maintain the continuity of the surface.

Bend Stiffness: This value determines how resistant the Bend springs are when deforming, depending on the value, they can be very rigid or very pliable.

Bend Map: With the 'Bend Map' option, users can assign a Weight Map that modulates the overall strength of the Bend as a multiplier. If 'Bend Stiffness' is set to 50%; with a Weigh Map value of 0.6 (60%), the final calculated Stiffness value would be 30%.

Ropes--

Ropes are Soft Body deformations applied to modo curves. These can be Curves , Bezier Curves, curves used as Hair Guides, or rendered curves utilizing the Item attribute.

Anchor Start/End: The 'Anchor Start' and 'Anchor End' options are simple toggles allowing easy pinning of either the start or end point of a curve (or both). See 'Pins' above for information on what Pinning is.

Advanced--

Force Sampling: Determines how force values are sampled and applied to the mesh--

single- A single call is made to the force and it is applied evenly to all vertices in the mesh.

singleWithShadowing- A single call is made to the force(from the objects center) and it is applied evenly to all vertices in the mesh. Intervening shadow items will modulate the force amount.

perVertex- Forces are queried individually for every vertex, based on it's current position.

perVertexWithShadowing- Forces are queried individually for every vertex based on it's current position. Intervening shadow objects will modulate the force amount.

Time Scale: Generally a simulation moves items in time based on real world values of simulated forces. Users can use the 'Time Scale' option to adjust the speed at which a simulation runs, per soft-body, simulating slow-motion or super-speed effects when the value is lowered or raised respectively.

Constraints

Constraints are basically joints, affecting the movement of dynamic objects in a scene, restricting their motion in very specific ways dependant on the type of constraint applied. recoil offers a variety of Constraint types providing users a means to produce very controlled dynamic simulations. Additionally, Constraints can be combined together (by way of locators) producing even more control to complex dynamic simulations.

Constraint Types			Special Constraints
Using Constraints	Box	Sliding Hinge	Motor
Pin	Universal	Make Constraints	Spring
Point	Slide		Anchor
Plane	Hinge		

Using Constraints

Constraints in recoil are simply joints that connect two dynamic bodies together. Each constraint provides a different type of joint, limiting the motions of the dynamic objects in very specific ways. Typically constraints are used to relate the dynamic motion of an 'Active Rigid Body' item to that of a static or animated 'Passive Rigid Body' item. For instance, if a user wanted to animate a pair of dice swaying from the rear view mirror of a car, the car itself (including the mirror) would be defined as a 'Passive Rigid Body', and the pair of dice within would be defined as a 'Active Rigid Body'. Next the user would assign a 'Point' constraint to the mirror *and* the pair of dice and position it near the mirror. If the car were animated to move as if it was travelling over rough terrain, running a subsequent recoil simulation would then produce an incredibly realistic secondary swinging motion to the pair of dice without the need to add a single keyframe.

Constraints are also a powerful way to provide control of Active Dynamic objects, relating the motion of one to that of another. This is handled in the same way as the above example, but both items would be designated as 'Active Rigid Bodies'. Both objects would then move relative to the motions of the other, controlled by the constraint joint. Furthermore, series' of constraints can be added to consecutive objects, providing linked motions for the entire length of items. This is applied by selecting the chain of items in order and then applying the constraint.

Applying Constraints

Constraints can be added to scenes like any other item using the Items lists 'Add Item' function. However, by selecting the target items the user wishes to join together and utilizing the recoil toolboxes many Constraint buttons, the constraint item will automatically link to the selected items. To link two items together, first select the master or parent item, this is typically the keyframed or static Passive Dynamic object, next select the child or auxiliary item, this will always be an *Active Dynamic* object. Next choose from one of the many Constraint options available, LMB+clicking on the button to add the item to the scene. When a Constraint is added this way, the Constraint item automatically populates the 'Body A' and 'Body B' options within the constraint, eliminating the need for the users to add them manually. It is also positioned at exactly the middle point between the centers of the two items. Depending on your intentions, this will likely not be the optimal position for the Constraint. The motion of the child item (Body B) will originate from the position of the Constraint item itself, so it will need to be positioned appropriately.

Resting Frame

It's important to understand that the positioning of Constraints is set at what is called the *resting frame* this is the initial position of the object, when it is fully at rest. This is typically frame 0, but can be a different frame for delayed simulations. Setting this position prior to when the simulation start is necessary to ensure that the proper results are produced.

Removing Constraints

Constraints are just items like any other in a scene and can be removed in the same way by simply selecting the target Constraint and invoking the menu bar command "Edit > Delete" or by pressing the Delete key on the keyboard.

Breaking Joints

Users can *break* a constraint once a certain force threshold is achieved, this essentially disables the constraint allowing connected objects to fall away from each. This is useful if, say for instance, two objects were tethered together by a chain, when the chain is stressed to a specific point, the links would fall apart, each link then acting as an individual dynamic bodies, and the tethered item would no longer be under the influence of the constraint. Clever rigging types could also link a morph of the chain to the 'isBroken' channel of the Constraint, so that once it breaks, the link itself would appear broken. To activate this function, toggle the 'Breakable' option of the Constraint, and set a lower threshold. Once that value or greater is achieved through a combination of the forces applied to the item, including gravity, Forces and any force imparted by the items own movements the link will break apart.

Pin Constraint

The 'Pin' constraint type is a rigid constraint meant to rigidly attach two or more dynamic items together; almost literally like pinning the two items together. This can be useful to attach together two object with dramatically different Mass values, such as a the metal head of a hammer attached to its wooden handle. The heavier head will affect the overall of the hammer. It is also useful to rigidly combined *shattered* objects that are meant to break apart when combined with the 'Breakable' function.

To apply a constraint automatically, users can select two items prior to invoking any of the Constraint types. The first item selected represents the master or parent item and will be populated as 'Body A' within the constraint, the second item selected will be the item that is constrained, also considered the auxiliary or child item and will be populated as 'Body B'. With the items selected, LMB+click on the 'Pin' constraint button, adding the constraint item to the 'Items List'. The default center location (position) of the Pin Constraint will be exactly half-way between the Body A and Body B objects with a line drawing to the centers of the constrained items.

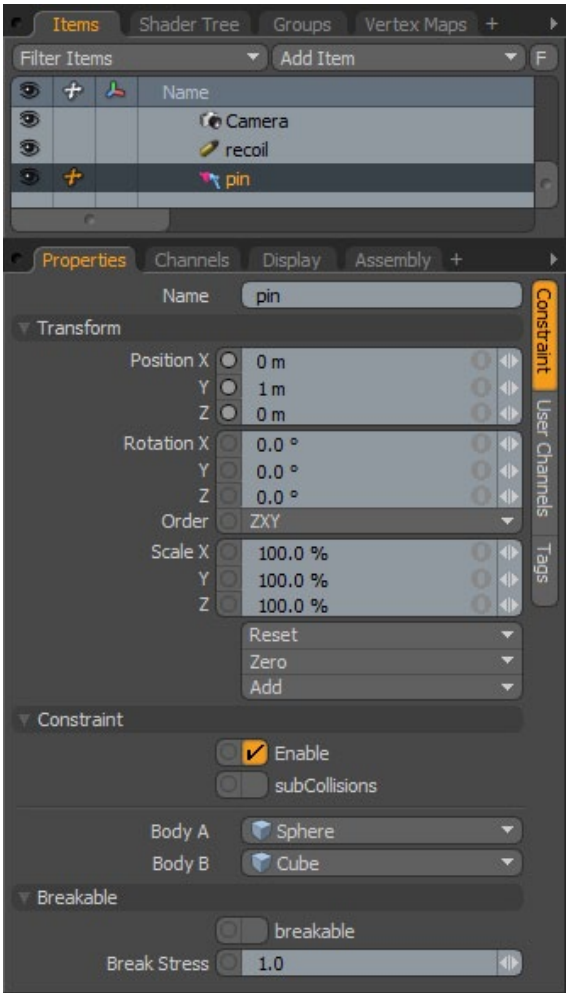
Name: This data field displays the current Constraint item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position X/Y/Z: The Position XYZ values represents the initial center for constraint item in 3D world space, it also represents the center of mass relative to the item it is constraining. Based on its initial position, once a dynamics simulation is invoked, the constraining item will rotate and move based on this initial (resting) position.

Rotation X/Y/Z: The Rotation XYZ value represents the initial (resting) rotation of the constraining item.

Order: The 'Order' option allows the user to set the order that rotations are applied to the item, each producing a different result. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.



Scale X/Y/Z: The Scale XYZ values have no direct affect over Constraints.

Reset: Resets the selected transform values to (0,0,0) returning the item back to its default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when keyframe animating complex motions, providing a layered approach to the applied transforms. There are no benefits to applying additional transforms to dynamic objects.

Constraint--

Enable: The 'Enable' option toggles the Constraint item on or off. When *enabled* the constraint item will be considered during a dynamics simulation, when *disabled*, the constraint item will be ignored. However, *disabled* constraints are persistent across modo sessions being saved with the scene, and retain their present settings.

SubCollisions: The 'SubCollisions' option, when *enabled*, allows constrained objects (especially those within a chain, not a literal chain but subsequent groupings) to collide with one another. When disabled (the default state), fewer calculation are required, however, constrained items may inter-penetrate each other. If this occurs users will want to *enable* SubCollisions.

Body A: The 'Body A' setting represents the item that is constrained to, this could also be considered the parent or master item.

Body B: The 'Body B' setting represent the item that is constrained, this could be considered the child or auxiliary item.

Breakable--

Breakable: The 'Breakable' option allows constraints to literally be broken apart when a certain stress threshold is reached, defined by the 'Break Stress' value. When the threshold is reached, the constraint will no longer have any affect on the constrained item, leaving it to be affected by any other forces present in the simulation.

Break Stress: The 'Break Stress' value determines the threshold when the constraint will no longer affect the constrained item. The 'Breakable' option must be *enabled* for this value to have any affect on the constraint.

Point Constraint

The Point Constraint attaches an object to a single point, providing an unrestricted dangling or swaying type motion when applied. To visualize this constraint, users can imagine the constrained item were tied to a piece of string which had been affixed to a specific point in 3D space, as the parent item moves, the constrained item will swing relative to the motions of the parent item.

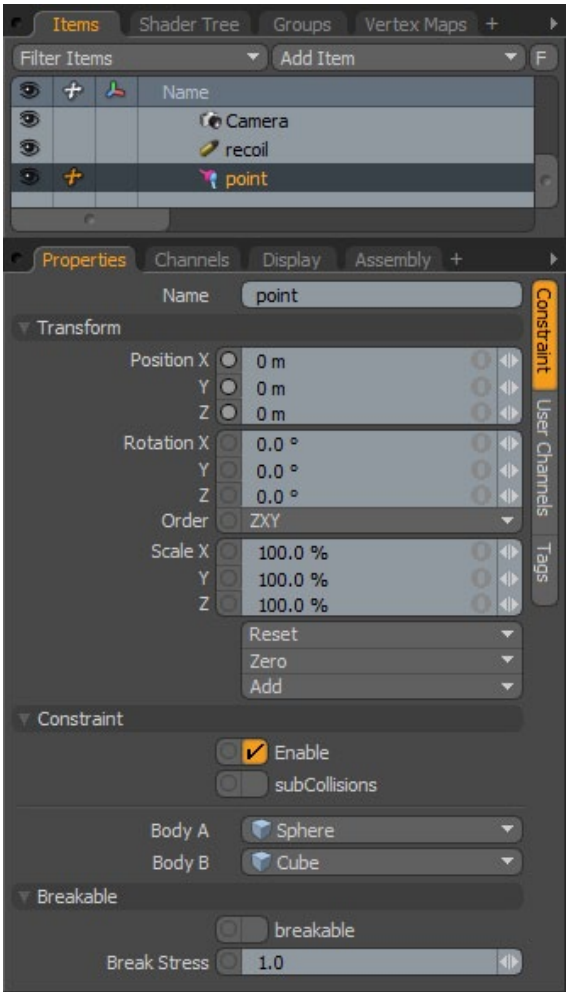
To apply a constraint automatically, users can select two items prior to invoking any of the Constraint types. The first item selected represents the master or parent item and will be populated as 'Body A' within the constraint, the second item selected will be the item that is constrained, also considered the auxiliary or child item and will be populated as 'Body B'. With the items selected, LMB+click on the 'Point' constraint button, adding the constraint item to the 'Items List'. The default center location (position) of the Point Constraint will be exactly half-way between the Body A and Body B objects with a line drawing to the centers of the constrained items. Depending on your intentions, this might not be the optimal position for the Constraint. The motion of the child item (Body B) will originate from the position of the Constraint item itself, so it may need to be positioned appropriately.

Name: This data field displays the current Constraint item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position X/Y/Z: The Position XYZ values represents the initial center for constraint item in 3D world space, it also represents the center of mass relative to the item it is constraining. Based on its initial position, once a dynamics simulation is invoked, the constraining item will rotate and move based on this initial (resting) position.

Rotation X/Y/Z: The Rotation XYZ value represents the initial (resting) rotation of the constraining item.



Order: The 'Order' option allows the user to set the order that rotations are applied to the item, each producing a different result. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale X/Y/Z: The Scale XYZ values have no direct affect over Constraints.

Reset: Resets the selected transform values to (0,0,0) returning the item back to its default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when keyframe animating complex motions, providing a layered approach to the applied transforms. There are no benefits to applying additional transforms to dynamic objects.

Constraint--

Enable: The 'Enable' option toggles the Constraint item on or off. When *enabled* the constraint item will be considered during a dynamics simulation, when *disabled*, the

constraint item will be ignored. However, disabled constraints are persistent across modo sessions being saved with the scene, and retain their present settings.

SubCollisions: The 'SubCollisions' option, when *enabled*, allows constrained objects (especially those within a chain, not a literal chain but subsequent groupings) to collide with one another. When disabled (the default state), fewer calculation are required, however, constrained items may inter-penetrate each other. If this occurs users will want to *enable* SubCollisions.

Body A: The 'Body A' setting represents the item that is constrained to, this could also be considered the parent or master item.

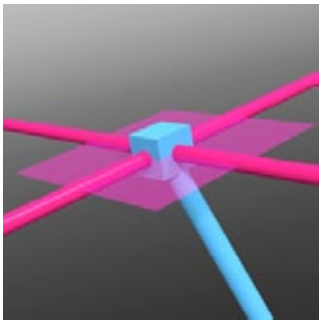
Body B: The 'Body B' setting represent the item that is constrained, this could be considered the child or auxiliary item.

Breakable--

Breakable: The 'Breakable' option allows constraints to literally be broken apart when a certain stress threshold is reached, defined by the 'Break Stress' value. When the threshold is reached, the constraint will no longer have any affect on the constrained item, leaving it to be affected by any other forces present in the simulation.

Break Stress: The 'Break Stress' value determines the threshold when the constraint will no longer affect the constrained item. The 'Breakable' option must be enabled for this value to have any affect on the constraint.

Plane Constraint



The Plane Constraint restricts the motion of an object to a flat X/Z plane (Y axis being the Up axis). This allows objects to slide over a surface. Users can control the size of the surface using the 'Limit' options and defining positions for each upper and lower side. The viewport representation for the constraint is a square denoting the edges of the plane, a diamond shape is present for each edge that is *limited*.

To automatically apply the Constraint, first select the item to constrain to, this will populate as the 'Body A' value when applied, and then select the constrained item, this will populate as 'Body B'. With the items selected, LMB+click on the 'Plane' constraint button, adding the constraint item to the 'Items List'. The default center location (position) of the Plane Constraint will be exactly half-way between the Body A and Body B objects with a line drawing to the center of the constrained items.

Name: This data field displays the current Constraint item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position X/Y/Z: The Position XYZ values represents the initial center for constraint item in 3D world space, it also represents the center of mass relative to the item it is constraining. Based on its initial position, once a dynamics simulation is invoked, the constraining item will rotate and move based on this initial (resting) position.

Rotation X/Y/Z: The Rotation XYZ value represents the initial (resting) rotation of the constraining item.

Order: The 'Order' option allows the user to set the order that rotations are applied to the item, each producing a different result. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale X/Y/Z: The Scale XYZ values have no direct affect over Constraints.

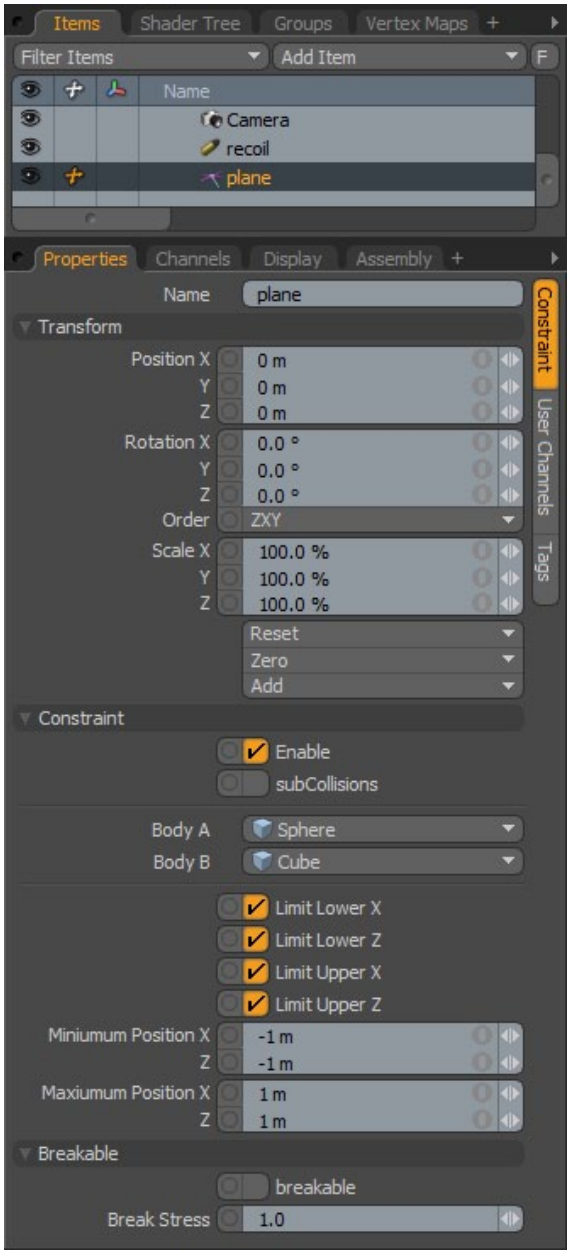
Reset: Resets the selected transform values to (0,0,0) returning the item back to its default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when keyframe animating complex motions, providing a layered approach to the applied transforms. There are no benefits to applying additional transforms to dynamic objects.

Constraint--

Enable: The 'Enable' option toggles the Constraint item on or off. When *enabled* the constraint item will be considered during a dynamics simulation, when *disabled*, the constraint item will be ignored. However, *disabled* constraints are persistent across modo sessions being saved with the scene, and retain their present settings.



SubCollisions: The 'SubCollisions' option, when *enabled*, allows constrained objects (especially those within a chain, not a literal chain but subsequent groupings) to collide with one another. When disabled (the default state), fewer calculation are required, however, constrained items may inter-penetrate each other. If this occurs users will want to *enable* SubCollisions.

Body A: The 'Body A' setting represents the item that is constrained to, this could also be considered the parent or master item.

Body B: The 'Body B' setting represent the item that is constrained, this could be considered the child or auxiliary item.

Limit Lower X/Z: When *enabled*, sets a boundary limit for the -X and -Z sides of the constraining plane. Represented by an upright diamond on the planes edge. When disabled, the plane will be infinite in that direction.

Limit Upper X/Z: When *enabled*, sets a boundary limit for the +X and +Z sides of the constraining plane. Represented by an upright diamond on the planes edge. When disabled, the plane will be infinite in that direction.

Minimum Position X/Z: When the 'Limit Lower X/Z' options are *enabled*, this value is the distance away from the center position of the constraint defining its edge. The constrained item will not be able to move further than this value.

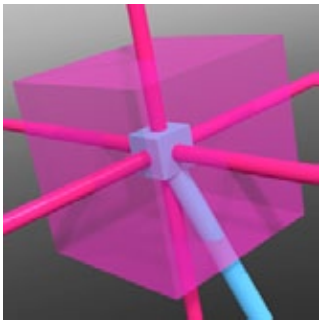
Maximum Position X/Z: When the 'Limit Upper X/Z' options are *enabled*, this value is the distance away from the center position of the constraint defining its edge. The constrained item will not be able to move further than this value.

Breakable--

Breakable: The 'Breakable' option allows constraints to literally be broken apart when a certain stress threshold is reached, defined by the 'Break Stress' value. When the threshold is reached, the constraint will no longer have any affect on the constrained item, leaving it to be affected by any other forces present in the simulation.

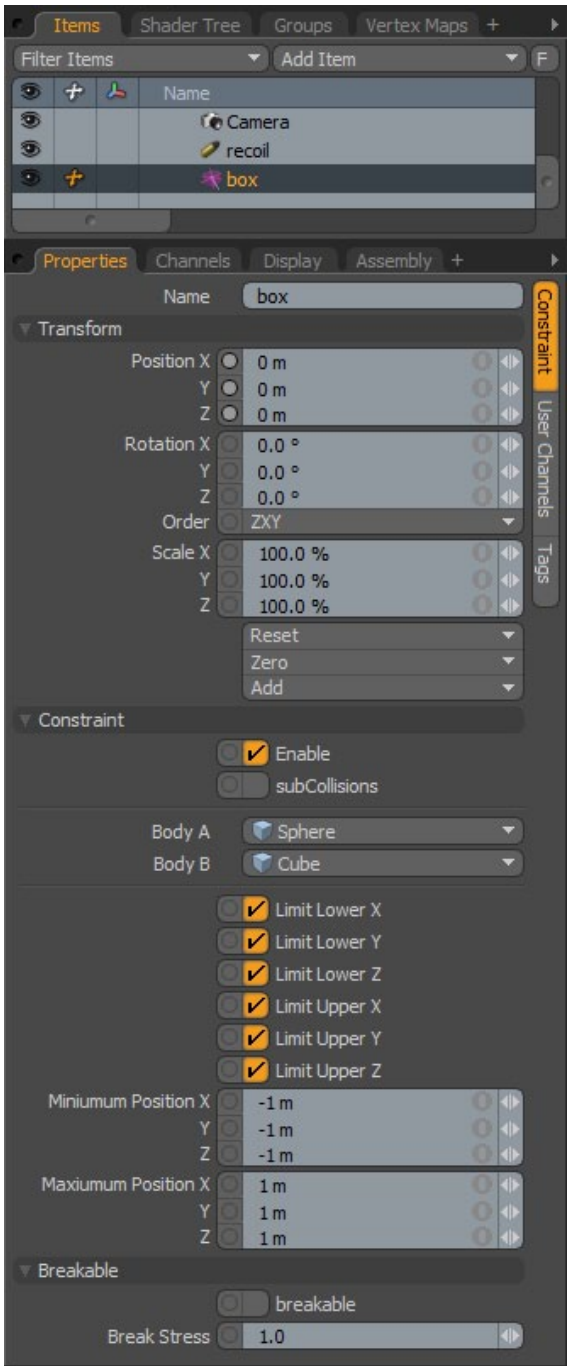
Break Stress: The 'Break Stress' value determines the threshold when the constraint will no longer affect the constrained item. The 'Breakable' option must be enabled for this value to have any affect on the constraint.

Box Constraint



The Box Constraint limits the movements of the constrained item to within the volume of a user defined cube. Not to be confused with collision detection, the Box Constraint merely restrict the movement of the constrained item to within the user defined boundary.

To apply a constraint automatically, users can select two items prior to invoking any of the Constraint types. The first item selected represents the master or parent item and will be populated as 'Body A' within the constraint, the second item selected will be the item that is constrained, also considered the auxiliary or child item and will be populated as 'Body B'. With the items selected, LMB+click on the 'Box' constraint button, adding the constraint item to the 'Items List'. The default center location (position) of the Box Constraint will be exactly half-way between the Body A and Body B objects with a line drawing to the centers of the constrained items. Depending on your intentions, this might not be the optimal position for the Constraint. The motion of the child item (Body B) will originate from the position of the Constraint item itself, so it may need to be positioned appropriately.



Name: This data field displays the current Constraint item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position X/Y/Z: The Position XYZ values represents the initial center for constraint item in 3D world space, it also represents the center of mass relative to the item it is constraining. Based on its initial position, once a dynamics simulation is invoked, the constraining item will rotate and move based on this initial (resting) position.

Rotation X/Y/Z: The Rotation XYZ value represents the initial (resting) rotation of the constraining item.

Order: The 'Order' option allows the user to set the order that rotations are applied to the item, each producing a different result. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale X/Y/Z: The Scale XYZ values have no direct affect over Constraints.

Reset: Resets the selected transform values to (0,0,0) returning the item back to its default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when keyframe animating complex motions, providing a layered approach to the applied transforms. There are no benefits to applying additional transforms to dynamic objects.

Constraint--

Enable: The 'Enable' option toggles the Constraint item on or off. When *enabled* the constraint item will be considered during a dynamics simulation, when *disabled*, the constraint item will be ignored. However, *disabled* constraints are persistent across modo sessions being saved with the scene, and retain their present settings.

SubCollisions: The 'SubCollisions' option, when *enabled*, allows constrained objects (especially those within a *chain*, not a literal chain but subsequent groupings) to collide with one another. When disabled (the default state), fewer calculation are required, however, constrained items may inter-penetrate each other. If this occurs users will want to *enable* SubCollisions.

Body A: The 'Body A' setting represents the item that is constrained to, this could also be considered the parent or master item.

Body B: The 'Body B' setting represent the item that is constrained, this could be considered the child or auxiliary item.

Limit Lower X/Y/Z: When *enabled*, sets a boundary limit for the -X, -Y and -Z sides of the constraining cube. Represented by an upright diamond on the cubes side. When disabled, the cube will be infinite in that direction.

Limit Upper X/Y/Z: When *enabled*, sets a boundary limit for the +X, +Y and +Z sides of the constraining cube. Represented by an upright diamond on the cubes side. When disabled, the cube will be infinite in that direction.

Minimum Position X/Y/Z: When the 'Limit Lower X/Y/Z' options are *enabled*, this value is the distance away from the center position of the constraint defining its edge. The constrained item will not be able to move further than this value.

Maximum Position X/Y/Z: When the 'Limit Upper X/Y/Z' options are *enabled*, this value is the distance away from the center position of the constraint defining its edge. The constrained item will not be able to move further than this value.

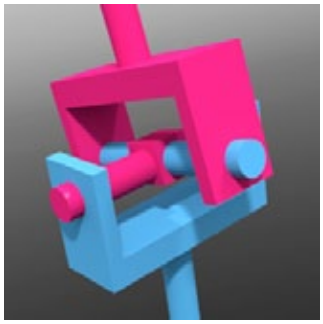
Breakable--

Breakable: The 'Breakable' option allows constraints to literally be broken apart when a certain stress threshold is reached, defined by the 'Break Stress' value. When the

threshold is reached, the constraint will no longer have any affect on the constrained item, leaving it to be affected by any other forces present in the simulation.

Break Stress: The 'Break Stress' value determines the threshold when the constraint will no longer affect the constrained item. The 'Breakable' option must be *enabled* for this value to have any affect on the constraint.

Universal Constraint



The Universal Constraint constrains the rotation of an object like that of a universal joint, a coupling that connects two rotating shafts allowing a freedom of movement for the constrained items from around the joints central point. Rotations by default are around the 'X' and 'Y' axes. This joint is often seen on a drive shaft for a vehicle. By default, this joint is represented in the viewport by two perpendicular fans representing the limited 'X' and 'Y' rotation values.

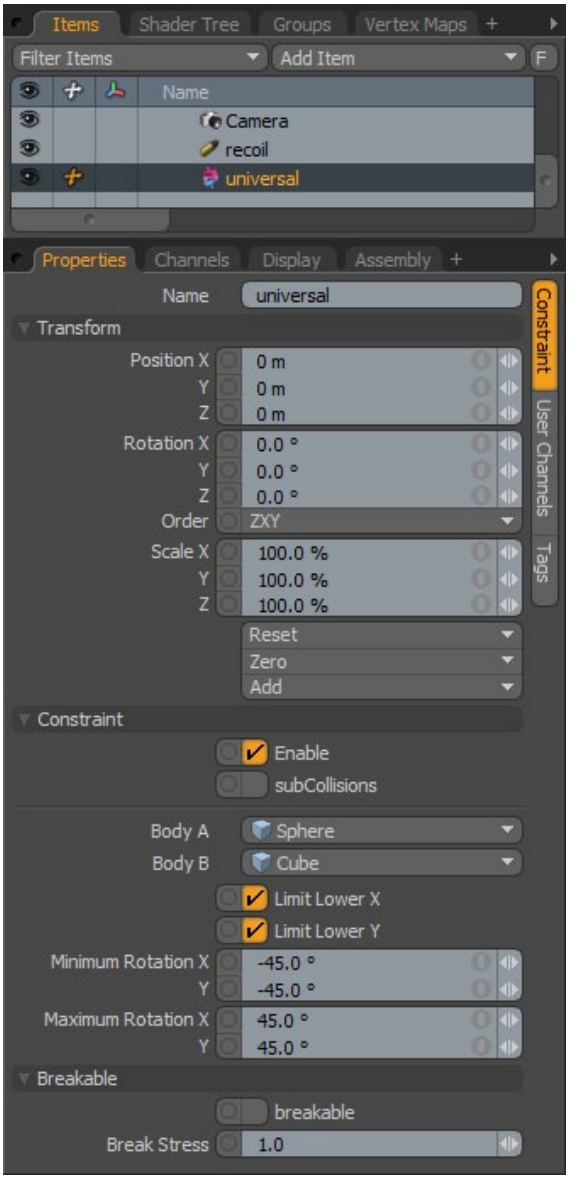
To apply a constraint automatically, users can select two items prior to invoking any of the Constraint types. The first item selected represents the master or parent item and will be populated as 'Body A' within the constraint, the second item selected will be the item that is constrained, also considered the auxiliary or child item and will be populated as 'Body B'. With the items selected, LMB+click on the 'Universal' constraint button, adding the constraint item to the 'Items List'. The default center location (position) of the Universal Constraint will be exactly half-way between the Body A and Body B objects with a line drawing to the centers of the constrained items. Depending on your intentions, this might not be the optimal position for the Constraint. The motion of the child item (Body B) will originate from the position of the Constraint item itself, so it may need to be positioned appropriately.

Name: This data field displays the current Constraint item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position X/Y/Z: The Position XYZ values represents the initial center for constraint item in 3D world space, it also represents the center of mass relative to the item it is constraining. Based on its initial position, once a dynamics simulation is invoked, the constraining item will rotate and move based on this initial (resting) position.

Rotation X/Y/Z: The Rotation XYZ value represents the initial (resting) rotation of the constraining item.



Order: The 'Order' option allows the user to set the order that rotations are applied to the item, each producing a different result. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale X/Y/Z: The Scale XYZ values have no direct affect over Constraints.

Reset: Resets the selected transform values to (0,0,0) returning the item back to its default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when keyframe animating complex motions, providing a layered approach to the applied transforms. There are no benefits to applying additional transforms to dynamic objects.

Constraint--

Enable: The 'Enable' option toggles the Constraint item on or off. When *enabled* the constraint item will

be considered during a dynamics simulation, when *disabled*, the constraint item will be ignored. However, *disabled* constraints are persistent across modo sessions being saved with the scene, and retain their present settings.

SubCollisions: The 'SubCollisions' option, when *enabled*, allows constrained objects (especially those within a chain, not a literal chain but subsequent groupings) to collide with one another. When *disabled* (the default state), fewer calculation are required, however, constrained items may inter-penetrate each other. If this occurs users will want to *enable* SubCollisions.

Body A: The 'Body A' setting represents the item that is constrained to, this could also be considered the parent or master item.

Body B: The 'Body B' setting represent the item that is constrained, this could be considered the child or auxiliary item.

Limit Lower X/Y: The Universal constraint offers rotations around a central point for two axes, the 'X' axis and the 'Y' axis. By default, these are limited to a 90° rotational range. Users can *disable* these limits individually for each axis, by toggling either of the two options.

Minimum Rotation X/Y: Sets the minimum rotational clamping value when either the 'Limit Lower X/Y' values are *enabled*.

Maximum Rotation X/Y: Sets the maximum rotational clamping value when either the 'Limit Lower X/Y' values are *enabled*.

Breakable--

Breakable: The 'Breakable' option allows constraints to literally be broken apart when a certain stress threshold is reached, defined by the 'Break Stress' value. When the threshold is reached, the constraint will no longer have any affect on the constrained item, leaving it to be affected by any other forces present in the simulation.

Break Stress: The 'Break Stress' value determines the threshold when the constraint will no longer affect the constrained item. The 'Breakable' option must be *enabled* for this value to have any affect on the constraint.

Slice Constraint

The Slide Constraint restricts the motion of an object to a single line, by default aligning itself to the X axis, allowing free rotation of the object around the rotation point.

To apply a constraint automatically, users can select two items prior to invoking any of the Constraint types. The first item selected represents the master or parent item and will be populated as 'Body A' within the constraint, the second item selected will be the item that is constrained, also considered the auxiliary or child item and will be populated as 'Body B'. With the items selected, LMB+click on the 'Slide' constraint button, adding the constraint item to the 'Items List'. The default center location (position) of the Slide Constraint will be exactly half-way between the Body A and Body B objects with a line drawing to the centers of the constrained items. Depending on your intentions, this might not be the optimal position for the Constraint. The motion of the child item (Body B) will originate from the position of the Constraint item itself, so it may need to be positioned appropriately.

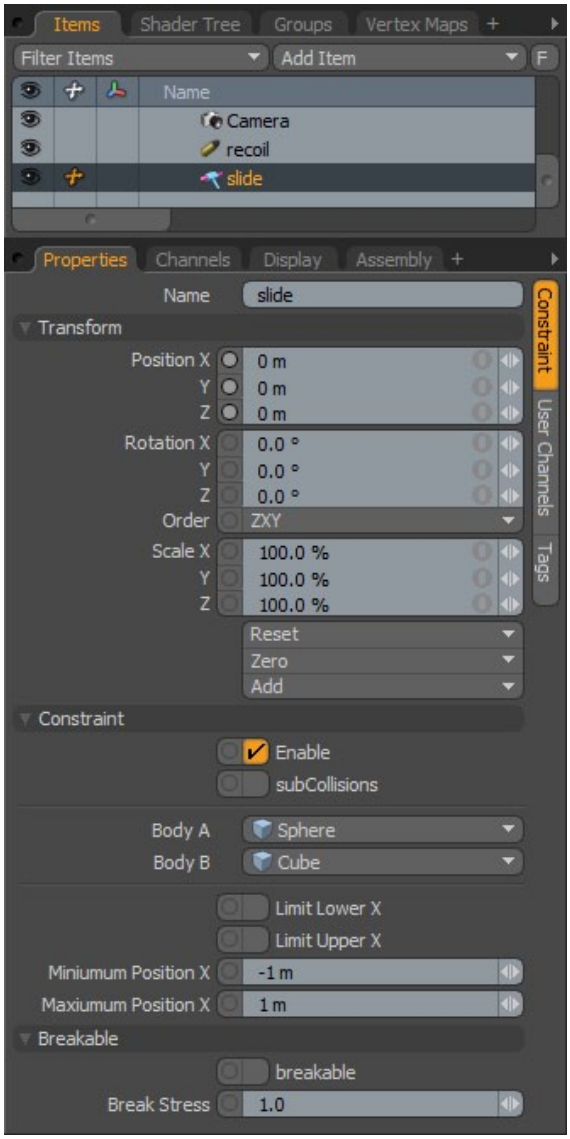
Name: This data field displays the current Constraint item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position X/Y/Z: The Position XYZ values represents the initial center for constraint item in 3D world space, it also represents the center of mass relative to the item it is constraining. Based on its initial position, once a dynamics simulation is invoked, the constraining item will rotate and move based on this initial (resting) position.

Rotation X/Y/Z: The Rotation XYZ value represents the initial (resting) rotation of the constraining item.

Order: The 'Order' option allows the user to set the order that rotations are applied to the item, each producing a different result. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.



Scale X/Y/Z: The Scale XYZ values have no direct affect over Constraints.

Reset: Resets the selected transform values to (0,0,0) returning the item back to its default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when keyframe animating complex motions, providing a layered approach to the applied transforms. There are no benefits to applying additional transforms to dynamic objects.

Constraint--

Enable: The 'Enable' option toggles the Constraint item on or off. When *enabled* the constraint item will be considered during a dynamics simulation, when *disabled*, the constraint item will be ignored. However, *disabled* constraints are persistent across modo sessions being saved with the scene, and retain their present settings.

SubCollisions: The 'SubCollisions' option, when *enabled*, allows constrained objects (especially those within a chain, not a literal *chain* but subsequent groupings) to collide with one another. When disabled (the default state), fewer calculation are required, however, constrained items may inter-penetrate each other. If this occurs users will want to *enable* SubCollisions.

Body A: The 'Body A' setting represents the item that is constrained to, this could also be considered the parent or master item.

Body B: The 'Body B' setting represent the item that is constrained, this could be considered the child or auxiliary item.

Limit Lower/Upper X: By default, the motion of the constrained item will not be limited across the length of the constraining line. Enabling the 'Upper' and/or 'Lower' limits allows users to restrict the motion to a specific range determined by the Min/Max position values.

Minimum/Maximum Position X: When the 'Limit Lower/Upper X' values are *enabled*, users can define a range as a distance value offset from the constraints center position.

Breakable--

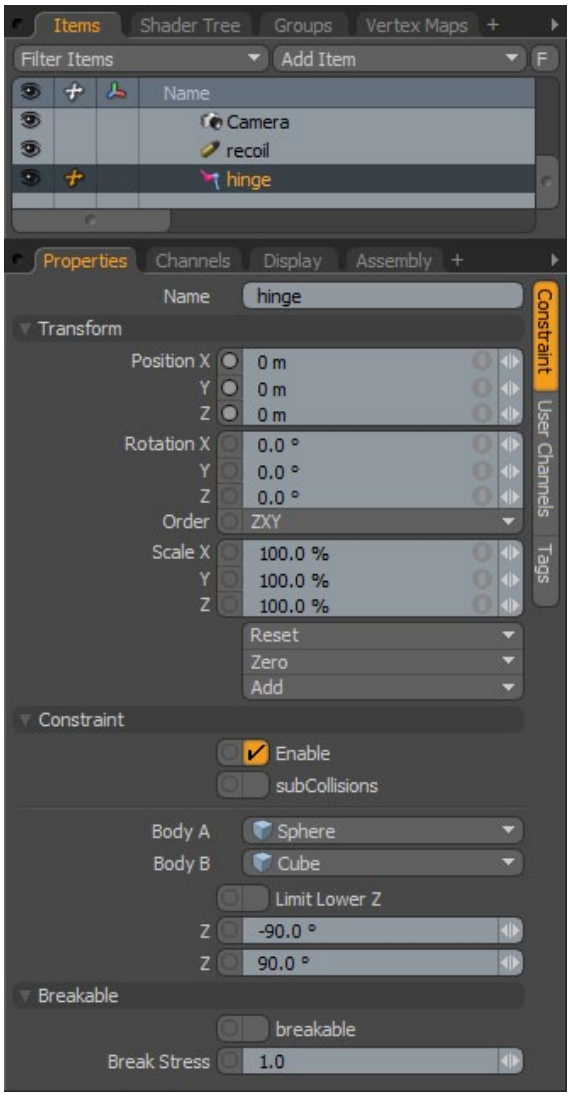
Breakable: The 'Breakable' option allows constraints to literally be broken apart when a certain stress threshold is reached, defined by the 'Break Stress' value. When the threshold is reached, the constraint will no longer have any affect on the constrained item, leaving it to be affected by any other forces present in the simulation.

Break Stress: The 'Break Stress' value determines the threshold when the constraint will no longer affect the constrained item. The 'Breakable' option must be *enabled* for this value to have any affect on the constraint.

Hinge Constraint

The Hinge Constraint limits the rotation movement of an object to swivel as if fixed by a hinge joint like that of a door. However, as a door hinge is limited by the collision of the opposing hinge plate, the hinge constraint offers unlimited rotation, much like a wheel axle. Users can add a rotation range limit to the joint to produce the more limited motion of the door hinge. The Hinge constraint is represented in the viewport as a circle around the point of rotation.

To apply a constraint automatically, users can select two items prior to invoking any of the Constraint types. The first item selected represents the master or parent item and will be populated as 'Body A' within the constraint, the second item selected will be the item that is constrained, also considered the auxiliary or child item and will be populated as 'Body B'. With the items selected, LMB+click on the 'Hinge' constraint button, adding the constraint item to the 'Items List'. The default center location (position) of the Hinge Constraint will be exactly half-way between the Body A and Body B objects with a line drawing to the centers of the constrained items. Depending on your intentions, this might not be the optimal position for the Constraint. The motion of the child item (Body B) will originate from the position of the Constraint item itself, so it may need to be positioned appropriately.



Name: This data field displays the current Constraint item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position X/Y/Z: The Position XYZ values represents the initial center for constraint item in 3D world space, it also represents the center of mass relative to the item it is constraining. Based on its initial position, once a dynamics simulation is invoked, the constraining item will rotate and move based on this initial (resting) position.

Rotation X/Y/Z: The Rotation XYZ value represents the initial (resting) rotation of the constraining item.

Order: The 'Order' option allows the user to set the order that rotations are applied to the item, each producing a different result. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale X/Y/Z: The Scale XYZ values have no direct affect over Constraints.

Reset: Resets the selected transform values to (0,0,0) returning the item back to its default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when keyframe animating complex motions, providing a layered approach to the applied transforms. There are no benefits to applying additional transforms to dynamic objects.

Constraint--

Enable: The 'Enable' option toggles the Constraint item on or off. When *enabled* the constraint item will be considered during a dynamics simulation, when *disabled*, the constraint item will be ignored. However, *disabled* constraints are persistent across modo sessions being saved with the scene, and retain their present settings.

SubCollisions: The 'SubCollisions' option, when *enabled*, allows constrained objects (especially those within a chain, not a literal *chain* but subsequent groupings) to collide with one another. When disabled (the default state), fewer calculation are required, however, constrained items may inter-penetrate each other. If this occurs users will want to *enable* SubCollisions.

Body A: The 'Body A' setting represents the item that is constrained to, this could also be considered the parent or master item.

Body B: The 'Body B' setting represent the item that is constrained, this could be considered the child or auxiliary item.

Limit Lower Z: The hinge Constraint is not initially limited in its rotation which by default is around the 'Z' axis. This way opposing objects can spin freely against each other, like a wheel or a propeller. By enabling the 'Limit Lower Z' option, users can limit the range or rotation that can be applied to the hinge constraint, mimicking the traditional range of motion of a door hinge that is allowed to only swing so far.

Z Minimum/Maximum: The first Minimum value sets the lower extreme of the of the rotation clamping, while the second Maximum value limits the upper extreme value.

Breakable--

Breakable: The 'Breakable' option allows constraints to literally be broken apart when a certain stress threshold is reached, defined by the 'Break Stress' value. When the

threshold is reached, the constraint will no longer have any affect on the constrained item, leaving it to be affected by any other forces present in the simulation.

Break Stress: The 'Break Stress' value determines the threshold when the constraint will no longer affect the constrained item. The 'Breakable' option must be *enabled* for this value to have any affect on the constraint.

Slide Hinge Constraint

The Slide Hinge Constraint combines the dynamic actions of both the 'Hinge' and the 'Slide' constraints, providing the rotation movement of the hinge with the addition of the linear back-and-forth sliding motion of the 'Slide'.

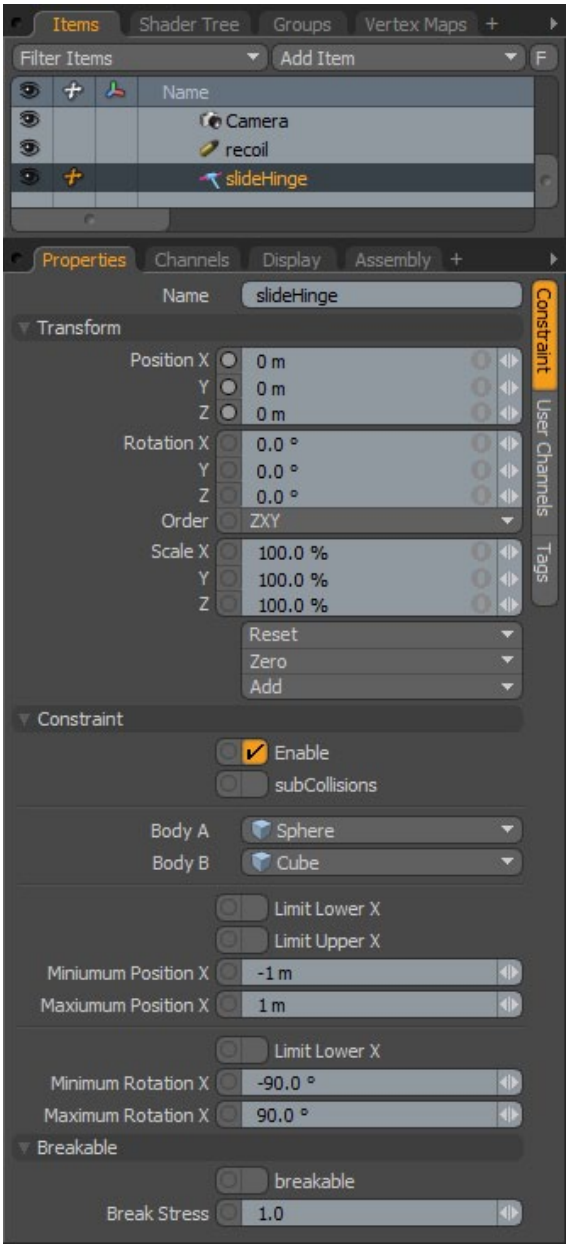
To apply a constraint automatically, users can select two items prior to invoking any of the Constraint types. The first item selected represents the master or parent item and will be populated as 'Body A' within the constraint, the second item selected will be the item that is constrained, also considered the auxiliary or child item and will be populated as 'Body B'. With the items selected, LMB+click on the 'Slide Hinge' constraint button, adding the constraint item to the 'Items List'. The default center location (position) of the Slide Hinge Constraint will be exactly half-way between the Body A and Body B objects with a line drawing to the centers of the constrained items. Depending on your intentions, this might not be the optimal position for the Constraint. The motion of the child item (Body B) will originate from the position of the Constraint item itself, so it may need to be positioned appropriately.

Name: This data field displays the current Constraint item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position X/Y/Z: The Position XYZ values represents the initial center for constraint item in 3D world space, it also represents the center of mass relative to the item it is

constraining. Based on its initial position, once a dynamics simulation is invoked, the constraining item will rotate and move based on this initial (resting) position.



Rotation X/Y/Z: The Rotation XYZ value represents the initial (resting) rotation of the constraining item.

Order: The 'Order' option allows the user to set the order that rotations are applied to the item, each producing a different result. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale X/Y/Z: The Scale XYZ values have no direct affect over Constraints.

Reset: Resets the selected transform values to (0,0,0) returning the item back to its default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when keyframe animating complex motions, providing a layered approach to the applied transforms. There are no benefits to applying additional transforms to dynamic objects.

Constraint--

Enable: The 'Enable' option toggles the Constraint item on or off. When *enabled* the constraint item will be considered during a dynamics simulation, when *disabled*, the constraint item will be ignored. However, disabled constraints are persistent across modo sessions being saved with the scene, and retain their present settings.

SubCollisions: The 'SubCollisions' option, when *enabled*, allows constrained objects (especially those within a chain, not a literal chain but subsequent groupings) to collide with one another. When *disabled* (the default state), fewer calculation are required, however, constrained items may inter-penetrate each other. If this occurs users will want to enable SubCollisions.

Body A: The 'Body A' setting represents the item that is constrained to, this could also be considered the parent or master item.

Body B: The 'Body B' setting represent the item that is constrained, this could be considered the child or auxiliary item.

Limit Lower/Upper X: By default, the motion of the constrained item will not be limited across the length of the constraining line. Enabling the 'Upper' and/or 'Lower' limits allows users to restrict the motion to a specific range determined by the Min/Max position values.

Minimum/Maximum Position X: When the 'Limit Lower/Upper X' values are *enabled*, users can define a range as a distance value offset from the constraints center position.

Limit Lower Z: The hinge part of this Constraint is not initially limited in its rotation which by default is around the 'Z' axis. By enabling the 'Limit Lower Z' option, users can limit the range or rotation that can be applied to the constraint, similar to the traditional range of motion of a door hinge that is allowed to only swing so far.

Z Minimum/Maximum: The first Minimum value sets the lower extreme of the of the rotation clamping, while the second Maximum value limits the upper extreme value.

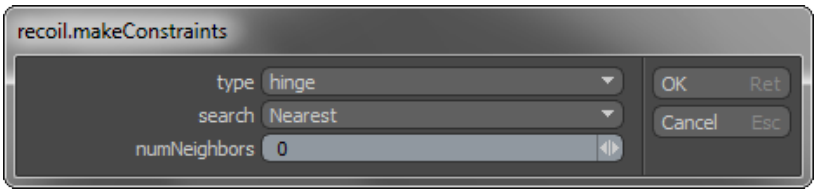
Breakable--

Breakable: The 'Breakable' option allows constraints to literally be broken apart when a certain stress threshold is reached, defined by the 'Break Stress' value. When the threshold is reached, the constraint will no longer have any affect on the constrained item, leaving it to be affected by any other forces present in the simulation.

Break Stress: The 'Break Stress' value determines the threshold when the constraint will no longer affect the constrained item. The 'Breakable' option must be *enabled* for this value to have any affect on the constraint.

Make Constraints

Users can easily apply Constraints to large numbers of objects either randomly or in-series using the 'Make Constraints' command found in the recoil toolbox or invoked from the menu bar under "recoil > Constraints > Make". Users can select the target items the constraints will apply to and then invoke the command opening the dialogue window below.



Make Constraints--

Type: This option determines the actual constraint type that will be applied to the objects. Select from the various Constraint available.

Search: This option defines how recoil searches for meshes to connect together with constraints, applies only to selected dynamic items.

Nearest- Searches from the origin (world center) outward. Constraint attachment order is determined by how close the items are to the origin.

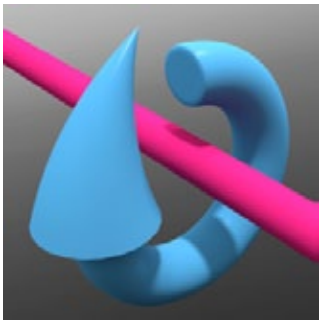
Farthest- Searches from the furthest items position inward toward the origin (world center). Constraint attachment order is determined by how far the items are from the origin.

Random- Randomly attaches constraints to bodies.

X/Y/Z- These are linear search options, dependant on the axis selected. The '+' axis values will start at the lowest position and work upward, and the '-' negative axis values will work at the highest value and work downward.

Num Neighbors: Generally two items are attached together in series with the constrain in between them, i.e., one item has one body constrained to it. The 'Num Neighbors' value provides a way to add additional constraints per item. Such as if the value were two, then each dynamic body would be constrained to two other dynamic bodies. Useful in setting up breakable objects when combined with the 'Breakable' and 'Stress' values of the Constraint items (See the particular Constraint type for more information. For shattered objects, the 'Pin Constraint' is useful in this regard.

Motor



The Motor item provides automatic motion without keyframes for dynamic objects. Like an engine, it can provide a normal forward or reverse motion using the Linear Target Velocity, or it can be used to rotate an object using the Angular Target Velocity.

To apply a Motor automatically, users can select two items prior to invoking the Motor constraint. The first item selected represents the master or parent item and will be populated as 'Body A' within the Motor item, the second item selected will be the item that receives the motion of the motor, also considered the auxiliary or child item and will be populated as 'Body B'. With the items selected, LMB+click on the 'Motor' button, adding the item to the 'Items List'. The default center location (position) of the Motor will be exactly half-way between the Body A and Body B objects with a line drawing to the centers of the constrained items. The Motors items position represents its center of rotation to the target item, so it will be likely users will need to re-position the item once it has been applied.

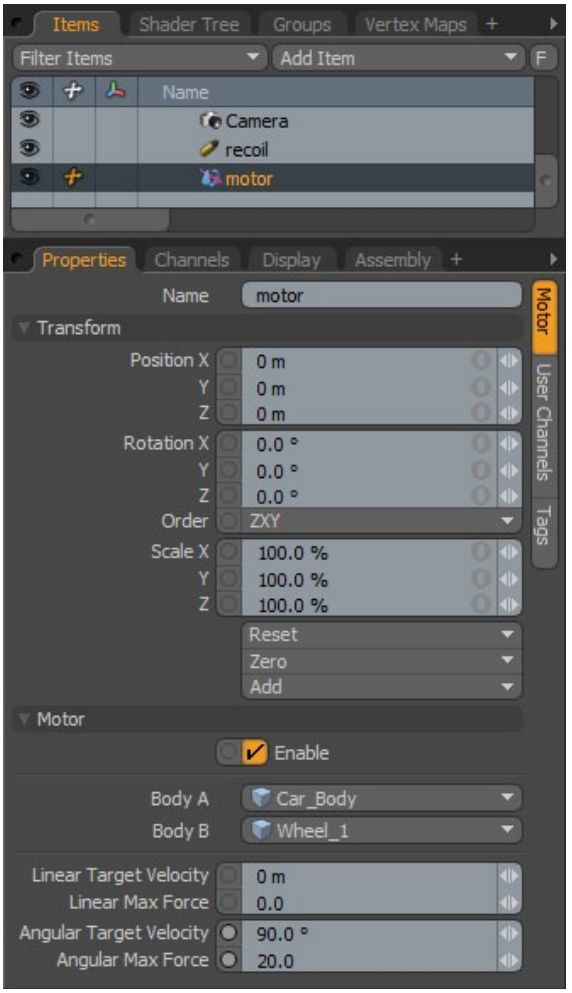
Removing Motor items can be done by simply selecting the item and pressing the 'Delete' key on the keyboard. To temporarily disable the effects of the Motor, toggle the 'Enable' option.

Name: This data field displays the current Motor items name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position X/Y/Z: The Position XYZ values represents the initial center for constraint item in 3D world space. Based on its initial position, once a dynamics simulation is invoked, the constraining item will rotate and move based on this initial (resting) position.

Rotation X/Y/Z: The Rotation XYZ value represents the initial (resting) rotation of the Motor item.



Order: The 'Order' option allows the user to set the order that rotations are applied to the item. However, since the rotations to the Motor are derived procedurally, this option is superfluous.

Scale X/Y/Z: The Scale XYZ values have no direct affect over Constraints.

Reset: Resets the selected transform values to (0,0,0) returning the item back to its default state.

Zero: Resets the chosen transform property values to '0'.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when keyframe animating complex motions, providing a layered approach to the applied transforms. However, there are no benefits to applying additional transforms to dynamic objects.

Motor--

Enable: The 'Enable' option toggles the Motor item on or off. When *enabled* the Motor will effect the target item during a dynamics simulation, when *disabled*, the Motor item will be ignored. However, *disabled* Motors are persistent across modo sessions being saved with the scene, and retain their present settings.

Body A: The 'Body A' setting represents the item that is constrained to, this could also be considered the parent or master item.

Body B: The 'Body B' setting represent the item that is constrained, this could be considered the child or auxiliary item. When a Linear or Angular force is a non-zero value, this is the object that receives the motion of the motor.

Linear Target Velocity: When this value is non-zero it applies a linear motion to the 'Body B' item, positive value move the item forward and negative values move it in

reverse. The direction of the motion is based on the resting position of the constraint. The motor will attempt to achieve the defined Target Velocity, taking into account forces, mass as well as any collisions it encounters.

Linear Max Force: The 'Linear Max Force' defines the strength of the motor. This can be thought of like horsepower in a way, the greater the force of the motion the more difficult it will be for opposing forces to affect the object. Such as if a car were moving up a steep grade and is affected by the pull of gravity. Users could increase the Maximum Force value allowing the car to move uphill more easily without changing its speed.

Angular Target Velocity: This setting is basically the same as 'Linear Target Velocity' but applies a rotating motion to the object instead.

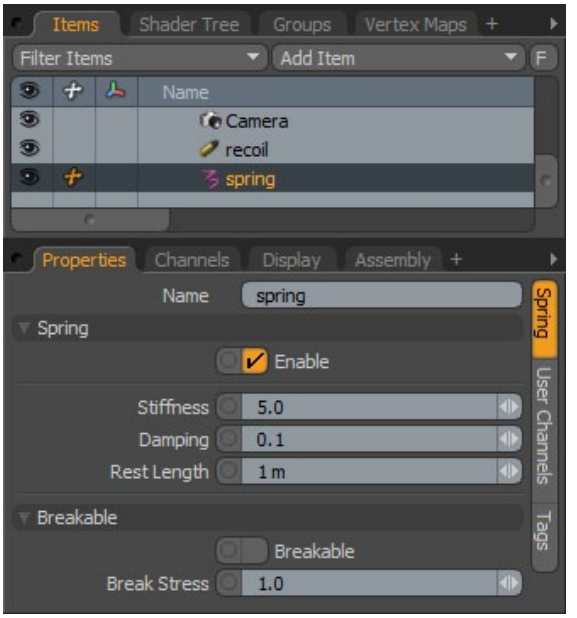
Angular Max Force: This setting is basically the same as 'Linear Max Force' but applies as a rotational force to the object instead.

Spring

The 'Spring' is a special constraint the mimics the motion of an actual coiled spring. The Spring constraint produces a motion similar to the 'Point Constraint' swinging and swaying, with the addition of the stretching and compression action of a spring. It is represented in the viewport by a coil that will extend and compress as objects move during a simulation.

To apply a Spring constraint automatically, users can select two items prior to invoking the Spring constraint option. The first item selected represents the master or parent item, typically this is the fixed or keyframe animated item. The second item selected will be the item that is constrained, also considered the auxiliary or child item and will be constrained to the motion of the Spring. With the items selected, LMB+click on the 'Spring' button in the recoil toolbox, adding the Spring item to the 'Items List'. By default, the spring will stretch between the center points of the two

selected objects. While Constraints can be added manually to a scene by way of the item list, automatically assigning them to items is the preferred method as it will also automatically define the constraints position relative to the selected items.



Name: This data field displays the current Spring items name. Users may easily change it by LMB-clicking within the field and typing the new name.

Spring--

Enable: The 'Enable' option toggles the Spring item on or off. When *enabled* the Spring will be considered during a dynamics simulation, when *disabled*, the Spring item will be ignored. However, *disabled* items are persistent across modo sessions being saved with the scene, and retain their present settings.

Stiffness: The amount of force it takes to compress the spring (shorten its length). A stiffer spring is harder to compress.

Damping: During a simulation, a spring will oscillate between extended and compressed. The 'Damping' effect reduces the amplitude of the springs oscillations.

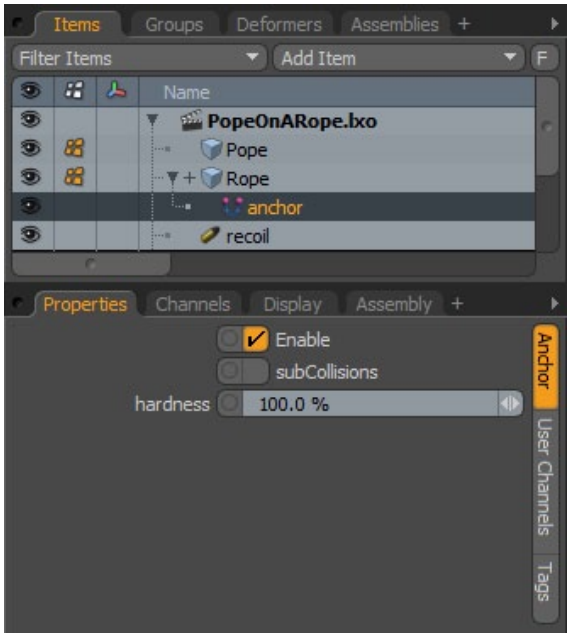
Rest Length: The 'Rest Length' represents the length of the spring when it is not in motion. This can also be considered the distance between the two constraining items when the spring reaches equilibrium.

Breakable: The 'Breakable' option allows the Spring to literally be broken apart when a certain stress threshold is reached, defined by the 'Break Stress' value. When the threshold is reached, the Spring will no longer have any affect on the constrained item, leaving it to be affected by any other forces present in the simulation.

Break Stress: The 'Break Stress' value determines the threshold when the Spring will no longer affect the constrained item. The 'Breakable' option must be *enabled* for this value to have any affect on the Spring.

Anchor

The 'Make Anchors' options is used to connect Soft Body items to Rigid Body items (or vice-versa). For simulations sake, it doesn't matter if the 'Rigid Body' item is *Active or Passive*. This could be useful to add some geometry to the end of a rope, for instance, or connect a flag to a flagpole. In order for the Anchor option to work, user must follow this specific workflow- First in 'Items' mode, users need to select the target Soft Body item AND the target Rigid Body item. With the items selected, change into the 'Vertices' mode (press '1' on the keyboard) and select at least one vertex on the Soft Body item, and at least one more on the Rigid Body (selection order isn't important). With the vertices selected, press the 'Make Anchors' option in the recoil toolbox. The two items are now rigidly connected together via an additional 'Anchor' item added to the 'Soft Body'. To break the connection, simply delete the resulting 'Anchor' item; to make adjustments, remove the Anchor and recreate it, following the above instructions.



Enable: The 'Enable' option toggles the Anchor item on or off. When *enabled* the Spring will be considered during a dynamics simulation, when *disabled*, the Spring item will be ignored. However, *disabled* items are persistent across modo sessions being saved with the scene, and retain their present settings.

SubCollisions: The 'SubCollisions' option, when *enabled*, allows constrained objects (especially those within a chain, not a literal chain but subsequent groupings) to collide with one another. When disabled (the default state), fewer calculation are required, however, constrained items may inter-penetrate each other. If this occurs users will want to *enable* SubCollisions.

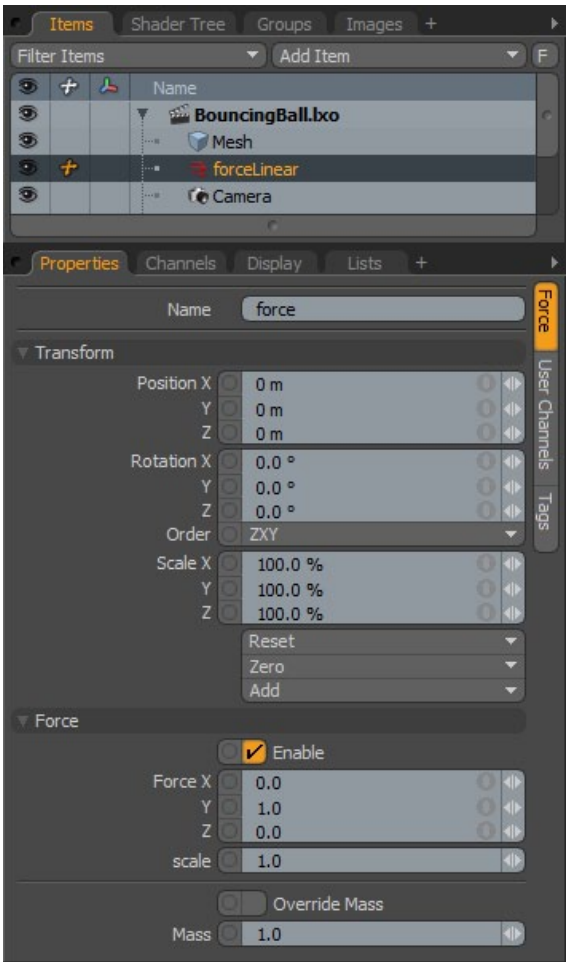
Hardness: The 'Hardness' option controls the rigidity of the Anchor between the elements. At 100%, elements will be solidly connected, At 0%, extreme forces will tend the anchor to *give and take*, but not break.

recoil Forces

In recoil, a force is any influence that causes a dynamic body to undergo a change in speed or direction. Six different types of Forces are available to affect dynamic objects, Select one of the types to see the associated documentation related to it-

Force Linear

Forces provide a way for users to further affect and control the motion of dynamic items in a scene. The 'Force Linear' items applies a constant force globally to all dynamic items in the direction specified by the 'Force X, Y, or Z value. Linear Force can be thought as being similar to wind, but without the turbulence associated with wind blowing. The higher the value, the stronger the force applied to the dynamic objects.



Users can add a 'Force Linear' item to the scene by selecting the "Forces > Linear" option in the recoil subtab of the Animation toolbox. Additionally, users may also use the menu bar command "recoil > Forces > Linear". Users may remove Force items the same way as any other item by simply selecting the item and pressing the 'Delete' key on the keyboard.

Name: This data field displays the current Force items name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position X/Y/Z: The Force Linear item globally affects dynamic items regardless of its position, therefore the Position X, Y & Z values in this case are superfluous and have no bearing on the forces affect.

Rotation X/Y/Z: Rotation Values

affect the direction of the linear force determined by the Force XYZ options allowing users to easily change the direction of the force applied to dynamic items.

Scale X/Y/Z: The Force Linear item globally affects dynamic items regardless of its scale, therefore the Scale X, Y & Z values in this case are superfluous. To adjust the strength of the Linear Force, adjust the 'Scale' value under the Force options.

Force--

Enable: The 'Enable' option toggles the Force item on or off. When *enabled* the force item will be considered during a dynamics simulation, when *disabled*, the force item will be ignored. However, *disabled* forces are persistent across modo sessions being saved with the scene, and retain their present settings.

Force X/Y/Z: This is a force *strength* value that acts as a multiplier on an objects motion in the direction specified. The higher the value the greater the strength of the force.

Scale: Overall scaler of the applied force. If the X, Y and Z values are all different, creating a specific direction, the Scale value gives a single channel option to adjust the overall strength without changing direction or adjusting multiple values.

Override Mass: When 'Override Mass' is *enabled*, recoil will use the specified 'Mass' value when calculating the force applied to items. This is useful if a user wanted all items in the scene to be effected by a linear force in the same way. Overriding the mass will cause all items to move the same amount regardless of size.

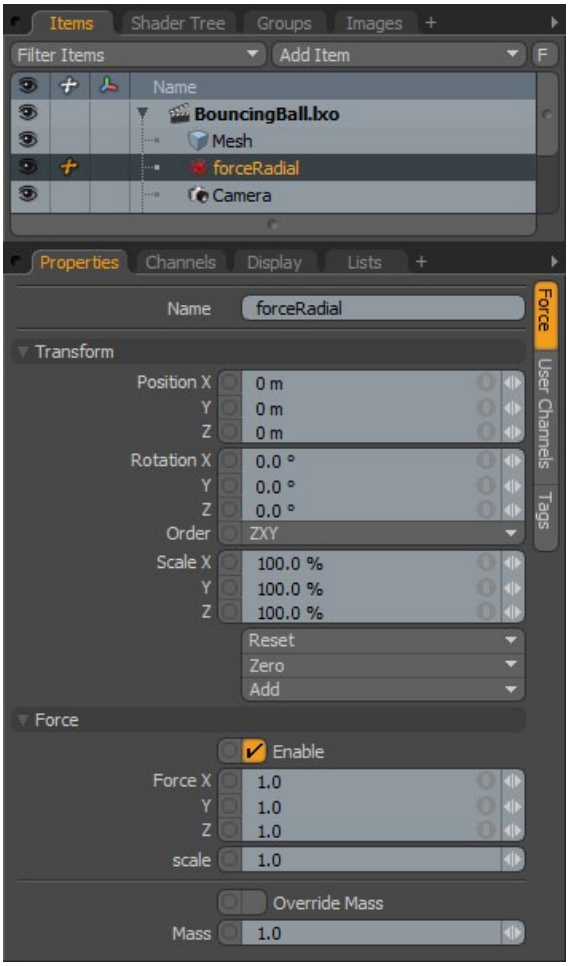
Mass: The mass value used when 'Override Mass' is *enabled*.

Force Radial

Forces provide a way for users to further affect and control dynamic items in a scene. The 'Force Radial' items applies a force that draws items toward or pushes then away from the force locator center (defined by the items center position). This force can be thought of like a magnet that repels or attracts dynamic objects. The higher the positive value, the stronger the push force applied, the greater the negative value, the stronger the attraction.

Users can add a 'Force Radial' item to the scene by selecting the "Forces > Radial" option in the recoil subtab of the Animation toolbox. Additionally, users may also use the menu bar command "recoil > Forces > Linear". Users may remove Force items the same way as any other item by simply selecting the item and pressing the 'Delete' key on the keyboard.

Name: This data field displays the current Force items name. Users may easily change it by LMB-clicking within the field and typing the new name.



Transform--

Position X/Y/Z: An Item transform that allows the user to numerically position the Force item in XYZ space, representing the point of the repulsion or attraction.

Rotation X/Y/Z: The Force Radial item effects items radially, therefore the Rotation X, Y & Z values are superfluous.

Scale X/Y/Z: The Force Radial item globally affects dynamic items regardless of its scale, therefore the Scale X, Y & Z values in this case are superfluous.

Force--

Enable: The 'Enable' option toggles the Force item on or off. When *enabled* the force item will be considered during a dynamics simulation, when *disabled*, the force item will be ignored. However, *disabled* forces are persistent across modo sessions being saved with the scene, and retain their present settings.

Force X/Y/Z: The 'Force X/Y/Z are direction multipliers on the drag forces applied to item motions on the X,Y or Z axes as designated. For example, setting a value of XYZ to 0,1,0 respectively would draw the items toward the XZ plane.

Scale: The 'Scale' function is also a multiplier, the same as 'Force X/Y/Z', but affects all axes simultaneously. This can act as a strength modifier to the amount of force applied, whether to attract or repel.

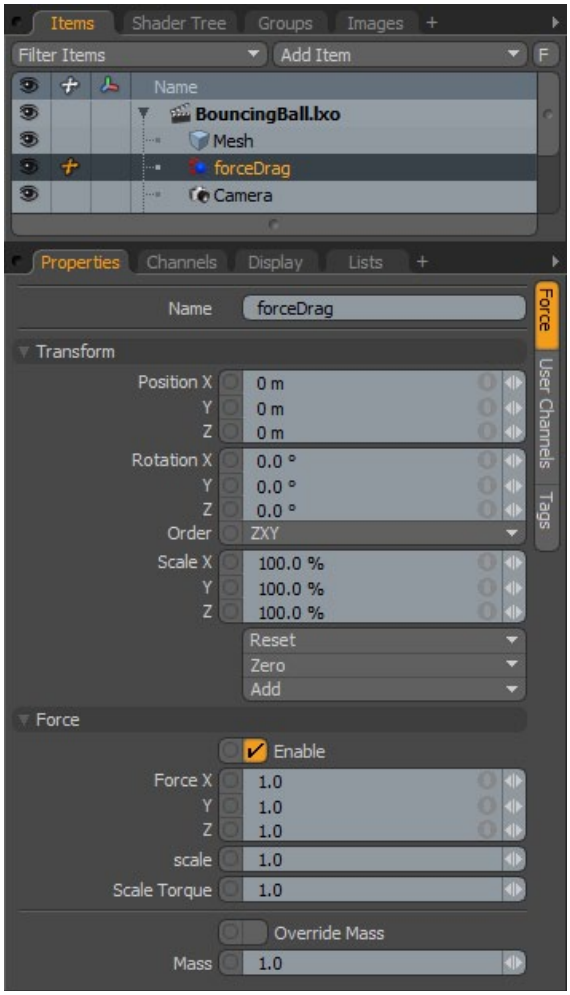
Override Mass: When 'Override Mass' is *enabled*, recoil will use the specified 'Mass' value when calculating the force applied to items. This is useful if a user wanted all

items in the scene to be effected by a linear force in the same way. Overriding the mass will cause all items to move the same amount regardless of size.

Mass: The mass value used when 'Override Mass' is *enabled*.

Force Drag

Forces provide a way for users to further affect and control dynamic items in a scene. The 'Force Drag' items applies a counter force to the objects current linear and rotational velocities. This is useful in slowing object motions, kind of like dragging it through molasses. The higher the value, the stronger the drag applied to the dynamic objects.



Users can add a 'Force Drag' item to the scene by selecting the "Forces > Drag" option in the recoil subtab of the Animation toolbox. Additionally, users may also use the menu bar command "recoil > Forces > Drag". Users may remove Force items the same way as any other item by simply selecting the item and pressing the 'Delete' key on the keyboard.

Name: This data field displays the current Force items name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position X/Y/Z: Defines the locator position for the Force Drag item. By default, the Force Drag item affects dynamic objects globally, so position of the item in the scene is irrelevant to its affect.

Rotation X/Y/Z: The Force Drag item globally affects dynamic items regardless of its rotation, therefore the Rotation X, Y & Z values in this case are superfluous.

Scale X/Y/Z: The Force Drag item globally affects dynamic items regardless of its scale, therefore the Scale X, Y & Z values in this case are superfluous. To adjust the amount of force attenuation, modify the 'Scale' option under 'Force'.

Force--

Enable: The 'Enable' option toggles the Force item on or off. When *enabled* the force item will be considered during a dynamics simulation, when *disabled*, the force item will be ignored. However, *disabled* forces are persistent across modo sessions being saved with the scene, and retain their present settings.

Force X/Y/Z: The 'Force X/Y/Z are direction multipliers on the drag forces applied to item motions on the X,Y or Z axes as designated.

Scale: The 'Scale' function is also a multiplier, the same as 'Force X/Y/Z', but affects all axes simultaneously. This can act as a strength modifier to the amount of drag.

Scale Torque: The 'Scale Torque' value can be thought of as a overall strength to the drag, the greater the value, the higher the amount of drag, but torque only affects the twisting and rotations of objects, and not linear motions.

Override Mass: When 'Override Mass' is *enabled*, recoil will use the specified 'Mass' value when calculating the force applied to items. This is useful if a user wanted all items in the scene to be effected by a linear force in the same way. Overriding the mass will cause all items to move the same amount regardless of size.

Mass: The mass value used when 'Override Mass' is *enabled*.

Force Turbulence

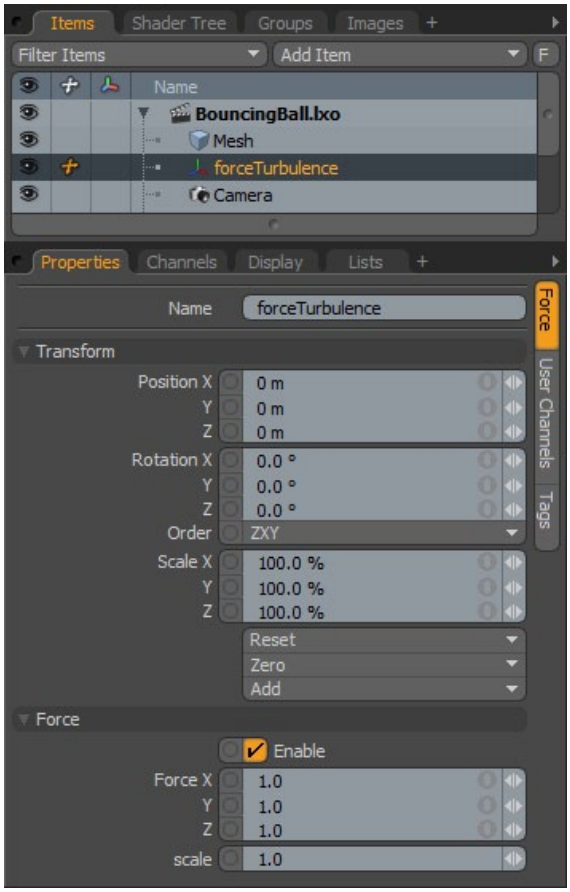
Forces provide a way for users to further affect and control dynamic items in a scene. The 'Force Turbulence' item applies a noise like function to a dynamic items movements producing an unsteady or agitated motion. Useful in adding a small or large amount of randomness to simulations, depending of the strength of the affect. The higher the value, the stronger the Turbulence applied to the dynamic objects.

Users can add a 'Force Turbulence' item to the scene by selecting the "Forces > Turbulence" option in the recoil subtab of the Animation toolbox. Additionally, users may also use the menu bar command "recoil > Forces >Turbulence". Users may remove Force items the same way as any other item by simply selecting the item and pressing the 'Delete' key on the keyboard.

Name: This data field displays the current Force items name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position X/Y/Z: The position can modify the way in which the Turbulence affects dynamic objects, as changing the positions changes the location of the turbulence field.



Rotation X/Y/Z: Determines the rotation of the Turbulence field as X,Y & Z values for each axis.

Scale X/Y/Z: Defines the scale of the turbulence field as X,Y & Z values for each axis.

Force--

Enable: The 'Enable' option toggles the Force item on or off. When *enabled* the force item will be considered during a dynamics simulation, when *disabled*, the force item will be ignored. However, *disabled* forces are persistent across modo sessions being saved with the scene, and retain their present settings.

Force X/Y/Z: The 'Force X/Y/Z are strength multipliers on the turbulence filed applied to item motions on the X,Y and/or Z axes as designated.

Scale: The 'Scale' function is also a multiplier, the same as 'Force X/Y/Z', but affects all axes simultaneously. This can act as an overall strength modifier to the amount of turbulence.

Override Mass: When 'Override Mass' is enabled, recoil will use the specified 'Mass' value when calculating the force applied to items. This is useful if a user wanted all items in the scene to be effected by a linear force in the same way. Overriding the mass will cause all items to move the same amount regardless of size.

Mass: The mass value used when 'Override Mass' is *enabled*.

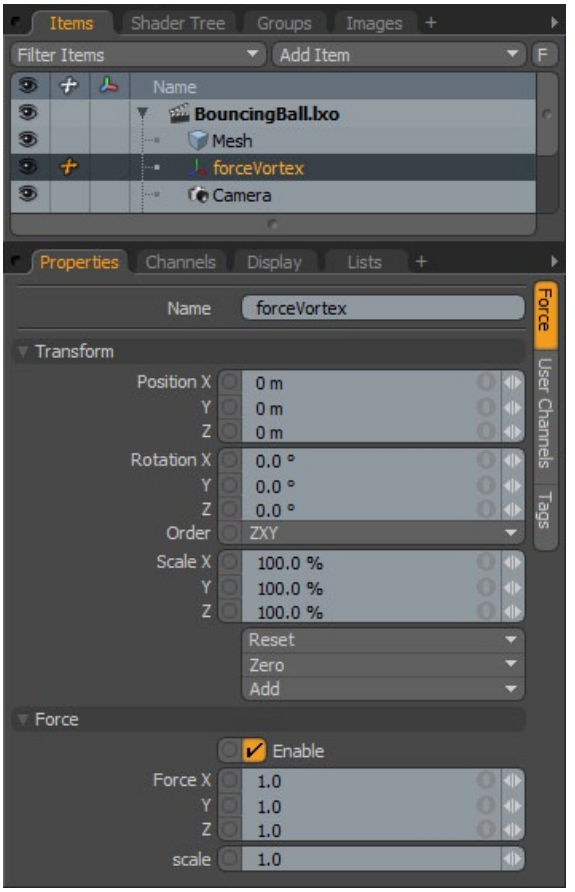
Seed: The 'Seed' attribute defines the initial value for the noise function, with every

different value producing a different random variation to the noise. If a Turbulence filed isn't producing the results you like, users can try adjusting this value to sample different randomized fields.

Octaves: The 'Octaves' value determines the number of layers for the noise function, where each additional layer adds complexity to the field, subsequently making finer details to the turbulent motion applied.

Frequency: 'Frequency' in the noise field determines the distance between the *hills* and *valleys* of the motion. Increasing the frequency values lessens the distance between these hills and valleys, producing an increasingly finer degree of undulation to the turbulence.

Amplitude: 'Amplitude' determines the strength of the Turbulence, or the height difference between the hills and valleys. Increasing the 'Amplitude' increases the overall intensity of the turbulence.



Force Vortex

Forces provide a way for users to further affect and control dynamic items in a scene. The 'Force Vortex' items applies a rotational swirling force, like a whirlpool, spinning dynamic items around the force locators center (defined by the items center position).

Users can add a 'Force Vortex' item to the scene by selecting the "Forces > Vortex" option in the recoil subtab of the Animation toolbox. Additionally, users may also use the menu bar command "recoil > Forces > Vortex". Users may remove Force items the same way as any other item by simply selecting the item and pressing the 'Delete' key on the keyboard.

Name: This data field displays the current Force items name. Users

may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position X/Y/Z: The 'Position' XY & Z values determine the *vortex line* - the center position for the vortex force, where all items affected by the vortex spin around this central point.

Rotation X/Y/Z: The rotational values can determine the angle that the vortex revolves around, defined by the X, Y & Z rotation values.

Scale X/Y/Z: The Force Vortex item globally affects dynamic items regardless of its scale, therefore the Scale X, Y & Z values in this case are superfluous. To adjust the strength of the force, modify the 'Scale' option under 'Force'.

Force--

Enable: The 'Enable' option toggles the Force item on or off. When *enabled* the force item will be considered during a dynamics simulation, when *disabled*, the force item will be ignored. However, *disabled* forces are persistent across modo sessions being saved with the scene, and retain their present settings.

Force X/Y/Z: The 'Force X/Y/Z are direction multipliers on the vortex force applied to item motions on the X,Y or Z axes as designated. Identical values for all 3 axes produce a smooth rotation, where different values will affect the directional strength, in effect scaling the shape of the vortex to an oval or oblong shape.

Scale: The 'Scale' function is also a multiplier, the same as 'Force X/Y/Z', but affects all axes simultaneously. This can act as a strength modifier to the amount of drag.

Force Newton

Forces provide a way for users to further affect and control dynamic items in a scene. The 'Force Newton' follows Newton's law of universal gravitation in that all objects are attracted to all other objects, where objects with greater mass will produce a greater amount of attraction than objects will smaller mass.

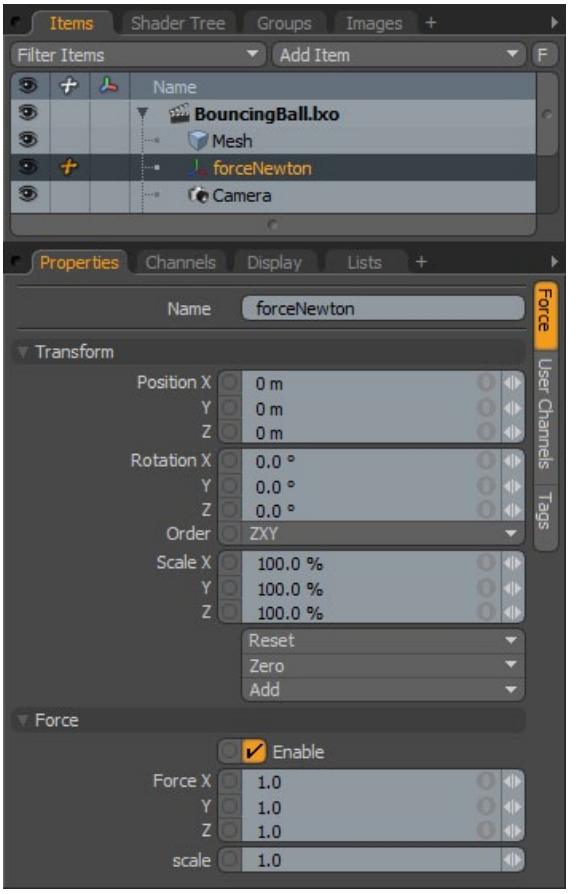
Users can add a 'Force Newton' item to the scene by selecting the "Forces > Newton" option in the recoil subtab of the Animation toolbox. Additionally, users may also use the menu bar command "recoil > Forces > Newton". Users may remove Force items the

same way as any other item by simply selecting the item and pressing the 'Delete' key on the keyboard.

Name: This data field displays the current Force items name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position X/Y/Z: The Force Newton item globally affects dynamic items regardless of its position, therefore the Position X, Y & Z values in this case are superfluous and have no bearing on the forces affect.



Rotation X/Y/Z: The Force Newton item globally affects dynamic items regardless of its rotation, therefore the Rotation X, Y & Z values in this case are superfluous and have no bearing on the forces affect.

Scale X/Y/Z: The Force Linear item globally affects dynamic items regardless of its scale, therefore the Scale X, Y & Z values in this case are superfluous. To adjust the strength of the Newton Force, adjust the 'Scale' value under the Force options.

Force--

Enable: The 'Enable' option toggles the Force item on or off. When *enabled* the force item will be considered during a dynamics simulation, when *disabled*, the force item will be ignored. However, *disabled* forces are persistent across modo sessions being saved with the scene, and retain their present settings.

Force X/Y/Z: This is a force *strength* value that acts as a multiplier on an objects motion in the direction specified. The higher the value the greater the strength of the force.

Scale: Overall scaler of the applied force. If the X, Y and Z values are all different, creating a specific direction, the Scale value gives a single channel option to adjust the overall strength without changing direction or adjusting multiple values.

recoil Falloffs

The Falloff items provides an attenuated area of influence limiting the application of attached force(s) to within the bounds of the Falloff. Essentially a falloff defines an area of influence for recoil forces, fading their influence across the Min and Max values. Users can also apply the negative of this, by *enabling* the 'Shape Exclude' option inverting the falloffs influence. The falloff shapes available are Spherical, Box, Cylinder and Plane.

Simply adding a Falloff to a scene will not produce any results. In order for a falloff to work, it must be *attached* to the Force it is attenuating. This can be one force, or several. Users can automatically attach the Falloff item to the Force by selecting the target Force item in the 3D viewport or 'Items' list prior to adding the Falloff using the toolbox button (Simply adding the items with any 'Add Item' command will not make the connection). Otherwise, once a Falloff is defined, users may select the target Force item, and then the associated Falloff and use the 'Attach' command also found in the toolbox (LMB+click on the 'Attach' button). This may also be used to attach the Falloff to multiple Force items simultaneously. Removal of the Falloff is done in a similar way by selecting the two target items and applying the 'Detach' command found in the recoil toolbox.

Falloff Spherical

Name: This data field displays the current Falloff items name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

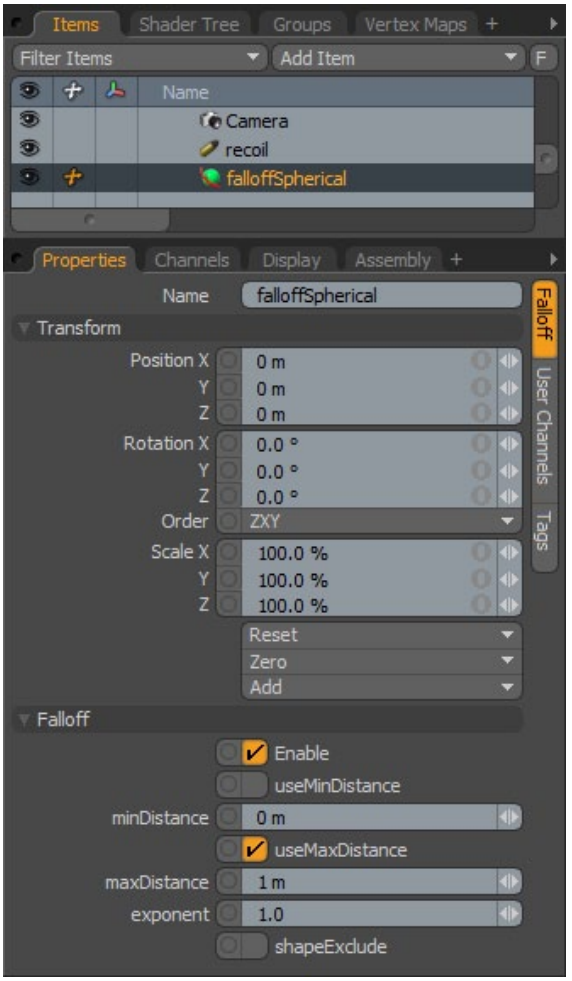
Position X/Y/Z: The Position XYZ values represents the initial center for the falloff item in 3D world space.

Rotation X/Y/Z: The Falloff Spherical item effects items radially, therefore the Rotation X, Y & Z values are superfluous.

Order: The 'Order' option allows the user to set the order that rotations are applied to the item, each producing a different result. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale X/Y/Z: The Scale XYZ values size the Falloffs sphere of influence, proportionally scaling the Min and Max distance values.

Reset: Resets the selected transform values to (0,0,0) returning the item back to its default state.



Zero: Resets the chosen transform property values to '0', leaving the items current position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when keyframe animating complex motions, providing a layered approach to the applied transforms.

Falloff--

Enable: The 'Enable' option toggles the Falloff item on or off. When enabled the Falloff item will be considered during a dynamics simulation, when disabled, the Falloff item will be ignored. However, disabled Falloffs are persistent across modo sessions being saved with the scene, and retain their present settings.

Use Minimum Distance: Setting a minimum distance means that that any item closer to the center of the falloff than the specified amount won't be effected by the Forces under the Falloffs influence.

Minimum Distance: Defines the minimum spherical distance area as a radius from the Falloffs center point when 'Use Min Distance' is enabled.

Use Maximum Distance: Setting a Maximum distance defines the size of the falloffs outer area of influence. This is essentially the size of the falloff. Items further away from the falloffs center than the Maximum distance setting will not be affected by it.

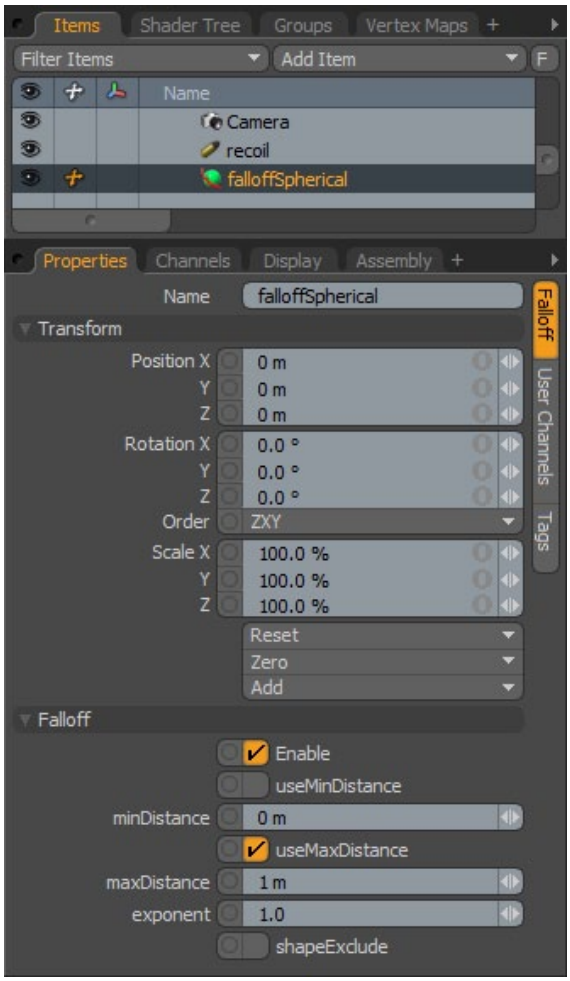
Maximum Distance: Defines the maximum spherical distance area as a radius from the Falloffs center point when 'Use Max Distance' is enabled.

Exponent: This raises the falloff amount to the specified value. This is used to achieve a smooth falloff between the min and max values.

Shape Exclude: Inverts the affects of the Falloff so that only items outside the specified falloff are affected.

Falloff Box

Name: This data field displays the current Falloff items name. Users may easily change it by LMB-clicking within the field and typing the new name.



Transform--

Position X/Y/Z: The Position XYZ values represents the initial center for the falloff item in 3D world space.

Rotation X/Y/Z: The Falloff Spherical item effects items radially, therefore the Rotation X, Y & Z values are superfluous.

Order: The 'Order' option allows the user to set the order that rotations are applied to the item, each producing a different result. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale X/Y/Z: The Scale XYZ values size the Falloffs sphere of influence, proportionally scaling the Min and Max distance values.

Reset: Resets the selected transform values to (0,0,0) returning the item back to its default state.

Zero: Resets the chosen transform property values to '0', leaving the items current position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list.

Useful when keyframe animating complex motions, providing a layered approach to the applied transforms.

Falloff--

Enable: The 'Enable' option toggles the Falloff item on or off. When enabled the Falloff item will be considered during a dynamics simulation, when disabled, the Falloff item will be ignored. However, disabled Falloffs are persistent across modo sessions being saved with the scene, and retain their present settings.

Use Minimum Distance: Setting a minimum distance means that that any item closer to the center of the falloff than the specified amount won't be effected by the Forces under the Falloffs influence.

Minimum Distance: Defines the minimum spherical distance area as a radius from the Falloffs center point when 'Use Min Distance' is enabled.

Use Maximum Distance: Setting a Maximum distance defines the size of the falloffs outer area of influence. This is essentially the size of the falloff. Items further away from the falloffs center than the Maximum distance setting will not be affected by it.

Maximum Distance: Defines the maximum spherical distance area as a radius from the Falloffs center point when 'Use Max Distance' is *enabled*.

Exponent: This raises the falloff amount to the specified value. This is used to achieve a smooth falloff between the min and max values.

Shape Exclude: Inverts the affects of the Falloff so that only items outside the specified falloff are affected.

Falloff Cylinder

Name: This data field displays the current Falloff items name. Users may easily change it by LMB-clicking within the field and typing the new name.

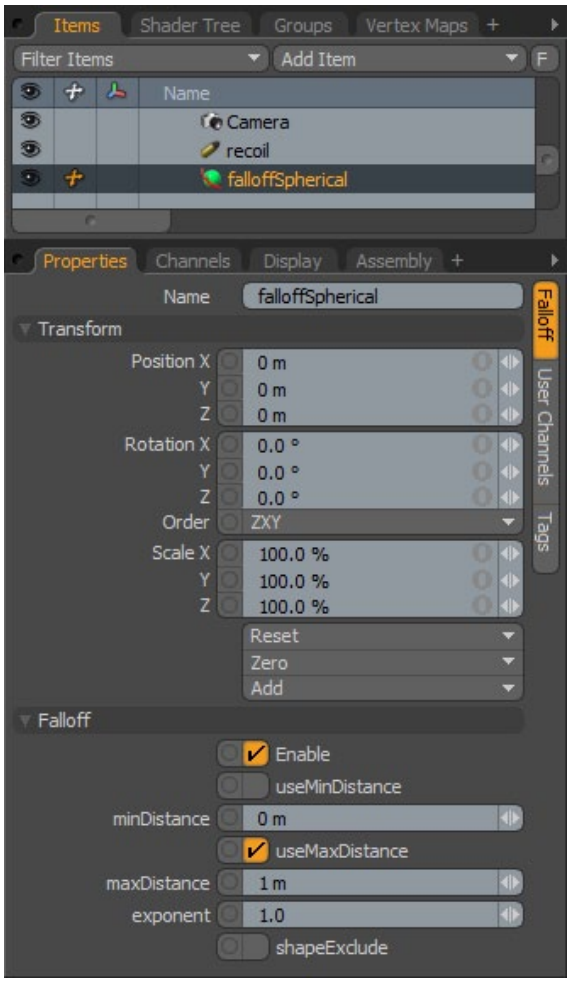
Transform--

Position X/Y/Z: The Position XYZ values represents the initial center for the falloff item in 3D world space.

Rotation X/Y/Z: The Falloff Spherical item effects items radially, therefore the Rotation X, Y & Z values are superfluous.

Order: The 'Order' option allows the user to set the order that rotations are applied to the item, each producing a different result. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale X/Y/Z: The Scale XYZ values size the Falloffs sphere of influence, proportionally scaling the Min and Max distance values.



Reset: Resets the selected transform values to (0,0,0) returning the item back to its default state.

Zero: Resets the chosen transform property values to '0', leaving the items current position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when keyframe animating complex motions, providing a layered approach to the applied transforms.

Falloff--

Enable: The 'Enable' option toggles the Falloff item on or off. When enabled the Falloff item will be considered during a dynamics simulation, when disabled, the Falloff item will be ignored. However, disabled Falloffs are persistent across modo sessions being saved with the scene, and retain their present settings.

Use Minimum Distance: Setting a minimum distance means that that any item closer to the center of the falloff than the specified amount won't be effected by the Forces under the Falloffs influence.

Minimum Distance: Defines the minimum spherical distance area as a radius from the Falloffs center point when 'Use Min Distance' is enabled.

Use Maximum Distance: Setting a Maximum distance defines the size of the falloffs outer area of influence. This is essentially the size of the falloff. Items further away from the falloffs center than the Maximum distance setting will not be affected by it.

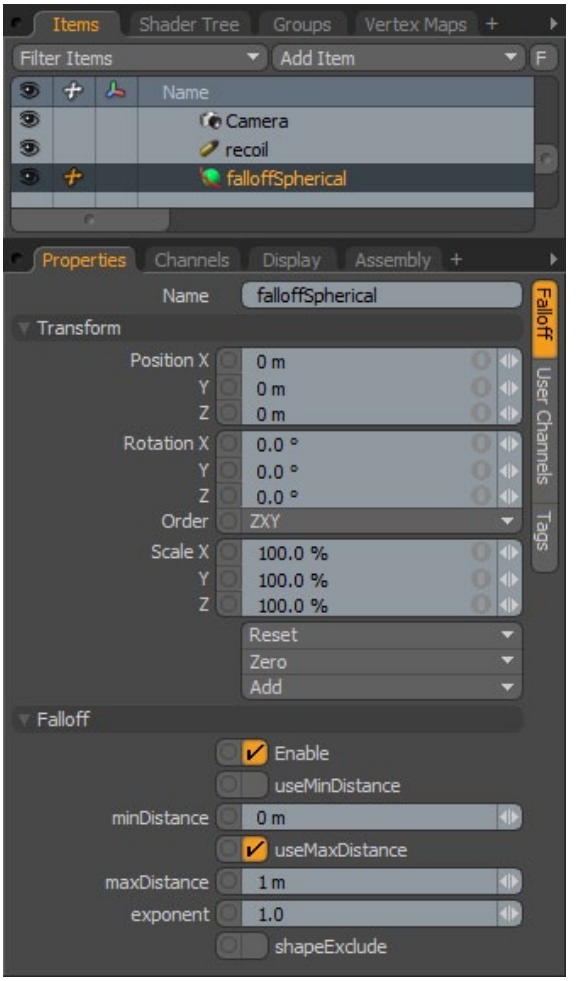
Maximum Distance: Defines the maximum spherical distance area as a radius from the Falloffs center point when 'Use Max Distance' is enabled.

Exponent: This raises the falloff amount to the specified value. This is used to achieve a smooth falloff between the min and max values.

Shape Exclude: Inverts the affects of the Falloff so that only items outside the specified falloff are affected.

Falloff Plane

Name: This data field displays the current Falloff items name. Users may easily change it by LMB-clicking within the field and typing the new name.



Transform--

Position X/Y/Z: The Position XYZ values represents the initial center for the falloff item in 3D world space.

Rotation X/Y/Z: The Falloff Spherical item effects items radially, therefore the Rotation X, Y & Z values are superfluous.

Order: The 'Order' option allows the user to set the order that rotations are applied to the item, each producing a different result. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale X/Y/Z: The Scale XYZ values size the Falloffs sphere of influence, proportionally scaling the Min and Max distance values.

Reset: Resets the selected transform values to (0,0,0) returning the item back to its default state.

Zero: Resets the chosen transform property values to '0', leaving the items current position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list.

Useful when keyframe animating complex motions, providing a layered approach to the applied transforms.

Falloff--

Enable: The 'Enable' option toggles the Falloff item on or off. When *enabled* the Falloff item will be considered during a dynamics simulation, when *disabled*, the Falloff item will be ignored. However, disabled Falloffs are persistent across modo sessions being saved with the scene, and retain their present settings.

Use Minimum Distance: Setting a minimum distance means that that any item closer to the center of the falloff than the specified amount won't be effected by the Forces under the Falloffs influence.

Minimum Distance: Defines the minimum spherical distance area as a radius from the Falloffs center point when 'Use Min Distance' is *enabled*.

Use Maximum Distance: Setting a Maximum distance defines the size of the falloffs outer area of influence. This is essentially the size of the falloff. Items further away from the falloffs center than the Maximum distance setting will not be affected by it.

Maximum Distance: Defines the maximum spherical distance area as a radius from the Falloffs center point when 'Use Max Distance' is enabled.

Exponent: This raises the falloff amount to the specified value. This is used to achieve a smooth falloff between the min and max values.

Shape Exclude: Inverts the affects of the Falloff so that only items outside the specified falloff are affected.

Baking Dynamics for Rendering

Once a recoil dynamics simulation has been refined and perfected, it is ready for rendering. You may have noticed however that each time modo is closed and then reopened, any scenes containing recoil dynamics need to be *simulated* before the dynamic motions show up in the 3D viewport. This is because the actual motion definitions aren't part of the scene, the simulation only exists in the recoil cache which is lost each time a user closes the scene (or clears it manually with the 'Clear Cache' command). There is no way to save or export the contents of the recoil cache, however users can freeze the recoil motions to the scene by 'Baking' the recoil simulation results. What this does is convert the motions calculated by recoil into individual keyframes. Once applied, recoil scenes can be rendered normally using the standard modo rendering commands.

The main benefit to freezing the simulations are that the motions of the dynamic object will now be part of the scene, no longer requiring recoil computations to be visible. Scenes will play back faster as well, as no dynamics are being calculated, especially beneficial for complex simulations. Freezing simulations also makes it possible to render scenes over a network or by command line; with the cache values being a local entity, network slaves would have no knowledge of the recoil motion and would therefore render incorrect results. It also makes it possible for users to exchange recoil scenes with other modo users who do not own recoil. Lastly, due to tiny deviations in recoils physics calculations, slight variations in object motions are possible across subsequent simulations; baking eliminates these variations producing precise repeatable results. User should be aware that for complex simulations containing many objects, the large number of keyframes generated can adversely affect file size as well as viewport performance. Users should also note that is is not currently possible to bake Deformations, such as those created by Soft Bodies.

INFO: Baking a recoil simulation disables the dynamic information from items, therefore it is recommended users save a separate version of their scene with dynamics settings enabled prior to baking. This will make it much easier than trying to remove the keyframes and re-enabling the dynamics should it be necessary to further modify the simulation down the road.

Bake recoil Cache--

There are two commands related to the baking of recoil simulations. Both are found within the menu bar, under the recoil menu as "Recoil > Cache > Bake" and "Recoil > Cache > Bake to Transform". For users convenience they are also found in the Properties viewport for dynamic objects under the 'Cache' section.

Cache > Bake

The "Cache > Bake" command will convert the dynamic motion into individual item keyframes at each frame for the Position and Rotation channels of the currently selected item(s), recreating every nuance of the recoil simulation result. In order for the command to work, there must be motion information in the cache for the target item. Users must run a recoil simulation prior to baking. To do so, users should run one of the 'Play' or 'Run' simulation buttons. Once it is finished simulating the dynamics, select the target items you wish to convert, and invoke the 'Bake' command.

Cache > Bake to Transform

The "Cache > Bake to Transform" command will convert the current position and rotational result into the objects transform properties. Essentially freezing the items current position and rotation but eliminating any motion associated with the recoil

simulation as no animation keyframes are generated. This is extremely useful when using recoil as a stacking mechanism for still renders, such as filling a gumball machine with gumballs. In order for the command to work, there must be motion information in the cache for the target item(s). Users must run a recoil simulation prior to baking to transforms. To do so, run one of the 'Play' or 'Run' simulation. Once it is finished simulating the dynamics, users should navigate to the appropriate frame in the timeline where they wish to freeze the items positions, select the target items to convert, and invoke the 'Bake to Transforms' command.

INFO: Dynamic items must be selected when invoking the 'Bake' commands, otherwise there will be no results.

Clear Recoil Cache

Recoil also offers an option to clear the simulation cache results for individual items. To use, simply select the target items you wish to remove the simulation results for and invoke the command in the menu bar under "recoil > Cache > Clear".

Wakeup Triggers

Of the various 'WakeOn' options available to wake up a sleeping dynamic body (essentially changing the objects state from *passive* to *active*) the 'Trigger' option provides users the ability to keyframe the exact time when they wish to wake up the object. When any 'Active Dynamic Bodies' are set as such, the 'Wakeup Triggers' scripts offer users some interesting means to automatically apply those keyframes to large groups of objects. This alone can create some interesting and amazingly complex effects very quickly and easily, especially when combined with an 'Impulse' value, providing some initial velocity or rotation to on object the moment it wakes up.

The commands are found in the recoil toolbox under the 'Triggers' heading, and within the menu bar under "recoil > Triggers > Wakeup Triggers". To apply the triggers, user must first define the target item(s) as an 'Active Dynamic Body' and in the 'Dynamic' properties panel, set the 'Wake On' attribute under the 'Sleep' heading to 'Trigger'. Next, simply select the target items and then invoke one of the trigger commands. The commands are as follows-

Randomize

This will apply a single keyframe to each selected dynamic body at a random interval of time across the length of the scenes total frame range. Generally, the keyframes for all

items will be spread across the entire spectrum, but this is dependant on the number of items selected, and the length of time the random function is applied to.

Randomize (Selection Order)

This provides a similar functionality to the 'Randomize' trigger, but orders the trigger keyframes in the same as the selection order, providing users control over how the triggers are applied the dynamic objects. Trigger keyframes are also evenly spaced across the length of the scenes total frame range.

Remove Trigger

Removes the trigger keyframe for the selected item, or when nothing is selected, will remove the trigger keyframes for all dynamic items.

Deformers

Applying Deformers

modo 601 introduces a number of powerful Deformers that allow users to twist, bend and otherwise distort their creations in many creative ways, providing simplicity and power in an easy to apply and use workflow. Of course each Deformer is *fully animateable* so deformations can happen over time (can you say Character Animation!!). These flexible Deformers can also be combined and stacked in the Deformers viewport, and can be controlled using the Deformer Falloffs. With so many options, it might feel overwhelming as to where to start with Deformers, but they are easy enough to understand once you grasp the basic philosophy.

Deformers in modo are made up of two basic parts, an 'Effector' and an 'Influence' (collectively called 'Deformers'). Effectors provide instructions on how to affect the target, while the Influence determines *what* is being affected. The reason for these two parts is to provide flexibility to users. Consider this scenario - a cartoony car needs to bend as it goes around a tight corner, the car is made up of multiple parts, including its four wheels, a Bend effector is applied to the body to distort it as it goes through the curve, but the wheels when included in the deformation produced undesirable results. Wanting to keep the wheels connected to the car as it bend, an additional Influence could be added and then connected to the exact same Bend Deformer, allowing it to change the items positions and rotation, but not actually deform the geometry itself, keeping the wheels in proper position relative to the bending car, all using a single simple to control Effector. Additional Deformers could then be attached to the wheels to further fine tune their local deformations all while under the influence of the prior Effector. Each Deformation is layered upon the previous producing and easy to understand workflow that provide and incredible amount of depth with

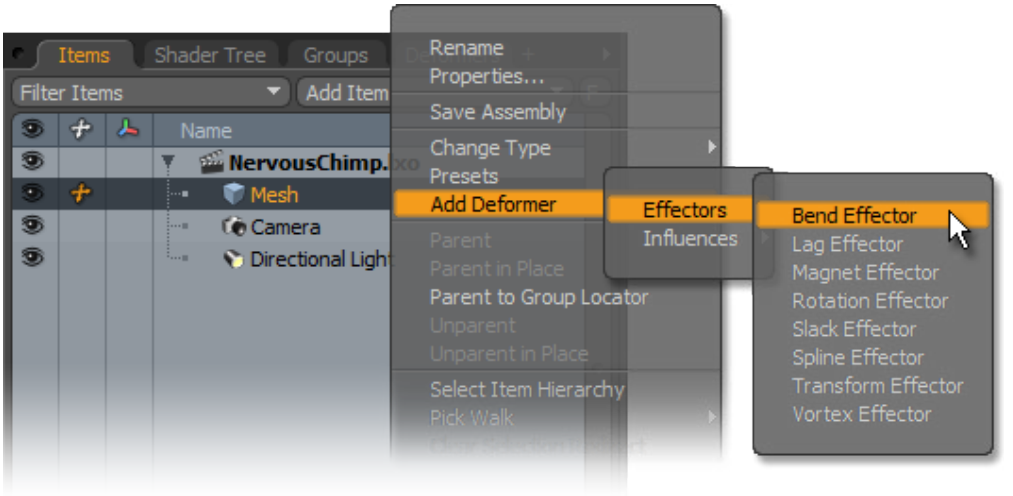
modo's acclaimed ease of use. That is merely one example, but should give users an introduction to the power of dealing with Deformers in this way.

modo offers a variety of options for working with Deformers, users just starting out with Deformers will want to follow the steps outlined below in how to apply them for specific situations. Deformers in modo are made up of multiple pieces that need to be connected together (such as through the Schematic view). By selecting a target mesh and applying the outlined commands, modo will automatically make these connections for you. More advanced users will likely add the 'Effectors' and 'Influences' separately and connect them manually in the Schematic view. When combined with Modifiers, modo provides a very deep and powerful rigging toolbox that allows the whole to be much greater than the sum of its parts.

Applying Deformers

modo offers some procedural deformers that work in a way like its modeling tool counterparts (Bend, Magnet and Vortex) that can be animated over time. For example if one wanted to animate a car antenna whipping in the wind, this could easily be achieved using a the Bend Deformer. (Combined with a Modifier modulating bend amount, a simply animation could be applied with very few keyframes.) There are also *reactive* Deformers, which are dependant on the objects keyframed motions, such as the 'Slack' Deformer or 'Lag' Deformer which produce their deformations based on the positions or motions of an item, providing interesting secondary motions to the animation. Deformers, such as the 'Rotation' and 'Transform' Effector provide users a riggable way to connect items in a controlled way.

Applying these Deformers can be done by simply selecting the appropriate target item in the Items list and then RMB+clicking on the layer name, opening a contextual menu and selecting the desired Deformer "Add Deformer > Effectors > Effector Type". They can also be assigned by selecting the item and then pressing the associated Deformer type in the 'Deformers' subtab of the 'Animation' and 'Setup' toolboxes (located within their respective interface tab). What this does is assigns an appropriate 'Effector' and a 'General Influence'. The 'Effector' determines how to deform the target, while the 'Influence' determines what to affect (by default the 'Entire Mesh'). Once the Deformer has been applied, users can select the Effector in the Items list and modify its attributes in the 'Properties' panel, deforming the target to their wishes.



For the procedural Deformers, the position of the Effector defines the origin of the of its influence over the target, so animating its position will change how it affects what its deforming. By selecting the Effector, its editable attributes are found in the Properties panel. By selecting the Influence, users can adjust what parts of the mesh are influenced.

Viewing Deformers in 3D viewports

Since the modeling and animation workflow is significantly different, there are some visibility options on the 3D viewports that are set appropriately for each discipline. When modeling, you most likely want to see your mesh in its relaxed, un-deformed state. Therefore in the Model, Model Quad, Paint and UV tabs the option to Enable Deformers is turned off. If you happen to be in one of these tabs when you add a deformer you will not see any effect. The Animate and Render tabs have this option on by default. If you wish to change the state in either case, you can simple press the '0' key with your mouse over a 3D viewport. At the bottom of the Viewport Properties you will find the toggle control for this option.

Deformer List

Deforming any Item in modo is applied through a variety of means, mainly using 'Effectors' that provide instructions on how to deform a target, and Influences that determines what it is to be deformed. Deformers are applied to items in the usual way, by selecting the target Item and the LMB+clicking on the desired deformer in the toolbox, automatically linking the elements together. Advanced users may prefer to manually add elements in the Deformers vieweport and connect them through the Schematic view

Deformers

Bend Effector

Magnet Effector

Slack Effector

Vortex Effector

Lag Effector

Influences

Gen. Influence

MDD Influence

Morph Influence

Softbody Influence

Item Influence

Bend Effector

Effectors are a means to affect a target item, specifically a Mesh layer, deforming it in specific ways. The 'Bend Effector' when assigned to an item will deform the object in a curving/arcing fashion. As the 'Angle' value is increased, the more pronounced the bending will be. The 'Bend Effectors' position relative to the mesh item it is deforming should be considered, as it's position provides the origin location of the Effectors influence.

Applying an Effector

While simply adding an Effectors with the 'Add Items' option is possible, placing the appropriate Effector in the Items list, it won't produce the expected results, as the Effector, being a separate item, need to be connected to the Mesh layer by way of a 'General Influence'. The General Influence acts upon the mesh, while the Effector itself produces the instructions on how the Influence should modify its target. This connection can be made automatically (while in 'Items' mode) by first selecting the target item(s) either in the Items list or any 3D viewport, and then RMB+clicking on the

item, opening the contextual menu and selecting the menu option "Add Deformer > Effectors > Choose Type". Additionally, users can select the target item(s) and press the Effector button in 'Deformers' toolbox subtab of the 'Layout' and 'Animate' interfaces and produce the same result. Once the Effector is created, users can position the Effector and modify its values to produce the deformed results. When applied to multiple items, each item will be deformed by the single Effector based on its relative position to the mesh.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. These values defines the origin of influence for the Effector.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. These values defines the direction of influence for the Effector.

Order: Allows the user to set the order that rotations are applied to the Effector item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item representation in the 3D viewport. To adjust the scale of the arc's influence, disable the 'Use Length' option and scale the 'Radius' value.

Deformation Mode: The 'Deformation Mode' determines how the transforms of the Effector translate into a deformation-

World- This is the default setting and determines the deformation based on the world position of the effector. In most cases, this provides the expected results.

Local- This setting applies the deformation relative to the Setup position of the item as if the item were still at the rest pose regardless of the position of the Effector. Useful for the order of operation workflow when rigging.

Local ** Rotation Modes- Identical to the 'Local' mode but only the specified rotation is applied applying a deformation that is more like a Rotation Effector with Parametric interpolation.

Reset: Resets the selected transform values to (0,0,0) returning the item back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Mesh position intact. This is done by adding a negative transform item to the Mesh items channels.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Bend Effector--

Enable: Toggles the effect of the layer on or off. When un-checked (disabled), the layer has no influence on the scene. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Axis: Determines the general directional Axis for the Effectors deformation, without the need to directly rotate the Effector item.

Angle: The 'Angle' value determines the amount of Bending deformation applied the target mesh as an arc. For longer or shorter bend lengths, change the Effectors position relative to its target, or disable the 'Use Length' option and adjust the 'Radius' value.

Use Length: When 'Use Length' is *enabled*, the Radius value is determined automatically by the Length option value.

Length: The 'Length' value determines the maximum range of influence along the chosen 'Axis', where values will attenuate along the curve, but those beyond the defined 'Length', mesh will be affected at 100% strength.

Radius: The 'Radius' value determines the size of the bending arc, and consequently its scale of influence.

Lag Effector

Effectors are a means to affect a target item, specifically a Mesh layer, deforming it in specific ways. The 'Lag Effector' when assigned to an item will deform the object along a directional vector, determined by the target items animation (essentially it will be deformed along the direction it is traveling). If the target item isn't animated, no effect will be visible. As the Lag values are increased, the greater the deformation will be.

Applying an Effector

While simply adding an Effectors with the 'Add Items' option is possible, placing the appropriate Effector in the Items list, it won't produce the expected results, as the Effector, being a separate item, need to be connected to the Mesh layer by way of a 'General Influence'. The General Influence acts upon the mesh, while the Effector itself produces the instructions on how the Influence should modify its target. This connection can be made automatically (while in 'Items' mode) by first selecting the target item(s) either in the Items list or any 3D viewport, and then RMB+clicking on the item, opening the contextual menu and selecting the menu option "Add Deformer > Effectors > Choose Type". Additionally, users can select the target item(s) and press the Effector button in 'Deformers' toolbox subtab of the 'Layout' and 'Animate' interfaces and produce the same result. Once the Effector is created, users can position the Effector and modify its values to produce the deformed results. When applied to multiple items, each item will be deformed by the single Effector based on its relative position to the mesh.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

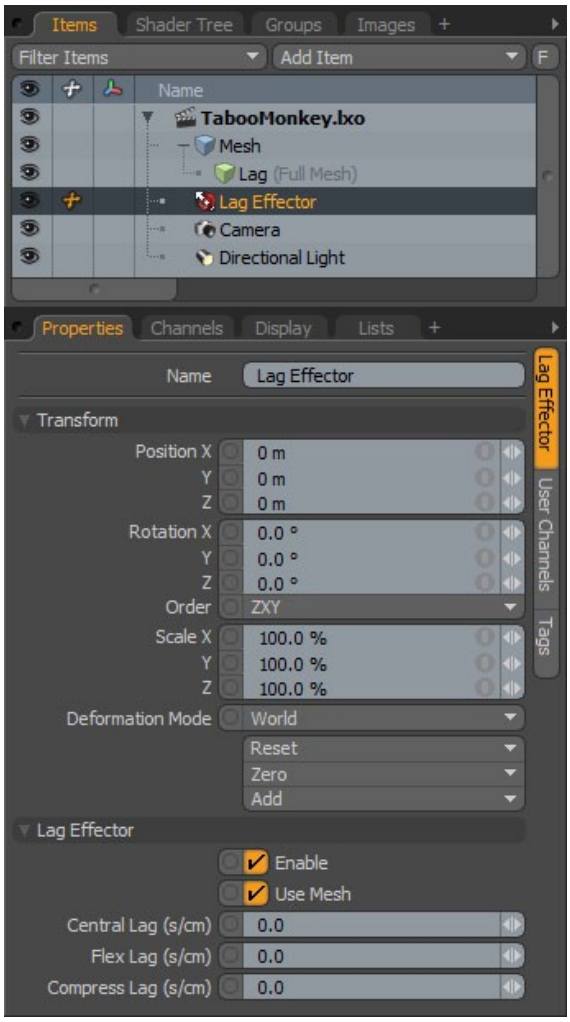
Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. These values defines the origin of influence for the Effector.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. These values defines the direction of influence for the Effector.

Order: Allows the user to set the order that rotations are applied to the Effector item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item representation in the 3D viewport. To adjust the scale of the arc's influence, disable the 'Use Length' option and scale the 'Radius' value.



Deformation Mode: The 'Deformation Mode' determines how the transforms of the Effector translate into a deformation-

World- This is the default setting and determines the deformation based on the world position of the effector. In most cases, this provides the expected results.

Local- This setting applies the deformation relative to the Setup position of the item as if the item were still at the rest pose regardless of the position of the Effector. Useful for the order of operation workflow when rigging.

Local ** Rotation Modes- Identical to the 'Local' mode but only the specified rotation is applied applying a deformation that is more like a Rotation Effector with Parametric interpolation.

Reset: Resets the selected transform values to (0,0,0) returning the item back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Mesh position intact. This is done by adding a negative transform item to the Mesh items channels.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Lag Effector--

Enable: Toggles the effect of the layer on or off. When unchecked (disabled), the layer has no influence on the scene. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Use Mesh: When *enabled*, this option sets the Lag effector to use the center of the mesh it is deforming, instead of the position of the Lag effector itself as the origin. It is not much different than parenting the effector to the mesh, but it is much easier.

Central Lag: This is the amount of time by which a point 1cm from the center will lag behind the center, so it is the amount of time lag per cm distance from the center. The distance is the absolute distance from the point to the origin location, be it Leg effector position or the Mesh's Center.

Flex Lag: Similar to the 'Central Lag', this is also the Lag amount per cm, but the distance is only the distance perpendicular to the line of motion (vector).

Compress Lag: This is the lag amount per cm, but the distance is measured only along the line of motion, giving a stretchiness just along the motion. The 'Flex' and 'Compress' lag amounts are components of the central lag, so by combining them in different levels, you can get good control over how motion deforms the mesh.

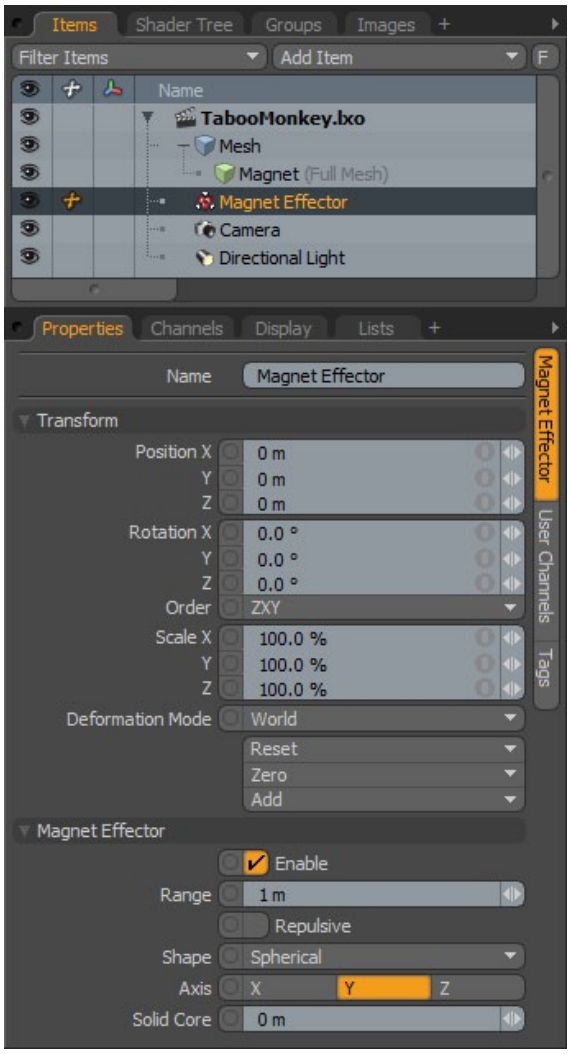
Magnet Effector

Effectors are a means to affect a target item, specifically a Mesh layer, deforming it in specific ways. The 'Magent Effector' when assigned to an item will draw points within its defined range toward the center of the Magnet itself. Enabling the 'Repulsize' value will push points outward.

Applying an Effector

While simply adding an Effectors with the 'Add Items' option is possible, placing the appropriate Effector in the Items list, it won't produce the expected results, as the Effector, being a separate item, need to be connected to the Mesh layer by way of a 'General Influence'. The General Influence acts upon the mesh, while the Effector itself produces the instructions on how the Influence should modify its target. This connection can be made automatically (*while in 'Items' mode*) by first selecting the target item(s) either in the Items list or any 3D viewport, and then RMB+clicking on the item, opening the contextual menu and selecting the menu option "Add Deformer > Effectors > Choose Type". Additionally, users can select the target item(s) and press the Effector button in 'Deformers' toolbox subtab of the 'Layout' and 'Animate' interfaces and produce the same result. Once the Effector is created, users can position the Effector and modify its values to produce the deformed results. When applied to multiple items, each item will be deformed by the single Effector based on its relative position to the mesh.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.



Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. These values defines the origin of influence for the Effector.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. These values defines the direction of influence for the Effector.

Order: Allows the user to set the order that rotations are applied to the Effector item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item representation in the 3D viewport. To adjust the scale of the arc's influence, disable the 'Use Length' option and scale the 'Radius' value.

Deformation Mode: The 'Deformation Mode' determines how the transforms of the Effector translate into a deformation-

World- This is the default setting and determines the deformation based on the world position of the effector. In most cases, this provides the expected results.

Local- This setting applies the deformation relative to the Setup position of the item as if the item were still at the rest pose regardless of the position of the Effector. Useful for the order of operation workflow when rigging.

Local ** Rotation Modes- Identical to the 'Local' mode but only the specified rotation is applied applying a deformation that is more like a Rotation Effector with Parametric interpolation.

Reset: Resets the selected transform values to (0,0,0) returning the item back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Mesh position intact. This is done by adding a negative transform item to the Mesh items channels.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Magnet Effector--

Enable: Toggles the effect of the layer on or off. When unchecked (disabled), the layer has no influence on the scene. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Range: This value determines the range of influence, value within the range will be deformed, those outside the range will not.

Repulsive: The Magnet Effector pulls points (attracts) toward the locator position, when 'Repulsive' is enabled, points will be pushed from the locator position.

Shape: This is the shape of the effector's field, it determines how points are moved to or away from the effector.

Planar- Points are attracted linearly to a directional axis based on the 'Range' value.

Sperical- Points within the 'Range value are attracted to the center of a round area of influence.

Cylindrical- Points within the 'Range' value are attracted to the centerline column of a 'Cylinder' chape.

Axis: Determines the axis direction for the 'Shape' option. Only really affects the 'Planar' and 'Cylindrical' options (as Sperical produces the same results on any axis).

Solid Core: This specifies the distance from the center of the magnet that will be considered solid, thus causing any points inside that range to be moved to the surface of the shape.

Slack Effector

Effectors are a means to affect a target item, specifically a Mesh layer, deforming it in specific ways. The 'Slack Effector' when assigned to an item will deform the target as if suspended between two points. As the two points move closer together, the deforming element in between will sag and droop, retaining its overall initial length; this can be visualized as a rope holding a boat to a dock, as the boat drifts back and forth the rope will droop and stretch between the boat and the dock, but always retain its overall length.

Applying an Effector

While simply adding an Effectors with the 'Add Items' option is possible, placing the appropriate Effector in the Items list, it won't produce the expected results, as the Effector, being a separate item, need to be connected to the Mesh layer by way of a 'General Influence'. The General Influence acts upon the mesh, while the Effector itself produces the instructions on how the Influence should modify its target.

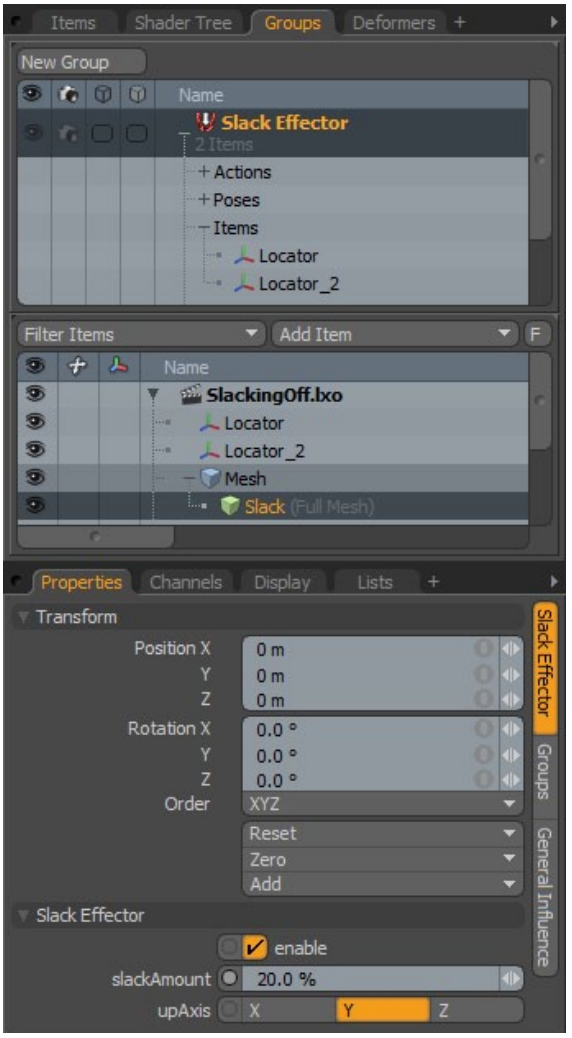
Items meant to be deformed by a 'Slack' effector require a little bit of initial setup to produce the intended result. First, the target item should be long and skinny in nature, like a rope, bar or chain links, and be a single Mesh item layer aligned with one of the three axes (X,Y or Z). Second, two additional Locator items need to be positioned at both ends, representing the start end end of the mesh. To automatically connect the Effector to the target, next select it in the Items list and then RMB+click on the layer, opening the contextual menu and then select the "Add Deformer > Effectors > Slack Effector" option. This creates the Effector not as an Item in the Items list (as others do) but Creates a special 'Group'. Next, users will need to select the 'Slack Effector' Group and the two previously created Locators, and them RMB+click on the 'Slack Effector' Group name and then select the "Add Items" command from the contextual menu. Once the Locators are added to the Group, the Slack Effector will immediately affect the target mesh. Within the 'Properties' viewport panel, users can adjust the related attributes.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the Effector item in XYZ space, though this has no affect upon the resulting deformations.

Rotation: An Item transform that allows the user to numerically set the rotation of the Effector item, though this has no affect upon the resulting deformations.



Order: Allows the user to set the order that rotations are applied to the Effector item.

Scale: An Item transform that allows the user to numerically set the size of the item representation in the 3D viewport, though this has no affect upon the resulting deformations.

Reset: Resets the selected transform values to (0,0,0) returning the item back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Mesh position intact. This is done by adding a negative transform item to the Mesh items channels.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Slack Effector--

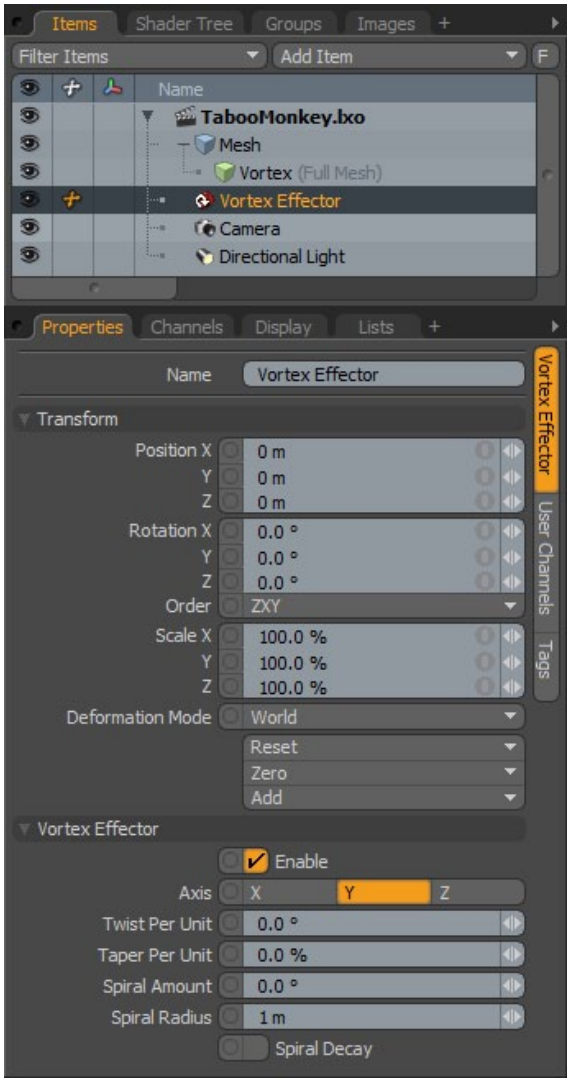
Enable: Toggles the effect of the layer on or off. When unchecked (disabled), the layer has no influence on the scene. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Slack Amount: Determines the amount of droop or sagging between the two Locators. Users can also move the end Locators to increase or reduce the amount of drooping/sagging.

Up Axis: Gravity always pulls downward in the real world (Y axis in modo space), but not all scenes are meant to simulate the real world. Users can adjust this setting to assign which axis direction the deformations is meant to be applied on.

Vortex Effector

Effectors are a means to affect a target item, specifically a Mesh layer, deforming it in specific ways. The 'Vortex Effector' when assigned to an item will deform the object in a twisting motions, with options for 'Decay' providing a spinning tornado type of deformation to the target item.



Applying an Effector

While simply adding an Effectors with the 'Add Items' option is possible, placing the appropriate Effector in the Items list, it won't produce the expected results, as the Effector, being a separate item, need to be connected to the Mesh layer by way of a 'General Influence'. The General Influence acts upon the mesh, while the Effector itself produces the instructions on how the Influence should modify its target. This connection can be made automatically (while in 'Items' mode) by first selecting the target item(s) either in the Items list or any 3D viewport, and then RMB+clicking on the item, opening the contextual menu and selecting the menu option "Add Deformer > Effectors > Choose Type". Additionally, users can select the target item(s) and press the Effector button in 'Deformers' toolbox subtab of the 'Layout' and 'Animate' interfaces and produce the same result. Once the Effector is created, users can position the Effector and modify its values to produce the deformed results. When applied to multiple items, each item will be deformed by the single Effector based on its relative position to the mesh.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. These values defines the origin of influence for the Effector.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. These values defines the direction of influence for the Effector.

Order: Allows the user to set the order that rotations are applied to the Effector item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item representation in the 3D viewport. To adjust the scale of the arc's influence, disable the 'Use Length' option and scale the 'Radius' value.

Deformation Mode: The 'Deformation Mode' determines how the transforms of the Effector translate into a deformation-

World- This is the default setting and determines the deformation based on the world position of the effector. In most cases, this provides the expected results.

Local- This setting applies the deformation relative to the Setup position of the item as if the item were still at the rest pose regardless of the position of the Effector. Useful for the order of operation workflow when rigging.

Local ** Rotation Modes- Identical to the 'Local' mode but only the specified rotation is applied applying a deformation that is more like a Rotation Effector with Parametric interpolation.

Reset: Resets the selected transform values to (0,0,0) returning the item back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Mesh position intact. This is done by adding a negative transform item to the Mesh items channels.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Vortex Effector--

Enable: Toggles the effect of the layer on or off. When unchecked (disabled), the layer has no influence on the scene. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Axis: This option determines the axis direction for the twisting motion of the Vortex.

Twist Per Unit: The amount of rotation added for every unit of distance from the base of the Vortex.

Taper Per Unit: The change in the radius for every unit of distance from the base of the Vortex.

Spiral Amount: This is the rotation amount added as we move away from the center axis of the cylinder. The angle will exactly equal the 'Spiral Amount' when the distance equals the 'Spiral Radius'.

Spiral Radius: This is the distance from the axis at which the spiral rotation added is equal to the Spiral Amount.

Spiral Decay: This toggle sets whether the spiral increases or decreases with distance along the axis.

General Influence

When applying many types of Deformers (such as the Bend, Lag or Vortex), each has its own 'Effector' type, but is then connected to the target with a 'General Influence'. The 'General Influence' allows users to assign any of the various vertex tagging methods to define a grouping of vertices for the Deformer to influence, limiting their application to specific user-defined areas of a mesh (or the Entire Mesh). These tagged vertices can then be transformed using locators to control the effect, providing users a direct way to influence geometry in a scene.

The most flexible of the tags types is the Weight Map type, which is easily created using the many Weights Tools and Brushes. They are the main way a mesh is deformed when animating with a skeleton. Each joint within the hierarchy is assigned a 'General Influence', with the type set to 'Weight Map' and the resulting weight map generated during the Bind process is assigned to the Influence using the 'Name' option. Once defined, any transforms applied to the joint will also transform the associated vertices. 'General Influences' are also the basis for the transformations Deformers - 'Rotation' and 'Transform' that are based on simple Locator transforms, mimicking the Weight Map deformers from previous versions of modo.

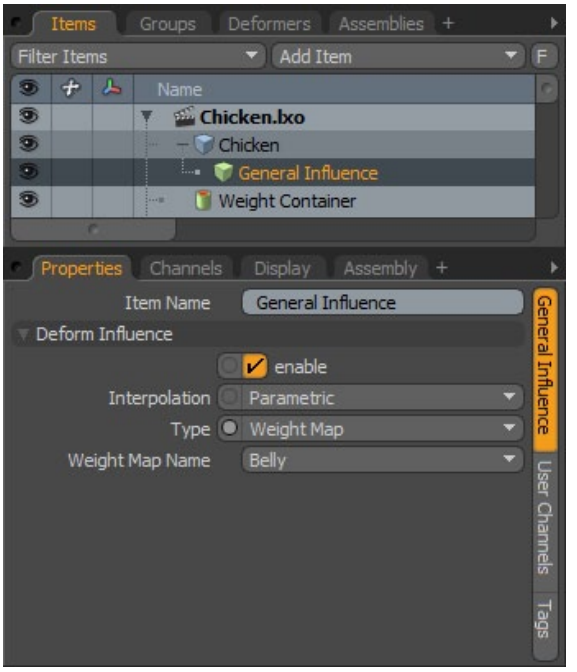
Applying a General Influence

'General Influences' are generated automatically when applying a 'Bend', 'Lag', 'Magnet', 'Rotation', 'Slack', 'Spline' 'Transform' or 'Vortex' Effectors, using the method outlines on the 'Working with Deformers' page of the documentation. The default for each is to influence the 'Entire Mesh', but users can define a polygon tag, and then

specify it by name in the 'Properties' panel to limit the area of influence to specific regions of a model.

Normalization

When binding Joints to a mesh, users will find the resulting 'General Influences' placed within a 'Normalization' folder in the 'Deformers' viewport. When two opposing Weight Maps are influencing the same surface, smoother deformations are produced when the transitions between the two are soft (such as the 'Bind' process creates). However, in these transition areas there is an anomaly; if a single vertex is influenced equally by two Weight Map values, such as 50% and 50% respectively, applying a 50% transform amount and then another at 50% doesn't mathematically add up to 100% for a full Deformation; when they are applied in sequence. This is resolved by the use of Normalization, in modo all 'Influences' that are placed within a Normalization folder have their weight values normalized, meaning all overlapping values are adjusted to equal to 100% when combined. Additionally, the deformations are calculated concurrently (at the same time) eliminating the anomaly with sequential calculation.



General Influence

Item Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Deform Influence--

Enable: The 'Enable' option toggles the Influence item on or off. When enabled the influence will affect its target, when disabled, its influence on the target will be ignored. However, a disabled influence is persistent across modo sessions and is saved with the scene, retaining its settings.

Interpolation: Influences in modo provide two different ways of calculating the resulting deformation-

Parametric- Vertex values are passed directly to the Effector for deformation producing smooth curved interpolations but with more calculation overhead.

Linear- Vertex values are calculated between undeformed and fully deformed and applied linearly between the two positions.

Type: The 'Type' option determines the type of Vertex Map tag-

Entire Mesh- Influences the entirety of the mesh within the current item layer.

Weight Map- Influences vertices with Weight Values above 0.

Vertex Selection Set- Influences all vertices within the defined Selection Set.

Material- Influences all vertices within the defined Material tag.

Part- Influences all vertices within the defined Part tag.

Polygon Selection Set- Influences all vertices within the defined Polygon Selection Set.

Name: This option changes, depending on the selected 'Type', allowing users to select from all the available tag types assigned to the Mesh item

Item Influence

Deformers are meant to twist, bend and otherwise distort a target mesh but sometimes users may wish to instead simply affect an items position, scale or rotation, while NOT distorting the mesh itself. This can be useful when a rigid item need to stay in the proper position relative to a Deforming surface, or when multiples of items need to transform in a uniform way, such as a tower of spheres that *bends* around on itself. In these cases the 'Item Influence' can be applied, as when it replaces the standard 'General Influence', only basic Item transform properties will be affected by the Effector.

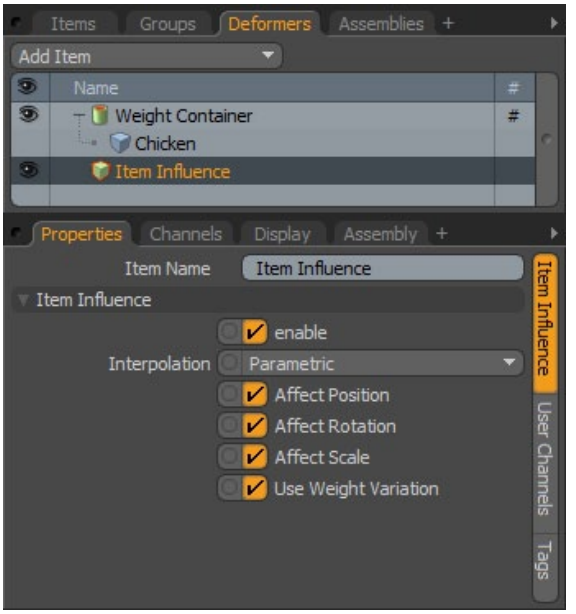
Applying an Item Influence

Being such a specialty case Influence, modo does not (currently) offer any means of *automatically* connecting the Influence to the target Items, therefore a degree of familiarity with the 'Schematic' view are required. Users will need to add an effector, the 'Item Influence' and the target Mesh item(s) manually to the Schematic view.

To do this, first switch to the 'Setup' interface tab, where the Schematic view shows up as a viewport directly below the 3D viewport. Next, the Mesh Item(s) can be added by selecting all the desired layers in the Items list and then LMB+click and dragging the layers into the Schematic view area. This creates an object representing each item in the view. Next, using the 'Add Deformer' button on the top of the Schematic window itself, create the desired 'Effector' (such as the Bend Effector) and then create the 'Item Influence' the same way.

Now that they are created and placed into the Schematic view, they need to be connected. On each side of the objects in the Schematic view are opposing lavender circles, the left side an input and the right an *output*. By LMB+clicking and dragging the circle on the right side, a line is created that can be dropped onto the input of another object. Drag a line from the Effectors *output* to the 'Item Influences' *input*, and then from the influences *output* to the Mesh items *input* (as illustrated below). One Effector and Influence can be used to affect multiples of Items, so if there are more than one, just drag out additional connections to every Mesh item that is meant to be influenced. From there users can adjust the settings of the Effector and it will properly transform the target Mesh(s).

Item Influence



Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Item Influence--

Enable: The 'Enable' option toggles the Influence item on or off. When enabled the influence will affect its target, when *disabled*, its influence on the target will be ignored. However, a *disabled* influence is persistent across modo sessions and is saved with the scene, retaining its settings.

Interpolation: Influences in modo provide two different ways of calculating the resulting deformation-

Parametric- Transform values are passed directly to the Effector for deformation producing smooth curved interpolations but with more calculation overhead.

Linear- Transform values are calculated between undeformed and fully deformed and applied linearly between the two positions.

Affect Position: When *enabled*, will influence the 'Position' transform channels of the target item.

Affect Rotation: When *enabled*, will influence the 'Rotation' transform channels of the target item.

Affect Scale: When *enabled*, will influence the 'Scale' transform channels of the target item.

Use Weight Variation:

MDD Influence

MDD files were invented by Daisuke Ino during his work on a dynamics engine built as a plug in to a host 3D application. These files are a very simple point cache that simply stores a location in space for every vertex in a model for every frame in the animation. The original intent for these files was to simply allow users to freeze a dynamics simulation so that the result could be saved to disc, reloaded and played back without having to simulate again. Soon Hollywood visual FX facilities realized an interesting benefit to these simple point cache files. They can be generated very easily for any kind of deformation animation. This not only allowed them to disable complex IK and bones calculations but also by creating MDD deformers in multiple applications they could transport animation done in one application into a completely different program for final rendering. Genius! Thanks Ino!

Today there are MDD exporters for almost every major 3D application on the market. This means that with MDD support inside modo, you can model in modo, perform complex animation or simulations in another host application and return that model with MDD files to modo for final frame out rendering.

Applying an MDD

Applying an MDD to a file, obviously first requires an MDD generated by another 3D package. A number of exporters are available for most every 3D program out there, a quick search using one of the many internet search engines should yield the necessary results. The base geometry for the MDD should have originated in modo, or be imported into modo for application. With the target item layer selected in the 'Items list, users can RMB+click on the item itself, opening a contextual menu. From the menu, select the "Add Deformer > Influences > MDD Influence" option. This opens an OS specific dialogue panel where users can select the saved MDD file. Pressing 'Open' applies the MDD Influence item to the target mesh, scrubbing in the viewport will display the results. Adjustments to the attributes, if necessary, are available in the Properties panel.

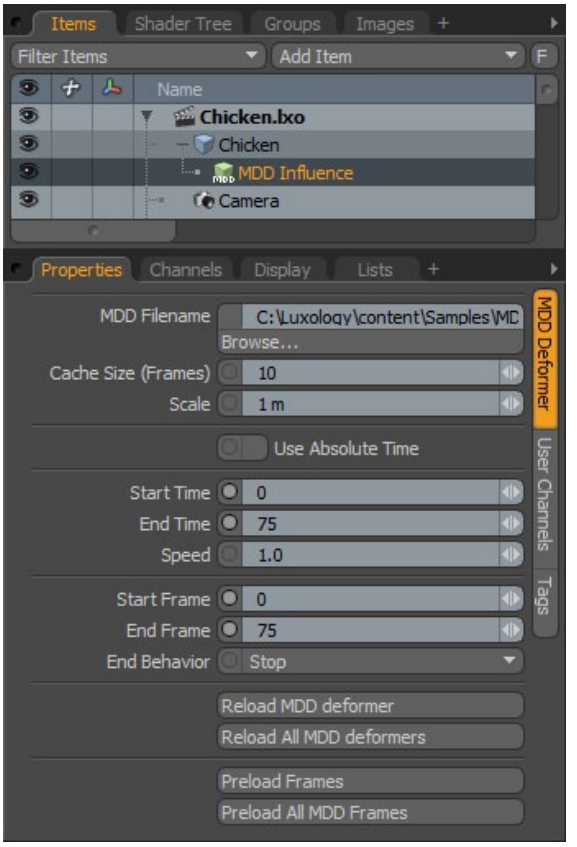
MDD Influence

MDD Filename: This field shows you the path and filename of the MDD that was applied at the time of creation.

Cache Size (Frames): The 'Cache Size' determines the number of frames from the sequence held in memory at one time, when this value is exceeded during playback, frames will have to be loaded from disc, slowing performance. If enough RAM is available, users may prefer to use the 'Load all MDD Frames' option below, increasing scene playback intransitivity.

Scale: The default scale is 1 meter. By changing this scale value you can increase or decrease the size of the item as it is deformed via the MDD.

Use Absolute Time: This option uses the times indicated in the file header of the MDD file rather than the modo time settings. This way users can export an animation with precise timing and replay it in modo without having to adjust any of the MDD time values.



Start/End Time: These values indicate when the MDD animation should be applied during your scene. By default the MDD will begin at frame 1 of your animation. However, if you want to delay the start frame you can simple change the Start Time. For instance, changing the Start Time to 2 will force the MDD file to delay until 2 seconds into the animation at which point the MDD will begin at its initial frame. This is a very convenient method for time shifting your MDD.

Speed: This parameter adjusts the playback rate of the selected MDD file. By default modo uses a one to one match of MDD frame to modo frame. To slow down the MDD playback simply lower the value below 1. To increase playback speed increase the number.

Start/End Frame: These values indicate the frames within the MDD that are to be used. By default when an MDD is applied modo reads the values from the file directly. If you wish to use only subset of the MDD data this is where you can set the desired in and out points.

End Behavior: If the MDD data is shorter than the modo animation the deformer needs to know what to do when it reaches the end of the MDD file. The default behavior of Stop simply holds the last frame of the MDD. There are also options to Repeat and Ping-Pong the playback.

Reload MDD (All): These options allow you to reload the selected MDD or All MDD files in a given project. With these commands you can load a project into modo, return to another application to modify the simulation or character animation, export the MDD file and then force modo to update to the most recent data without having to reload the entire project.

Preload Frames (All): The 'Preload Frames' option will load all the MDD frames into memory for the entire sequence on the selected item. This is a good way to improve playback speed as the deformer won't have to load the data for every frame. The 'All' option loads frames for all MDD Influence items in the scene but may require a lot of memory.

Morph Influence

Morph maps offer users a means of saving alternate user-defined deformations for any mesh. Individual morph map deformations can be animated over time providing the ability to animate say for instance facial expression. In order to animate any morph map, a morph deformer must first be applied to the model.

Applying a Morph

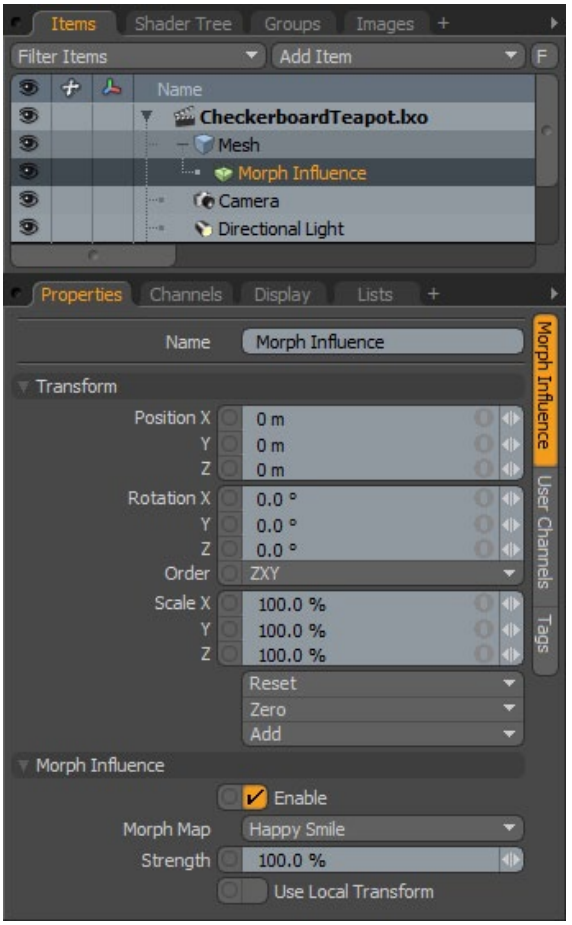
Before a morph can be applied, it needs to be created. New morph maps are added using the 'Vertex Map List' viewport. Any modifications made to the map while selected will be stored in that Morph Map. For more information on creating Morph Maps, please reference the 'Working with Vertex Maps' page of the documentation under the Morphs section.

To apply a morph to a mesh, for the purposes of animation, users can select the target Mesh item in the Items list and RMB+click on the item name to open the contextual menu. In the menu, select "Add Deformer > Influences > Morph Influence", this automatically add the Morph Influence item, connecting it to the mesh. Users can also select the 'Morph' option from the 'Animate' and 'Setup' toolboxes, under the 'Deformers' subtab. Once applied, users can access the attributes by selecting the

Morph item, and viewing them within the Properties panel. Each Morph Influence will support a single morph map, so for multiple maps, additional Influences will need to be added. Combined morph maps are additive, so all will blend together appropriately providing predictable results. Additionally, the morph 'Strength' value can be driven beyond 100% (or into negative values) producing interesting results.

TIP: One of the most confusing aspects of working with Morphs is the fact that when modeling a Morph appears automatically in the 3D viewport when simply selected, but does not under the Animation and Render interface tabs. In order to actually see and animate a Morph, users need to apply the Morph Influence and set its strength to some value above 0. Additionally, the viewport option 'Enable Deformers' need to be selected.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.



Transform--

(Transform values will change the position of the Influence items locator, but has no affect over the Morph itself)

Position: An Item transform that allows the user to numerically position the influence item in XYZ space.

Rotation: An Item transform that allows the user to numerically set the rotation of the influence item.

Order: Allows the user to set the order that rotations are applied to the Influence item.

Scale: An Item transform that allows the user to numerically set the size of the influence item.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Item position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Morph Influence--

Enable: Toggles the effect of the layer on or off. When unchecked (disabled), the layer has no influence on the scene. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Morph Map: This option is used to select the target Morph vertex map.

Strength: This control allows users to modulate the strength of the Morph. At 100%, mesh will be deformed fully at the morph state.

Use Local Transforms: Adjusts the offsets of the Morph so that they deform in the local coordinates. This is useful in creating corrective Morphs when coupled with other deformers.

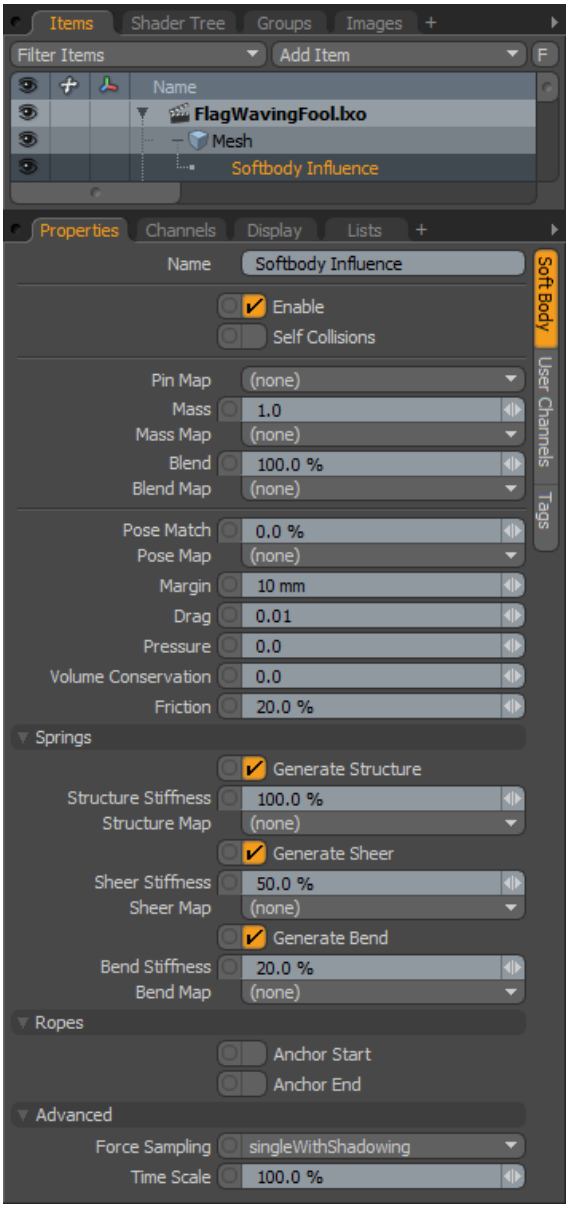
Soft Body Influence

The recoil Soft Body item is governed by the recoil solver. Users should reference that section of the documentation for information on dynamics and simulation. The Soft Body Influence allows users to deform geometry as if it were made of a soft, pliable material. Unlike Rigid bodies that do not deform at all, soft body items are calculated as if springs were attached between adjoining vertices, providing a flex and stretch type warping of the surface. With it, it is possible to simulate a wide variety of materials by adjusting the various properties, from soft billowy fabric, to wiggly gel-like surfaces, to stiff cardboard-like surfaces. With the addition of Pin Maps, Soft Bodies that react to animation are also possible, such as the thin flexible membrane surface on a bat wings as it flies, the Deformation of the wings flapping is applied through a Skeleton and Joint animation, with the secondary membrane motion applied as a Soft Body, utilizing a Pin map.

Apply Soft Body Deformer

Applying a Soft Body deformer only requires that the target mesh item be selected, then simply LMB+clicking the 'Make Soft Body' button in the 'recoil' submenu, of the toolbox in the 'Setup' and 'Animate' interface tabs. This applies the 'Soft Body Influence' item, linking it to the mesh. From there, users can access the attributes of the item within the 'Properties' panel. In order to see the results of the Soft Body

deformations users will need to 'Run a Simulation', this action actually calculates the deformations based on the settings of the Deformer. For more information on working with dynamic items, please reference the recoil overview page of the documentation.



Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Enable: Toggles dynamics on and off for the item. When enabled, modo will calculate dynamics for the item when the timeline is played (or the 'Run Simulation' command is invoked). When disabled item will be ignored for dynamics and collision calculations.

Self Collisions: When enabled, items will be able to collide with themselves (such as when one end of a sheet is passing through another). This option is only necessary for surfaces that inter-penetrate, as it will slow down the simulation.

Pin Map: A 'Pin Map' is simply a Weight Map (or Weight Container) that controls what areas of a surface are affected by the Soft Body Deformations. Limiting them to specific areas can be useful to create a wide variety of organic effects, such as adding secondary motions to characters. Vertices are either fully pinned or not pinned, as every non zero value is treated as a 1.0 value (values aren't attenuated).

Mass: The 'Mass' option controls the overall mass or weight applied to the item. The amount entered is divided by the number of vertices and the same amount is applied each one.

Mass Map: The 'Mass Map' is simply a Weight Map (or Weight Container) that controls how the mass value is applied to the mesh per vertex. The Vertex Map values are normalized by the Mass value. Example: if the 'Mass' is 5.0, and in the mass map there are three vertices with weight values of 1.0, 0.5, 0.5, those vertices would masses of 2.5, 1.25, 1.25

Blend: The 'Blend' option controls how much of the mesh moves toward the simulated positions. A value of 100% means that the points in the mesh exactly match the simulated positions attenuating towards 0% where item would not deform at all (and act as if pinned).

Blend Map: The 'Mass Map' is simply a Weight Map (or Weight Container) that controls how the 'Blend' value is applied to the mesh per vertex.

Pose: The 'Pose' option controls to what degree a surface tries to maintain its original, undeformed shape (this differs from 'Blend' in that Blend maintains the original vertex positions, while 'Pose' maintains the original shape).

Pose Map: The 'Pose Map' is simply a Weight Map (or Weight Container) that controls how the 'Pose' value is applied to the mesh per vertex.

Margin: The 'Margin' determines a distance away for the surface that is added to the size of the object, useful for controlling collision detection.

Drag: The 'Drag' option determines the loss of energy on a surface that is in motion (say for jiggling type of effects).

Pressure: The 'Pressure' value exerts a force upon the overall volume of an item, similar to how pressure would react on an object in the real world, increasing values would add overall volume, inflating it and decreasing (negative) values would reduce its volume, deflating it. This option requires that the target mesh be completely watertight (no seams).

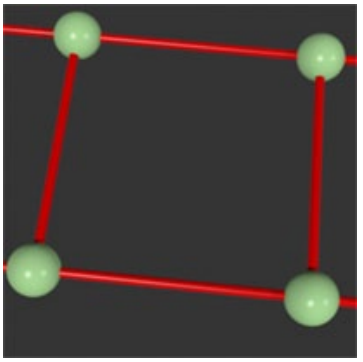
Volume Conservation: The 'Pressure' option tries to maintain an overall internal volume during the deformation simulation. So, if one side of an object is squashed, the other should balloon out. This option requires that the target mesh be completely watertight (no seams).

Friction: The amount an object will slide against another object. A value of 0.0 would offer no resistance, while a value of 1.0 would keep the object from sliding entirely.

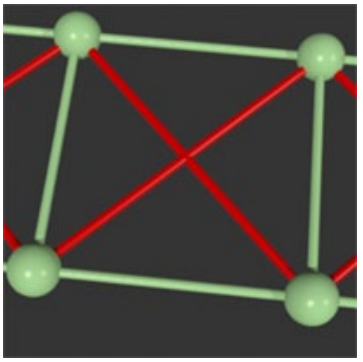
Springs--

The Springs controls the rigidity of the object, and determine wether of not it is stiff or squishy and to what degree the surface maintains its original, undeformed shape.

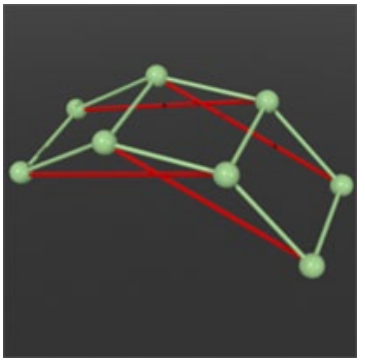
There are three types of springs that all provide different types of structure to the surface. All spring values are defined by the base value multiplied by the value in any associated Weight Map, if it has one.



Structure Springs



Shear Springs



Bend Springs

Generate Structure: When 'Generate Structure' is enabled, springs are generated that follow along the lines of the mesh, connecting the vertices along their polygon edges.

Structure Stiffness: This value determines how resistant the Structure springs are when deforming, depending on the value, they can be very rigid or very pliable.

Structure Map: With the 'Structure Map' option, users can assign a Weight Map that modulates Structure Stiffness as a multiplier. If 'Structure Stiffness' is set to 50%; with a Weigh Map value of 0.6 (60%), the final calculated Stiffness value would be 30%.

Generate Shear: When 'Generate Shear' is enabled, springs are generated that add extra connections to the inside of a face, on a quad polygon, the springs will form an 'X' shape. In a triangle, no additional springs are created. In an Ngon, a spring connected to each non-neighboring vertex. These springs help prevent the polygon from collapsing in on itself.

Shear Stiffness: This value determines how resistant the Shear springs are when deforming, depending on the value, they can be very rigid or very pliable.

Shear Map: With the 'Shear Map' option, users can assign a Weight Map that modulates the overall strength of the Shear as a multiplier. If 'Shear Stiffness' is set to 50%; with a Weigh Map value of 0.6 (60%), the final calculated Stiffness value would be 30%.

Generate Bend: When 'Generate Bend' is enabled, springs are generated that help an object maintain it's shape, following along the Structure Springs. These springs work to

maintain the shape of the object by applying rotational force to maintain the continuity of the surface.

Bend Stiffness: This value determines how resistant the Bend springs are when deforming, depending on the value, they can be very rigid or very pliable.

Bend Map: With the 'Bend Map' option, users can assign a Weight Map that modulates the overall strength of the Bend as a multiplier. If 'Bend Stiffness' is set to 50%; with a Weigh Map value of 0.6 (60%), the final calculated Stiffness value would be 30%.

Ropes--

Ropes are Soft Body deformations applied to modo curves. These can be Curves , Bezier Curves, curves used as Hair Guides, or rendered curves utilizing the Item attribute.

Anchor Start/End: The 'Anchor Start' and 'Anchor End' options are simple toggles allowing easy pinning of either the start or end point of a curve (or both). See 'Pins' above for information on what Pinning is.

Advanced--

Force Sampling: Determines how force values are sampled and applied to the mesh--

single- A single call is made to the force and it is applied evenly to all vertices in the mesh.

singleWithShadowing- A single call is made to the force(from the objects center) and it is applied evenly to all vertices in the mesh. Intervening shadow items will modulate the force amount.

perVertex- Forces are queried individually for every vertex, based on it's current position.

perVertexWithShadowing- Forces are queried individually for every vertex based on it's current position. Intervening shadow objects will modulate the force amount.

Time Scale: Generally a simulation moves items in time based on real world values of simulated forces. Users can use the 'Time Scale' option to adjust the speed at which a simulation runs, per soft-body, simulating slow-motion or super-speed effects when the value is lowered or raised respectively.

Deformer Falloffs

Falloffs serve a variety of functions relating to deforming Items and meshes. They can be used to attenuate the affects of Deformers on a mesh, much like modeling Falloffs do for tools. They can be used as Deformers as well, defining an area of influence that then modulates the surface based on the Falloffs shape and size. Additionally, they can be used to limit how global Forces act upon dynamic items in a recoil simulation. Falloffs are applied to the target based on their intended function.

Falloffs

Capsule Falloff Linear Falloff Radial Falloff Spline Falloff

For documentation on Falloffs, please turn to Falloffs section in Item Reference, Scene Layout Chapter.

Inverse Kinematics

Full Body IK Setup

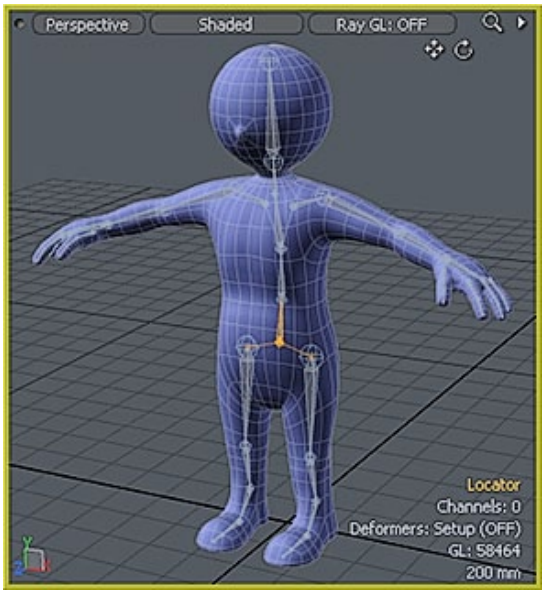
There are two methods for animating hierarchies of elements (such as a character) - forward kinematics (FK) and inverse kinematics (IK). With forward kinematics, one starts animating by positioning the root element and then adjusting all the element down the chain. For example say you wanted a characters arm to reach for an object on a table. With FK, you would first position the shoulder joint, then the elbow and finally the wrist. If lucky, it would only take a few attempts to get the hand properly positioned to the object. IK works in reverse and is far more intuitive, simply grab the hand and bring to the object and the wrist, the elbow and shoulder joints adjust automatically, maintaining their proper orientation toward the goal, in this case, the hand.

modo has implemented an IK system that is fairly automatic and extremely robust. modo's Full Body IK is an automatic IK system that will work in a wide variety of rigging situations. While it is commonly used for characters, it can be applied to any hierarchy chain of locator based items. The setup of the chains is done easily by simply selecting the target elements and applying the commands found in the toolbox of the 'Setup' and 'Animate' interface tabs under the 'Inverse Kinematics' subtab.

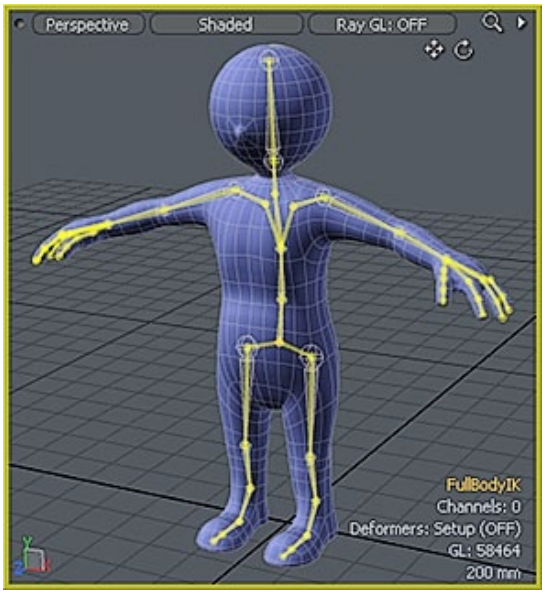
Assigning IK

The first step in using IK in modo is to assign it to the items that you want it to work with, this needs to be done in 'Setup' mode as this will define the rest positions for each joint. This can be done to a hierarchy of items, or to a full Skeleton, and can be applied either before or after the

binding process. There are two methods of doing this, either to an entire hierarchy of items or to specific sections of a hierarchy. To assign IK to an entire hierarchy select the Root item of the chain and LMB+click click on the 'Assign IK' button. All items in the hierarchy from the root outward will have IK assigned to them.

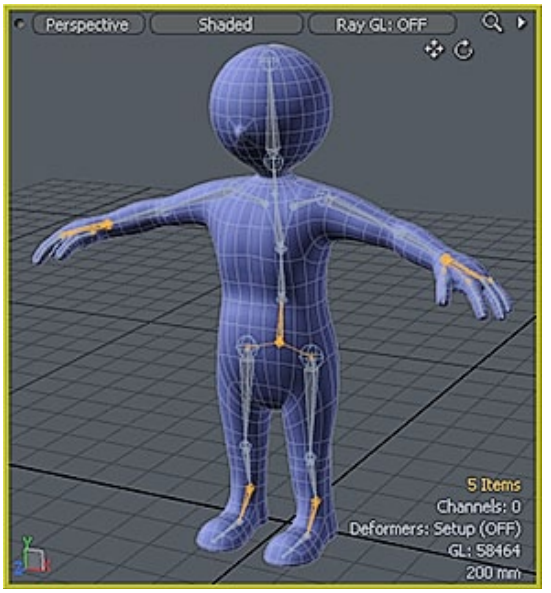


Select the 'Root' element of the hierarchy

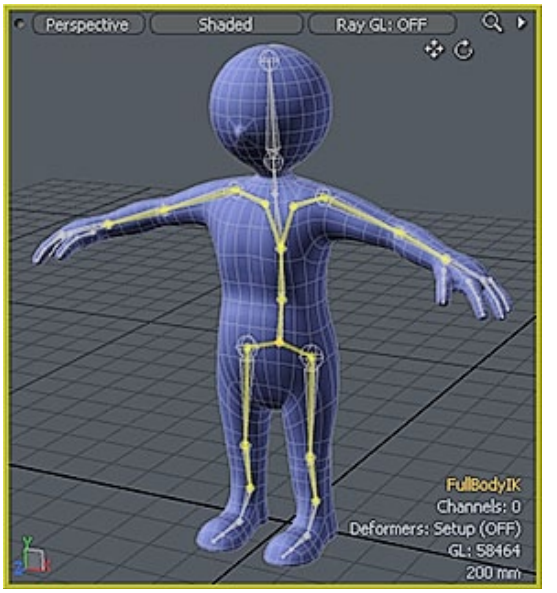


'Apply IK' applies IK to the entire chain

To assign IK to just specific parts of a hierarchy, select the Root item first, and then select the additional item(s) where the IK will be terminated and again LMB+click on the 'Assign IK' button.

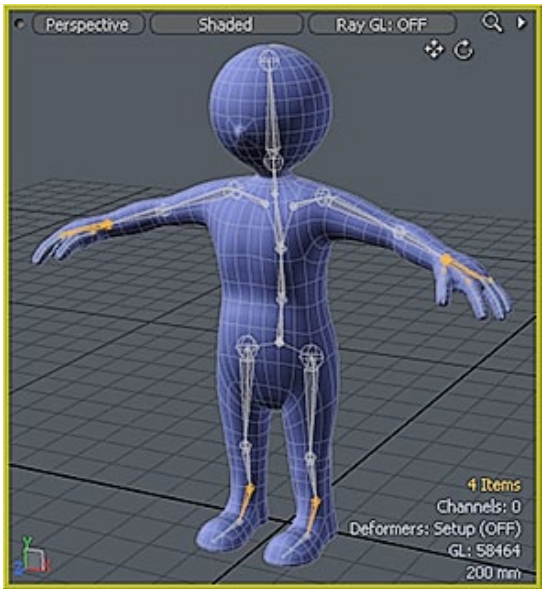


Select the 'Root' and 'Terminating' elements

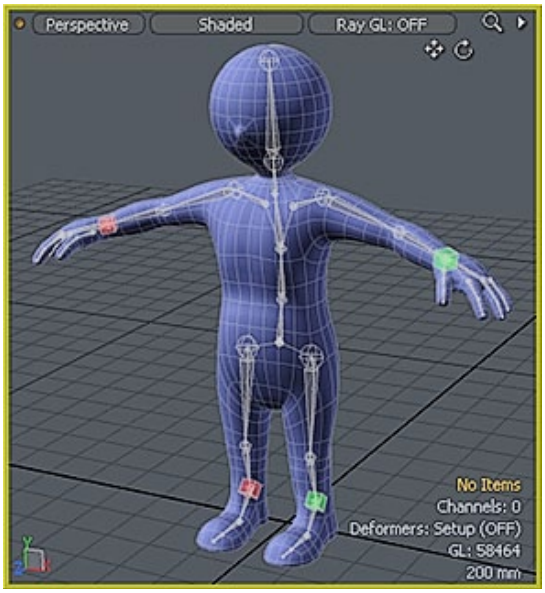


IK is applied between the selected elements

After IK has been assigned in this way, modo retains the selection of the terminating elements, users can simply add goal items by LMB+clicking on the 'Add Goal' button. Now when the Goals are transformed, the chain will adjust its potions and rotations toward the goal. You can then exit 'Setup' and begin to position the chain.



The terminating elements remain selected



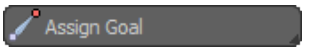
The 'Add Goal' command adds goals to each

All the items in the chain must be part of the same hierarchy, and all items must be 3D locator type items (such as joints created by the Skeleton tool, Locators or Mesh items). If any other type of item is encountered in the hierarchy, or another IK chain is present, the IK will terminate at that item. Any hierarchies that contain scaled items will need to have their scale frozen before assigning the IK. With the IK defined, an IK solver is created and parented to the Root item, called "FullBodyIK". This item contains the settings that control the overall behavior of the chain which can be found in Item Properties when selected (Documented on the 'Full Body IK' page).

Goals

With the IK chains defined, the target elements will still be controlled directly via FK (forward kinematics); users will still be able to individually transform each item in the chain. The IK is activated by the addition of 'Goals' and 'Tasks'. A goal is a general term used for items that drive IK chains, manipulating a goal item causes the rotations of items in the IK chain to change so the position and\or rotation of the end item matches that of the goal item. Goal items can be added to any item belonging to an IK chain apart from the chain Root and can have multiple goals, the IK solver will try to find the best solution to satisfy all the goals.

The simplest method of adding a goal is to select the item in the IK chain that represents the part of the chain that should move toward the goal, such is in an arm, this would be the hand/wrist joint (giving the user FK control over the fingers). Next, LMB+click on the 'Add Goal' button in the IK toolbar. This creates a small cube item (as a locator shape) at the position and orientation of the selected chain item, and adds position and orientation tasks to the selected joint, properly connecting them together with Channel Links. The cube is also given an initial custom color to help identify it, green for the positive side and red for the negative side.



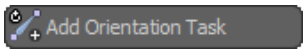
If you have an existing item that you wish to use as a goal then you can use the 'Assign Goal' commands to attach it to a chain. Select the target chain item and the item to be used as a goal (in any order) and choose one of the 'Assign Goal' commands from the tool choice menu on the IK toolbar (RMB+click on the 'Assign Goal' button for additional options). These allow an item to be assigned as a 'Position Goal', an 'Orientation Goal' or both.

Removing a goal item is achieved by deleting the goal item and then selecting the chain item that the goal was connected to and removing the 'Tasks' via the buttons in the IK toolbar.

Tasks

IK assigned to a chain only comes into effect when "Tasks" are added to one or more items in the chain. Tasks are related to Goals, as they are the instructions to the joint to retain a Position, Rotation or Both. For a 'Position Task' the IK solver will attempt to keep the chain item at the world location the item had when the task was added. For a 'Rotation Task' the IK solver will attempt to keep the chain item at the same world orientation that it had when the task was added. 'Tasks' are implemented as groups of channels added to the joint. Each task has a set of FK channels and a Matrix channel. These channels are used to set the task transforms (in world space), so they can be driven by Modifiers, Channel links or other means. Multiple items in a chain can also have tasks added to them, the IK solver will try to find the best solution to satisfy all the tasks on the chain. Settings for each task on a chain item can be found in the Item Properties form.

To assign a 'Position Task', simply select the target Joint(s), and LMB+click on the 'Add Positions Task' button. This will add additional Position Task options to the Joints attributes, found in the Properties panel with the Joint is selected.



To assign an 'Orientation Task', simply select the target Joint(s), and LMB+click on the 'Add Orientation Task' button. This will add additional Orientation Task options to the Joints attributes, found in the Properties panel with the Joint is selected.

Adjusting the Chain

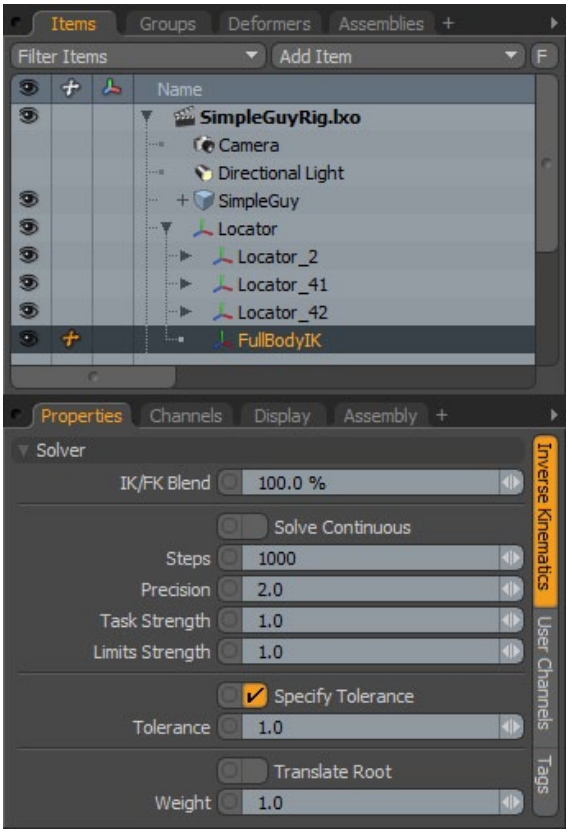
There will be times when the chain items need to be adjusted, such as repositioning a knee or elbow joint for example. When this happens, IK on the chain needs to be disabled before the chain items can be freely edited by the transform tools. Currently this is achieved by turning off the 'Enable' checkbox in each of the chain tasks. Once the chain items have been adjusted IK can be turned back on by re-enabling the chain tasks. Remember to update the location and\or orientation of any goal items that are driving the tasks beforehand. This can be applied easily by using the 'Match Position' and\or 'Match Rotation' commands in the Setup tab of the toolbar. If tasks are not driven by goals the chain will update itself to match the values of the task channels, these may need to be edited manually depending on your setup.

Removing IK

IK can be removed easily from the hierarchy by selecting the 'FullBodyIK' item and then pressing the 'Remove IK' button in the Inverse Kinematics toolbox. This removes the IK settings from each individual item, and deletes the 'FullBodyIK' solver item from the scene.

Full Body IK

Inverse Kinematics, or IK for short, is a system where users can positions and rotate a goal item and the IK hierarchy chain will *reach toward* that goal, making for a faster and more intuitive animation workflow. Modo provides a Full Body IK option that is easy to setup and use. The set-up of the chain is covered on the "Full Body IK Setup" page of the documentation.



Solver--

When any IK chain is applied to a hierarchy, an IK Solver item is created and parented to the Root of the hierarchy, named by default "FullBodyIK". It contains the settings that control the IK for the entire chain (all children items of the root).

IK/FK Blend: Using the IK/FK Blend option, users can control to what degree an the chain is affected by the IK solver, attenuating between the Fully Automatic IK solution at 100% toward fully manual keyframe positioning (FK) at 0%. Items need to be keyframed in their driven state prior to blending toward FK, otherwise the item will interpolate toward their last keyframed position.

Solve Continuous: Specifies whether the IK solving will start from the chain's rest pose (Off) or from the previous IK solution (On). The differences are quite subtle but the latter can be effective for things like tails or tentacles.

Steps: This is the maximum number of times the IK solver will evaluate while trying to satisfy all the tasks on a chain and reach a final solution. If a satisfactory solution is found before the maximum number of steps is reached then the solver will stop early.

Precision: The precision of the solver controls how accurately the solver will try to match the tasks on the chain. Higher values will decrease damping but require more

calculations and may reduce performance depending on the complexity of the chain. Lower values will increase damping and can result in less accurate solving.

Task Strength: Specifies the overall strength of the tasks assigned to this chain, i.e. how much influence the tasks have on the IK solving.

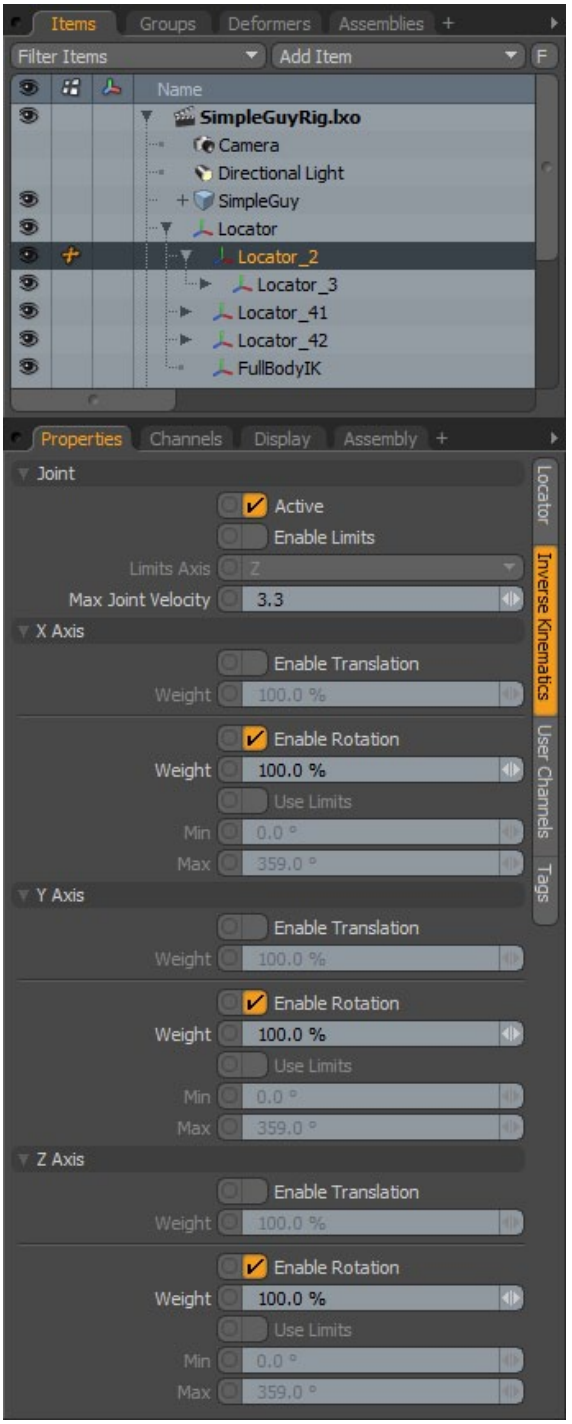
Limits Strength: When joints approach their Min or Max rotation angles they will become stiffer and hence less likely to rotate further. This setting controls the overall strength of the limits on the chain, lower values will result in softer damping of the rotation while higher values will make the rotation of the joints decrease more abruptly. For very simple chains this value will need to be raised before the limits will come into effect.

Specify Tolerance: When enabled the value set under 'Tolerance' (see below) will be used while solving. Otherwise the tolerance will be calculated internally based on the size and complexity of the chain and it's tasks.

Tolerance: Then 'Specify Tolerance' is enabled, this option determines how close a chain needs to match it's tasks before the solver will consider the solution to be completed. Lower values will cause the joints to match their tasks more closely but will require more steps to calculate.

Translate Root: When enabled, this option specifies whether the Chain Root item is allowed to move when solving the IK.

Weight: This controls how freely the Chain Root will move when the setting above is enabled. Zero will prevent it moving while 1.0 will allow it to move freely. This setting can be raised above one if you want to increase the likelihood of the root moving even more.



Joint Settings

Joint--

Active: When enabled, this option specifies that the joint will be affected by the IK solving, disable the option to have it ignore IK.

Enable Limits: Limits on a joint can be used to restrict it's rotation on each axis. This setting is used by the IK system to determine whether limits should be active on the joint.

Limits Axis: When limits are used on a joint this control should be set to the axis of the joint that limits will be calculated from. Normally this is the axis that is pointing towards the joint's first child joint, i.e. the axis running along the IK segment.

Max Joint Velocity: The "Max Joint Velocity' setting is used to adjust the amount of rotation a joint is allowed to make for each iteration of the IK solver. Larger values allow the joint to rotate more. For very simple chains, e.g. a two segment arm or leg, this setting should be increased as there are fewer joints to spread the rotations across. On more complex chains this can be used to reduce jitter that may occur when goals or tasks are moved a long way out of reach.

X/Y/Z Axis--

Each of the three axes provide identical options allowing users to enable/disable translations and rotations and limit the amount of movement.

Enable Translation: When this is enabled the joint's child is allowed to move on the specified axis of the current joint while trying to reach it's goal (Not applicable to the chain root).

Weight: The 'Weight' option can be used to control the amount of joint translation, zero percent will disable movement, 100% allows full movement. The value can be increased above 100% to make movement more likely, especially when several joints have translation active.

Enable Rotations: This option is used to toggle rotation on the specified axis.

Weight: This control can be used to adjust the freedom of rotation round this axis. At zero percent the joint will not rotate around this axis, at 100% full rotation is allowed. Raising this value above 100% will make this axis more likely to rotate.

Use Limits: The Use Limits options will force the IK only rotate within the specified values, producing a hard stop when the Min or Max values are reached.

Min/Max: The 'Min' and 'Max' values determine the range that a joint is allowed to rotate within. Only applies when 'Use Limits' is enabled.

Task Options

When Goals are added to a Joint, additional 'Task' options are added to the target Joint to adjust how the joint changes to reach toward the goal. Tasks are related to Goals, as they are the instructions to the joint to retain a Position, Rotation or Both. For a 'Position Task' the IK solver will attempt to keep the chain item at the world location the item had when the task was added. For a 'Rotation Task' the IK solver will attempt to keep the chain item at the same world orientation that it had when the task was added.

Position Task--

Enable: Controls whether the task is currently active or not.

Weight: This is the strength of the task, i.e. how much effect it has on the chain. When there are multiple tasks on a chain this can be used to make particular tasks have more or less influence on the solution.

Precision: This controls how closely the solver will try to match the joint to it's task.

Number of Joints: Specifies the number of joints towards the root that this task will affect. Set to -1 to affect all joints.

Orientation Task--

Enable: Controls whether the task is currently active or not.

Weight: This is the strength of the task, i.e. how much effect it has on the chain. When there are multiple tasks on a chain this can be used to make particular tasks have more or less influence on the solution.

Precision: This controls how closely the solver will try to match the joint to it's task.

Number of Joints: Specifies the number of joints towards the root that this task will affect. Set to -1 to affect all joints.

Weighting

Weight Container

Vertex Weight maps can be used to modulate a variety of function within modo. When used for animated deformations (essential for character animation), each joint is assigned a specific Weight Map that defines its area of influence upon the mesh. While being incredibly useful and tremendously powerful, the drawback of a traditional embedded Weight Map is exactly that, its part of the mesh and when the mesh isn't available, then the weight map isn't either. The Weight Container is exactly the same thing as a traditional Weight Map, storing weight values for vertices, but its been

separated from the mesh, becoming an item of its own. This is useful when rigging, where the weight values can be abstracted from the geometry. This way a rig can be created and attached to the geometry, users can then simply apply weight to the model, storing them in the Container and it will instantly be affected by the rig. Additionally, since the Weight Container is its own item for storing values, it can store weights for more than a single item, further streamlining the rigging process.

Creating a Weight Container

Weight Containers can be easily added to a scene by using the 'Add Items ' function of the Items list -"Add Items > Deformers > Weight Container" Before any weights can be created they need to be linked to the target mesh(es) There are a number of commands that facilitate these connections. Located in the 'Setup' and 'Animation' interface tabs in the toolbox, under the 'Deformers' subtab. The commands are-

Add Weight Container

This command adds a weight container to the items list. If any items are selected when the command is invoked, they will automatically be added to the container. Additionally, if vertices are selected, they will be automatically assigned a weight value of 1.

Add Item

This command will add any selected Items to any selected Weight Container. Select any number of Items and a Weight Container prior to invoking. Additionally, if vertices are selected, they will be automatically assigned a weight value of 1.

Remove Item

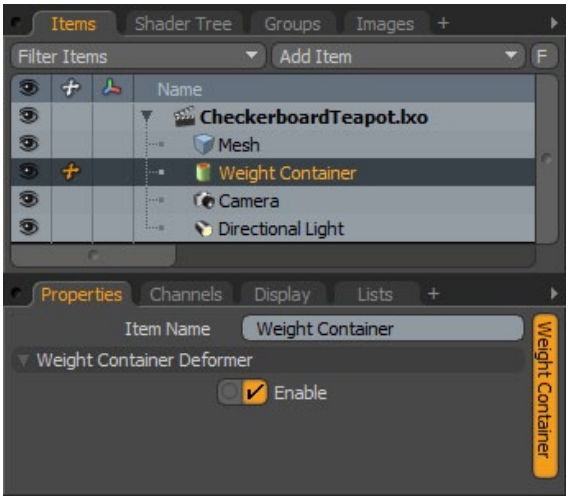
This command will remove any selected Items from any selected Weight Container. Select any number of Items and a Weight Container prior to invoking.

Add Points

This command will add any selected vertices to any selected Weight Container automatically assigning them a value of 1.

Remove Points

This command will add any selected Items to any selected Weight Container. Select any number of Items and a Weight Container prior to invoking. Additionally, if vertices are selected, they will be automatically assigned a weight value of 1.



Item Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Weight Container Deformer--

Enable: Toggles the effect of the Container item on or off. When unchecked (disabled), the layer has no influence on the scene. However, disabled layers are still saved with the scene and its values are persistent across modo sessions.

Weighting Tools

The 'Weighting' tools provides users a way to quickly and interactively adjust 'Vertex Weight' maps used for animation, including those created automatically using modo's 'Bind' command. The Weight Map values are adjusted using modo's familiar brush based workflow and is combined with a number of useful commands, providing users a powerful way to optimize and correct deformations related to weights. To see the results of the tools, users will need to select the target weight (this is done automatically when a Joint is selected) and make sure the 3D viewport is set to either the 'Vertex Map' viewport style or has the 'Show Weight Maps' option enabled, displaying the Weight Map as an overlay to the viewport.

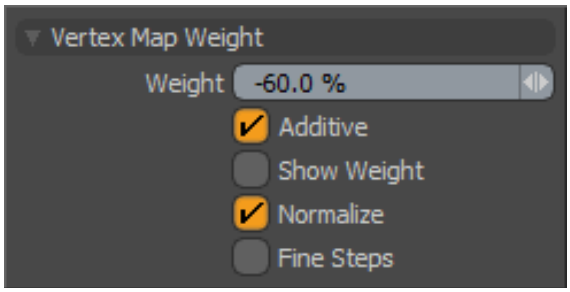
Using the tools

The Weighting tools are located in the 'Setup' and 'Animation' interface tabs, under the 'Weighting' subtab of each toolbox. In order to use the Vertex Map tools, users will first need to select the target Mesh item layer in the Items List. Users will also need to create and/or select the appropriate 'Weight Map' in the Vertex Map list viewport. Please reference the 'Working with Vertex Maps' page of the documentation for

information on creating vertex maps. In the case of editing Weight Maps automatically created during the Bind process

Adjust Weights--

The Adjust Weights tool enables users to haul (off handle drag) in the viewport to adjust the Weight values interactively. LMB+click and drag to the left or right increase or decreases the overall values, with additional options to Add to the existing values, display the numeric values of each vertex and Normalize the target map. Adjustments are made to the entire Map, or can be isolated to specific sections with a component selection.



Weight: This value determines the present Weight Value being applied to the the target Vertex Map.

Additive: When the 'Additive' option is *enabled*, the Weight values will be added to existing map values instead of replacing them.

Show Weight: When the 'Show Weight' option is *enabled*, small numbers are drawn in the 3D viewport at each vertex position for the target item representing the current weight value for the associated vertex.

Normalize: When the Normalize' option is enabled, adjusted weight values will take in to account the Weight values of other Weight Maps present for the target vertices and adjust them accordingly so the combined Weight values never exceed 1.0 (100%).

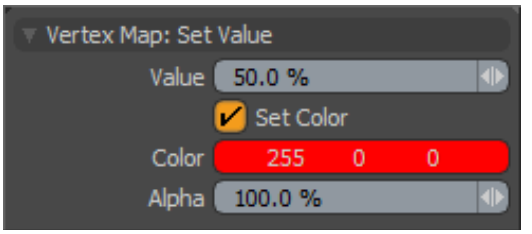
Fine Steps: When the 'Fine Steps' option is *enabled*, Weight values are applied in a much finer degree of precision, providing precise decimal values to the target.

Isolate Weights

The Isolate Weights command is a one shot command that eliminates all Weight values outside of a selected area, therefore isolating the selected area. It is especially useful in eliminating crosstalk between neighboring surfaces with distance based weight maps, by simply selecting the area that is to remain and then pressing the 'Isolate Weights' command, removing all other Weight values from the map.

Paint

The Paint Weights tool will paint Weight Map values using the same familiar workflow as painting texture maps. With the target Weight Map selected, simply define a Value to paint, RMB+click in the 3D viewport to set a brush size and the LMB drag over the surface to apply the weight value. Press and hold the 'Ctrl' key to remove the weight amount when painting. The Paint Weights tool also does double duty to paint Vertex Color Maps, users need only to select the appropriate map type before painting with the tool.



Paint Value: The 'Paint Value' option determines the strength of the Painting action. Higher values will saturate the canvas more quickly than lower values. Set to 100% to quickly overlay an area, and to 5% or under to make subtle adjustments.

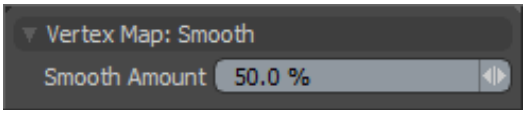
Set Color: When applying values to a Vertex Color Map, the 'Set Color' option enabled automatically. Users can then define a 'Color' value and apply it to the target map.

Color: Defines an RGB value that is applied to the target Vertex Color Map.

Alpha: This value determines how the total opacity of the color applied to the Vertex Color Map.

Smooth

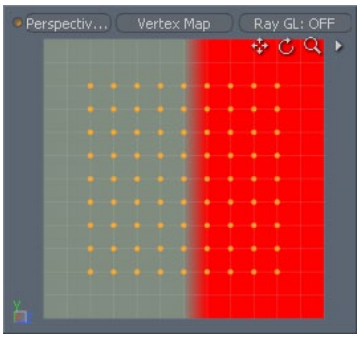
The Smooth tool will blur, or soften Weight Map values, smoothing out their values across the surface they are applied to. With the target Weight Map selected, simply define a Smooth Amount to apply, RMB+click in the 3D viewport to set a brush size and the LMB drag over the surface to smooth the weight values. Press and hold the 'Ctrl' key to increase contrast (sharpen) neighboring weight values.



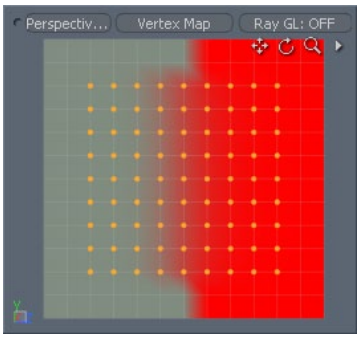
Smooth Amount: The 'Smooth Amount' option controls the strength of the smoothing applied to the Vertex Map. Higher values (>50%) will produce increasingly softer results, where lower values (<50%) will produce less noticeable softening.

Smooth Selected

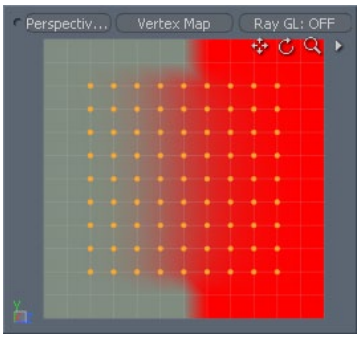
The 'Smooth Selected' command will automatically soften the Vertex Map values of any selected geometry component. Additional applications will continue to further smooth the values, similar to a blur filter applied to an image. Also works well when applied using any of the available 'Falloffs'.



Initial Vertex Selection.



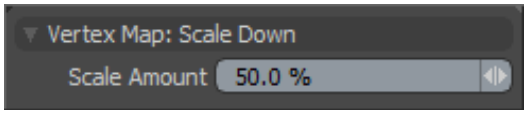
Smooth Selected Applied Once.



Smooth Selected Applied a Second Time.

Scale Down

The "Scale Down' tool will interactively adjust the values in a map down toward '0' (zero), sort of like slowly reducing the values toward '0' (zero). Pressing and holding the 'Ctrl' key will Scale Up the values. The overall effect is sort of similar to dodging and burning in an image editor.



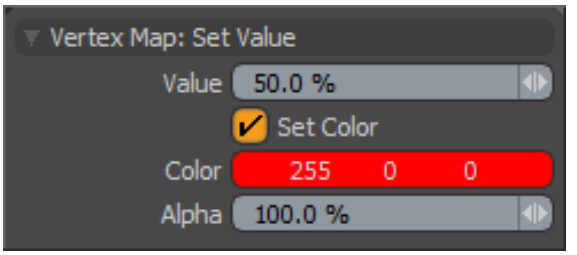
Scale Amount: The 'Scale Amount' determines the strength of the scaling applied to the values.

Erase

The 'Erase' tool will remove any values in the target map that the brush touches. This isn't just lowering the value of the map to 0%, but completely removing the values for affected vertices. No property form for this tool as there are no user adjustable values necessary.

Set Value

The 'Set Value' tool is sort of like the paint tool combined with the erase too, in that it affects any vertex it touches 100% with no fading or attenuation. Pressing and holding the 'Ctrl' key will erase the value.



Value: When editing Weight Map values, this option determines the Weight Value applied.

Set Color: When painting onto Vertex Color Maps, this option is enabled automatically allowing users to specify a color to apply to the map.

Color: Defines an RGB value that is applied to the target Vertex Color Map.

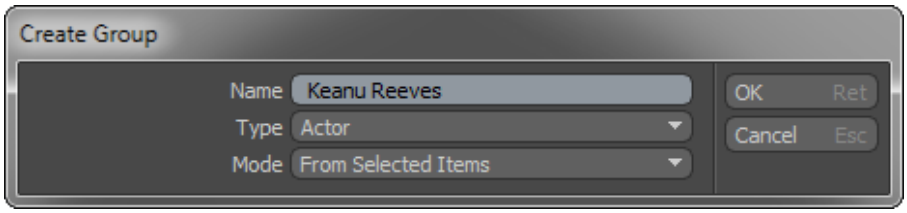
Alpha: Defines an Alpha value that is applied to the target RGBA Vertex Color Map.

Actors Actions & Poses

Across the bottom of the 'Animation' interface, directly under the keyframing and timeline transport controls is located the Actor, Action and Pose control bar. These controls provide you with a streamlined method of animating in modo. An Actor needn't be a character either, it is simply a special type of 'Group, as created in the 'Groups' viewport, that stores Items and their associated Channels. This allows you to quickly assign keyframes for all associated Channels, then save the Channels values as alternate layers using the 'Action' function, where each 'Action' is an alternate of the entire timeline (Actions could also be considered separate takes of a scene). Additionally, the 'Pose' function stores specific keyframe values that can then be applied at any point along the timeline with a single click.

Creating an Actor

To create an 'Actor', make sure to select all the items you want to include and then LMB+click on the 'New' button next to the 'Actor' controls. This opens the 'Create Group' dialogue-



Make sure that the 'Type' option is set to 'Actor'. The 'Mode' menu lets you choose which items or channels will be added to the new 'Actor'. In this case, choose the 'From Selected Items' entry. What this means is that all selected items will be automatically added to the 'Actor' (Group) when it is created. LMB+click 'OK' to create the Actor. If you need to add more Items or Channels to an Actor, or remove Items from the Actor, this can be done from the 'Groups' viewport.

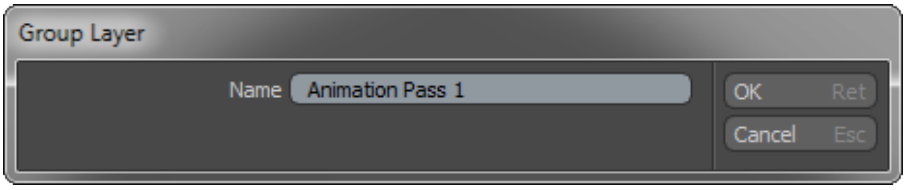
Setting Keyframes

With an 'Actor' defined, users can simply LMB+click on the 'Key' button to define keyframes for every animated channel belonging to each item in the 'Actor'. Make sure to define at least one keyframe for each channel you wish to work with before

hand (For Characters this would typically be the position and rotation Channels). For information on creating keyframes, reference the 'Working with the Timeline' page of the documentation.

Creating Actions

Action serve to store keyframes for the entire timeline per 'Actor', what this means is that each separate action can be either a completely separate animation, or a subtle reworking of an existing animation. In movie terms, each 'Action' is its own take of the scene. To create a new 'Action' layer make sure the 'Actor' control menu shows the particular Actor for which a new animation is desired. Next, LMB+click on the 'New' button next to the 'Actions' control opening the 'Group Layer' dialogue-

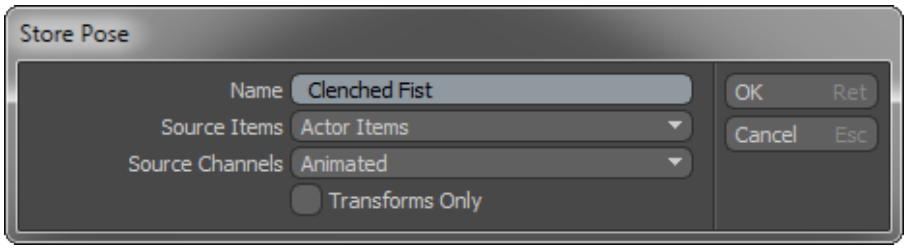


Enter a name for the animation and click OK to create the 'Action'. Now when you animate the items belonging to the current 'Actor' the animation will be stored in the current 'Action' as shown in the Action control menu. Note that if you're animating in a scene with multiple Actors, the current Actor will switch automatically when you select any item belonging to an 'Actor' and that Actor's last active 'Action' will become the current Action. Any number of alternate animations can be created for an 'Actor' by repeating the above steps. You can choose which action to view or edit by simply selecting it from the 'Action' control menu. Deleting an Action is done in the 'Groups' viewport by selecting the Action and pressing the 'Delete' key or choosing 'Delete' from the RMB+click context menu.

TIP: If you have an animation that you like but want to experiment with a bit, you can do this without affecting the original animation by duplicating the 'Action' in the Groups viewport and making your changes to the copy. To do this, go to the Groups viewport and expand the 'Actor'. Then expand the 'Actions' section and RMB+click on the Action and choose 'Duplicate' from the context menu. You can then compare the two animations by switching between them using the 'Action' control in the Animation interface.

Creating Poses

A Pose stores a static snapshot of the current channel values of the Actor contents. For example, you might create Poses for a character's hand positions, such as "Point", "Fist", etc. These Poses can then be quickly applied to the Actor at any point in the timeline. To create a 'Pose' position the items you want to store prior to creating it, example- setting the fingers in a hand to a fist. Next LMB+click on the 'Store' button next to the 'Pose' control menu to bring up the 'Store Pose' dialog and enter a name for the Pose.



The 'Source Channels' menu allows you to choose which channels belonging to the Actor will be stored in the 'Pose'. If 'Transforms Only' is *enabled*, then only position, rotation and scale channels will be stored. Once you have made your choice, LMB+click OK to create the Pose.

Note: The items in the actor will not be affected when creating a Pose, their channel values are copied not cut. This allows further Poses to be created using the current values as a starting point.

Applying a Pose

Go to the position on the timeline where you want to apply the Pose and choose the desired 'Pose' from the popup menu and LMB+click on 'Set' button. Once the values are set, they can be treated like any other keyframe value.

Shading, Lighting & Rendering

Shading, Lighting and Rendering in modo are the necessary functions to produce a final rendered image. Lighting plays an important role in illuminating the scene and setting the mood. Shading defines the look of the modeled surfaces as either shiny or dull, reflective or transparent and rendering calculated the combination of the two to produce the final image.

Applying Materials

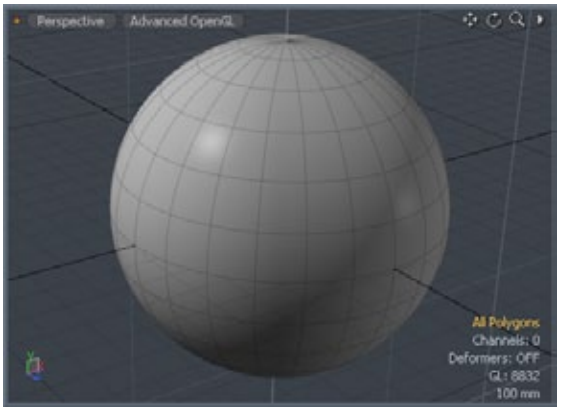
Defining Polygon Tags

All surfacing in modo begins with Polygon Tags, which are simply user defined labels associating any number of polygons together as a group. Though the various tags serve several purposes, they act mainly as a means of generating shader tree masks for applying various material attributes to specific surfaces in a scene, in other words, polygon tags are the main component in applying surfacing to the geometry in a scene. The main tag type to facilitate this is a 'Material' tag, additionally 'Parts' and 'Selection Set' tags offer additional functionality for layering materials for different effects. Users should note that 'Material' and 'Part' tags are mutually exclusive, meaning a polygon can only belong to a single material or part tag, while polygons can belong to any number of 'Selection Sets'.

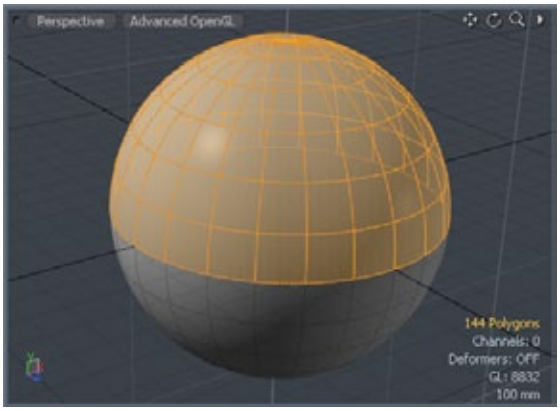
Assigning Material Tags

When rendering in modo, all surface and shading attributes are processed via the 'Shader Tree' to produce the final result. When creating a new scene in modo, you may have noticed it always contains a 'base material' and a 'base shader' item. Modifying any value of either item will affect the entire scene. That's because the shader tree works independent of the geometry with each layer acting like a blanket covering every surface in the scene until it has been instructed to limit itself to a specific surface with a mask. Material tags are the simplest means of generating these types of masks, users don't need to think about the mask itself, as they are generated automatically, but it is an important concept to understand for modo's shading system. Lets go over how to assign a material tag.

In order to assign a tag, you will need some geometry in the scene. Here I have created a default 1 meter sphere.

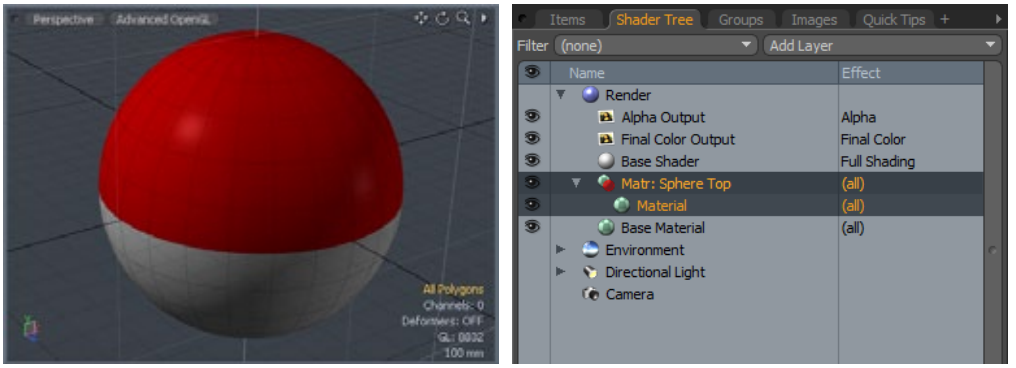


In 'Polygons' mode, select the polygons you wish to apply the material tag to, here I have selected the upper half of the sphere.



Select the menu bar command "Texture > Assign Material Group" or press the 'm' keyboard shortcut to open the 'Polygon Set Material' dialog box. Here you define the 'Name' of the tag as well as set some basic surfacing attributes. Additionally, you have the option to set any material tag as the default value within this dialog box. Here I have chosen a red color and named my surface 'Sphere Top'.

Once you have modified the values to your liking, LMB-click the 'OK' button or press the 'Return' (Enter) key on the keyboard to assign the material to the selection. In doing so, you can see that a group item " " was added to the shader tree with a new material item " " associated to it. If you can't see the material item, click the arrow widget " " next to the named material group to make it visible. Now any any changes made to this new material item will be limited to only polygons in the scene with the tag 'Sphere Top'.



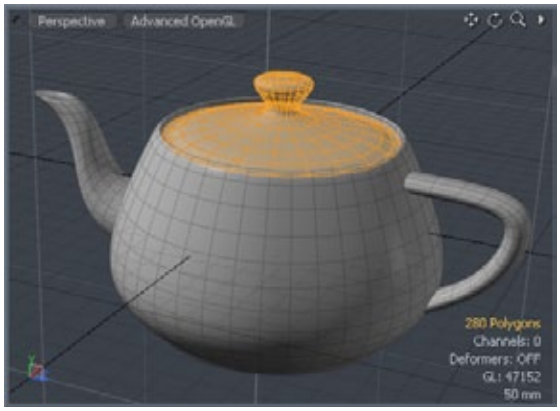
You can continue recursively adding additional material tags to define all the various surfaces in a scene in the same way, just remember that a polygon can only have a single material tag at a time, so if you accidentally assign a second material, it will overwrite the previous value. To fix this, simply select the offending polygon(s) and reassign the previous tag. If you wish to layer materials that require additional selection groups (such as a label onto a bottle of wine), users can define these groupings through the addition of 'Part' tags and 'Selection Sets'.

TIP: In complex scenes, users should consider a strategy for assigning material tags to the various surfaces. Descriptive names are often better than cryptic ones. Consider the case of a car model. One naming scheme might title materials based on their type, such as "Glass", "Rubber", "Leather" and "Chrome", or by what the item itself is; "Windshield", "Tire", "Seat" and "Bumper". Both are much better than "Upper Front", "Inside", "Lower Left" and "Underside" because when you hand off a file or return to it at a later date, the descriptive names will still make sense.

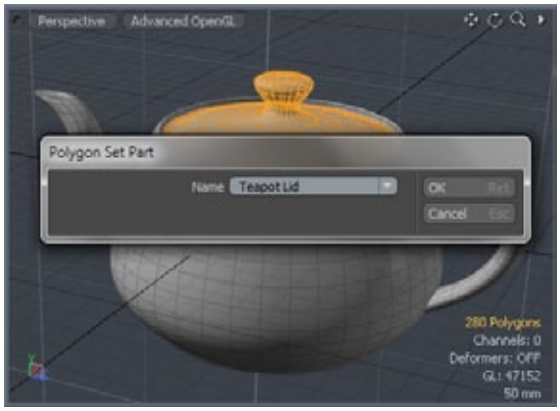
Part Tags

Parts work in a similar fashion to 'Material' tags offering an additional means of defining polygon groupings. Parts can be a useful way to define additional surfacing layers in the shader tree, especially in cases where the surface layer spans across two or more material tags boundaries. They can also be useful when exporting geometry to other applications. Parts can be easily defined in the following manner.

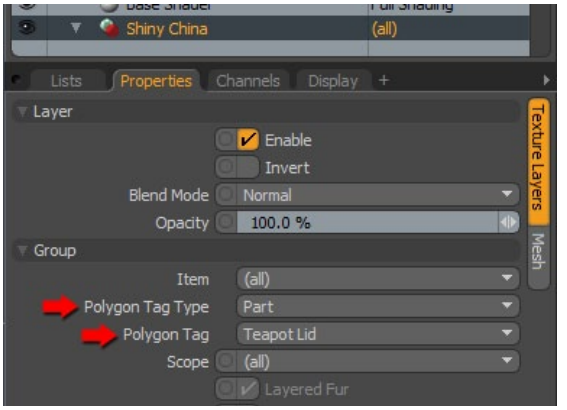
In 'Polygons' mode, select the geometry you wish to assign the part tag to, in this case the lid of the teapot.



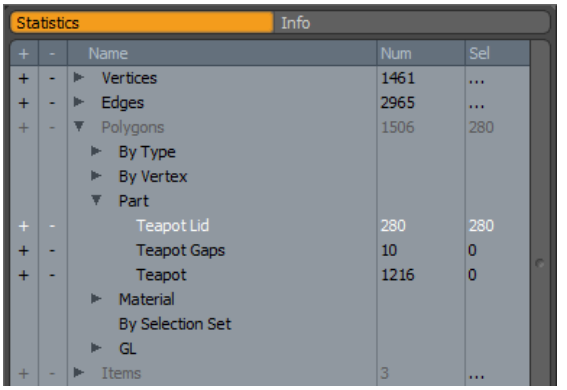
Select the menu bar command "Geometry > Polygon > Set Part..." to open the 'Polygon Set Part' dialog box. You can assign a new name or use the drop down context menu choose an existing part tag from the list. LMB-click the 'OK' button or press the 'Return' (Enter) key on the keyboard to assign the Part tag to the selection.



Once applied, part tags can be recalled in two ways. If you create a shader tree group item, within its properties panel, one can define the 'Polygon Tag Type' as a 'Part', and in the 'Polygon Tag' drop down menu, select the a part you defined in the previous step. When defined in this way, the affect of any shader tree items added to this group would be confined to the polygons grouped by the tag, masking the layers in the same manner as a material tag.



The second way to recall a 'Part' is to use the 'Statistics' viewport panel under the 'Lists' tab. Within the viewport, under the "Polygon > Part" section, users can recall part selections by using the +/- buttons in the left column of the panel. Here I clicked on the '+' symbol to activate the 'Teapot Lid' selection. Pressing the '-' button would deselect any polygons that contained that tag.

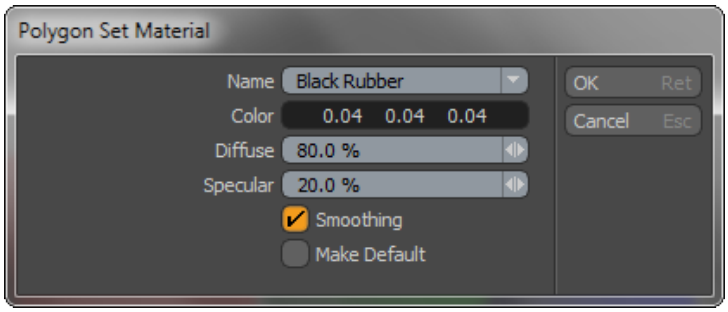


TIP: Users can add tags to curve elements in the same way that tags are added to polygons. Simply select the Curve itself while in 'Polygons' mode and apply the tags as demonstrated above. This can be useful when defining alternate/additional guides when styling a 'Fur' layer.

Polygon Set

The 'Polygon Set Material' function assigns a material tag to the current polygon selection (or, if nothing is selected, all polygons in the active item layers), additionally creating a 'Material Group' item " " in the shader tree with the same name and proper tag assignment, and a single child 'Material' " ". Once defined, any modifications made to properties of the child material will be limited to only those polygons that hold that specific tag. Please reference Defining Polygon Tags for more information on specifying and applying tags and the Shader Tree page for adding and modifying additional layers to the shader tree.

Users can apply a material tag several ways. The command is available in the menu bar under "Geometry > Polygon > Assign Material Group". Users can also use the 'Set Material' button of the polygon toolbox in the modeling tabs of the interface. Finally, users can use the 'M' keyboard shortcut to assign material tags to a polygon. When invoked, the following dialog opens--



Polygon Set Material

Name: The 'Name' function allows users to either define a new name for the "Material Tag" by simply typing into the name input field, or, by LMB clicking the arrow widget to the right of the field and selecting from a list of tags defined previously within the scene.

Color: The 'Color' option allows users the ability to specify the 'Diffuse Color' value for the material item assigned to the material group.

Diffuse: The 'Diffuse' option allows users the ability to specify the 'Diffuse Amount' value for the material item assigned to the material group.

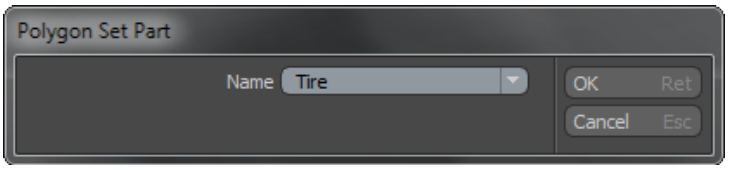
Specular: The 'Specular' option allows users the ability to specify the 'Specular Amount' value for the material item assigned to the material group.

Smoothing: Use this field to toggle the 'Smoothing' percentage value between 100% and 0% in the material item assigned to the material group, producing either smoothed or faceted geometry shading.

Make Default: When "Make Default" is enabled, all subsequent polygons that are created will default to that Material Tag name, inheriting the material setting automatically instead of the receiving the setting defined by the default base material.

Polygon Set Part

Parts work in a similar fashion to 'Material' tags offering an additional means of defining polygon groupings. Parts can be a useful way to define additional surfacing layers in the shader tree, especially in cases where the surface layer spans across two or more material tags boundaries. Parts can also be useful when exporting geometry to other applications. Users can assign 'Part' tags from the menu bar command "Geometry > Polygon > Set Part". When invoked, the following dialog opens--



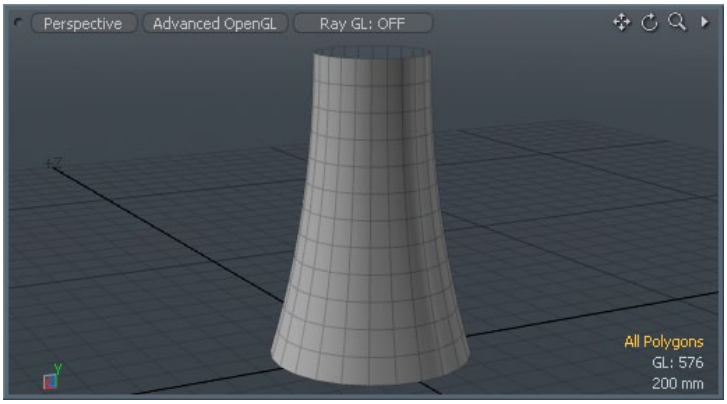
Name: The 'Name' function allows users to either define a new name for the "Part Tag" by simply typing into the name input field, or, by LMB clicking the arrow widget to the right of the field and selecting from a list of tags defined previously within the scene.

Polygon Set Smoothing Group

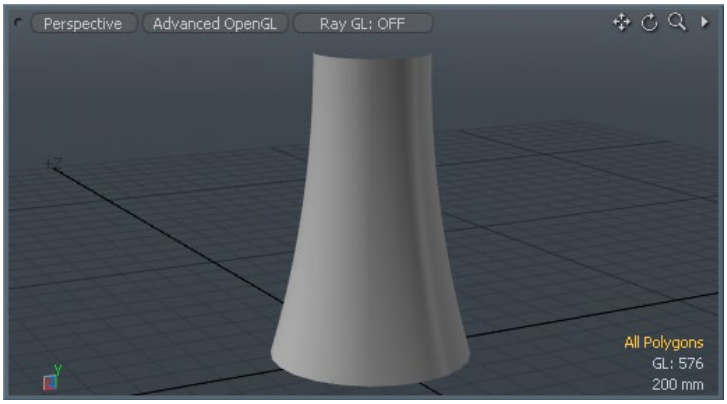
The modo 'Material' item allows users to define the smoothing 'Angle' and 'Amount' for a surface, determining if adjacent polygons are shaded smooth or creased (producing a faceted surface). This option is set based on their difference of angle which in some cases may not provide users with enough control. This is especially true in the case of lower resolution game models where users will prefer to set smoothing based on groups. The borders between these groups determine where creases will appear. The interior of the smoothing group will continue to be based on the material smoothing settings or vertex normal map as modo smoothing groups only add creases, they do not remove them.

modo assigns smoothing groups by applying a specific type of polygon tag with an alphanumeric character name. Users can access the command to tag a polygon as such in the menu bar under "Geometry > Polygon > Set Smoothing Group". Invoking the command opens the Polygon Tag dialog--

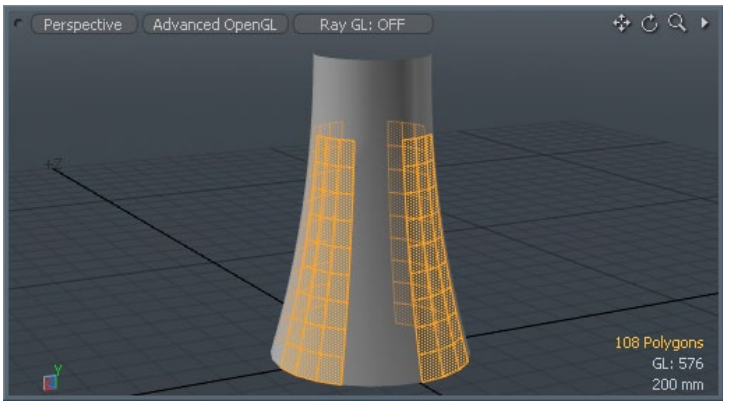
When defining a smoothing group, users must assign alphanumeric characters (A-B, 0-9) representing sets. Smoothing group tags are not treated as names, instead each tag is considered a collection of characters or set. If two adjacent group tags contain the same character then they are smoothed. They are only creased if adjacent smoothing group tags don't contain any of the same characters. Here's an example--



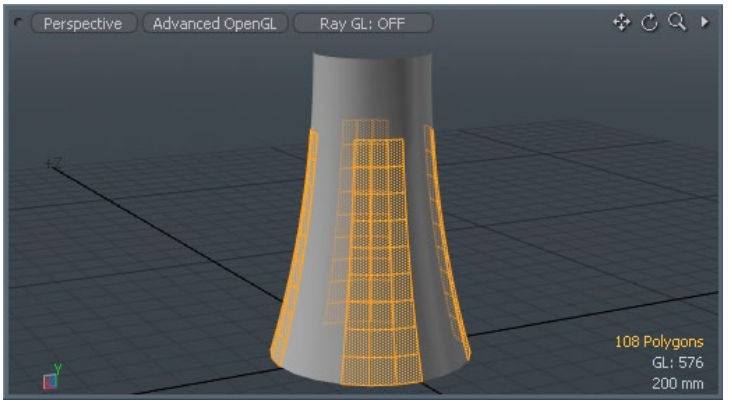
Here is some cylindrical geometry that requires creased edges.



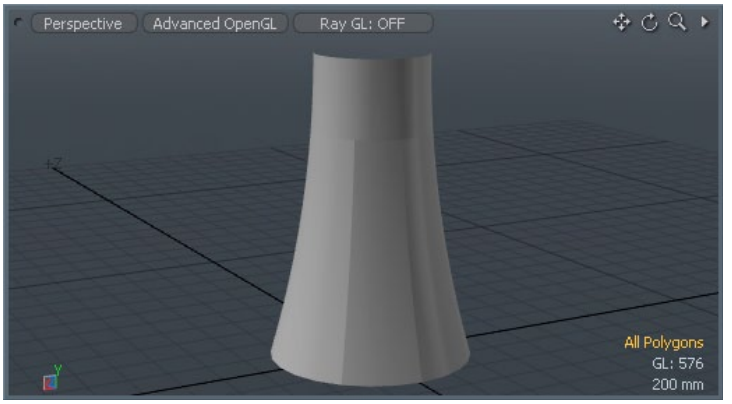
Turning off the edges (numeric pad /) allows one to better see the smoothing.



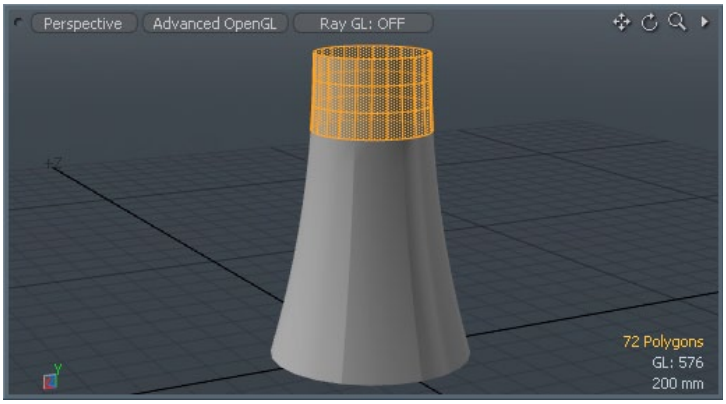
Sections of geometry are selected and tagged with a smoothing group name 'A'.



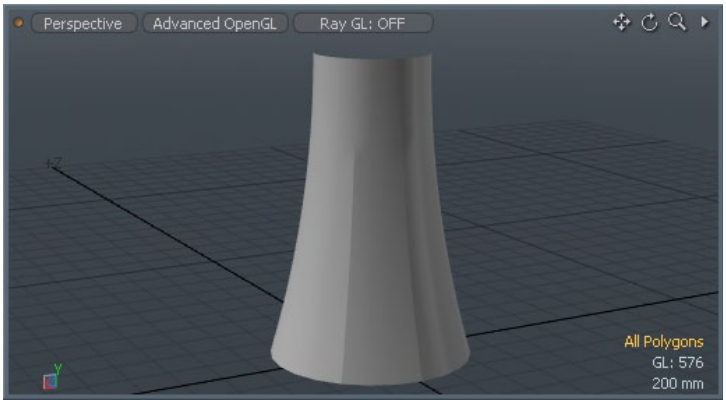
Alternating Sections are selected and tagged 'B'.



The border between adjacent sections are now creased.



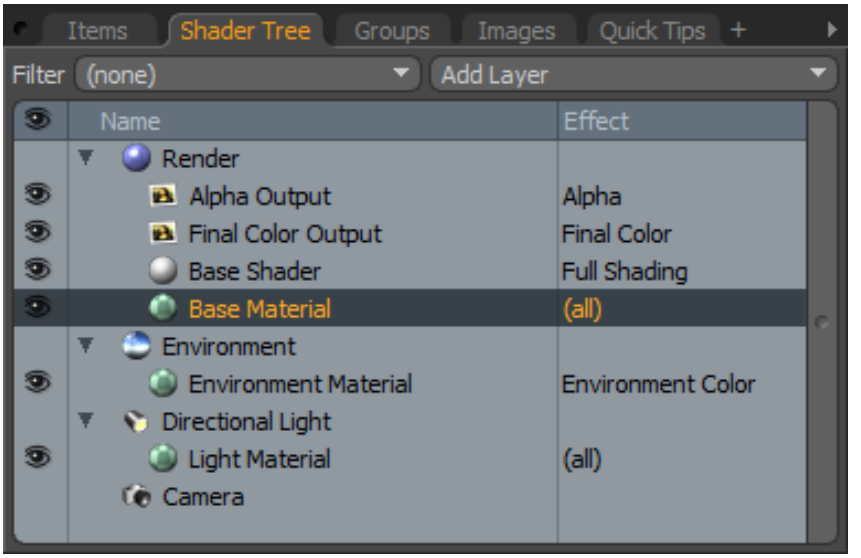
The top section is now selected and assigned the smoothing group name 'AB'.



Since the top section belongs to both the 'A' and 'B' sets, it is smoothed by both.

Shader Tree

Before working with the shader tree, please make sure you are comfortable with the viewport UI, by referring to the Shader Tree Viewport page of the documentation.



Understanding the Shader Tree

All the shading and surfacing of any scene is handled by the shader tree. This is where specific surface attributes are specified, such as what color a surface is, or how shiny it is. It is also where settings for global illumination, environments, and how the scene is rendered are specified. The shader tree is so named because it is conceptually similar to a tree where the 'Render' item "" can be considered the trunk with each individual 'Material Group' item "" the branches. Each branch can contain other 'material groups' nested inside for different layering effects. Eventually, all these layers, nested or otherwise, feed into the 'Render' item, this information is then fed into any 'Render Outputs' "" producing the final rendered image when users activate any of the render commands. To produce the final result, layers are processed in a bottom to top fashion. Anyone familiar with Photoshop's layers palette will be instantly familiar with how modo's shader tree works as any opaque layer above another will always supercede the underlying layers, obscuring them. They can only be made visible by either changing the transparency of the upper layer, changing the blending mode or by applying a mask to the layer.

Adding & Modifying Shader Tree Item Layers

While the shader tree does control the shading and rendering of any scene, the main functionality of the shader tree itself is for adding, removing, organizing and otherwise managing the many shader tree item layers. When adding new items to the tree, the various item layers can be added by LMB-clicking the 'Add Layer' option of the full viewport window and selecting an item from the pop-up context menu. New items are always added to the top of the list in the shader tree directly below the 'Base Shader' " " item. In cases where a 'Material Group' " " Item is selected, new layers are added within that subgroup, directly below the group item itself. For information regarding which types of layers can be added, please see the Shader Tree Item Layers page of the documentation.

Any of the editable properties related to item layers appear within the 'Properties' viewport when the layer is selected in the shader tree. Selected layers are indicated by the dark bar highlighting the layer name. Multiple non-consecutive layers can be selected by holding down the 'Ctrl' key on the keyboard and LMB clicking subsequent layers in the list. Consecutive layers can be selected automatically by holding down the 'Shift' key on the keyboard, and LMB clicking on a new layer; any layers between the current and active selections will be selected as well. Multiple layer selections of identical item types allow for gang editing of property values, eliminating the need to individually adjust item properties.

For organizational purposes, users may wish to rename layers, which can be accomplished by selecting the target layer and choosing the menu bar command "Item > Rename", this opens a dialog box where users can enter the new name, LMB+clicking OK assigns the new name, but be aware, if multiple items are selected, including anything in the Item List, all selected items will receive the new name. To avoid this from happening, it is better to use the 'Rename' option from the popup context menu when RMB+clicking on the layer name itself. Finally, users may also inline rename a layer; simply LMB+click-pause-click directly on the layer name and then type a new name inline, pressing the "enter' key when done to accept the change.

Any layer within the shader tree can repositioned by LMB click and drag the layer, moving it to a new position. The orange line that appears when dragged, indicates the position the item will take when the mouse button is released, dropping the item. When nesting layers within a material group, the layer will inherit the mask, if any, specified by the material group item. Layer positions are an important concept of the shader tree as they control the order in which items are calculated at render time, higher layers in the tree supercede lower layers.

Layers can be easily removed by selecting the target layer in the shader tree and from the menu bar, select the command "Texture > Delete Shader Tree Layer". Also, by RMB clicking the layer to open the contextual menu and selecting 'Delete'. Finally, selected layers can be removed by pressing the 'delete' key on the keyboard.

TIP: When assigning layer order, remember the tree works bottom up; when setting a layers 'Effect' to 'Layer Mask', it will only mask the texture layer directly above it. Also, gradients that use other layers for 'Input Parameters' need to be below the gradient itself so the layer is calculated first to feed into the gradient. If a layers effect is dependant on another, and it isn't working, sometimes changing the layer order can solve the issue.

Hierarchies

Hierarchies are quite important to the shader tree; as this is how associations are made to material groups, which generate the masks necessary to limit surfacing to individual sections of geometry. Any item that is a child of a material group will inherit the group items mask settings. Hierarchies are denoted by the small indentation in subsequent item layers. Material Group items that contain child layers are preceded by a small triangle " ▶ " icon. LMB clicking the arrow will toggle the hide/reveal action of the nested layers. Successive nested material group items will be further indented indicating their association to the parent item. The Parent/Child associations are created by dropping a layer onto another closed group, or between two layers of an open group. When adding new layers, selecting the parent material group item prior to selecting the new item layer type will nest the new layer inside the group. Finally, if a texture layer, procedural, image based or otherwise, is dropped onto another texture layer, the dropped layer automatically becomes a layer mask to the parent. Masked layers are preceded by a small "+" icon, click the icon to toggle the view of the items mask which is now also indented to show its relationship to the parent item.

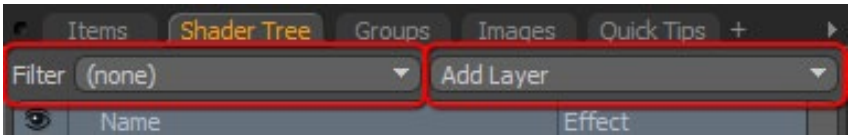
TIP: The key to a smooth workflow is organization when it comes to the shader tree; if left unattended, materials and groups could quickly stack up leaving a long unwieldy list that is difficult to navigate. Multiple material groups can be further grouped by multi-selecting several layers using the shift and ctrl modifier keys. Once selected, press 'Ctrl-G' on the keyboard to move the highlighted layers into another group. When used this way, the groups properties are set to (all); when set as such, the group merely provides a means to organize multiple material groups. The resulting 'Group Item' can be renamed either by RMB clicking the title and choosing 'Rename' from the contextual menu, or by 'click pause click' for inline editing. Grouping items layers can greatly simplify the visual appearance of the shader tree list, making it much easier to navigate. Users can quickly expand and contract all nested layers within a hierarchy by pressing the 'Shift' key when toggling the arrow ' ▶ ' icon.

Specifying Layer Effect

By default, any new texture layer added to the shader tree will modulate the 'diffuse color' channel, meaning the layer will affect the apparent visual color of the surface. Should you wish to effect the surface differently, such as specifying a layer as a bump or displacement map, this is specified via the 'effect' column of the shader tree. A layer can be set to modulate a specific effect, by way of a RMB click over the effect name in the effect column and selecting a new type from the pop up contextual menu. Individually, all of the Texture Effects are covered on their respective documentation pages. Whenever a texture layer is created, such as an 'Image Map' " " or a procedural texture " ", an associated 'Texture Locator' is added as well to the 'Item List'. Texture Locators hold position and projection settings specific to the texture layer and define how it is applied to the surface, while also providing a means to animate the texture using basic transforms applied with keyframes to the locator itself.

Filtering the Item List View

Across the top of the viewport are two buttons, one that allows users to filter how layers are displayed within the viewport, and a second used to add additional item layers to the tree. For more info on adding layers, please see the Shader Items page of the documentation.



The 'Filter' function always defaults to the '(none)' value, which displays all items within the shader tree. Alternate options available in the menu can be selected by clicking the small triangle widget to the right of the bar. When any of the other options are selected, they temporarily toggle the visibility of the items in the list, without removing or disabling them, making it easier to locate a specific layer, or to make selections for gang editing multiple items easier.

The options include--

Material-- Displays only Material Items " " in the shader tree.

Texture-- Displays only Texture Items " , " in the shader tree.

Group-- Displays only Material Group Items " " in the shader tree.

Shader-- Displays only Shader Items " " in the shader tree.

Render Output-- Displays only Render Output Items " " in the shader tree.

Fur Materials-- Displays only Fur Material Items " " in the shader tree.

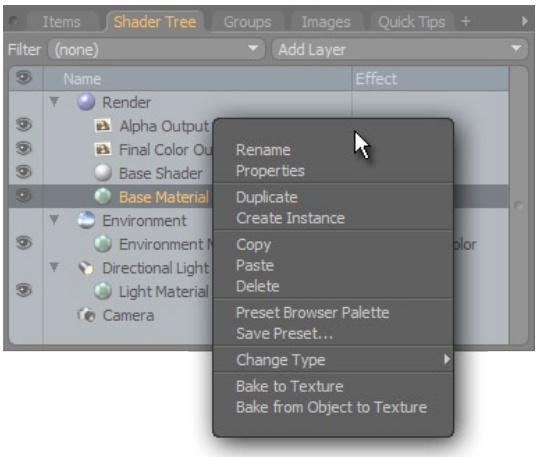
Filter on Selection-- When this option is enabled, the filtering will only display filtered items from the selected item down the branch.

Toggling Layers

Sometimes a user may wish to toggle the effect a layer has on a scene without removing the item from the list. This can be accomplished by LMB clicking over the eye icon " " within the visibility column on the left side of the viewport. Toggling a layer disable/enable its affect on shading in the scene, but the item itself will retain all settings and remains part of the shader tree list when scenes are saved or closed. The ability to toggle layers can be extremely helpful in a variety of scenarios, including creating various shading passes that users would want to enable and disable when rendering, applying a variety of textural options when rendering out conceptual product stills, and experimenting with alternate shading options while exploring looks for a scene.

Context Menu

Additional options for working with the item layers are available by RMB clicking on an item layer name and choosing an option for the pop up contextual menu.



Rename: Allows the user to assign a new name to the specific layer.

Properties: Opens the 'Properties' viewport with the selected layers corresponding settings.

Duplicate: This command will create a copy of a layer with identical settings. Changes

in the original item are not reflected in the duplicate layer, nor are modification to the duplicate reflect in its original layer.

Create Instance: Creates a clone of a layer that references the original; changes to one layer automatically propagate to all instances. Especially useful when separate objects with identical properties reside within different masks. When instances are used, changing one items attributes automatically duplicates the changes on the instance.

Copy/Paste/Delete: Entire layers, as well as the settings of individual layers themselves can be copied, pasted and/or deleted using these commands. When pasting values from one layer to another, the layers must be of the same type, i.e. material item to material item.

Preset Browsers Palette: This command opens a 'Preset Browser' palette.

Save Preset: Any layer in the shader tree can act as a preset. When one selects 'Save Preset...'' a file save dialog is presented asking where you'd like to save your preset. If you want the preset to show up automatically in the default structure, you'll need to navigate to the directory where your presets are stored and save within that directory, one can also create a new directory for saving presets, but make sure to use the '(add path)' command so you can see them within the browser viewport. Often a surface made up of multiple texture layers, in that case make sure to create the preset from the material group " ", as that will include all associated texture layers within the preset. Don't forget that Fur is made up from texture layers so making hair presets is just as easy as making surface presets.

Change Type: This command offers the ability to convert a layer of one type, such as an image map to another type such as a procedural. Associated 'Texture Locators' will retain their settings if applicable but layers themselves will lose their settings.

Bake to Texture: This command offers the ability to generate a texture map image of a layer. Baking textures from inside the Shader Tree is useful for baking a single "Effect" such as Color or Diffuse into an image whereas the "Bake" command from the Render menu bakes all light and texture contributors into a final shaded image. It is reasonable to think of the Shader Tree level of baking as similar to "Flattening" layers in Photoshop. Only layers of the same "Effect" setting underneath the texture layer that is triggering the bake will contribute to the baked layer. Once "baked" this texture can remain in the Shader Tree as a pre-computed result of all contributors which results in a savings of render time and provides a real-time view of the effect. The 'Bake to Texture' command requires a UV map on the baked surface as well as a blank image map to bake into. For more on baking textures, see the Baking Images Workflow page of the documentation.

Bake From Object to Texture: This command offers the ability to use the UV map of a low polygon count model to bake a texture map from a higher polygon count model. Maps can be used for any of the various 'effect' including diffuse color, displacement and normal. Like 'Bake to Texture, a UV map is required and a blank image map to bake into. Both surfaces must have different material name and cannot share the same UV map. When calculating a displacement map, the 'distance setting' from the popup requestor should be used as the 'Displacement Distance' setting in the material dialog box when applied. For more on baking textures, see the Baking Images Workflow page of the documentation.

Shader Tree Items

Every new scene in modo, by default, provides a few basic shader tree layers to get users started. The 'Material' Item " " created when users assign a material tag is sufficient for many surfacing tasks, however, for reproducing more complex surfaces, additional shader items will eventually need to be added, such as applying a bitmap image to produce the label on a can of soda.

The following items may be added to any scene from within the shader tree viewport, by LMB-clicking the 'Add Layer' option of the full viewport window and selecting an item from the pop-up context menu. For information regarding adding and otherwise working with Shader Tree Items Layers, please reference the the Shader Tree page of the documentation.

Shader Tree Items

- Material Group
- Material

Custom Materials

- Cel Edges
- Cel Material
- Hair Material
- Halftone Material
- Skin Material

E:modo Textures

- Texture Samples

Image Map

- Image Map

Processing

- Constant
- Gradient
- Occlusion
- Process
- Weight Texture

Special

- Fur Material
- Render Ouput
- Shader
- Surface Generator

Textures

- Cellular
- Checker
- Dots
- Grid
- Noise
- Ripples
- RPC Texture
- Weave
- Wood

Other

- MatCaps

Texture Locator

- Texture Locator

Material Group

Material Groups can be thought of as containers that hold any of the various shader tree items, such as Shaders, Materials, Textures, and even Render Outputs. These containers can be useful for simply organizing the shader tree, but are incredibly powerful when used to control how surfaces are applied to polygons at render time. Referred to in in the Shader Tree 'Add Layer' function as a 'Group', here they are referred to as a 'Material Group' to avoid confusion with 'Groups'.

Surfacing in modo is separated from the geometry. Individually, Shader Tree item layers affect an entire scene, until limited to specific areas by way of a mask. The masks themselves are procedurally created, and defined by 'Polygon Tags'. Tags are simply bits of information tied to a polygon that associate it with a particular grouping. The most common tag is a 'Material' tag that, when applied to any polygon selection using the 'M' key shortcut (or 'Texture > Assign Material Group...' in the menu bar) automatically creates an appropriately named 'Material Group' layer in the shader tree with the proper tag specification and a 'Material' sub item. Once created, changes made to the various layers under the 'Material Group' will only affect the polygons with that tag. Since polygons can only have one type of material tag at a time, additional layers masks can be applied using 'Selection Sets', polygon 'Parts' and item masks.

These masks that the Material Groups define are based on the settings under the 'Group' header. They can be used individually, or in combination, and while the functions may appear simple, they are an efficient means of surfacing a scene. Understanding what each does is an important step in understanding the shader tree itself, and realizing the power that the shader tree provides.

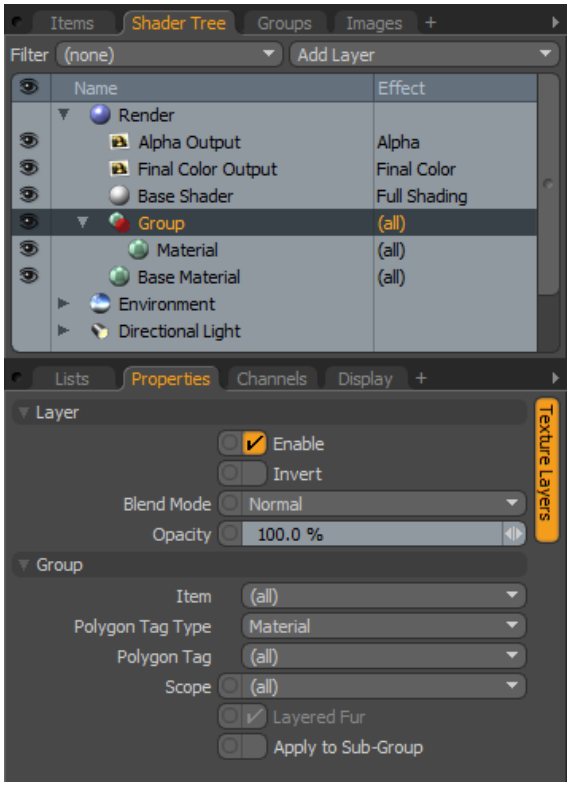
Layer--

Enable: Toggles the effect of the group (and its contained layers) on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled groups are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the group producing a negative effect.

Blend Mode: Affects blending between different groups, allowing user the ability to stack several groups for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current group. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Group--

Item: The 'Item' selector can be used to limit the application of the layers contained within a material group to a specific mesh item (these are the mesh item layers of the Item List). When defined, these are considered an 'Item Mask'. The '(all)' setting applies the materials to the entire scene. This setting can also be used in combination with a polygon tag, further limiting the surface.

Polygon Tag Type: Users can specify the type of polygon tag; several types are possible. Each limits or masks the application of the shading items contained within the group to only the polygons tagged as such.

Material-- The most common polygon tag type, Material Groups with this type of tag and automatically created when using the 'M' keyboard shortcut. Polygons can only have one Material Tag at a time.

Part-- Parts tags are similar to a selection set, but like a material tag, a polygon can only belong to one 'Part' group at a time.

Selection Set-- Selection Sets can be useful for recalling previous polygon selections, however, when specified here as a tag type, the selection set will mask the material groups layers to the associated polygons of the specified selection set. One polygon can belong to many different selection sets.

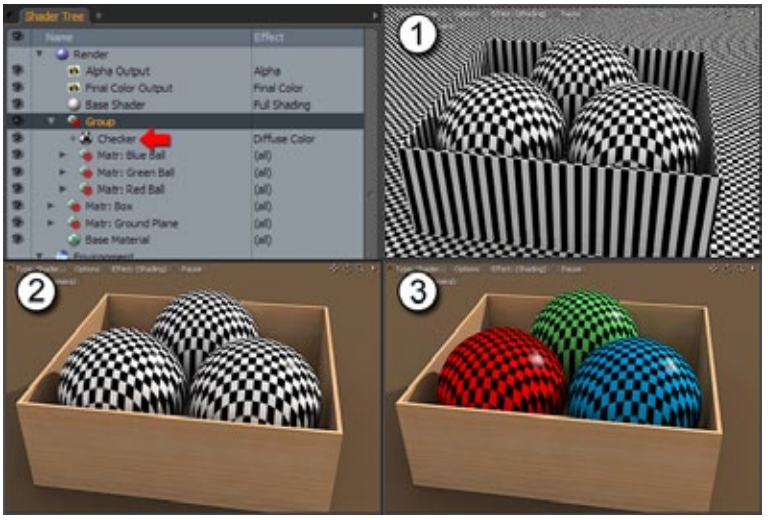
Polygon Tag: When any type of *tag* is created, be it a 'Material', a 'Part' or a Selection Set, the user is presented with a dialog box requesting a name. All tags of the defined type contained in the scene will appear here, where users can use this drop down to select the particular named tag they wish to use.

Scope: The affect of the material groups layers can be further limited to only 'Fur' materials, just the polygonal 'Surface' or to both with 'All'. As a fur material typically derives its surfacing from the underlying polygons (technically from the surfaces material item layer), duplicating a material group and setting the scope to 'Fur' will

allow the user to specify surfacing attributes exclusive to the fur layer(s). This is how it is possible to have green grass growing from a brown dirt surface.

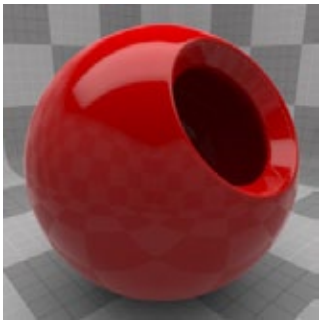
Layered Fur: Enabling this checkbox will allow multiple 'Fur' item layers with different settings on a single surface.

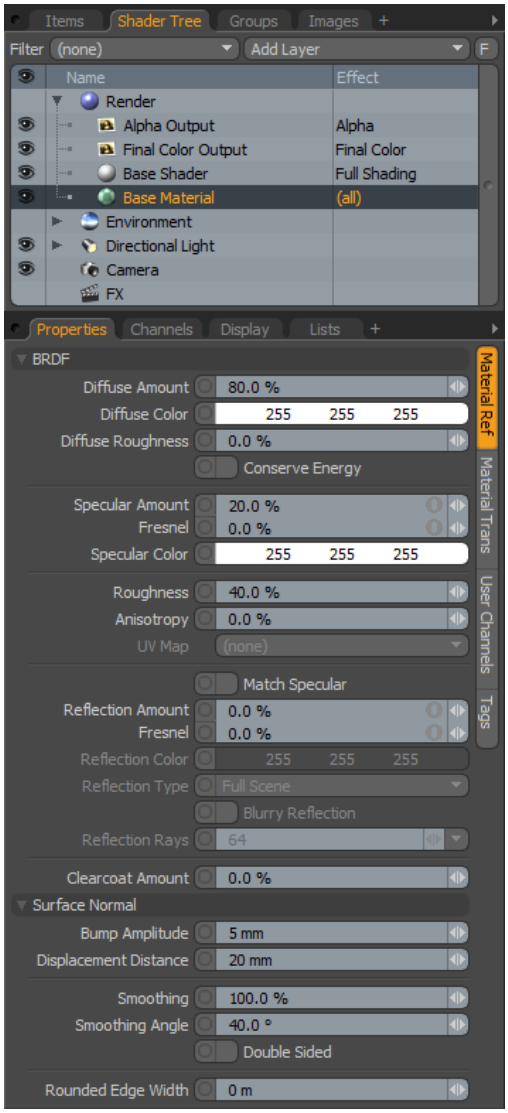
Apply to Sub-Group: Using undefined 'Material Groups' (Groups where no tags are specified) can be a convenient way to organize shader tree layers. With the addition of the 'Apply to Sub Group' option, they can also be an effective way to apply a setting to a number of material groups all at once. In the example below, there are 3 colored spheres, a box, and a ground plane. In the shader tree, one can see that the 3 colored spheres are grouped together into an undefined 'Material Group'. Directly below the group parent, a checkerboard texture has been added (next to the red arrow). Since it is above all the other layers in the tree and the group above it is undefined, it affects the entire scene; as illustrated by example one. By simply enabling the 'Apply to Sub Groups' option to the top 'Material Group' layer, the checkerboard effect is limited to those material groups that are descendants of the group as illustrated by example two. By setting the checkerboard pattern to 'Normal Multiply' as the blending mode in example three, one can quickly see how powerful the 'Apply to Sub Group option can be for quickly and easily applying a single texture layer to multiple surfaces.



TIP: As item layers and materials add up in the shader tree, it can quickly become a long list that gets confusing and tedious to edit. While there are many functions that speed and simplify the task of locating certain layers (such as the material selection mode), good Shader Tree organization is important. Users can select (highlight) multiple related layers and press the keyboard shortcut 'Ctrl+G' which will automatically group together the selected items. In this case the Material Group creates no mask for shading, but is simply used as an organizational tool. Renaming the layer will further help in identifying it in later sessions.

Material





Every new scene, by default, contains a single material layer called the 'Base Material' that effectively shades the entire scene until material tags are defined by the user, limiting materials and shading to specific surfaces. The 'Material' layer contains all the basic properties necessary to produce a rendered surface. There are many settings inside the 'Material' layer and as such, its property form is divided into two sub tabs 'Material Ref' (Reflected) and 'Material Trans' (Transmissive). The first tab contains all material attributes that impact how much light is reflected from the surface such as Diffuse, Specular and Mirror reflected light. The Transmissive section controls how light is transmitted through a material such as Transparency and Subsurface Scattering. In addition to these settings, nearly all material properties can be controlled via a texture -bitmap or procedural. Create a new texture layers and set the 'Effect' to the material property you wish to control. For more on adding layers and setting their 'Effect', please reference the Shader Tree page of the documentation.

Material Ref

BRDF--

Diffuse Amount: In the real world, all objects absorb certain wavelengths of light while reflecting others. The light that is reflected is what we see as the color of the object. The 'Diffuse Amount' acts as a multiplier to the 'Diffuse Color' setting, determining how much of that light is reflected from or absorbed by the surface. A setting of 100% would return the same value as the diffuse color value when lit by a pure white light at 100% intensity. A 'Diffuse Amount' of 50% would appear half as bright under the same light as it would only return half of the light bounced from the surface, and so on.

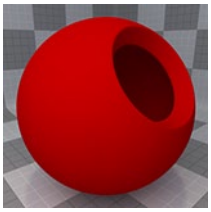
Diffuse Color: The 'Diffuse Color' setting defines the color of the surface when lit by pure white light sources be it direct or indirect. This is effectively the color of your object, representing the light that bounces from the surface. Diffuse colors can also be driven beyond traditional color space into high dynamic range values by applying Diffuse Amounts over 100%.

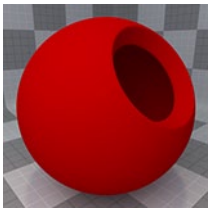
Diffuse Roughness: Extremely matte surfaces, such as stone, concrete or plaster, reflect light very differently than smooth or glossy surfaces, such as glass or plastic. This is because on a microscopic level the matte surface is actually quite rough, spreading the light outwards, making the surface appear more flat. The 'Diffuse Roughness' setting aims to simulate this microscopic roughness effect by modifying the the standard shading model, producing a more beleivable matte surface. A value of 0% disables the effect, while values above zero increasingly produce a rougher, more matte surface. Since 'Diffuse Rougness' simulates a lack of Specular, it is best applied when the 'Specular Amount' is 0%. -- (image examples below)

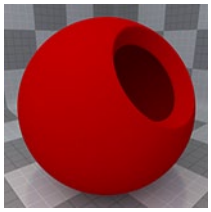
Conserve Energy: When enabled, the 'Conserve Energy' setting, in its most simple description, maintains a physically accurate blend of diffuse and specular light reflection over the surface of your object. As reflection amount increases at glancing angles (like when the 'Fresnel' settings are used), Conserve Energy realistically adjusts the amount of diffuse to maintain an accurate balance between the two producing extremely realistic results.

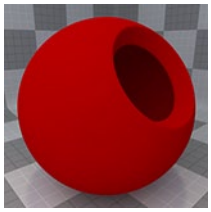
Conserve Energy affects three material channels. It changes the Specular Amount (which is derived itself only by direct light source items), it reduces the Diffuse Amount based on the Specular Amount and the Fresnel setting and it also reduces Diffuse Amount as the 'Reflection Amount' goes up. All of these things result in a more physically accurate and realistic result.

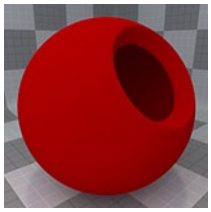
Diffuse Roughness


Diffuse Roughness 0%

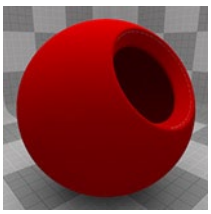

Diffuse Roughness 25%

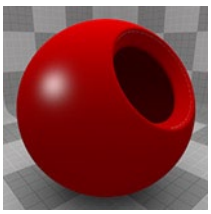

Diffuse Roughness 50%

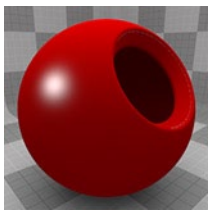

Diffuse Roughness 75%

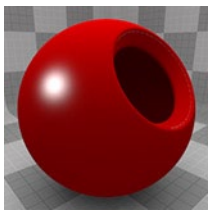

Diffuse Roughness 100%

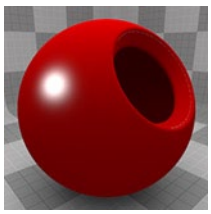
Specular Amount: In the real world specular highlights are the reflections of very bright light sources on a surface. How these reflections spread over a surface tell a lot about that surface, be it shiny or dull, smooth or rough. 3D programs simulate this reflection with these 5 settings- Specular Amount, Fresnel, Specular Color, Roughness and Anisotropy. Generated only from direct light sources in a scene, the 'Specular Amount' is multiplied with the 'Specular Color' to create the overall rendered result. Increasing the value will result in a more intense highlight while decreasing the value will mute the effect. When this value is 0% the Specular Color, Roughness and Anisotropy controls become disabled and specularity is not calculated.


Specular Amount 0%


Specular Amount 25%

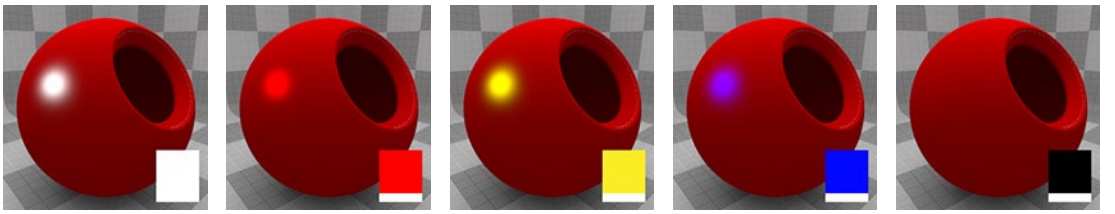

Specular Amount 50%


Specular Amount 75%


Specular Amount 100%

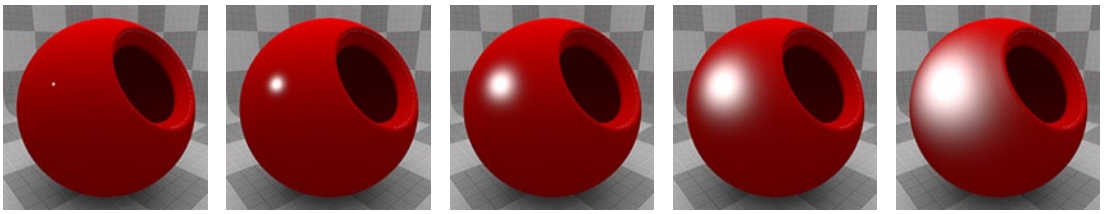
Fresnel: At glancing angles (those perpendicular to the camera), reflectance values increase. The fresnel setting realistically increases the specular amount at these glancing angles producing a physically correct surface.

Specular Color: The 'Specular Color' function defines an RGB value that tints the specular highlight.



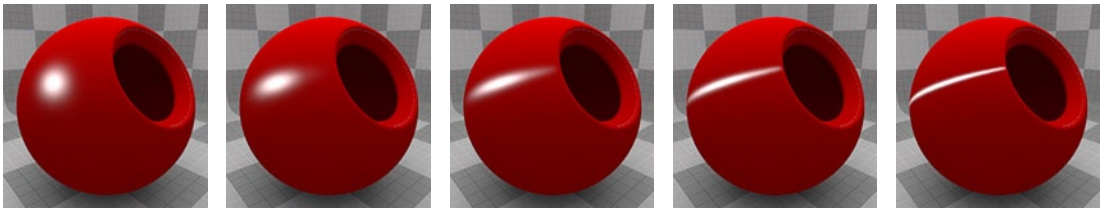
Specular Color | Specular Color | Specular Color | Specular Color | Specular Color

Roughness: Most surfaces are very different from each other --a rubber ball, a plastic cup, a terra cotta brick. These are all effectively smooth, but on a microscopic scale their surfaces are very different; having a direct affect on the way light reflects off of each one. The roughness setting simulates these microscopic differences making surfaces appear more dull or rough, or shiny and smooth. The 'Roughness' setting affects both the Specular highlight as well as ray traced and environmental reflections (reflection roughness only occurs when the Blurry Reflection option is active). Increasing this value will widen the reflection, creating a much broader specular highlight. When used with Blurry Reflections, a very high value will yield a very distorted reflection. In this case the number of Reflection Rays will most likely need to be increased to reduce graininess.



Roughness 0% | Roughness 25% | Roughness 50% | Roughness 75% | Roughness 100%

Anisotropy: Millions of microscopic scratches on a surface will distort highlights and reflections over their surface parallel to the directions of the scratches. The 'Anisotropy' value affects the highlights, simulating this effect. Anisotropy is controlled by a UV map or further controlled by a texture maps when the layers effect is set to the Anisotropy Direction. It can be caused by many man-made processes, such as machining metal and also appears in textiles where shiny threads spread the highlight over a surface.

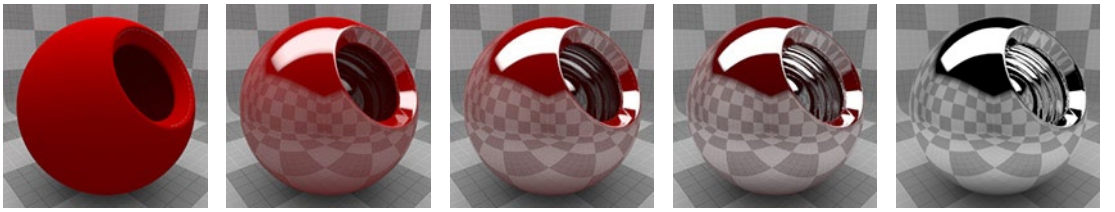


Anisotropy 0% Anisotropy 25% Anisotropy 50% Anisotropy 75% Anisotropy 100%

UV Map: The UV map selector allows the user to choose which map will determine the anisotropic direction. The "U" direction (horizontal) defines the direction of the *virtual scratches*.

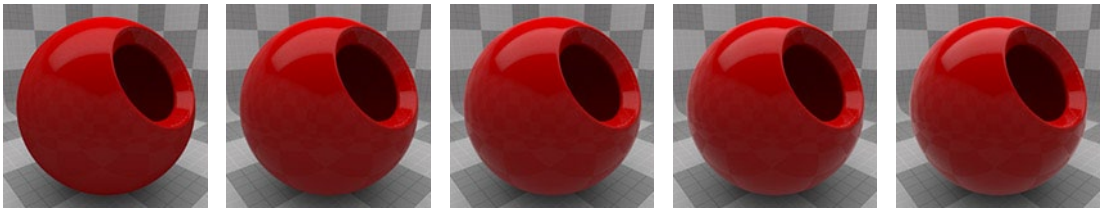
Match Specular: This toggle option, when enabled, will match the 'Reflection Amount' and 'Fresnel' values to the like settings under Specular Amount. A specular highlight is simply the reflection generated by modo's direct light sources, so naturally, to be correct, both would need identical values. With this checkbox, users will only need to adjust the values once. Additionally, this option takes into account any Texture layers with an effect defined as 'Specular Amount'/'Specular Color'. Users should note that when this option is *enabled*, all reflective shading is derived from the Specular channel so texture layers defined as 'Reflection Amount'/'Reflection Color' will be ignored.

Reflection Amount: The 'Reflection Amount' simply determines how reflective a surface is. At 100% the surface is effectively a perfect mirror whereas lower values will create a more muted reflection effect. Reflection is an additive effect, so reducing the diffuse amount is usually necessary to avoid overly bright surfaces. A surface is said to be 'physically correct' when Reflection amount and Diffuse Amount together equal 100%. Conserve energy will do this automatically and is enabled in the examples below.



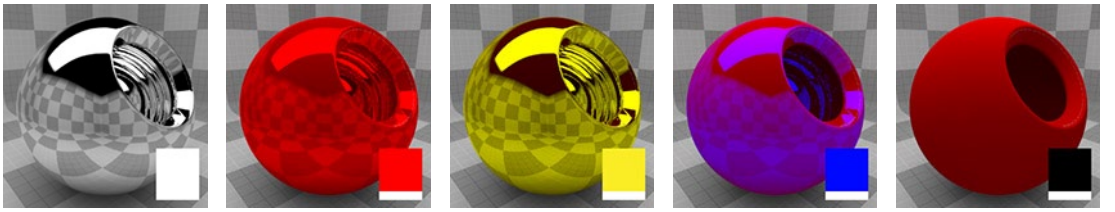
Reflection Amount 0% | Reflection Amount 25% | Reflection Amount 50% | Reflection Amount 75% | Reflection Amount 100%

Fresnel: At glancing angles (those perpendicular to the camera), reflectance values increase. The fresnel (pronounced fra-nell) setting realistically increases the reflection amount at these glancing angles producing a physically correct surface.



Fresnel 0% | Fresnel 25% | Fresnel 50% | Fresnel 75% | Fresnel 100%

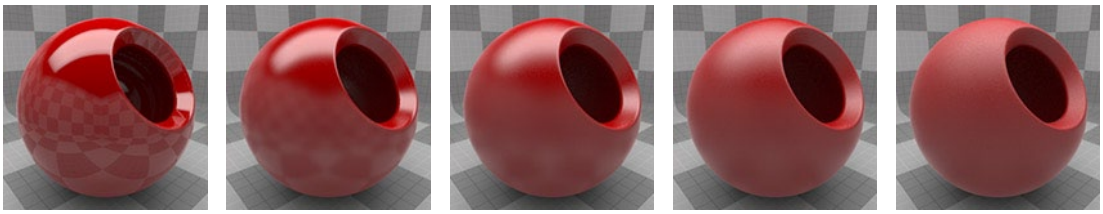
Reflection Color: The Reflection Color determines the tint on reflected light. By default the Color is white which reflects light without tint so that at 100% Reflection value, anything reflected in the surface would appear as if you were seeing it in a mirror. Setting the Reflection to something other than white will tint all reflected elements with that color.



Reflection Color | Reflection Color | Reflection Color | Reflection Color | Reflection Color

Reflection Type: The Reflection type popup allows the user to choose between 'Full Scene' reflection evaluation or 'Environment' only. This setting is effectively switching between ray traced reflections and an environment map technique. The Full Scene option will be more accurate at the expense of greater render times. Environment will utilize only the settings in the 'Environment' item.

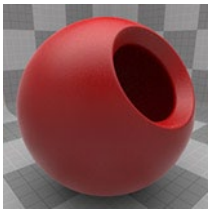
Blurry Reflection: When activated this feature will create a blurred reflection with the blur intensity being driven by the Roughness setting. It is important to understand that blurry reflections cause multiple reflection rays to be evaluated, each ray will be perturbed from the actual surface normal for every pixel. The higher the Roughness value the more reflection rays will be required to reduce noise in the reflection. If you consider that when the roughness value is increased the angle of each ray also increases you can imagine that the resulting sample from each ray can be quite different from one another. As such a higher number of rays is required to get an accurate average sample.

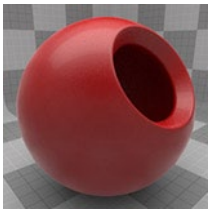


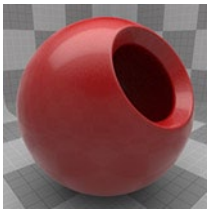
Blurry Reflection 0% | Blurry Reflection 25% | Blurry Reflection 50% | Blurry Reflection 75% | Blurry Reflection 100%

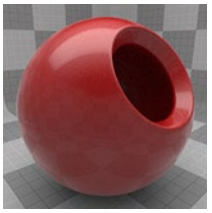
Reflection Rays: This value us only active when using Blurry Reflections. Increasing the Reflection Rays amount will improve the accuracy of the blurred reflection and reduce "grain" at the expense of computation time. Typically larger Roughness settings will require larger Reflection Ray settings as a wide spread of blur will need more samples to be accurate. If render times are a concern you can decrease the Roughness value rather than simply increasing the number of rays.

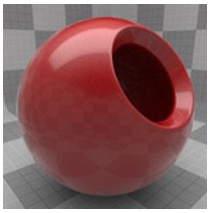
Clearcoat Amount: The 'Clearcoat' setting allows the user to easily add an additional unblurred and untinted reflective layer to a surface, simplifying the simulation of multiple layers of paint, especially automotive. Many metallic colors spread a wide colored highlight over the surface yet still exhibit a smooth reflective layer as a result of an additional transparent layer called a 'Clearcoat'. A setting of 0% effectively turns the clearcoat function off, increasing values add the reflective effect. Chief Scientist Allen Hastings says that in most cases that use Clearcoat, you'll just want to leave this setting at 100%.


Clearcoat Amount 0%


Clearcoat Amount 25%

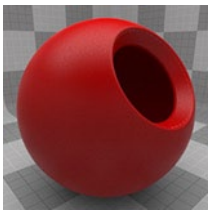

Clearcoat Amount 50%

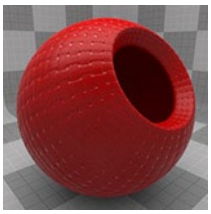

Clearcoat Amount 75%

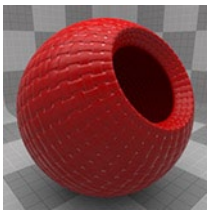

Clearcoat Amount 100%

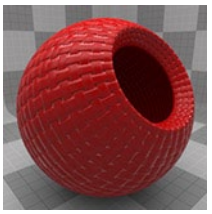
Surface Normal--

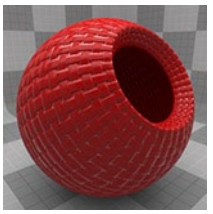
Bump Amplitude: Bump maps are a means of creating high frequency details on a surface that would be difficult, if not impossible to model. Bump mapping itself is a technique that modifies a rays direction when shading a surface giving the illusion that a surface has more detail than it actually does. The Bump Amplitude value allows a single control per material for modifying the amount of bump on a surface defined as a measurement. The size defined is relative to the surfaces actual size. The actual bump details come from the texture layer(s) driving the effect and the Bump Amplitude setting acts as a multiplier to create the final result. Please reference the 'Shader Tree' page of the documentation for information on adding layers and setting their effect.


Bump Amplitude 0mm


Bump Amplitude 2.5mm


Bump Amplitude 5mm

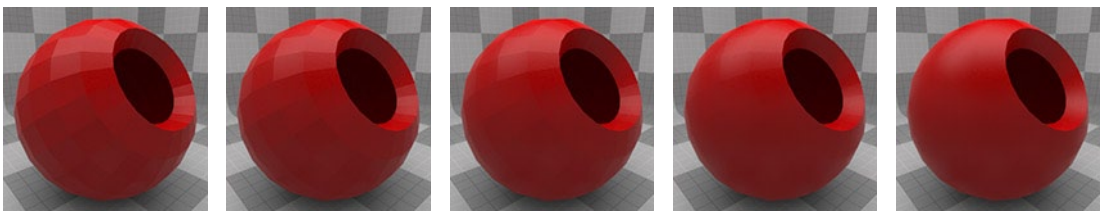

Bump Amplitude 7.5mm


Bump Amplitude 10mm

Displacement Distance: Similar to Bump maps, Displacement is a means to add detail to a surface that would be difficult, if not impossible to model otherwise. The maps themselves can be generated in modo using the sculpting toolset, or in an external application such as ZBrush™ or Mudbox™; even procedural textures are useful as Displacement sources. The 'Displacement Distance' setting determines the maximum range vertices can be created or moved away from the original mesh surface during micro-polygon tessellation. With a value of 2 cm, no vertex can be displaced more than 2 cm away from the original surface. Depending on the mesh's scale, this value may need to be increased significantly. Further, when textures are applied to the displacement channel, should the combined texture value become greater than 100%, the resulting surface will appear to be *clipped*.

The maximum displacement distance is different from other material parameters because the renderer needs to know it way in advance, before any textures are evaluated. At the start of rendering, all surfaces are just empty bounding boxes. The vertices and polygons of a particular surface are only created in the geometry cache when a ray pierces the bounding box of the surface. For displaced surfaces, these bounding boxes must be "padded" with the maximum displacement distance in order to avoid holes, since displacement can cause geometry that would normally appear in a later bucket to show up in an earlier one. For this reason, the actual distance a vertex moves can never exceed the displacement distance, therefore texture values outside the range of -100% to 100% are always clipped.

Smoothing: Smoothing is a shading technique to give the illusion that geometry has more detail than it actually does. Without smoothing the geometric faces would appear faceted or flat. The smoothing amount is set as a percentage rather than a simple toggle so that you have the flexibility to "fade" in the smoothing amount or even drive it above 100% or below 0% to create interesting effects. At values over 100% the smoothing effect will result in "pillowing" around each polygon.



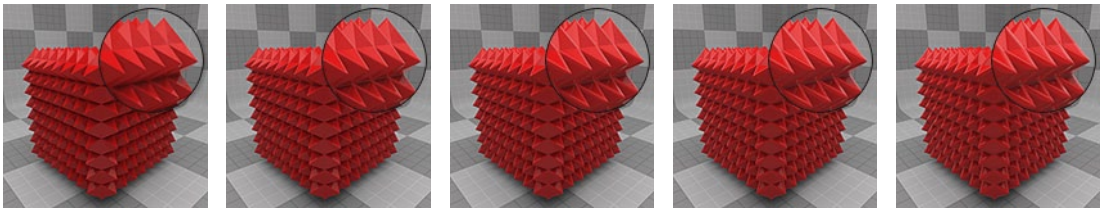
Smoothing 0% Smoothing 25% Smoothing 50% Smoothing 75% Smoothing 100%

Smoothing Angle: The Smoothing Angle determines which faces should receive smoothing. The smoothing angle sets a maximum angle tolerance between two polygons based on their *normals*. If the angle between the polygons is greater than the

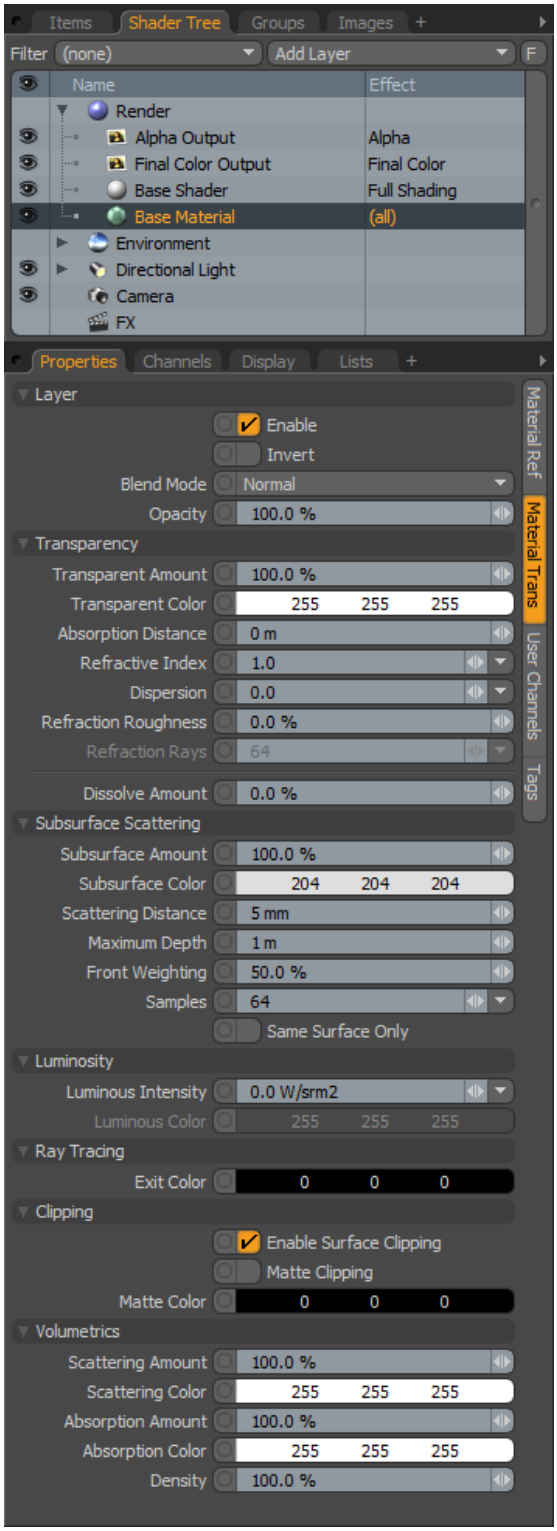
smoothing angle no smoothing will occur. This is useful so that hard edges, such as cubes, are not smoothed across. The default value of 40° is a good general case value.

Double Sided: The Double Sided toggle enables shading to occur from both sides of a polygon regardless of the normal direction. By default this value is off so that polygons that do not face the camera seem invisible as they only have one side. It should be noted that in some cases double sided polygons can cause render errors with ray tracing of shadows and other effects. The Double Sided flag should be used sparingly and never to correct poorly modeled meshes.

Rounded Edge Width: The 'Rounded Edge Width' option adjusts the rendered result of surface normals at a polygons edge to blend them with adjoining polygon normals, giving the impression of a small rounded edge between the two. To explain it more simply it shades polygons edges at render time to look as if a small rounded bevel has been applied. Users can define an Edge Width to determine the amount or 'roundness' but keep in mind it is merely a shading trick and wont round the edges of the actual geometry, nor change the objects silhouette in the rendered image, so it is best to keep set the length to just a few pixels in width in the final rendered image.



Edge Width 0mm | Edge Width 1mm | Edge Width 2mm | Edge Width 3mm | Edge Width 4mm



Material Trans

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Transparency--

Transparency Amount: The 'Transparency Amount' controls how transparent or opaque a surface is. 0% is completely opaque and disables the setting altogether, settings above 0% gradually increase how transparent the surface is ramping toward 100% which is completely transparent. -- (image examples on following pages)

Transparency Color: The 'Transparent Color' determines what color a ray is tinted as they penetrate the transparent surface. Since many transparent materials such as glass should have no diffuse amount whatsoever, the transparent color is used for tinting the material. At the default, pure white, a material that is 100% transparent will become completely invisible unless there are reflections and/or specular highlights. By changing the transparency color, the material will now appear (as modo will simply tint whatever the camera sees behind the surface). -- (image examples on following pages)

Absorption Distance: Colored transparent surfaces don't often exhibit even coloration. Thin areas will appear colorless while thicker areas will tint with color. The 'Absorption Distance' controls this effect, determining how far a ray must travel to get 100% of the Transparency Color. The falloff is determined by Beer's Law, just as in the real world. -- (image examples on following pages)

Refractive Index: The Refractive Index determines how far the a light ray is bent when it travels through a material. The higher the value the more distorted the refracted items will become. For instance, a value of 1.0 is equivalent to light traveling through a vacuum and will not bend the rays at all. Increasing the value to 1.333 would approximate the amount of distortion that occurs when light travels through water. This setting requires the surface have some amount of transparency active. -- (image examples on following pages)

--REFRACTIVE INDEX TABLE--

These settings may be applied to the Refractive Index setting in the materials panel.

Gemstones

Agate 1.544 - 1.553
Alexandrite 1.746 - 1.755
Almandine 1.75 - 1.83
Amber 1.539 - 1.545
Amethyst 1.532 - 1.554
Ammolite 1.52 -1.68
Andalusite 1.629 - 1.650
Apatite 1.632 - 1.42
Aquamarine 1.567-1.590
Axenite 1.674 - 1.704
Beryl 1.57 - 1.60
Beryl, Red 1.570 - 1.598
Chalcedony 1.544 - 1.553
Chrome Tourmaline, 1.61 - 1.64
Citrine 1.532 - 1.554
Clinohumite 1.625 - 1.675
Coral 1.486 - 1.658
Crystal 2.000
Crysoberyl, Catseye 1.746 - 1.755
Danburite 1.627 - 1.641

Diamond 2.417

Emerald 1.560 - 1.605

Emerald Catseye 1.560 - 1.605

Flourite 1.434

Garnet, Grossular 1.72 - 1.80

Garnet, Andradite 1.88 - 1.94

Garnet, Demantiod 1.880 - 1.9

Garnet, Mandarin 1.790 - 1.8

Garnet, Pyrope 1.73 - 1.76

Garnet, Rhodolite 1.740 - 1.770

Garnet, Tsavorite 1.739 - 1.744

Garnet, Uvarovite 1.74 - 1.87

Hauyn 1.490 - 1.505

Iolite 1.522 - 1.578

Jade, Jadeite 1.64 - 1.667

Jade, Nephrite 1.600 - 1.641

Jet 1.660

Kunzite 1.660 - 1.676

Labradorite 1.560 - 1.572

Lapis Lazuli 1.50 - 1.55

Moonstone 1.518 - 1.526

Morganite 1.585 - 1.594

Obsidian 1.50

Opal, Black 1.440 - 1.460

Opal, Fire 1.430 - 1.460

Opal, White 1.440 - 1.460

Oregon Sunstone 1.560 - 1.572

Padparadja 1.760 - 1.773

Pearl 1.53 - 1.69

Peridot 1.635 - 1.690

Quartz 1.544 - 1.553

Ruby 1.757 - 1.779

Sapphire 1.757 - 1.779

Sapphire, Star 1.760 - 1.773

Spessarite 1.79 - 1.81

Spinel 1.712 - 1.717

Spinel, Blue 1.712 - 1.747

Spinel, Red 1.708 - 1.735

Star Ruby 1.76 - 1.773

Tanzanite 1.690-1.7

Tanzanite 1.692 - 1.700

Topaz 1.607 - 1.627

Topaz, Imperial 1.605-1.640

Tourmaline 1.603 - 1.655

Tourmaline, Blue 1.61 - 1.64

Tourmaline, Catseye 1.61 - 1.64

Tourmaline, Green 1.61 - 1.64

Tourmaline, Paraiba 1.61 - 1.65

Tourmaline, Red 1.61 - 1.64

Zircon 1.777 - 1.987

Zirconia, Cubic 2.173 - 2.21

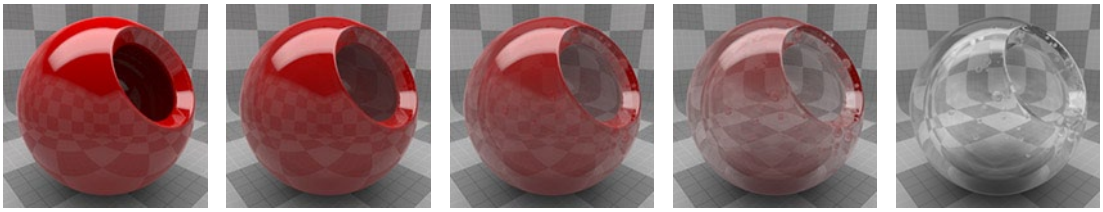
Common Liquids

Acetone 1.36
Alcohol, Ethyl (grain) 1.36
Alcohol, Methyl (wood) 1.329
Beer 1.345
Carbonated Beverages 1.34 - 1.356
Fruit Juice
Chlorine (liq) 1.385
Cranberry Juice (25%) 1.351
Glycerin 1.473
Honey, 13% water content 1.504
Honey, 17% water content 1.494
Honey, 21% water content 1.484
Ice 1.309
Milk 1.35
Oil, Clove 1.535
Oil, Lemon 1.481
Oil, Neroli 1.482
Oil, Orange 1.473
Oil, Safflower 1.466
Oil, vegetable (50 C) 1.47
Oil of Wintergreen 1.536
Rum, White 1.361
Shampoo 1.362
Sugar Solution 30% 1.38
Sugar Solution 80% 1.49
Turpentine 1.472
Vodka 1.363
Water (0 C) 1.33346
Water (100 C) 1.31766
Water (20 C) 1.33283
Whisky 1.356

Common Transparent Materials

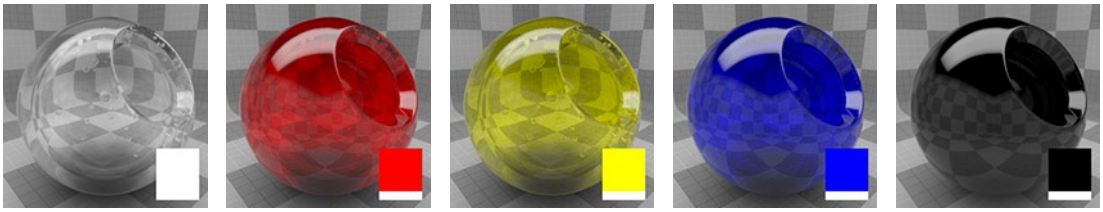
Eye, Aqueous humor 1.33
Eye, Cornea 1.38
Eye, Lens 1.41
Eye, Vitreous humor 1.34
Glass, Arsenic Trisulfide 2.04
Glass, Crown (common) 1.52
Glass, Flint, 29% lead 1.569
Glass, Flint, 55% lead 1.669
Glass, Flint, 71% lead 1.805
Glass, Fused Silica 1.459
Glass, Pyrex 1.474
Lucite 1.495
Nylon 1.53
Obsidian 1.50
Plastic 1.460 - 1.55
Plexiglas 1.488
Salt 1.516

Transparency Amount



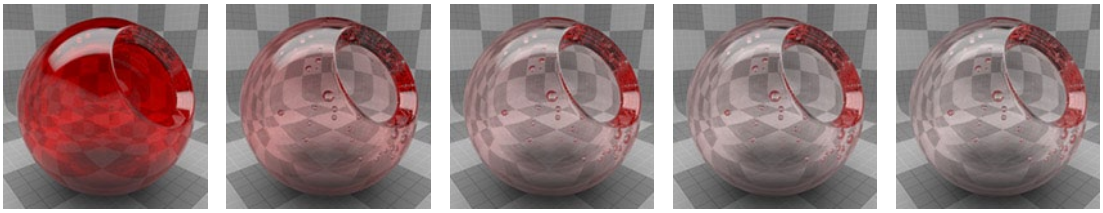
Trans. Amount 0% | Trans. Amount 25% | Trans. Amount 50% | Trans. Amount 75% | Trans. Amount 100%

Transparency Color



Transparency Color | Transparency Color | Transparency Color | Transparency Color | Transparency Color

Absorption Distance

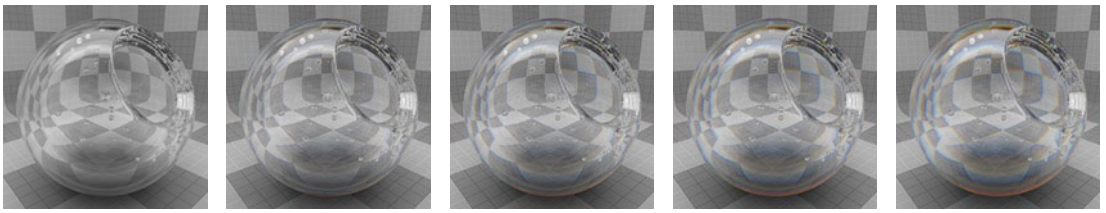


Absorption Dist. 100mm | Absorption Dist. 200mm | Absorption Dist. 300mm | Absorption Dist. 400mm | Absorption Dist. 500mm

Refractive Index

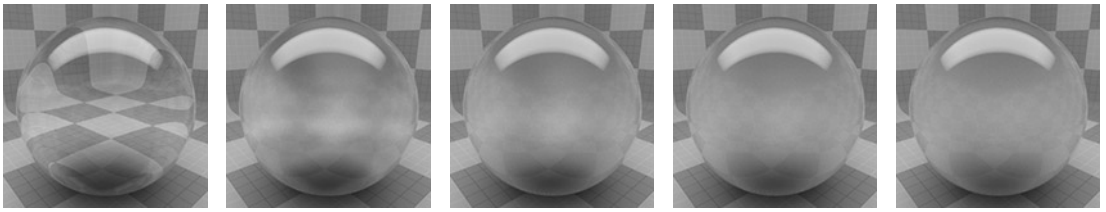
Refractive Index 1.0 | Refractive Index 1.5 | Refractive Index 2.0 | Refractive Index 2.5 | Refractive Index 3.0

Dispersion: Dispersion is the difference of Refractive Index between various wavelengths of lights. Pure white light in the real world contains all colors of the spectrum, as it passes through solid objects, the refraction of different wavelengths varies creating a rainbow effect. You've likely witnessed this when shining a bright light through a prism. This effect in gemstones is usually referred to as 'fire'. In modo, it is calculated as the difference between a high refractive index (at the violet end of the spectrum) and low refractive index (at the red end of the spectrum). A typical dispersion value for glass would be around .02 whereas diamond would be .044. Negative dispersion values will invert to the usual curve.



Dispersion 0 | Dispersion 0.025 | Dispersion 0.05 | Dispersion 0.075 | Dispersion 0.1

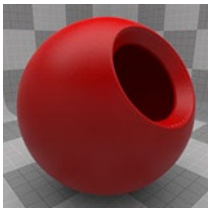
Refraction Roughness: Some transparent surfaces are better described as translucent or cloudy -- light passes through the surface, but is diffused in such a way as to make opposing items blurry or even completely indistinguishable. Frosted glass exhibits this effect, items close to the surface are clear, but as they recede from view, edges and details soften increasingly. The 'Refraction Roughness' calculates this effect by firing a number of jittered rays for each refraction ray and averages the result. A value of 0% disables the effect, values above 0% increase the blurriness of items viewed through the surface.

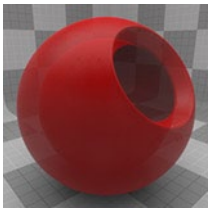


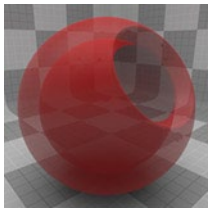
Ref. Roughness 0% | Ref. Roughness 5% | Ref. Roughness 10% | Ref. Roughness 15% | Ref. Roughness 20%

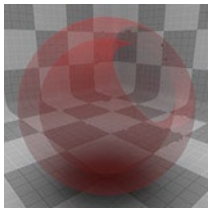
Refraction Rays: The 'Refraction Rays' setting specifies the number of rays modo uses to calculate the Refraction Roughness setting. As the Refraction Roughness amount grows, you will need to increase the number of rays necessary to produce a smooth effect, though at the expense of longer render times.

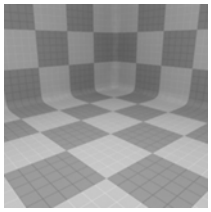
Dissolve Amount: The 'Dissolve Amount' will allow one to fade an object completely from view. A setting of 100% would make the surface completely invisible. The Dissolve amount is different from transparency as it is a convenient way to fade all aspects of a surface simultaneously.


Dissolve Amount 0%


Dissolve Amount 25%


Dissolve Amount 50%


Dissolve Amount 75%


Dissolve Amount 100%

Subsurface Scattering--

Subsurface scattering is the effect of light bouncing around inside a surface and "tinting" prior to exiting. This is often most obvious in materials like marble but is essential to creating realistic wax or fleshy materials. Subsurface scattering also helps simulated liquids such as milk to appear more natural. In humans "SSS" is most often witnessed when thin areas on the ears seem to turn red from back light as the light is being tinted by the tissues underneath the skin itself.

Subsurface Amount: Subsurface scattering contributes to the diffuse shading of a material. The 'Subsurface Amount' is a percentage value that determines the contribution of diffuse shading made by subsurface scattering. It is a good idea to keep this in mind in order to balance the percentages between Diffuse Color and Subsurface Amount depending on which effect you prefer for final shading. Also, combining high Diffuse Amounts and high Subsurface Amounts can yield unnatural shading results. Hey, if that's what you're into, there is nothing wrong with that.

Subsurface Amount 0%

Subsurface Amount 25%

Subsurface Amount 50%

Subsurface Amount 75%

Subsurface Amount 100%

Subsurface Color: The Subsurface Color is the color value that will be used to tint the light as it is attenuated through the material. This setting and the Scattering Distance work together to determine how far the light must travel to be tinted.

Subsurface Color Subsurface Color Subsurface Color Subsurface Color Subsurface Color

Scattering Distance: The 'Scattering Distance' setting determines how far light must travel through the surface prior to exiting in order to be fully tinted by the Subsurface Color. If the light travels beyond the scattering distance it will be further attenuated according to the original Subsurface color.

Maximum Depth: The 'Maximum Depth' option is a fast way to approximates an opaque solid core within the Subsurface layer (example - a bone inside a finger) by clipping any scattering samples that reach beyond 'Maximum Depth' value specified. Be aware that this may lead to a slightly darker scattering because the clipped samples exclusion from the solution, but an increase in the number of total 'Samples' can help to reduce or eliminate this effect. A setting of zero disabled the option.

Front Weighting: The 'Front Weighting' value sets a bias for the SSS rays. At 50%, SSS rays will scatter evenly forward and backward though a surface. Higher values bias the scattered rays toward the light source that emitted them creating a more waxy/flat look to SSS, while values lower than 50% bias the scattered rays away from the light source darkening the overall SSS effect.

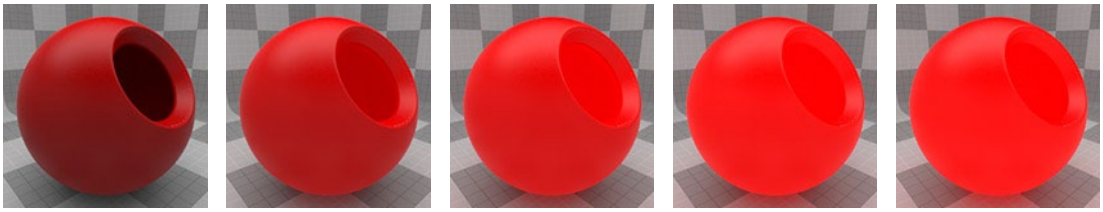
Front Weighting 0% Front Weighting 25% Front Weighting 50% Front Weighting 75% Front Weighting 100%

Samples: The Subsurface 'Samples' controls the quality of the subsurface shading. Increasing this value will improve the quality of the effect at the expense of render speed. The default setting is 64.

Same Surface Only: When the 'Same Surface Only' toggle is enabled, modo's subsurface scattering works as it previously did, by limiting the SSS sample cache values to affect the shading to a single surface. This was to prevent certain artifacts that would occur with very thin surfaces or intersections with non-SSS surfaces. This optimization however limits adjoining SSS surfaces from sharing values resulting in a slight dark border between them; in which case, 'Same Surface Only' should be disabled.

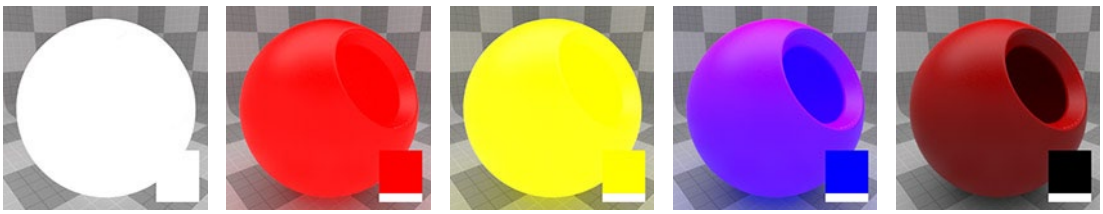
Luminosity--

Luminous Intensity: The Luminous Intensity is multiplier that drives the brightness of the Luminous color as it is emitted from the surface. This is different than Diffuse Amount in that Diffuse requires some amount of light to hit the surface whereas Luminous comes from the surface itself. It is accurate to think of Luminous surfaces as light sources - particularly when using Global Illumination. The Luminous Intensity is multiplied to the Luminous Color. By default the Value is zero which results in the Luminous color control being disabled and as such it is not calculated.



Luminous Intensity 0 | Luminous Intensity 0.25 | Luminous Intensity 0.5 | Luminous Intensity 0.75 | Luminous Intensity 1

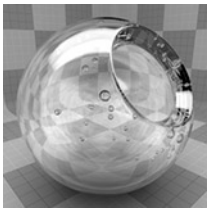
Luminous Color: The 'Luminous Color' setting allows the user to specify a color value for light emitted from a surface as specified by the 'Luminous Intensity' setting.



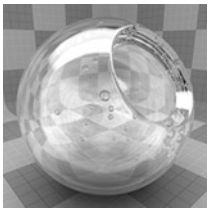
Luminous Color | Luminous Color | Luminous Color | Luminous Color | Luminous Color

Ray Tracing--

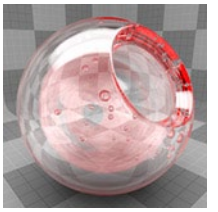
Exit Color: Originally when a series of refraction rays hit the maximum refraction depth, modo would return black as the final value, so in scenes with low refraction depth values would exhibit lots of dark areas. The 'Exit Color' allows you to now specify the color modo returns once it has reached its ray threshold. The color is still attenuated with the initial ray bounces --colors, absorption and such, however this setting can reduce a lot of the dark areas present with lower refraction depth values. In the first example below, there is a good amount of black revealed because of the low refraction depth. In the second image, the exit color was changed to a medium gray, simulating the background color resulting in a more accurate representation. Finally, the last image changes the exit color to red make obvious the effects of the exit color setting.



Exit Color (Default)



Exit Color Gray



Exit Color Red

Clipping--

Enable Surface Clipping: *Enabled* by default, Surface Clipping refers to the removal of part of the scene as defined by the 'Clipping Plane' settings of the Camera Item. Users can define a distance where all (or selected) surfaces will be clipped, removing them from the rendered image while leaving the scene geometry intact. In order for Surface Clipping to work, it must first be *enabled* in the Camera Item, under the 'Clipping' attributes. There users can define the distance value as well, determining the distance away from the camera where the scene will clip in the rendered image. Users can disable the clipping effect on a per-material basis by *disabling* this option. Users should note that a similar effect can also be obtained through the use of 'Render Booleans' that is not camera view dependant.

Matte Clipping: 'Matte Clipping' provides users a means to colorize interior areas of clipped geometry producing a filled area of color, the color being determined by the 'Matte Color' value.

Matte Color: Determines the color of clipped interior areas when Clipping is *enabled* in both the Material clipping attributes *and* in the Camera Item.

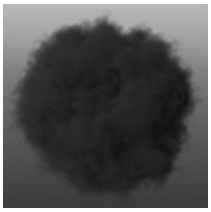
Volumetric--

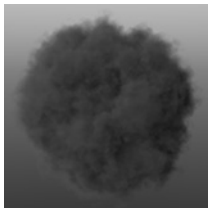
When light shines on particular matter suspended in the air, such as water vapor, smoke, dust or even pollution, the particles become visible, this is known in computer graphics as a *volumetric effect*. Two things then happen to the light; part of it is bounced back in all directions (this is called *scattering*) and part of it is absorbed (this is called *absorption*). The density parameter is the particle density of the volume, so the higher the density the more apparent *particles* there will be. When the density is increased, the scattering and the *absorption* will increase as well, so your volume becomes both more opaque and more luminous. However, when density is increased, the volume becomes more opaque, and the light scattered inside is more attenuated so the net result in terms of overall luminosity is hard to predict, the final result depends on the balance of scattering vs absorption vs density. In simple terms one can say that scattering is similar to luminosity and absorption is similar to opacity. This is however more subtle than that since the overall luminosity also depends on how much absorption happens inside the volume.

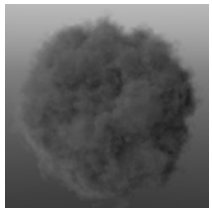
The following settings only apply to the 'Volume' item and otherwise will have no affect on the surface. Likewise, the other surfacing attributes, such as Diffuse Color and Reflection Amount will have no affect on a Volumetric object.

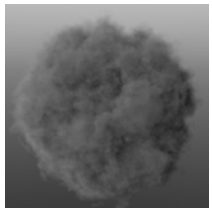
Scattering Amount: Scattering defines the amount of light that is reflected from the volume. This can also be described as how luminous the volume is, where a cloud is puffy and white, it is because it reflects back a lot of the light it receives (also absorbing very little). Conversely, a dense smoke cloud from an oil fire would reflect back very little light looking very dark and ominous. A value of 0% would reflect back no light received from the source, up toward 100% (the default value) where the volume reflects back all of the light it receives.

Scattering Amt. 0%

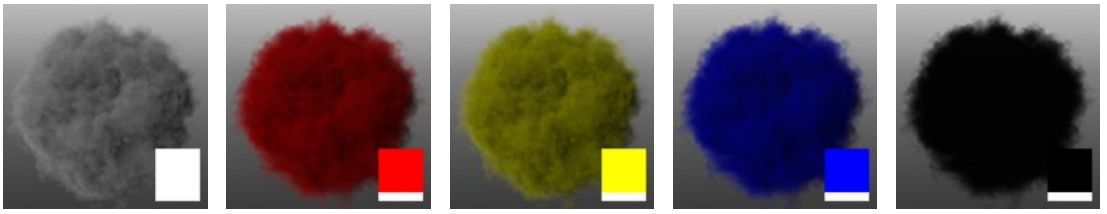

Scattering Amt. 25%


Scattering Amt. 50%


Scattering Amt. 75%

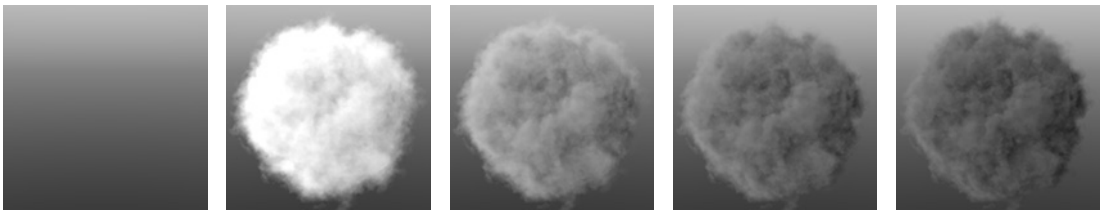

Scattering Amt. 100%

Scattering Color: The 'Scattering Color' represents the apparent color of the volume, representing the light that is reflected from the particulate matter.



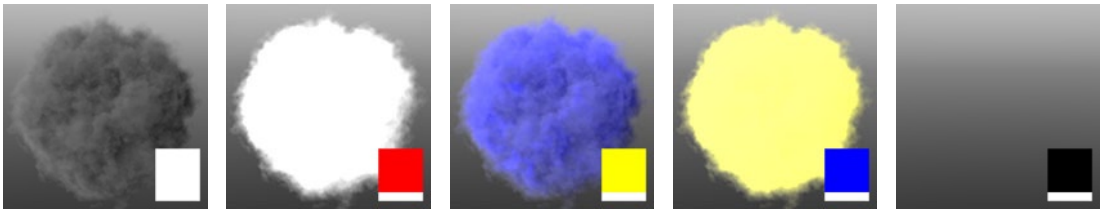
Scattering Color Scattering Color Scattering Color Scattering Color Scattering Color

Absorption Amount: The 'Absorption Amount' determines how much light is absorbed into the particular matter. If low amounts of light are absorbed, then more light is reflected back producing a brighter, more luminous volume, where when more light is absorbed, then the volume will appear darker and more powdery/dust like.



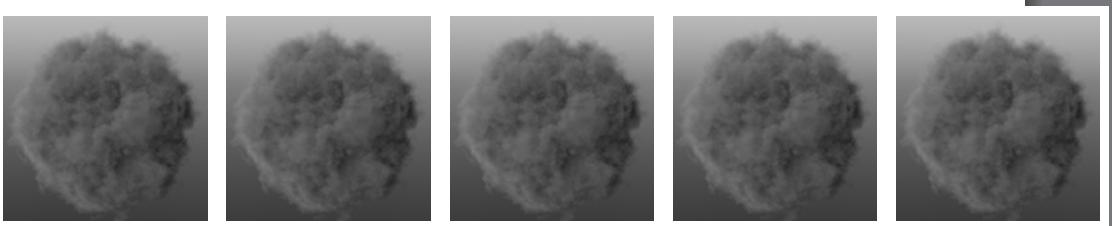
Absorption Amt. 0% Absorption Amt. 25% Absorption Amt. 50% Absorption Amt. 75% Absorption Amt. 100%

Absorption Color: The 'Absorption Color' determines the color that is absorbed into the volume...



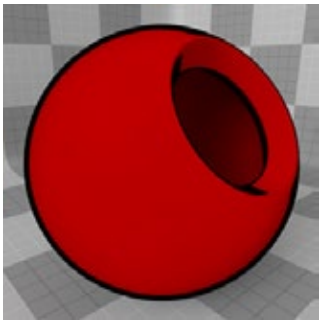
Absorption Color Absorption Color Absorption Color Absorption Color Absorption Color

Density: The 'Density' value determines the thickness or opacity of the volume, controlling the apparent amount of particular matter that is visible.



Density 0% Density 25% Density 50% Density 75% Density 100%

Cel Edges

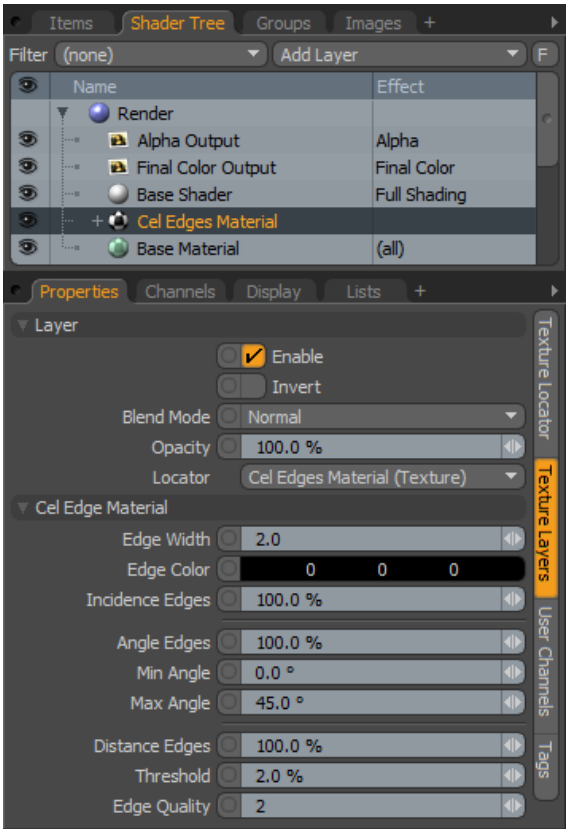


In the early days of animation, all art was hand drawn on transparent acetate called a 'Cel' and painted with flat solid colors. This produced a unique look that is still highly desirable even today. The 'Cel Edges' material attempts to mimic the hand drawn look of the outlines of these cartoons, creating colored strokes that follow surface contours and edges. When combined with the 'Cel Material' item, a low-fidelity *cartoony* look can be obtained. Users can add the material to any surface using the 'Add Layer' function of the Shader Tree, found in the menu under "Custom Materials > Cel Edges Material".

For information regarding adding and working with Shader Tree Items Layers, please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.



Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (way the texture is applied) to the surface. The 'Locator' option sets that association. Users can choose alternate locators, however, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Cel Edge Material--

Edge Width: This value determines the overall width of the edge stroke. Defined as a pixel value, users must be careful to ensure the adjust this value when changing frame resolution because a two pixel contour will appear half as thick when the render resolution is doubled.

Edge Color: This value determines the color of the actual stroke.

Incidence Edges: This value determines the amount that the 'Incidence Edges will contribute to the final Cel edge. Incidence edges are those that appear at the steepest glancing angles perpendicular to the camera. This is useful in obtaining a more organic thick and thin stroke as the incidence angle will change depending on the surfaces position relative to the camera.

Angle Edges: This value determines amount that interior contour edges within the Min and Max Angle values will contribute to the final Cel Edge.

Min/Max Angle: These two values determine the angle spread between opposing polygonal surfaces where a stroke will be rendered. Typically only an adjustment to the 'Max' value will be necessary. Lowering the value will produce more contour edges and raising it will produce progressively less.

Distance Edges: The 'Distance Edges' function detects which interior areas of a surface are considered edges, when surface contours overlap. Edges are shaded when two adjacent sampling points are greater than the given threshold.

Threshold: TheThreshold value is used to determine the distance between adjacent sampling points for 'Distance Edges'. Increasing it will produce fewer interior edges for a surface, while decreasing it will produce more edges.

Edge Quality: This value determines the overall quality of the Cel edges. Increasing this value will produce cleaner edges as more edge determining rays will be fired, but this will be at the expense of longer render times.

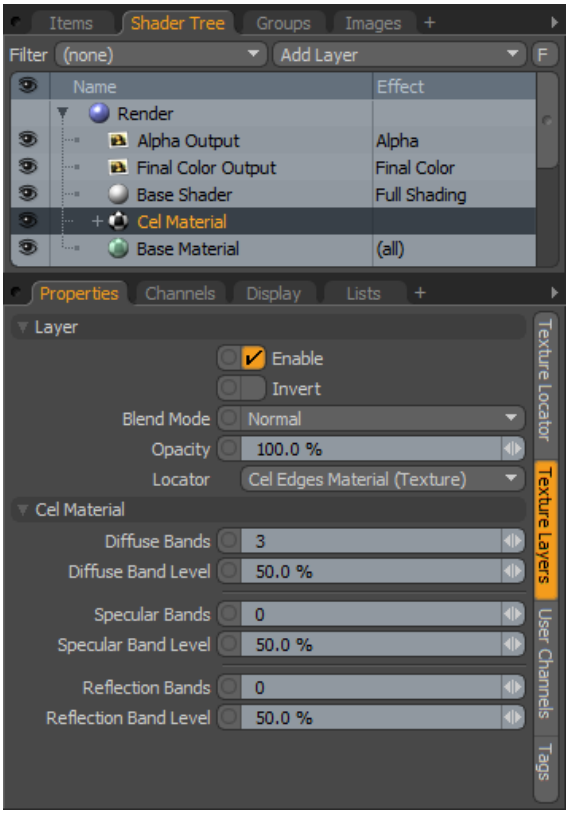
Cel Material

In the early days of animation, all art was hand drawn on transparent acetate called a 'Cel' and painted with flat solid colors. This produced a unique look that is still highly desirable even today. The 'Cel Material' attempts to mimic the flat posterized shading of these cartoons, creating flat colored bands that follow surface contours and edges producing a unique stylized surface that looks more two-dimensional. When combined with the 'Cel Edges Material' item, a low-fidelity *cartoony* look can be obtained. Users can add the material to any surface using the 'Add Layer' function of the Shader Tree, found in the menu under "Custom Materials > Cel Material".

For information regarding adding and working with Shader Tree Items Layers, please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (*disabled*), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.



Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (way the texture is applied) to the surface. The 'Locator' option sets that association. Users can choose alternate locators, however, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Cel Material--

Diffuse Bands: Determines the number of bands or stepped transitions representing the Diffuse shading across the surface. A value of '0' disables the cel shading effect.

Diffuse Band Level: This is an offset value, making it possible to adjust the overall brightness of the Diffuse Bands.

Specular Bands: Determines the number of bands or stepped transitions representing the Specular shading across the surface. A value of '0' disables the Cel shading effect for specular.

Specular Band Level: This is an offset value, making it possible to adjust the overall brightness of the Specular Bands.

Reflection Bands: Determines the number of bands or stepped transitions representing the Reflection shading across the surface. A value of '0' disables the Cel shading effect for reflections.

Reflection Band Level: This is an offset value, making it possible to adjust the overall brightness of the Reflection Bands.

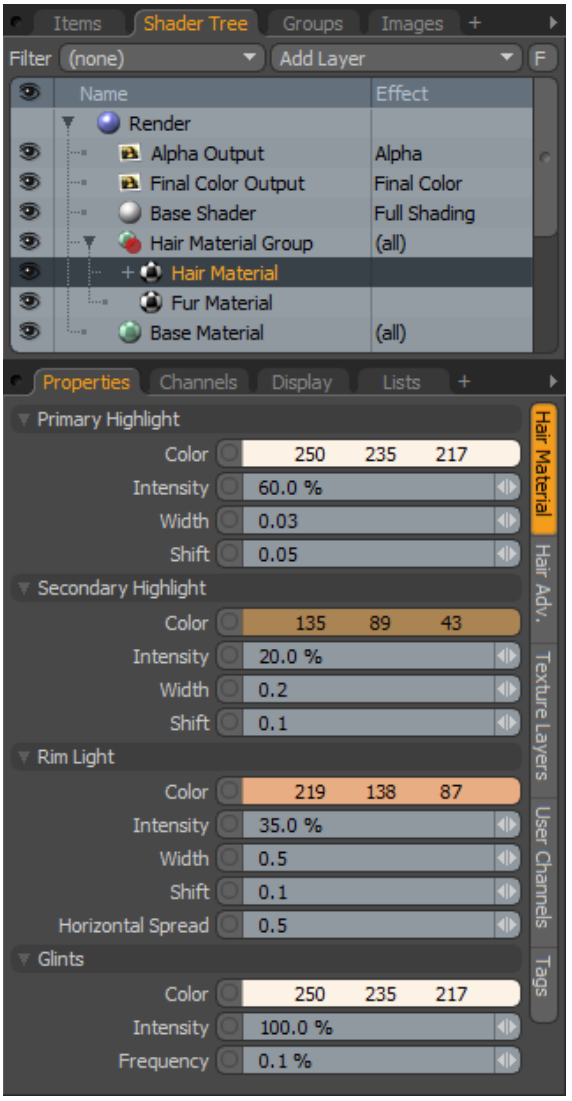
Hair Material

The 'Hair Material' is designed to replace the default BRDF shading calculations of the standard Material item when applied to fur. Fur, or hair as the case may be, is anisotropic, meaning its shading is directionally dependant and requires a different type of shading model to look correct. On a microscopic level, the surface of hair is quite complex and this affects how light reflects off of it and how it scatters through it, producing a unique look that is difficult to simulate without a dedicated hair shader.

modo has implemented a hair shading system uses a special model for quickly approximating multiple scattering layers, which is way too expensive to compute otherwise with brute force. The model is called *Dual Scattering Approximation*, and is based off this technical paper. The basic idea is that multiple scattering is essentially broken into two parts: global scattering -lighting of the overall volume and local scattering which is the shading on individual fibers that are visible.

The 'Hair Material' item is added to the Shader Tree using the 'Add Layer' function -"Add Layer > Custom Materials > Hair Material". For more information on adding and manipulating Shader Tree layers, please reference that page of the documentation. For

proper results, the placement of the layer is important, as the Hair Shader itself must reside in the same Group Mask as the 'Fur Material' item it is shading. Once a 'Hair Material' is added, its settings will supercede those of any Material item rendering it unnecessary. Once positioned, selecting the 'Hair Shader ' item will reveal its attributes in the 'Properties' viewport where its settings may be adjusted to fine tune the look of the rendered hair.



Primary Highlight--

The 'Primary Highlight' represents the main light colored highlight that is reflected off the surface of the hair, the more apparent this highlight is, the more *sheen* the hair will have.

Color: R,G and B values that determine the color of the 'Primary Highlight'.

Intensity: The 'Intensity' option acts as a multiplier controlling the strength of the 'Primary Highlight'.

Width: This option controls the size of the highlight along the hairs length.

Shift: This option controls the position of the highlight along the hairs length, basically simulating the tiny shingle-like cuticles of the hair's surface that modify the angle of the reflected highlight.

Secondary Highlight--

The 'Secondary Highlight' represents light that is refracted into the hair strand, hitting the back surface and reflecting back towards the front. Realistically, the magnitude of the longitudinal shift for the 'Secondary Highlight' should be larger than for the 'Primary

Highlight' because the light path will be more disrupted with additional refractions and reflections.

Color: R,G and B values that determine the color of the 'Secondary Highlight' shading.

Intensity: The 'Intensity' option acts as a multiplier controlling the strength of the 'Secondary Highlight'.

Width: This option controls the size of the highlight along the hairs length.

Shift: This option controls the position of the highlight along the hairs length, basically simulating the tiny shingle-like cuticles of the hair's surface that modify the angle of the reflected highlight.

Rim Light--

The 'Rim Light' represents light that is transmitted through the hair similar to the effects of subsurface scattering.

Color: R,G and B values that determine the color of the 'Rim Light' shading.

Intensity: The 'Intensity' option acts as a multiplier controlling the strength of the 'Rim Light'.

Width: This option controls the size of the highlight along the hairs length.

Shift: This option controls the position of the 'Rim Light along the hairs length, basically simulating the tiny shingle-like cuticles of the hair's surface that modify the angle of the reflected highlight.

Horizontal Spread: Controls how broad or tight the Rim Light highlight is, similar to a specular highlight in a sense.

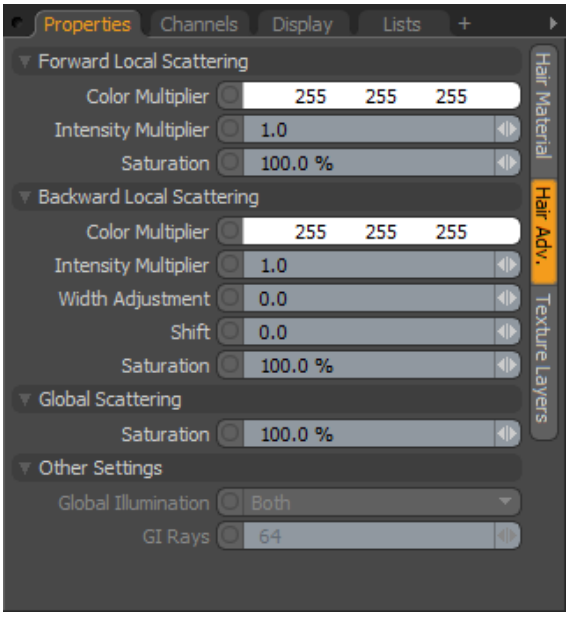
Glints--

Glints are representative of tiny caustics refracted through the hair, basically this is how hair looks shimmery.

Color: R,G and B values that determine the color of the 'Glints' shading.

Intensity: The 'Intensity' option acts as a multiplier controlling the strength of the 'Glints'.

Frequency: Determines how often a glint is shaded, controlling the overall density of 'Glints'.



Hair Shader Advanced

Forward Local Scattering--

This controls how multiple fibers in a small area will scatter light forward, or away from the source of light. It contributes to the appearance of rim lighting, but it is wider, darker, and the color is more saturated than direct rim lighting. This is because the light is scattering multiple times, making it spread out more.

Color Multiplier: This setting allows the user to adjust the color of the forward local scattering, by multiplying the color values. So when each color channel is set to 1.0 (when the color is white) the forward local scattering color is only dependent on the colors in the hair material settings overall.

Intensity Multiplier: This allows the user to make the forward local scattering brighter or darker, by multiplying the value.

Saturation: This allows the user to control the color intensity of the Forward Local Scattering, making it more or less colorful.

Backward Local Scattering--

The backward scattering adjustments represent areas where the hair is front lit, but the specific hair strand is in shadow.

Color Multiplier: This setting allows the user to adjust the color of the forward local scattering, by multiplying the color values. So when each color channel is set to 1.0 (when the color is white) the forward local scattering color is only dependent on the colors in the hair material settings overall.

Intensity Multiplier: This allows the user to make the forward local scattering brighter or darker, by multiplying the value.

Width Adjustment: This option controls the size of the highlight along the hairs length, relative to the initial width, adjusting it wider or thinner.

Shift: This option controls the position of the highlight along the hairs length, relative to the initial 'Shift value, moving the color of the backward scattering along the length of the hair strand.

Saturation: This allows the user to control the color intensity of the Backward Local Scattering, making it more or less colorful.

Global Scattering--

Global scattering controls the way in which light transmits, scatters, and spreads through a whole volume of hair. It is brightest and least spread out closest to the light, and gets dimmer, more spread out, and more colorful as the light transmits through more strands of hair. It will affect both forward and backward multiple scattering.

Saturation: This makes the global scattering less colorful. In particular, if the rim lighting setting of a hair is too bright or wide, in can cause the color of global scattering to blow out because some color channels lose very little energy as they're transmitted through the hair while other channels lose exponentially more energy. This can be used to compensate for that effect.

Other Settings--

Global Illumination: When illuminating hair (or fur) using Global Illumination, users can adjust the 'Global Illumination' settings of the hair shader to control how the GI rays are used--

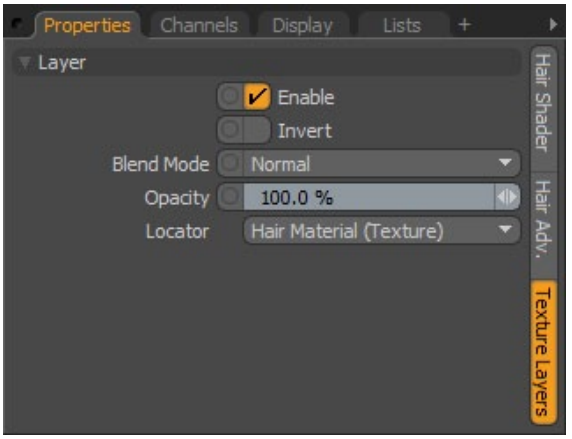
None- This means that the color of the hair is only affected by direct and ambient lighting. It also doesn't bounce light onto other surfaces.

Receive Indirect Lighting- Receiving indirect lighting means that light will bounce off other surfaces onto the hair, causing the hair to be lit indirectly. This is done with Monte-Carlo global illumination, by casting rays in every direction, instead of just towards a light.

Cast Indirect Lighting- This means that the hair will contribute indirect lighting onto other surfaces. In other words, indirect rays being fired from another surface will cause the hair shading to be evaluated if they hit the hair surface, instead of just appearing black.

Both- This does both Receiving and Casting of indirect lighting.

GI Rays: This controls how many rays will be used for calculating indirect lighting, either in the 'Receive' or 'Both' modes. More rays will produce smoother results at the expense of longer render times.



Texture Layers

Layer--

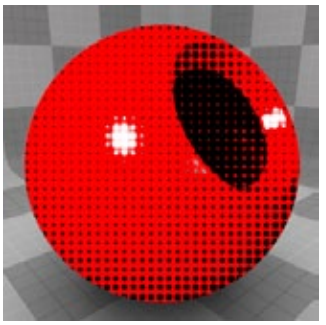
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When unchecked (*disabled*), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the illumination of the hair/fur producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

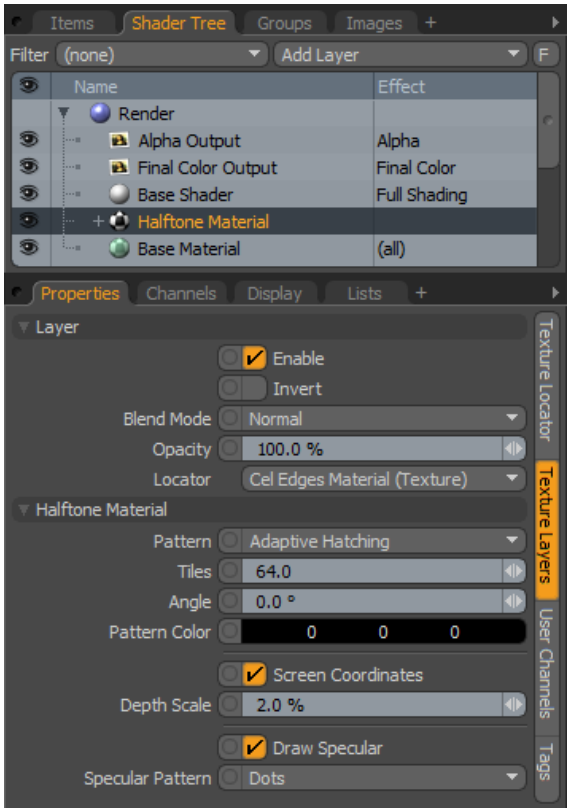
Halftone Material



The Halftone Material stylizes the target surface replacing the smooth shading with a black user defined pattern laid over its diffuse color, simulating the look of a low-fidelity image printed in a newspaper or magazine. Users can add the material to any surface using the 'Add Layer' function of the Shader Tree, found in the menu under "Custom Materials > Halftone Material". The direction and contouring of the lines can be dependant on the surfaces UV map, defined in the Texture Locator item, or by enabling the 'Screen Coordinates' option, the lines will simply follow along the user-

defined "Angle' value relative to the frame.

For information regarding adding and working with Shader Tree Items Layers, please reference the the Shader Tree page of the documentation.



Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When unchecked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (way the texture is applied) to the surface. The 'Locator' option sets that association. Users can choose alternate locators, however, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Halftone Material--

Pattern: Determines the design motif used to create the halftone pattern across the surface.

Dots- Applies a series of variable sized circles simulating surface shading.

Lines- Applies a series of lines of variable width that simulates surface shading.

Crosshatch- Applies a series of perpendicular lines of variable width that simulates surface shading.

Adaptive Hatching- Simulates the technique of *woodcut* shading where the lines can't get very thin. At the lightest shading, it draws a dashed line which becomes more solid and thicker as it gets darker. A perpendicular cross hatch is added for the darkest tones.

Tiles: Determines the size or spacing of the halftone Pattern. Smaller values will result in finer pattern spacing, while larger values will produce a larger more apparent pattern. Users should note that this value is relative to the frame, so scaling of the frame's resolution will affect the Halftone result. When doubling the frame resolution, users should also double the 'Tiles' value to produce the same resulting image.

Pattern Color: Determines the color of the halftone pattern added to the surfaces Diffuse color.

Screen Coordinates: When this option is enabled, the 'Pattern' will draw relative the the frame itself, based on the user defined 'Angle' value. When disabled, the UV map defined in the associated 'Texture Locator' determines the direction and angle of the halftone pattern, giving users ultimate control of halftone pattern results.

Depth Scale: When the 'Screen Coordinates' option is *enabled*, the 'Depth Scale' value can be applied to simulate the Pattern contouring when using a UV map without using a UV map. This is done by scaling the 'Tiles' values as the surface recedes into the distance. The Depth Scale' amount controls how much the depth affects the 'Tiles' scaling.

Draw Specular: When this option is *enabled*, any Specular highlights created by direct light sources in the scene will be added as an additional white pattern overlaid on the surface.

Specular Pattern: Determines the design motif used to create the Specular overlay halftone pattern.

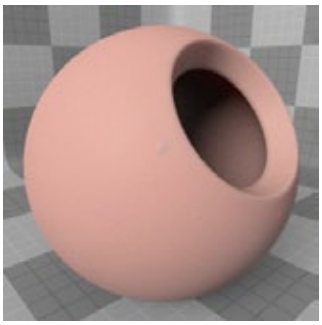
Dots- Applies a series of variable sized circles simulating surface shading.

Lines- Applies a series of lines of variable width that simulates surface shading.

Crosshatch- Applies a series of perpendicular lines of variable width that simulates surface shading.

Adaptive Hatching- Simulates the technique of woodcut shading where the lines can't get very thin. At the lightest shading, it draws a dashed line which becomes more solid and thicker as it gets darker. A perpendicular cross hatch is added for the darkest tones.

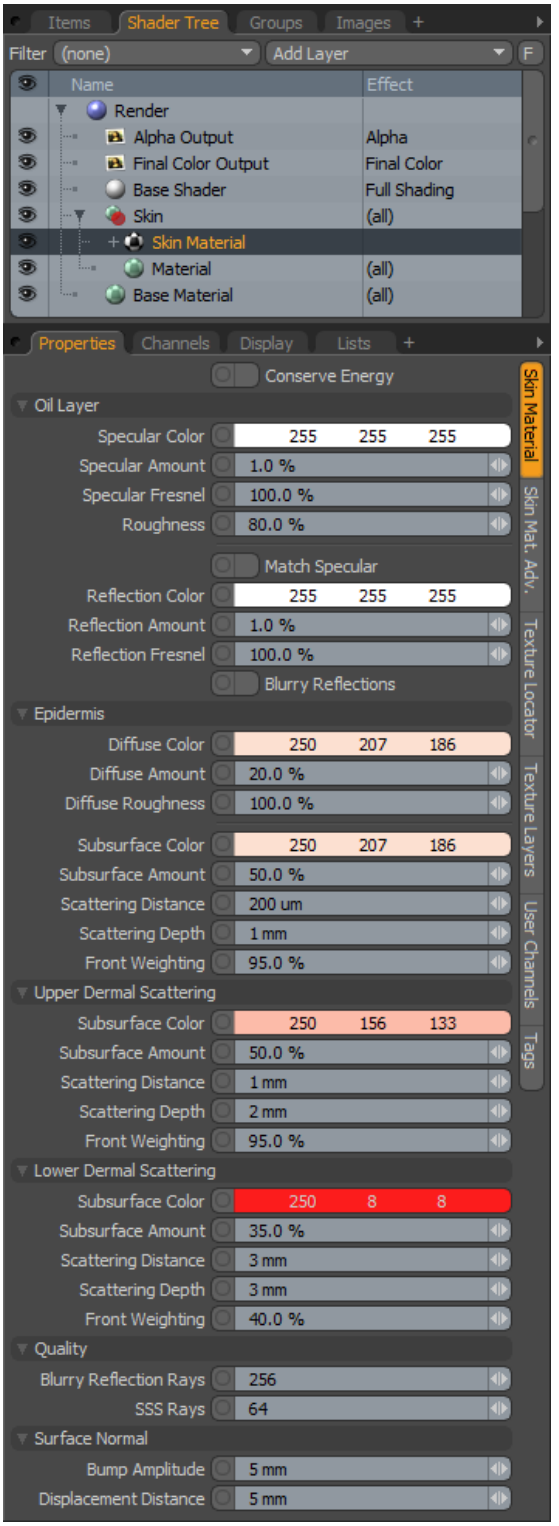
Skin Material



The 'Skin Material' is designed to replace the default BRDF shading calculations of the standard Material item when applied to a surface. Human skin is a rather complex organ, made up of multiple layers and materials, each absorbing, scattering and reflecting light differently creating a very subtle but distinct look that we as humans are intimately familiar with. In order to simulate such a complex surface a special Material shader was required. The material is a made up of mostly familiar options, that combine to simulate the subtle nuance that is *skin*.

The 'Skin Material' item is added to the Shader Tree using the 'Add Layer' function -"Add Layer > Custom Materials > Skin Material". For more information on adding and manipulating Shader Tree layers, please reference that page of the documentation. For proper results, the Skin Material should be placed above the Material item it is replacing (feel free to *disable* or *remove* the Material item). Once positioned, selecting the 'Skin Material' item will reveal its attributes in the 'Properties' viewport where its settings may be adjusted to fine tune the look of the rendered skin. Additionally, skin is not a consistent color across its entire surface and users will want to add texture layers, set to the appropriate 'Effect' to represent the natural variations inherent in skin, necessary in producing the most realistic results.

Conserve Energy: When enabled, the 'Conserve Energy' setting maintains a physically accurate blend of 'Diffuse' and 'Specular' light reflecting over the skins surface. It also accurately spreads the energy of each contribution to the oil and each dermal layer so that the overall contribution is never more than 100%.



Oil Layer--

The uppermost layer of skin excretes an oil that produces a shiny, specular reflection on its surface. Depending on the amount on the skin, it becomes less or more reflective. Skin that is flushed, say from exercise or exertion will be shinier than clean skin or skin with powder make-up on, so the Amount and Roughness will need to be adjusted depending on the desired look.

Specular Color: Determines the color of the highlight created from reflections of direct light sources.

Specular Amount: Determines the strength of the Specular highlight.

Specular Fresnel: At glancing angles (those perpendicular to the camera), reflectance values of skin increase. The Specular Fresnel setting realistically increases the specular amount at these glancing angles producing a physically correct surface.

Roughness: The 'Roughness' setting determines the size of the *spread* for both 'Specular' and 'Blurry Reflections' when *enabled*. A clay pot will reflect light very differently than a porcelain cup, though the surfaces appear to both be smooth, on a microscopic level the clay pot is much rougher than the porcelain and this affects how the light is reflected, spreading the reflection across the surface. A higher 'Roughness' value will produce these rougher more

spread out highlights, while lower values produce tighter more focused highlights.

Match Specular: When *enabled*, the Reflection options will be automatically adjusted to to match the equivalent Specular settings.

Reflection Color: Determines the tinting or colorization of Reflections on the surface.

Reflection Amount: Determines the amount of the environment reflections that will appear on the surface, like with the specular setting, oilier surfaces will appear more reflective.

Reflection Fresnel: At glancing angles (those perpendicular to the camera), reflectance values of skin increase. The Reflection Fresnel setting realistically increases the reflection amount at these glancing angles producing a physically correct surface.

Blurry Reflections: When *enabled*, reflections will take in to account the 'Roughness' option and spread reflections across the surface, such as would happen with real skin.

Epidermis--

The Epidermis is the outer layer of effectively dead skin cells. This layer is quite thin and on a microscopic level looks very much like tiny overlapping scales. This layer is mainly responsible for the color of the skin, depending on the actual amounts of melanin present in the cells.

Diffuse Color: Determines the Diffuse Color of the skins surface, this is the apparent visual color of the skins outer surface that reflects back when illuminated.

Diffuse Amount: Determines the amount of light reflected back from the skins surface, the rest of the light is absorbed for SSS.

Diffuse Roughness: The 'Diffuse Roughness' option represents a lack of specular on a Epidermal surface so reading the Diffuse shading outwards across the surface simulating the tiny microscopic variations of the individual skin cells.

Subsurface Color: Determines the Color of the light that is absorbed, scattered and the reflected back out. The epidermis is fairly translucent and the subsurface color is basically the same color as the Diffuse Color for human skin.

Subsurface Amount: Determines the amount of scattered light that is reflected back out. The subsurface scattering is combined with the Diffuse color to produce the final Epidermal layer.

Scattering Distance: Determines the distance that light scatters through the surface before reflecting back out. Deeper rays will attenuate toward the SSS color, reaching

the full SSS color at the Scattering Distance value.

Maximum Depth: Determines the maximum depth where a ray can contribute to the final solution, in essence clipping the SSS to approximate a solid core. A value of '0' disabled the option.

Front Weighting: The 'Front Weighting' option biases the SSS ray forward or backward. Settings between 0-49 bias the SSS rays backwards, 51-100 bias the rays forward and setting of 50% distributes the rays equally in both directions.

Upper Dermal Scattering--

The upper dermal layer contains tissue for cushioning the Epidermis, and also houses the nerve endings, hair follicles, sweat glands and blood vessels. Because of these constructs the light scattered from here will be differently colored than the epidermal layer.

Subsurface Amount: Determines the amount of scattered light that is reflected back out. The Upper Dermal Scattering is combined with the Epidermal layers solution to produce the final shaded result.

Scattering Distance: Determines the distance that light scatters through the surface before reflecting back out. Deeper rays will attenuate toward the SSS color, reaching the full SSS color at the Scattering Distance value.

Maximum Depth: Determines the maximum depth where a ray can contribute to the final solution, in essence clipping the SSS to approximate a solid core. A value of '0' disabled the option.

Front Weighting: The 'Front Weighting' option biases the SSS ray forward or backward. Settings between 0-49 bias the SSS rays backwards, 51-100 bias the rays forward and setting of 50% distributes the rays equally in both directions.

Lower Dermal Scattering--

The Lower Dermal area contains more blood vessels and typically is the layer that contributes the most red because of this.

Subsurface Amount: Determines the amount of scattered light that is reflected back out. The Lower Dermal Scattering is combined with the Upper Dermal and Epidermal layers to produce the final shaded result.

Scattering Distance: Determines the distance that light scatters through the surface before reflecting back out. Deeper rays will attenuate toward the SSS color, reaching the full SSS color at the Scattering Distance value.

Maximum Depth: Determines the maximum depth where a ray can contribute to the final solution, in essence clipping the SSS to approximate a solid core. A value of '0' disabled the option.

Front Weighting: The 'Front Weighting' option biases the SSS ray forward or backward. Settings between 0-49 bias the SSS rays backwards, 51-100 bias the rays forward and setting of 50% distributes the rays equally in both directions.

Quality--

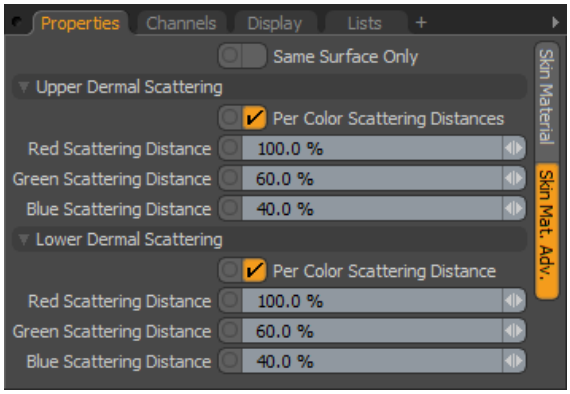
Blurry Reflection Rays: Defines the number of rays used to calculate the 'Blurry Reflections' when that option is enabled. Higher ray counts will produce smoother, more realistic results but will also result in longer calculation times when rendering.

SSS Rays: Defines the number of rays used to calculate the Subsurface Scattering results. Higher ray counts will produce more accurate SSS results but will also take longer to calculate.

Surface Normal--

Bump Amplitude: Determines the maximum strength of any bump map applied to the skin surface.

Displacement Distance: Determines the maximum displacement distance when any type of Displacement texture layer is used.



Skin Material Advanced

Upper Dermal Scattering--

In real skin, each wavelength of light penetrates the skin to a different depth, which attenuates the colors slightly as rays travel further into the surface. For the most accurate results users can enable and adjust these settings, but they will also increase render times as individual rays will need to be fired for each wavelength calculation.

Per Color Scattering Distances: *Enable* this option to calculate per color Scattering for SSS of the Upper Dermal layer.

Red/Green/Blue: Users can adjust, how each wavelength attenuates through the surface, Red being the highest wavelength should penetrate further than the

lower wavelengths of green and then blue. Values of 100% will travel the maximum 'Scattering Distance' specified.

Lower Dermal Scattering--

Per Color Scattering Distances: *Enable* this option to calculate per color Scattering for SSS of the Lower Dermal layer.

Red/Green/Blue: Users can adjust, how each wavelength attenuates through the surface, Red being the highest wavelength should penetrate further than the lower wavelengths of green and then blue. Values of 100% will travel the maximum 'Scattering Distance' specified.

Enhance:modo Textures

Enhance:modo textures are a collection of 147 advanced procedurals layers, many with additional parameters that further extend the collection. The complete shader set allows the creation of an infinite number of different textures, patterns and surfaces. Being that they are all procedurally based, they have no fixed resolution, providing an near infinite level of detail, even when applied with a UV map. In addition all of the textures parameters can be animated so that many different special effects can be generated.

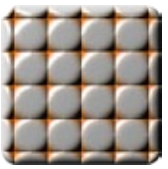
The following items may be added to any scene from within the Shader Tree viewport, by LMB-clicking the 'Add Layer' option of the full viewport window and selecting an item from the pop-up context menu. For information regarding adding and otherwise working with Shader Tree items layers, please reference the the Shader Tree page of the documentation. Specific documentation for Geometric, Noise, Organic, Skins, Tiles and Other are on the pages following each chart of example tiles.

Geometric

Box

Circular

Corners



Cubic

Dimples

Grid

Iris

Linear

Polygon

Radial

Ring

Rnd Linear

Spiral

Star

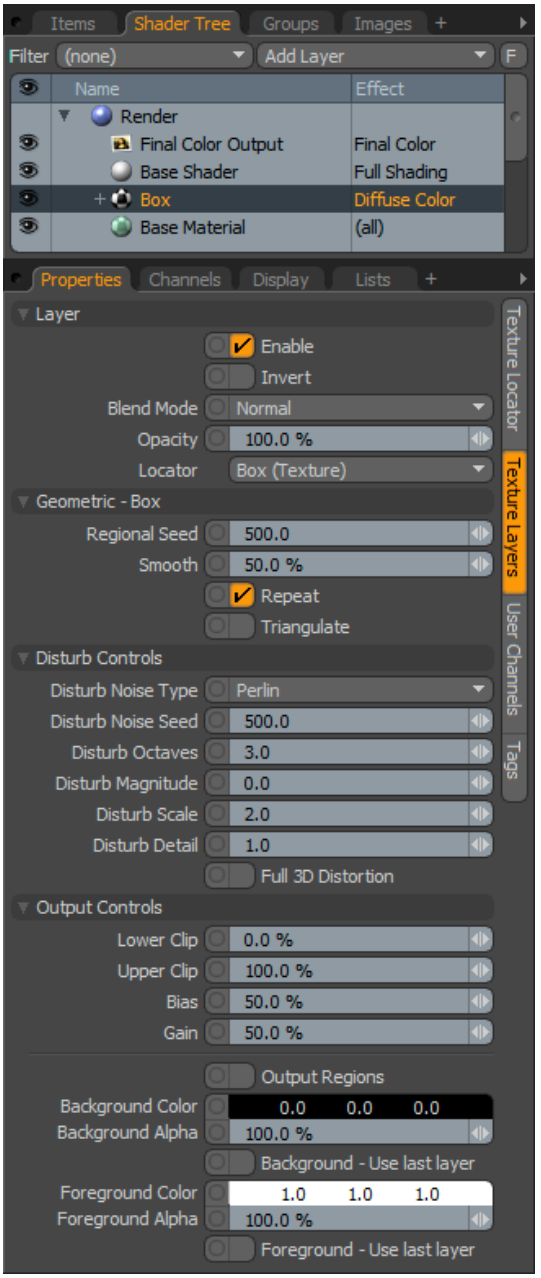
Box

The Box texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Box texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a series of adjustable *box*-shaped patterns.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.



Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Geometric - Box --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Smooth: This value determines the amount of smoothing around the edges of the box. Higher values will result in smoother edges.

Repeat: This option, when selected, causes the texture to repeat indefinitely.

Triangulate: This option, when selected, causes the pattern to have a triangle-based edging pattern, as opposed to the default sawtooth blend.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Disturb Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Disturb Magnitude: This value determines the strength of the distortion effect on the base procedural.

Disturb Scale: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

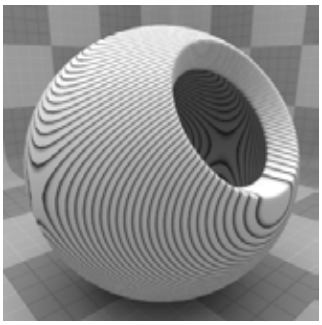
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Circular



The Circular texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Circular texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a series of adjustable *circular* patterns.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

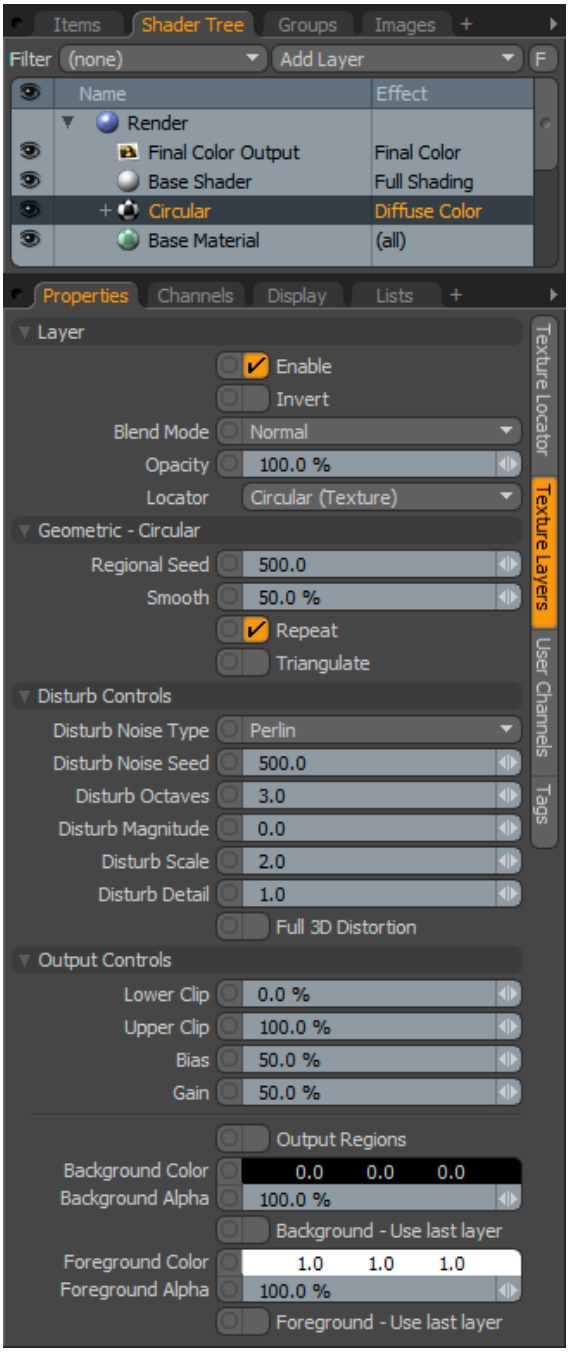
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Geometric - Circular --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Smooth: This value determines the amount of smoothing around the edges of the circle. Higher values will result in smoother edges.

Repeat: This option, when selected, causes the texture to repeat indefinitely.

Triangulate: This option, when selected, causes the pattern to have a triangle-based edging pattern, as opposed to the default sawtooth blend.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Disturb Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Disturb Magnitude: This value determines the strength of the distortion effect on the base procedural.

Disturb Scale: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Corners

The Corners texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Corners texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a series of adjustable *corner* patterns.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

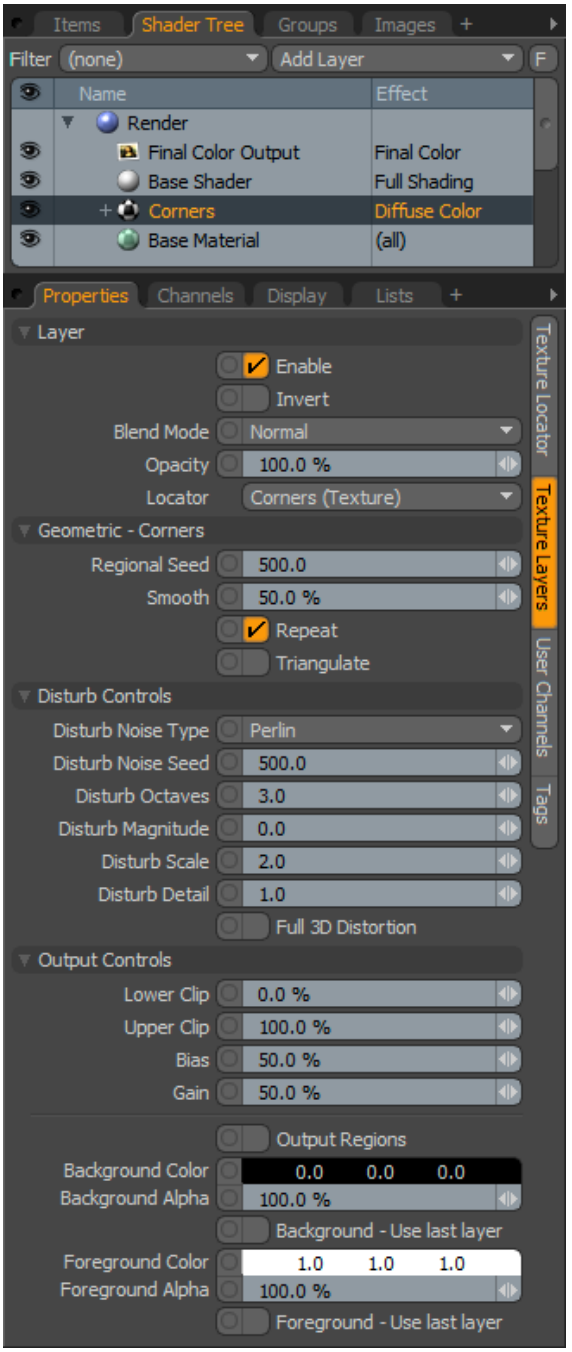
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Geometric - Corners --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Smooth: This value determines the amount of smoothing around the edges of the corners. Higher values will result in smoother edges.

Repeat: This option, when selected, causes the texture to repeat indefinitely.

Triangulate: This option, when selected, causes the pattern to have a triangle-based edging pattern, as opposed to the default sawtooth blend.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Disturb Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Disturb Magnitude: This value determines the strength of the distortion effect on the base procedural.

Disturb Scale: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

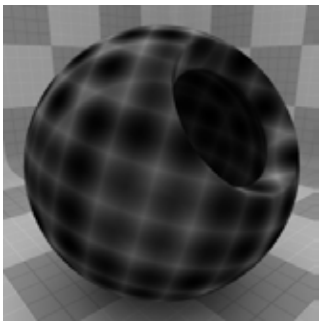
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Cubic



The Cubic texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Cubic texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a series of adjustable *cubic* patterns.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

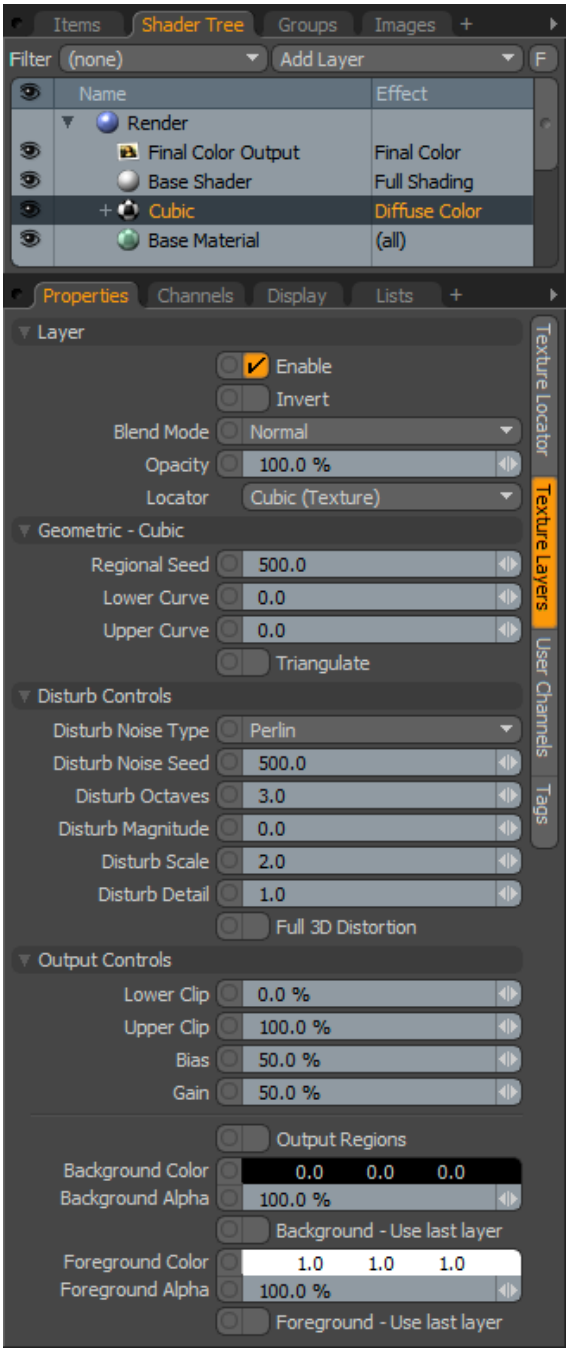
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Geometric - Cubic --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Lower Curve: This value determines the shape of the curve at the bottom of each cube.

Upper Curve: This value determines the shape of the curve at the bottom of each cube.

Triangulate: This option, when selected, causes the pattern to have a triangle-based edging pattern, as opposed to the default sawtooth blend.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Disturb Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Disturb Magnitude: This value determines the strength of the distortion effect on the base procedural.

Disturb Scale: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

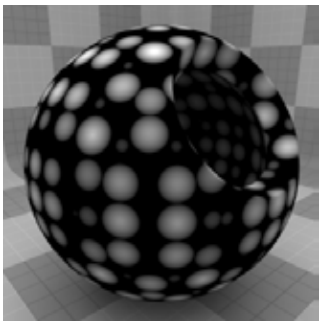
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Dimples



The Dimples texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Dimples texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a series of adjustable *dimple* patterns.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

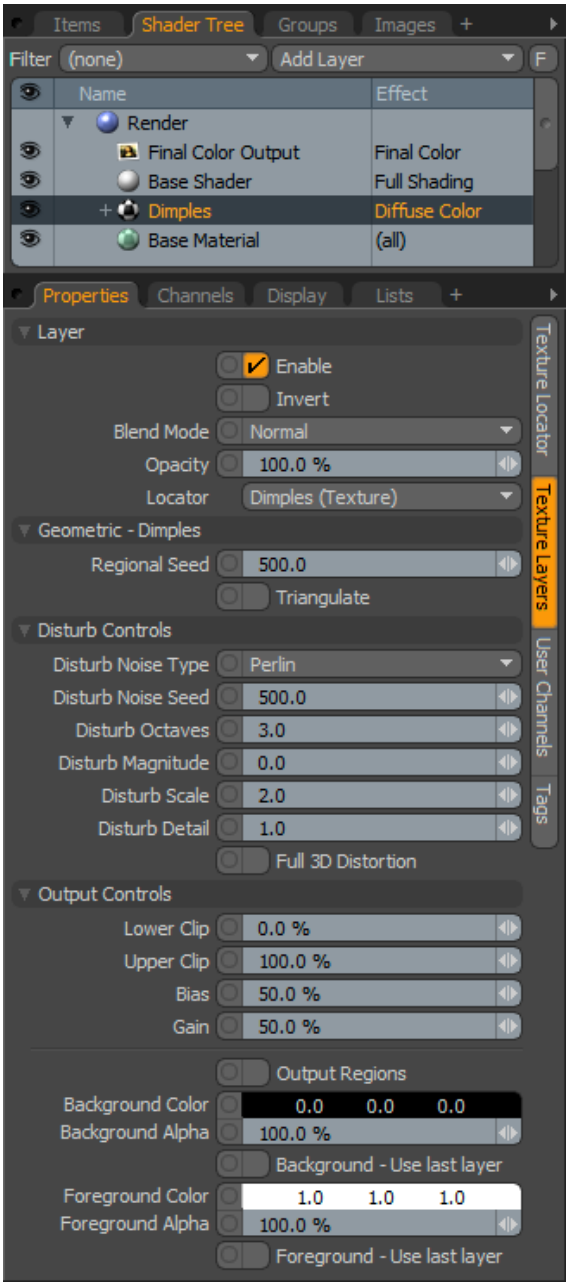
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Geometric - Dimples --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Triangulate: This option, when selected, causes the pattern to have a triangle-based edging pattern, as opposed to the default sawtooth blend.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Disturb Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Disturb Magnitude: This value determines the strength of the distortion effect on the base procedural.

Disturb Scale: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

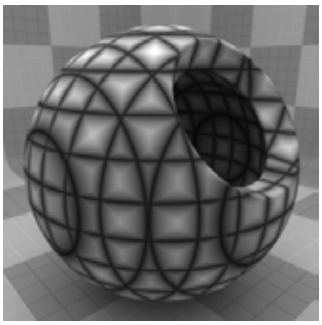
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Grid



The Grid texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Grid texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a series of adjustable *grid* patterns.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

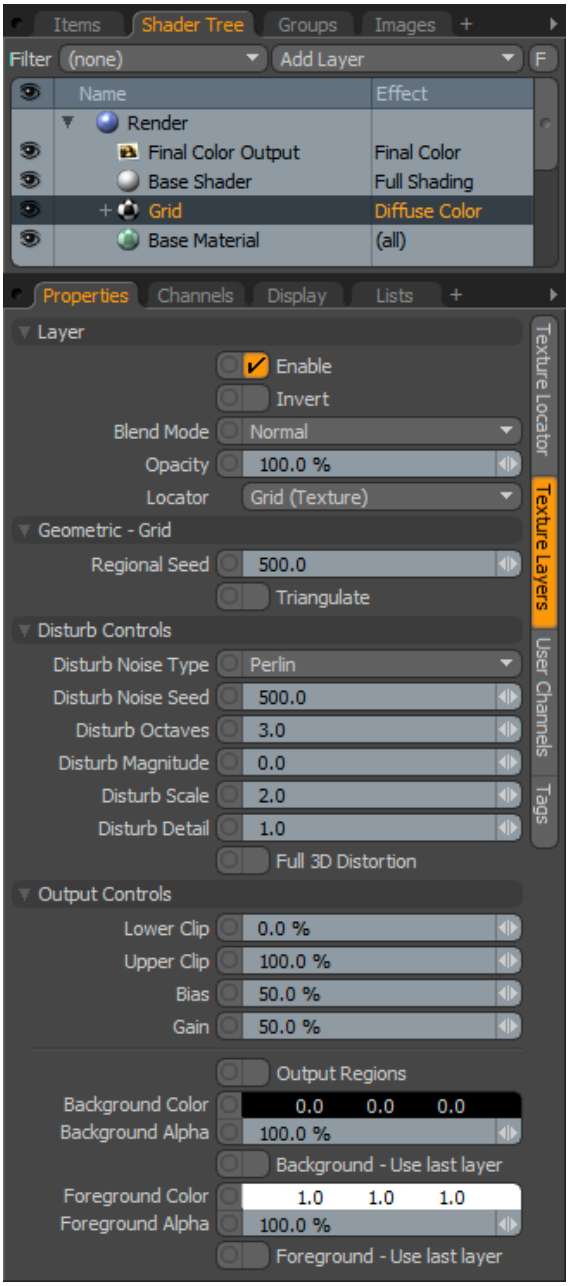
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Geometric - Grid --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Triangulate: This option, when selected, causes the pattern to have a triangle-based edging pattern, as opposed to the default sawtooth blend.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Disturb Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Disturb Magnitude: This value determines the strength of the distortion effect on the base procedural.

Disturb Scale: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value,

users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

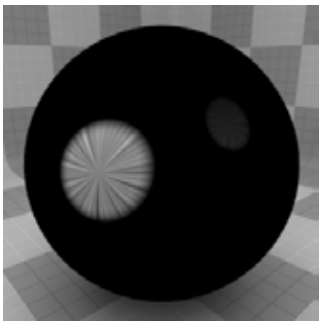
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Iris



The Iris texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Iris texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a pattern similar in shape and texture to an *iris*. This texture is great for making eyeballs or for adding a lens flare type-element.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

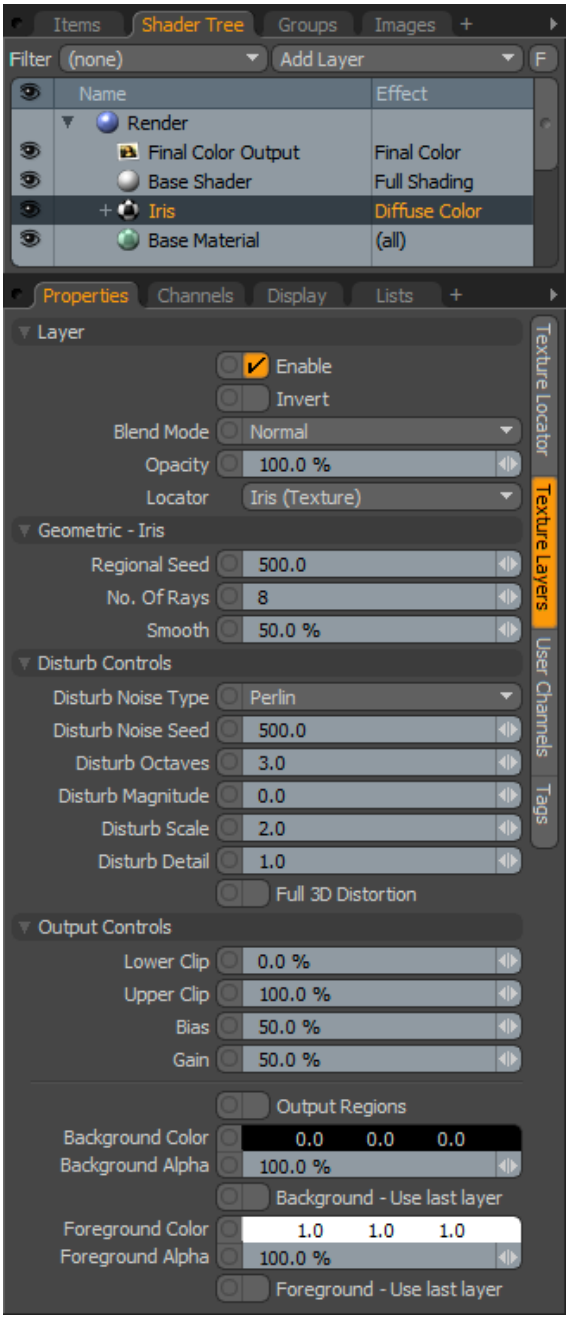
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Geometric - Iris --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Number Of Rays: This value determines the number of rays in the iris pattern.

Smooth: This value determines the amount of smoothing around the edges of the iris. Higher values will result in smoother edges.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Disturb Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Disturb Magnitude: This value determines the strength of the distortion effect on the base procedural.

Disturb Scale: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

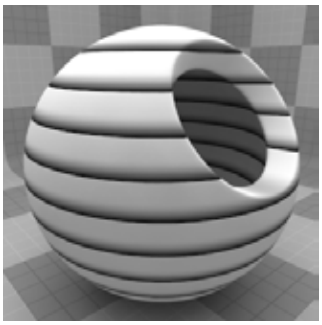
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Linear



The Linear texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Linear texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a series of adjustable *linear* patterns.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

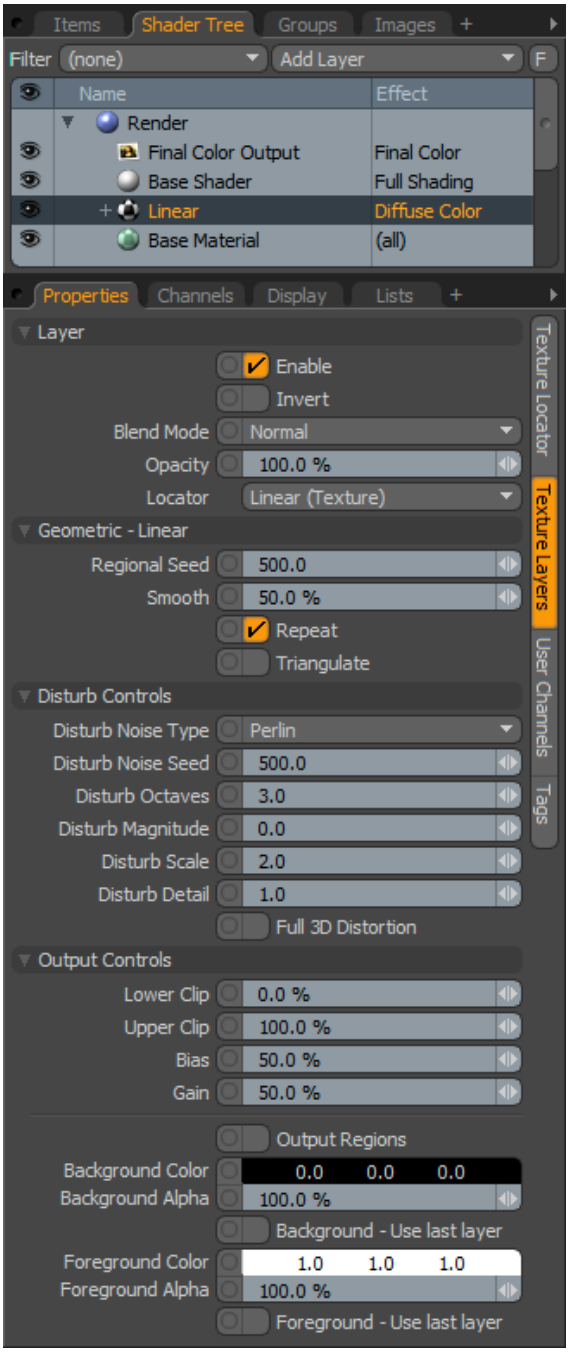
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Geometric - Linear --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Smooth: This value determines the amount of smoothing around the edges of the liner patterns. Higher values will result in smoother edges.

Repeat: This option, when selected, causes the texture to repeat indefinitely.

Triangulate: This option, when selected, causes the pattern to have a triangle-based edging pattern, as opposed to the default sawtooth blend.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Disturb Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Disturb Magnitude: This value determines the strength of the distortion effect on the base procedural.

Disturb Scale: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value,

truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

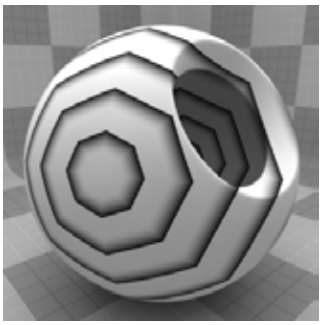
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Polygon



The Polygon texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Polygon texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a series of adjustable *polygonal* patterns.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

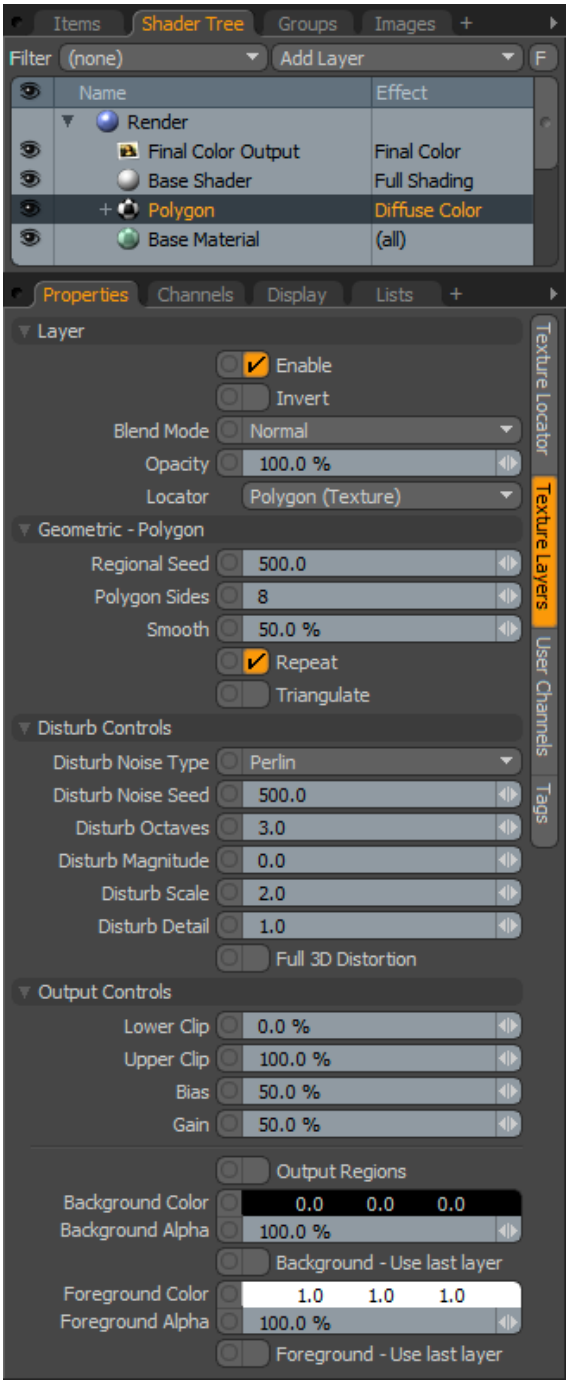
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Geometric - Polygon --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Polygon Sides: This value determines the number of sides on the polygon(s).

Smooth: This value determines the amount of smoothing around the edges of the polygon. Higher values will result in smoother edges.

Repeat: This option, when selected, causes the texture to repeat indefinitely.

Triangulate: This option, when selected, causes the pattern to have a triangle-based edging pattern, as opposed to the default sawtooth blend.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Disturb Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Disturb Magnitude: This value determines the strength of the distortion effect on the base procedural.

Disturb Scale: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value,

truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

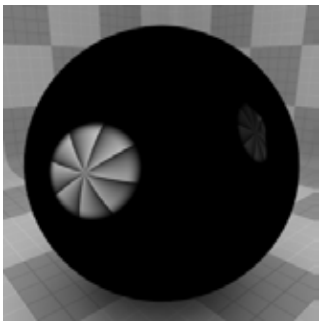
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Radial



The Radial texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Radial texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides a customizable *radial* pattern.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

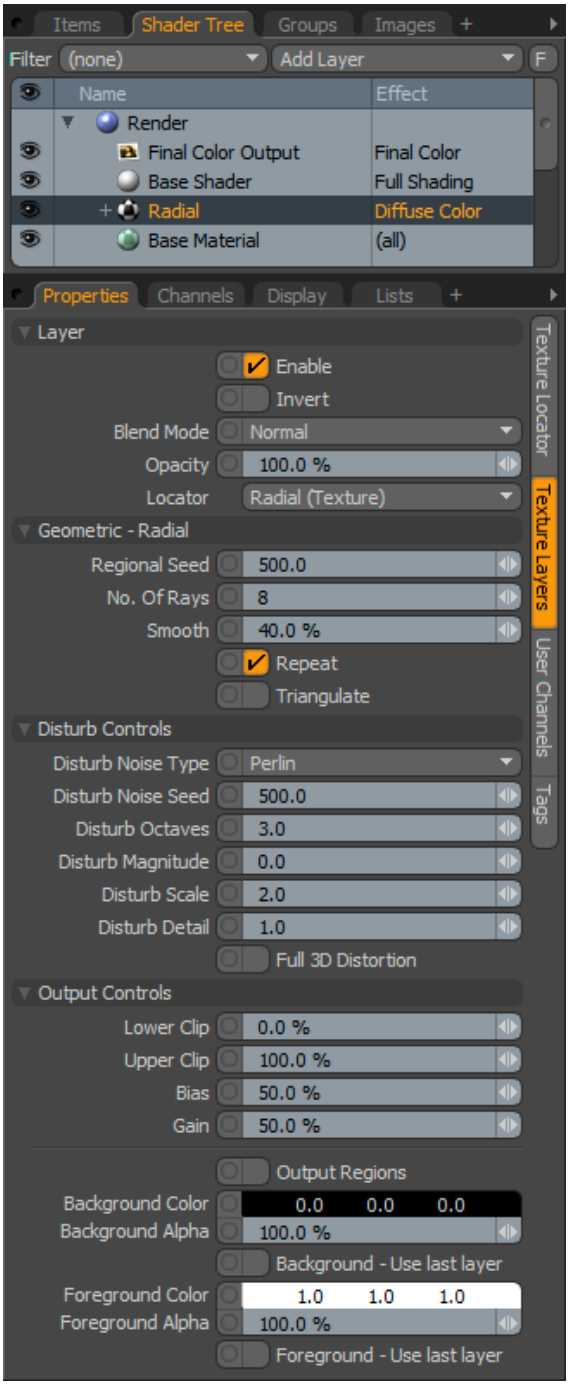
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Geometric - Radial --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Number Of Rays : This value determines the number of rays in the radial pattern.

Smooth: This value determines the amount of smoothing around the edges of the radial pattern. Higher values will result in smoother edges.

Repeat: This option, when selected, causes the texture to repeat indefinitely.

Triangulate: This option, when selected, causes the pattern to have a triangle-based edging pattern, as opposed to the default sawtooth blend.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Disturb Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Disturb Magnitude: This value determines the strength of the distortion effect on the base procedural.

Disturb Scale: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

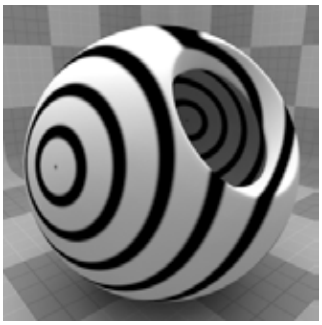
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Ring



The Ring texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Ring texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a series of adjustable *ring* patterns.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

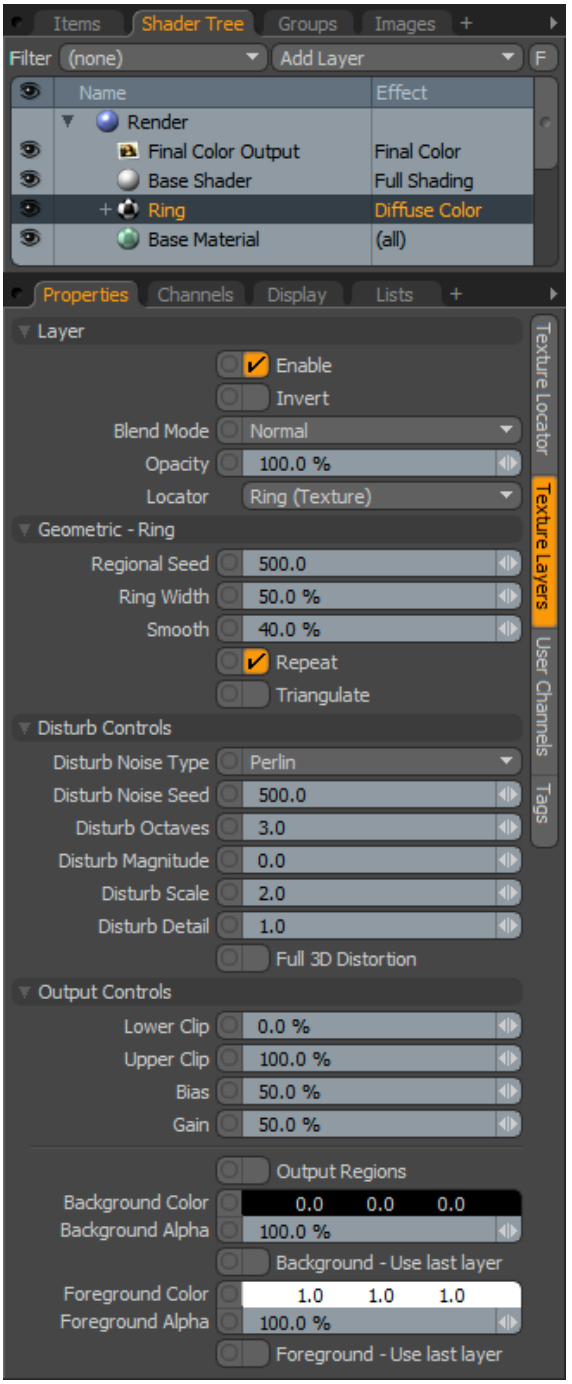
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Geometric - Ring --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Ring Width: This value determines the width of the individual rings.

Smooth: This value determines the amount of smoothing around the edges of the ring. Higher values will result in smoother edges.

Repeat: This option, when selected, causes the texture to repeat indefinitely.

Triangulate: This option, when selected, causes the pattern to have a triangle-based edging pattern, as opposed to the default sawtooth blend.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Disturb Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Disturb Magnitude: This value determines the strength of the distortion effect on the base procedural.

Disturb Scale: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value,

truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

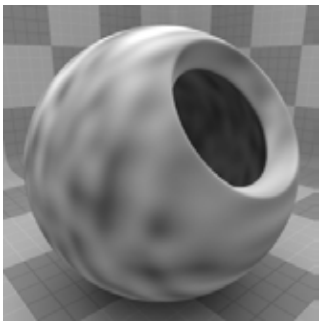
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

RndLinear (Random Linear)



The Random Linear texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Random Linear texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a *randomized linear* pattern.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

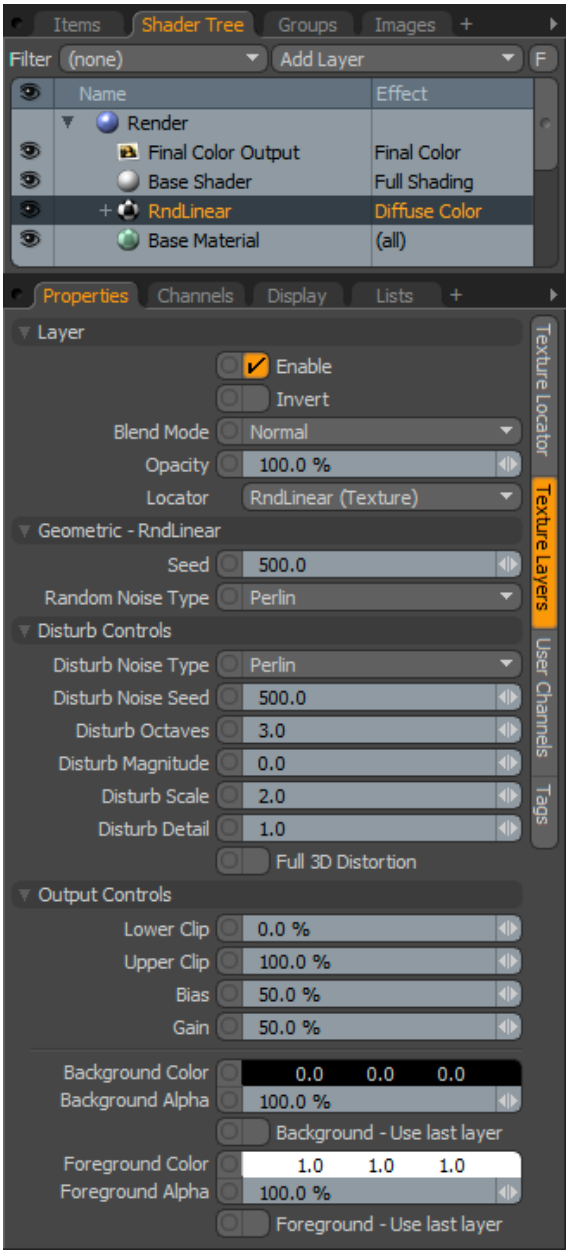
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Geometric - Random Linear --

Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Random Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Disturb Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Disturb Magnitude: This value determines the strength of the distortion effect on the base procedural.

Disturb Scale: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value,

truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

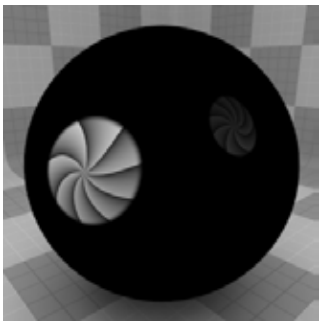
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Spiral



The Spiral texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Spiral texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides a customizable *spiral* pattern.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

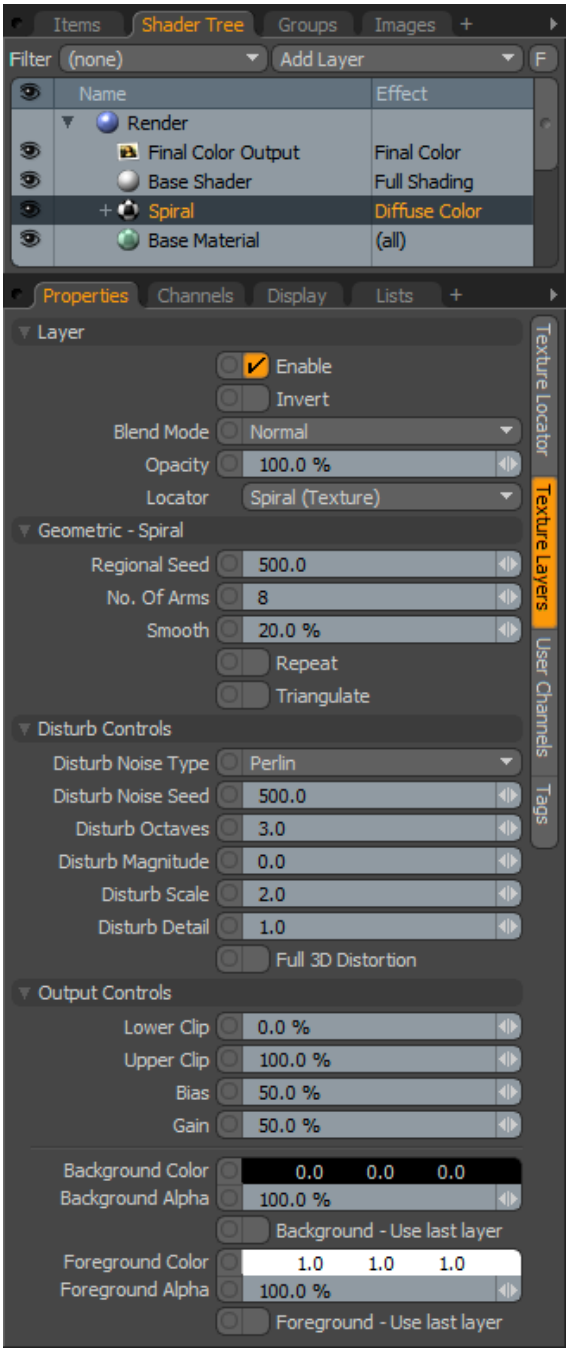
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Geometric - Spiral --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Number Of Arms : This value determines the number of arms in the spiral pattern.

Smooth: This value determines the amount of smoothing around the edges of the spiral. Higher values will result in smoother edges.

Repeat: This option, when selected, causes the texture to repeat indefinitely.

Triangulate: This option, when selected, causes the pattern to have a triangle-based edging pattern, as opposed to the default sawtooth blend.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Disturb Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Disturb Magnitude: This value determines the strength of the distortion effect on the base procedural.

Disturb Scale: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Star

The Star texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Star texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides a customizable *star* pattern.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

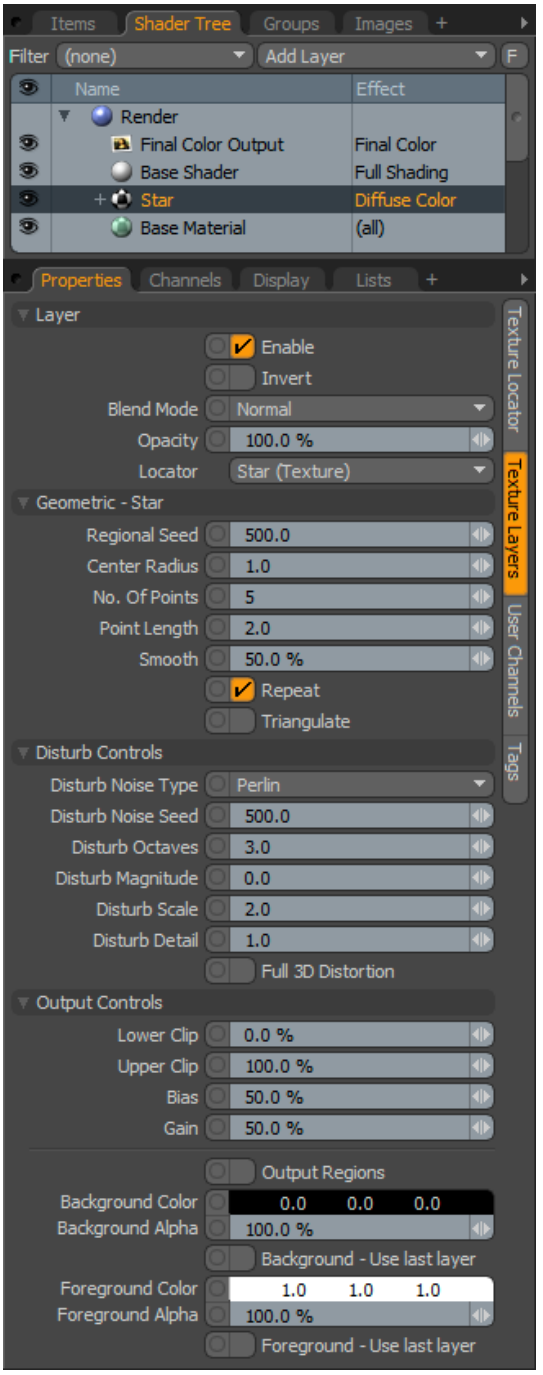
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Geometric - Star --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Center Radius: This value determines the radius at the center of the star.

Number Of Points: This value determines the number of points in the star pattern.

Point Length: This value determines the length of the indivudal points on the star.

Smooth: This value determines the amount of smoothing around the edges of the star. Higher values will result in smoother edges.

Repeat: This option, when selected, causes the texture to repeat indefinitely.

Triangulate: This option, when selected, causes the pattern to have a triangle-based edging pattern, as opposed to the default sawtooth blend.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Disturb Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Disturb Magnitude: This value determines the strength of the distortion effect on the base procedural.

Disturb Scale: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Noise

Agate

Bozo

Cruddy

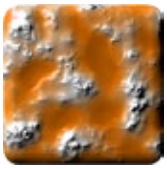
Dented

Etched

FBM

FlowBozo

Granite


Hybrid

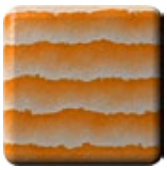
Lump

Marble Noise

Marble Vein

Scar

Scruffed

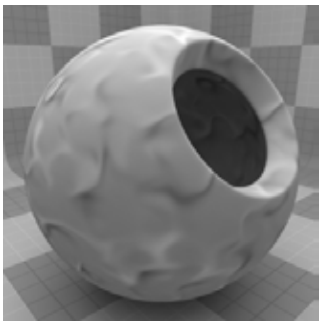

Strata

Stucco

Vector Bozo

Wrapped fBm

Agate



The Agate texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Agate texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader emulates the color patterns in certain types of marble, *agate*, or jade.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

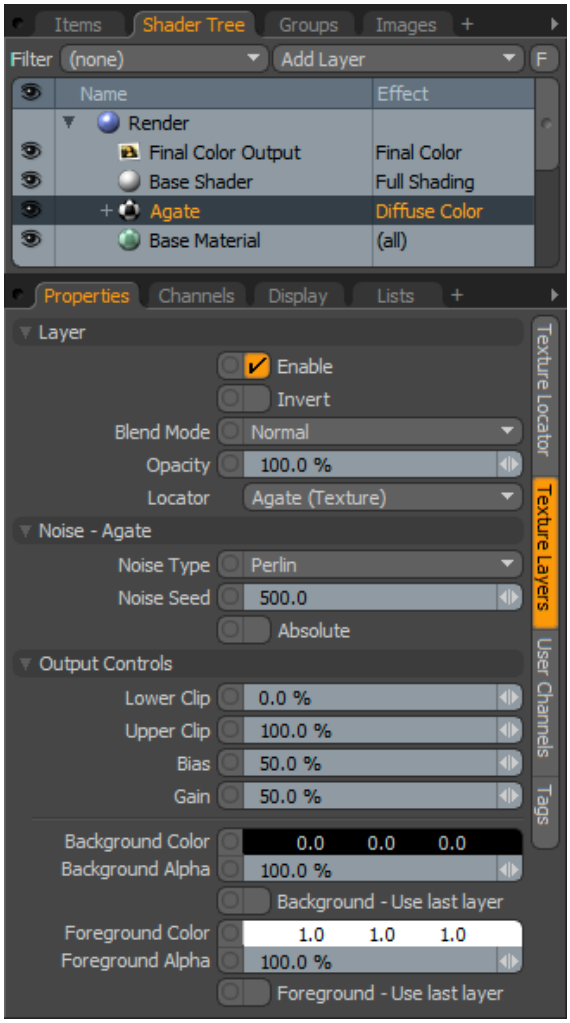
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - Agate --

Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Absolute: This option determines whether or not the Absolute value for the noise is used, providing a stronger, more defined variation on the default texture.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value,

truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

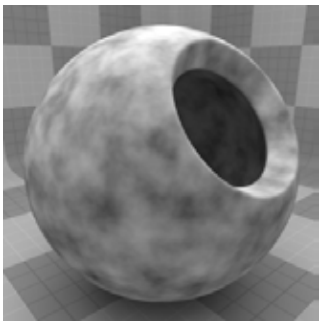
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Bozo



The Bozo texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Bozo texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader is ideal for textures like clouds, smoke or fog. A plastic stippled effect can also be created using very small *bozo* bumps. Adjusting the shader to draw larger bumps can create wave or ripple effects.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

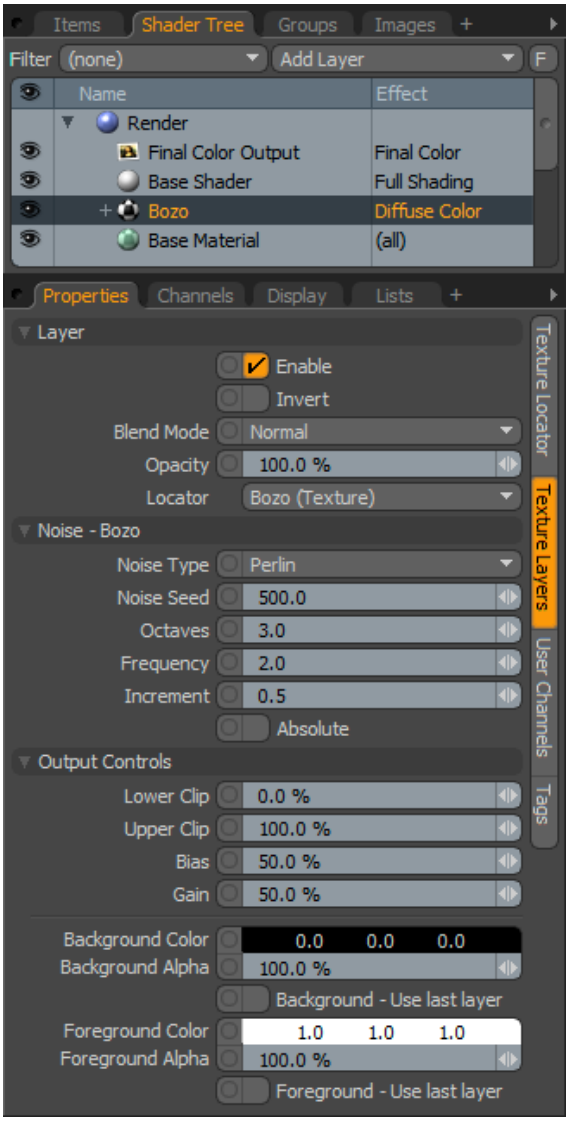
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - Bozo --

Noise Type: This option determines the look of the distortion by providing several Noise function types.

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this

number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Absolute: This option determines whether or not the Absolute value for the noise is used, providing a stronger, more defined variation on the default texture.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

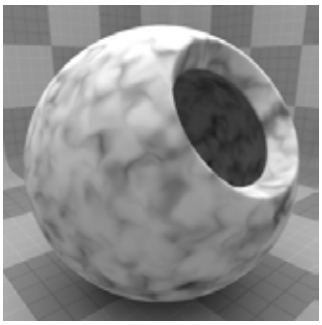
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Cruddy



The Cruddy texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Cruddy texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader creates a very rough crusty, *cruddy* type of pattern, much like the accumulation of dirt or rust on a surface.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

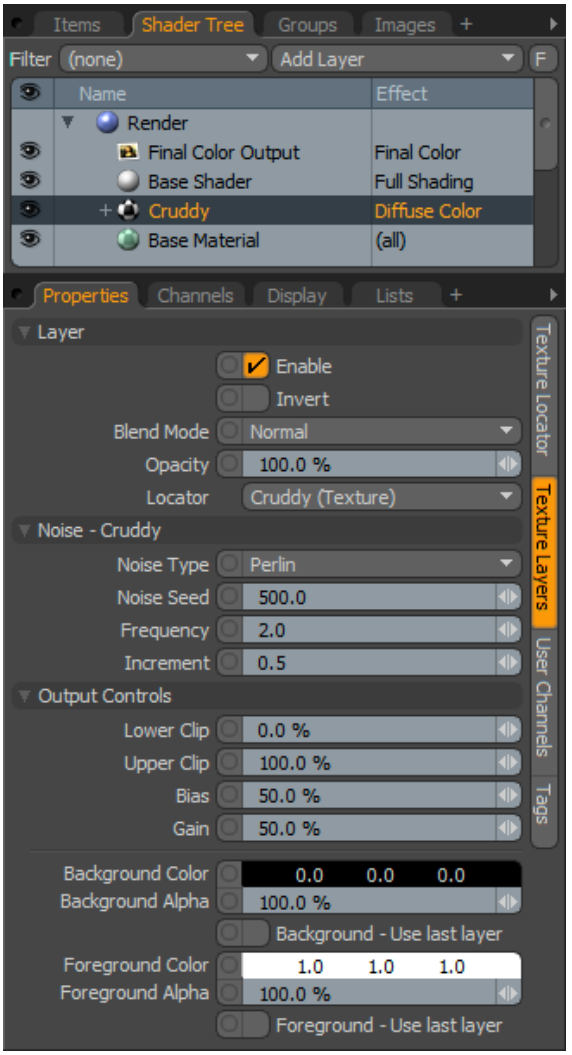
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - Cruddy --

Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

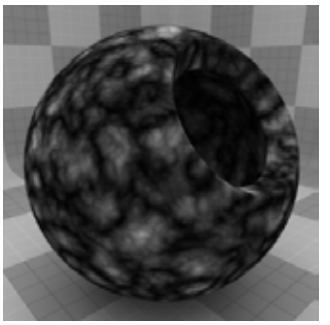
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Dented



The Dented texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Dented texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader can be used to provide variety of bumps that resemble *dents*. The bumps range from the kind of dents you might find in "beaten up" sheet metal to the deep craggy craters of an asteroid. Alternatively, the dents can be inverted to create pimples or detailed growths.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

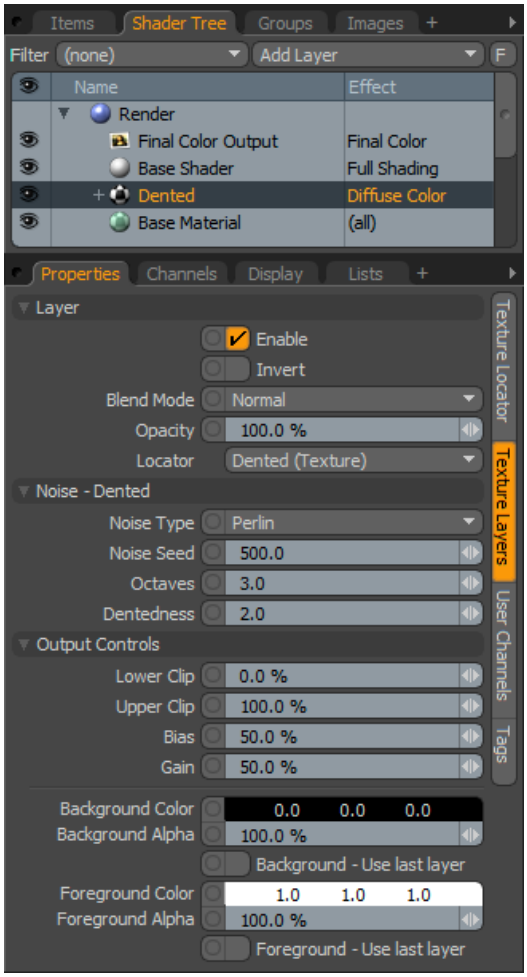
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - Dented --

Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Dentendness: This value controls how dented the texture appears, with higher values generating smaller, more detailed dents.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

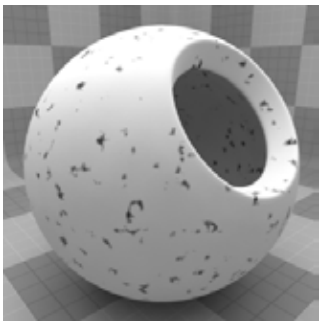
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Etched



The Etched texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Etched texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides an *etched* appearance, as though the surface has been treated with acid.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

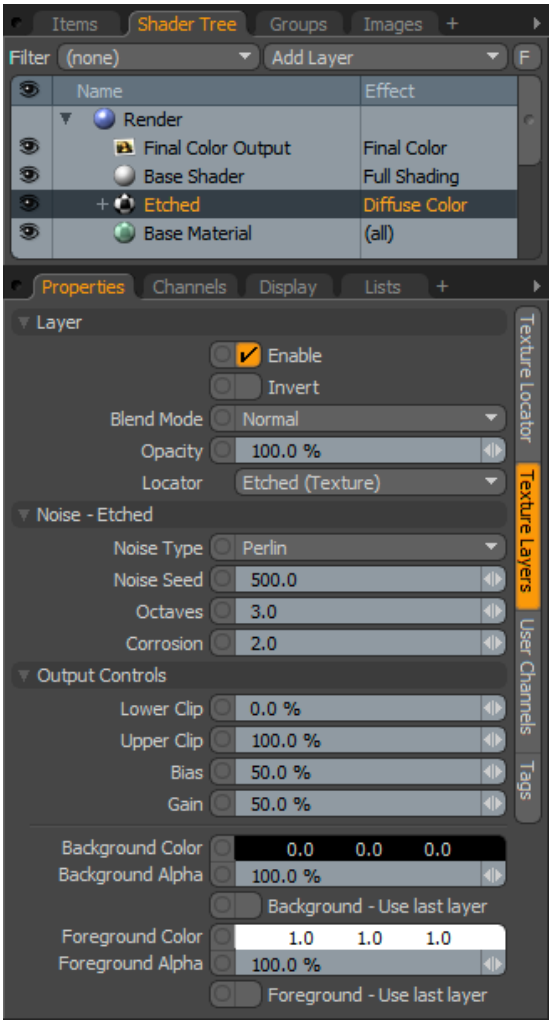
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - Etched --

Noise Type: This option determines the look of the distortion by providing several Noise function types.

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Corrosion: This value controls the amount of corrosion used to create the texture. As this number is increased, more corrosion is introduced into the texture.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

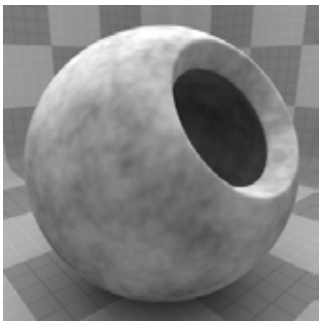
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

FBM



The FBM texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The FBM texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides noisy bumps based on *Fractal Brownian Motion* (FBM). This textures is great for noisy, organic surfaces such as rocks or rough stippled surfaces.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

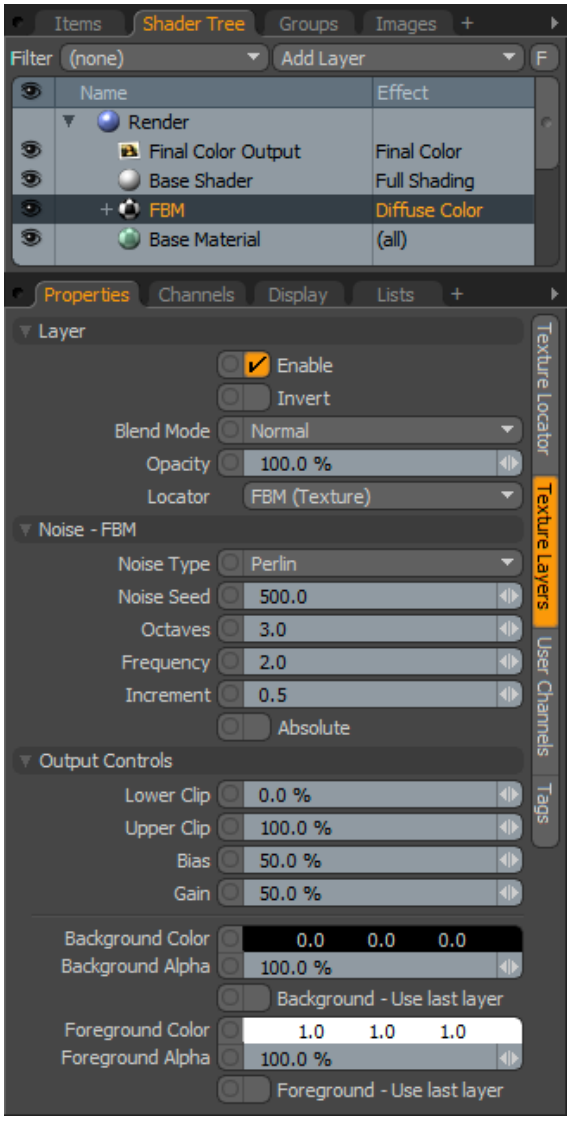
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - FBM --

Noise Type: This option determines the look of the distortion by providing several Noise function types.

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Absolute: This option determines whether or not the Absolute value for the noise is used, providing a stronger, more defined variation on the default texture.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

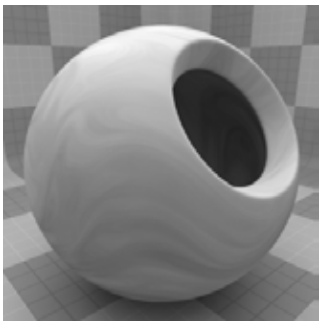
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Flow Bozo



The Flow Bozo texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Flow Bozo texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader creates a pattern similar to *flowing* gas or liquid.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

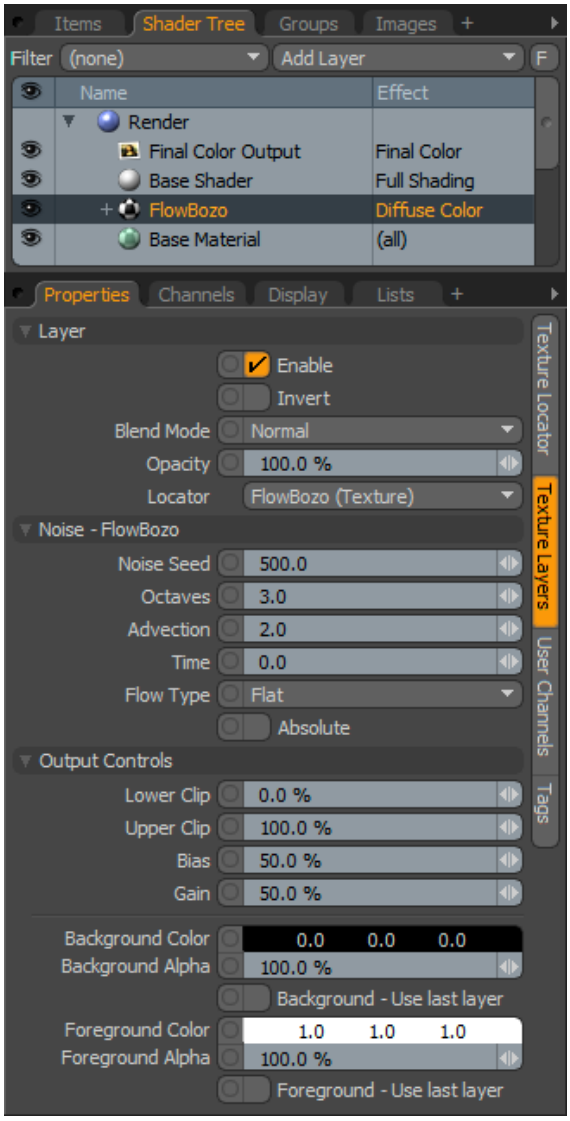
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - Flow Bozo --

Noise Type: This option determines the look of the distortion by providing several Noise function types.

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Advection: This value controls the noise of the flow effect, allowing you to stir up the texture.

Time: This value allows you to adjust the noise at selected key frames to control the animation of the flow.

Flow Type: This option controls the type of flow when using the flow time to create an animation. Choose from Flat, Square or Inverse.

Absolute: This option determines whether or not the Absolute value for the noise is used, providing a stronger, more defined variation on the default texture.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

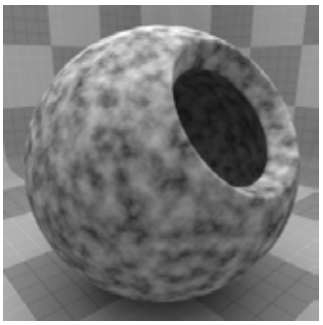
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Granite



The Granite texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Granite texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a pattern similar to *granite*. This texture can also be used for making star fields by defining very narrow color bands.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

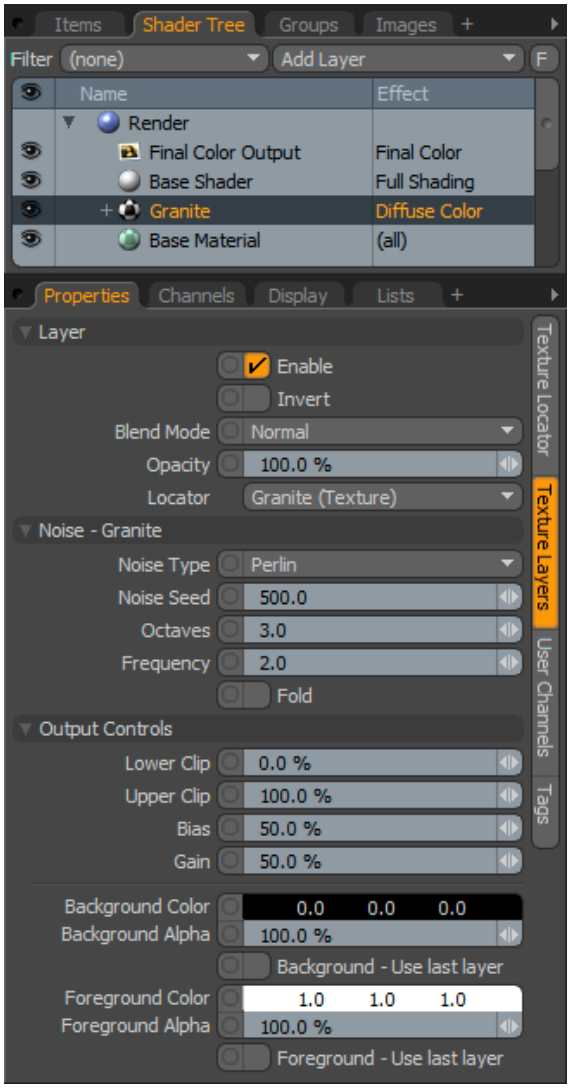
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - Granite --

Noise Type: This option determines the look of the distortion by providing several Noise function types.

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Fold: This option creates a folded variant, resulting in a denser noise pattern.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

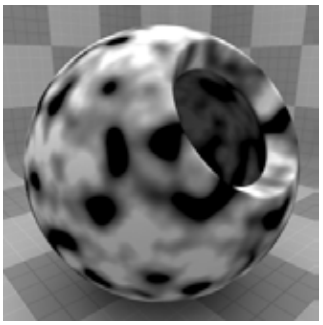
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Hybrid



The Hybrid texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Hybrid texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader is similar to the noisy bumps provided by the FMB (Fractal Brownian Motion) texture, with the *Hybrid* texture having smoother valleys, making it well suited for craggy mountain tops.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

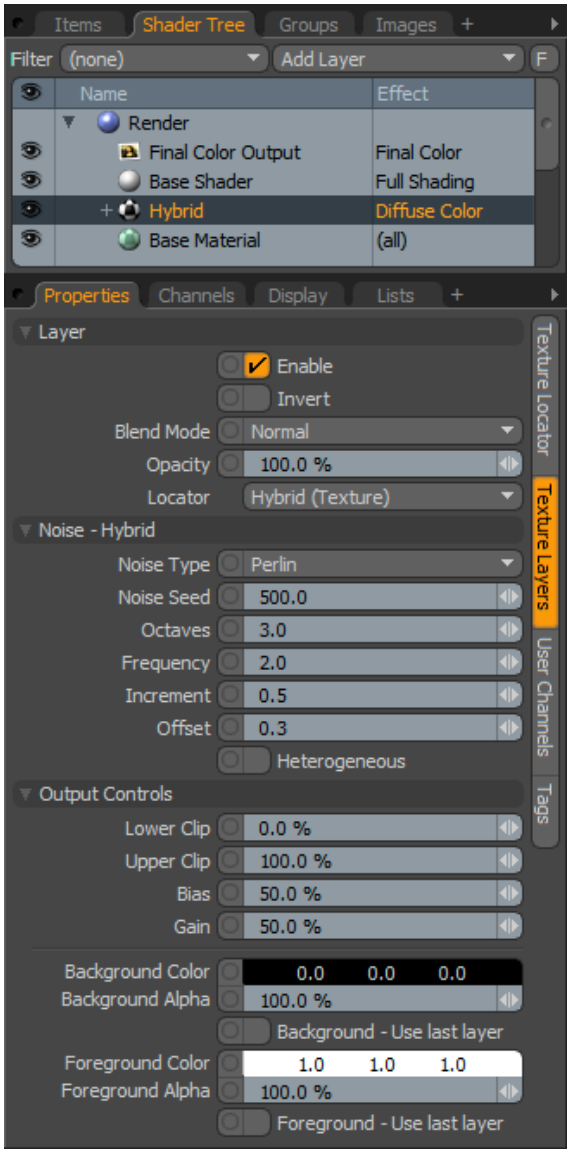
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - Hybrid --

Noise Type: This option determines the look of the distortion by providing several Noise function types.

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this

number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Offset: This value controls the detail of the texture. When the control is at a low value, the detail is very sharp, becoming increasingly faded as the parameter is increased.

Heterogeneous: When this option is enabled, the texture will switch from using the default fractal model to a heterogeneous terrain model.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

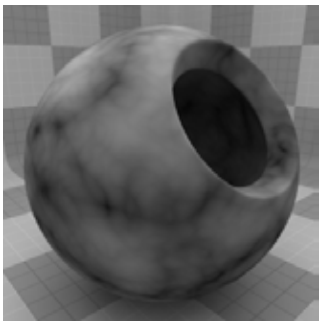
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Lump



The Lump texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Lump texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides *lumpy* noise, similar to billowing clouds.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

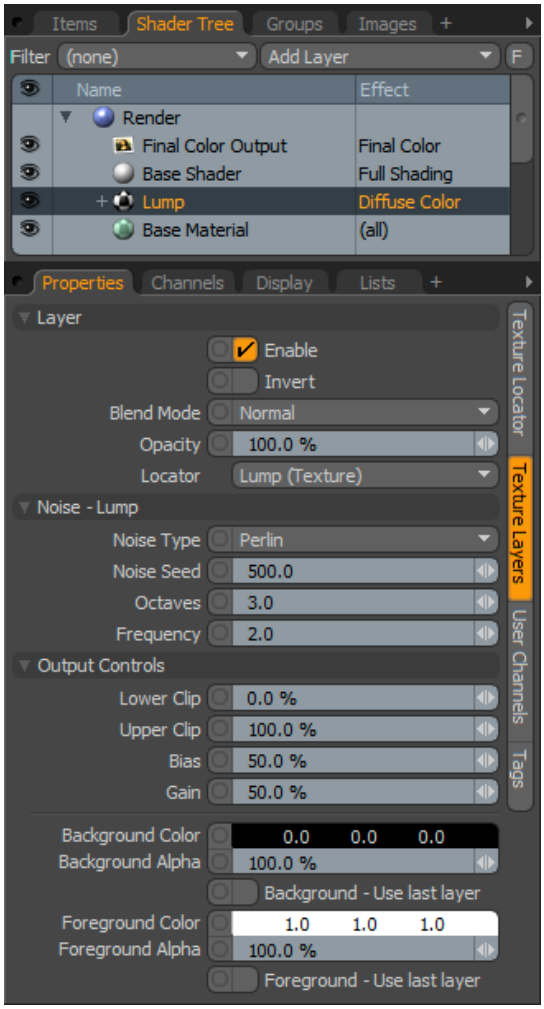
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - Lump --

Noise Type: This option determines the look of the distortion by providing several Noise function types.

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

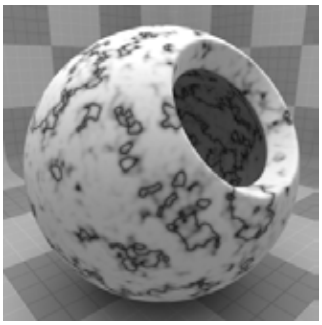
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Marble Noise



The Marble Noise texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Marble Noise texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides a *marble* effect pattern. However, unlike the Marble Vein shader, this is a noise based texture, giving it much more detail.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

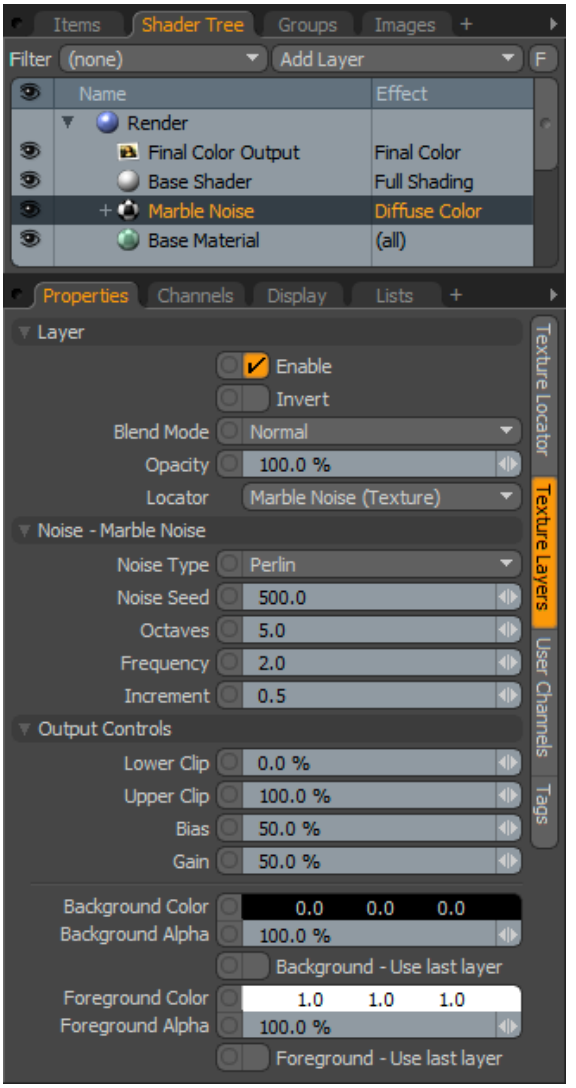
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - Marble Noise --

Noise Type: This option determines the look of the distortion by providing several Noise function types.

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed

noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

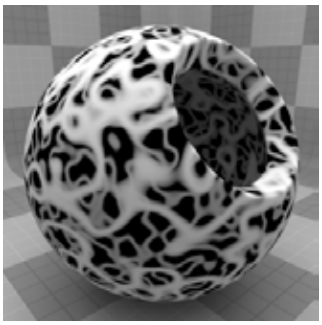
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Marble Vein



The Marble Vein texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Marble Vein texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides a *veiny marble* effect pattern. This texture provides a less detailed texture than the Marble Noise shader.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

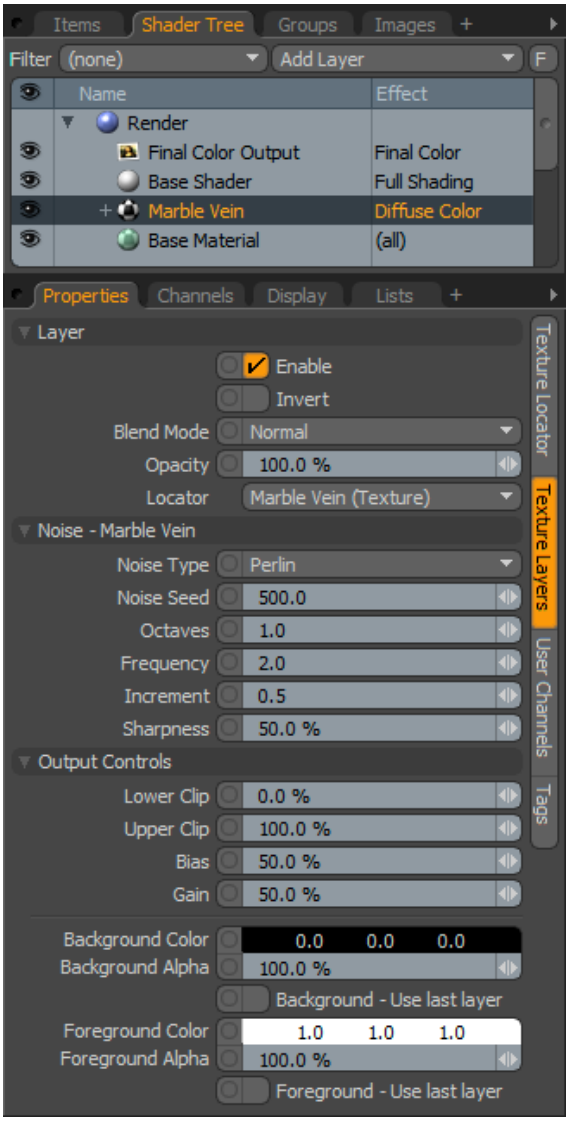
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - Marble Vein --

Noise Type: This option determines the look of the distortion by providing several Noise function types.

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Sharpness: This value controls the point at which the texture becomes more detailed.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

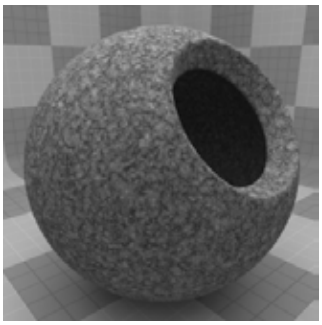
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Scar



The Scar texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Scar texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this shader provides a noisy ripped pattern rather like *scar* tissue or lava rock.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

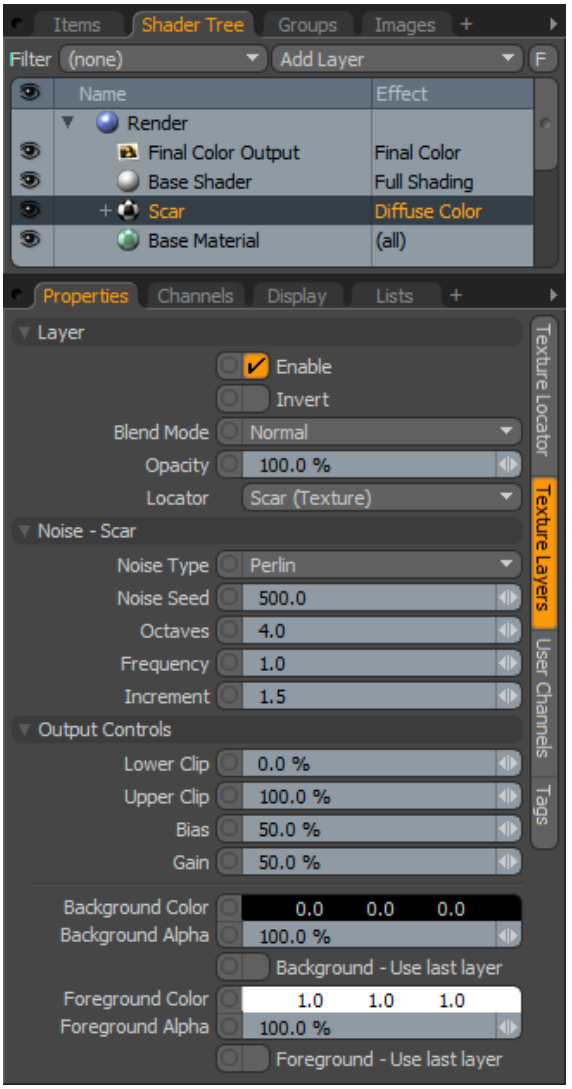
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - Scar --

Noise Type: This option determines the look of the distortion by providing several Noise function types.

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

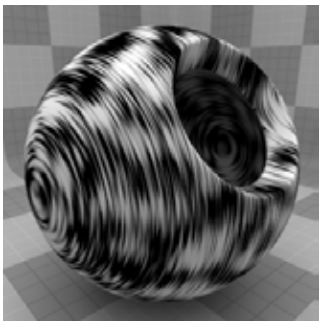
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Scruffed



The Scruffed texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Scruffed texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a surface that looks like it has been *scruffed* and scraped along its side.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

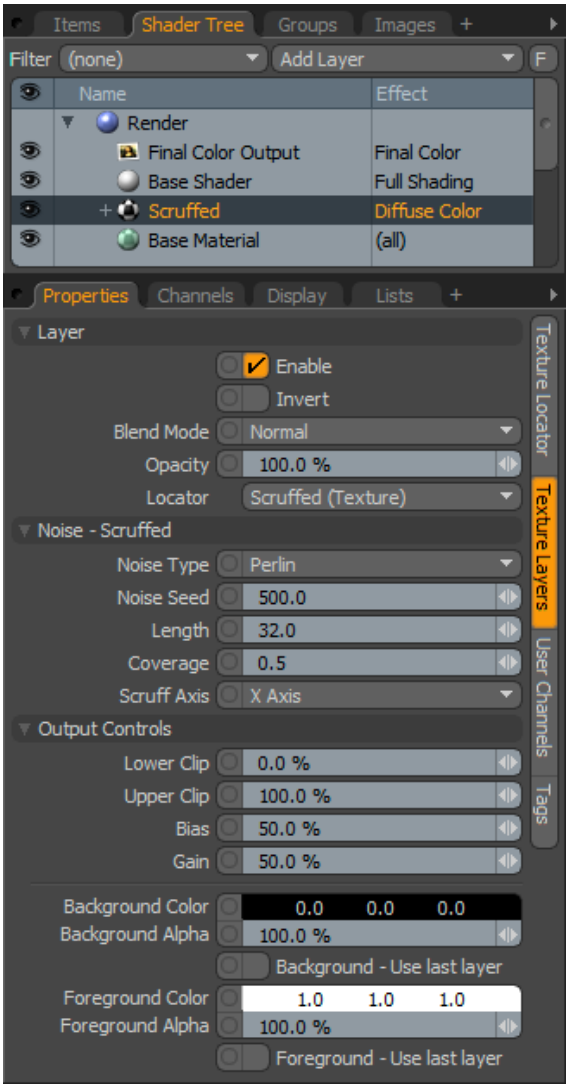
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - Scruffed --

Noise Type: This option determines the look of the distortion by providing several Noise function types.

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Length: This value controls the amount of the foreground color. Decreasing this value increases the overall coverage of the foreground color.

Coverage: This value controls the amount of the foreground color. Decreasing this value increases the overall coverage of the background color.

Scruff Axis: This value controls the axis along which the layers run.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

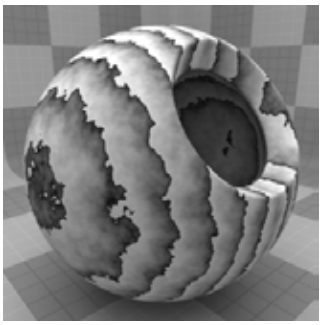
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Strata



The Strata texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Strata texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides a pattern similar to the layers, or *strata*, of sedimentary rock.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

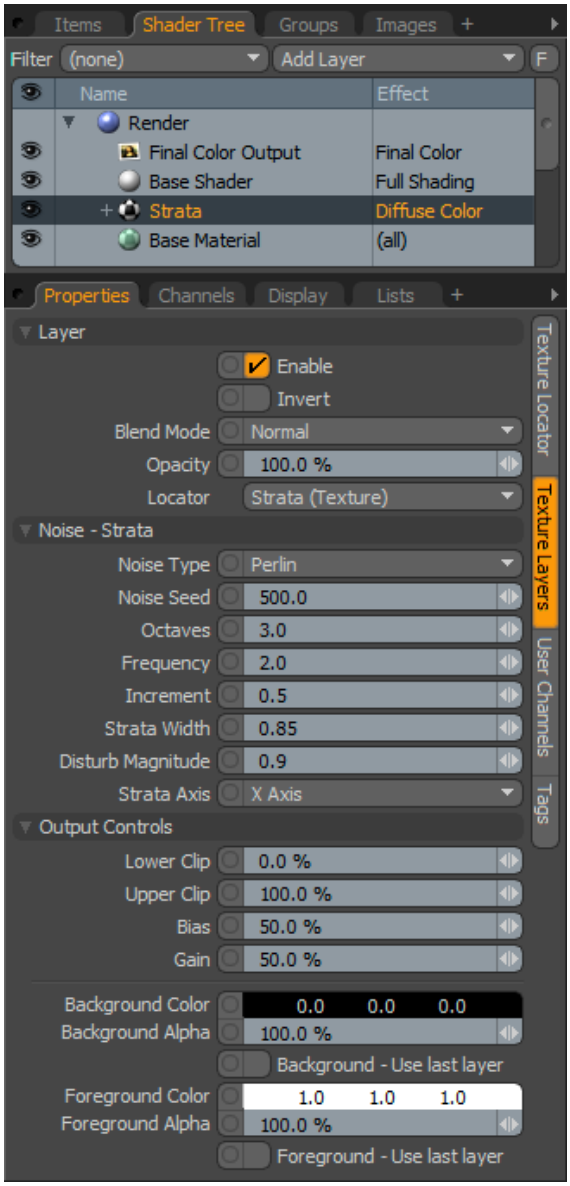
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - Strata --

Noise Type: This option determines the look of the distortion by providing several Noise function types.

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Strata Width: This value controls the amount of the foreground color. Decreasing this value increases the overall coverage of the foreground color.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Strata Axis: This value controls the axis along which the layers run.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

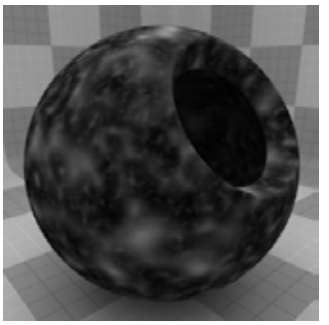
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Stucco



The Stucco texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Stucco texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides *stucco*-type bumps, similar to those found on rendered walls or stippled ceilings.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - Stucco --

Noise Type: This option determines the look of the distortion by providing several Noise function types.

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Power: This value controls the intensity of the stipples. As this number is increased, the background color becomes more prevalent, along with delineation between colors.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

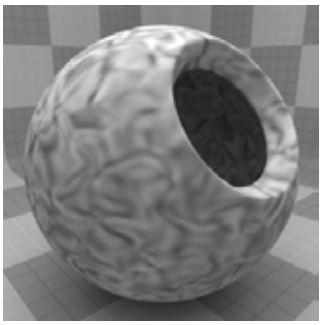
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Vector Bozo



The Vector Bozo texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Vector Bozo texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a noise pattern with vector-based disturbances.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

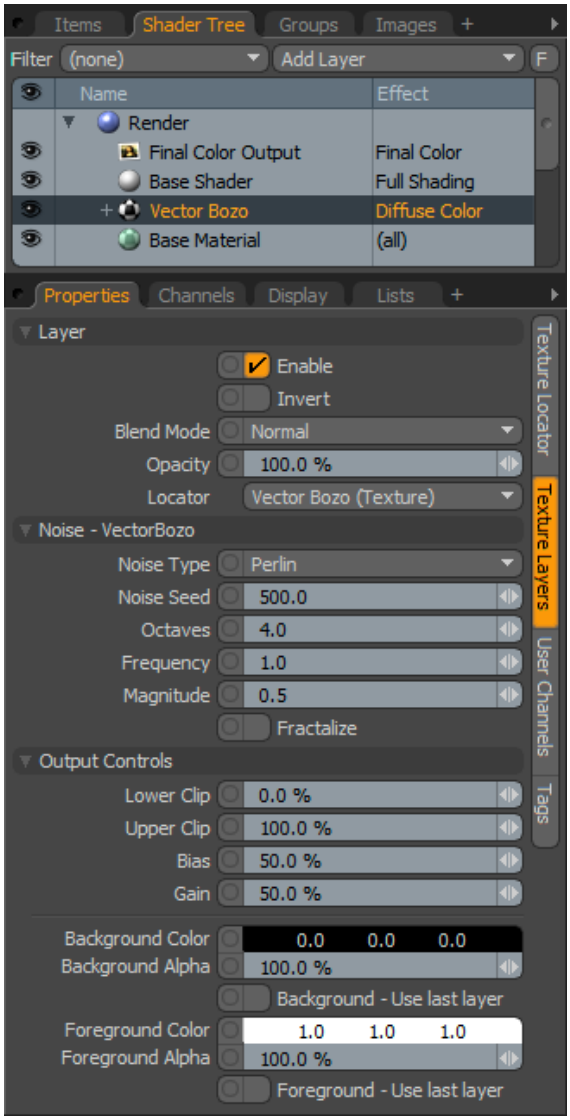
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - Vector Bozo --

Noise Type: This option determines the look of the distortion by providing several Noise function types.

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Magnitude: This value controls the magnitude or strength of each iteration of the noise. Increasing this value tightens the delineation between the background and foreground color.

Fractalize: This value determines whether or not the texture uses the fractalize algorithm, based on Fractal Brownian Motion (FBM).

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value,

truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

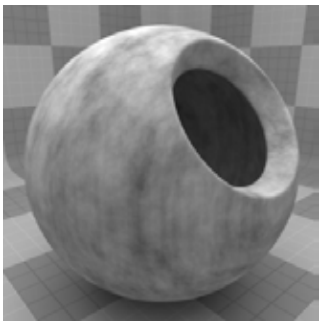
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Wrapped FBM



The Wrapped FBM texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Wrapped FBM texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader is the same as the FBM (*Fractal Brownian Motion*) texture except that the noise is wrapped radially.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise - Wrapped FBM --

Noise Type: This option determines the look of the distortion by providing several Noise function types.

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing

this value creates more turbulence, "stirring up" the pattern.

Magnitude: This value controls the magnitude or strength of each iteration of the noise. Increasing this value tightens the delineation between the background and foreground color.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

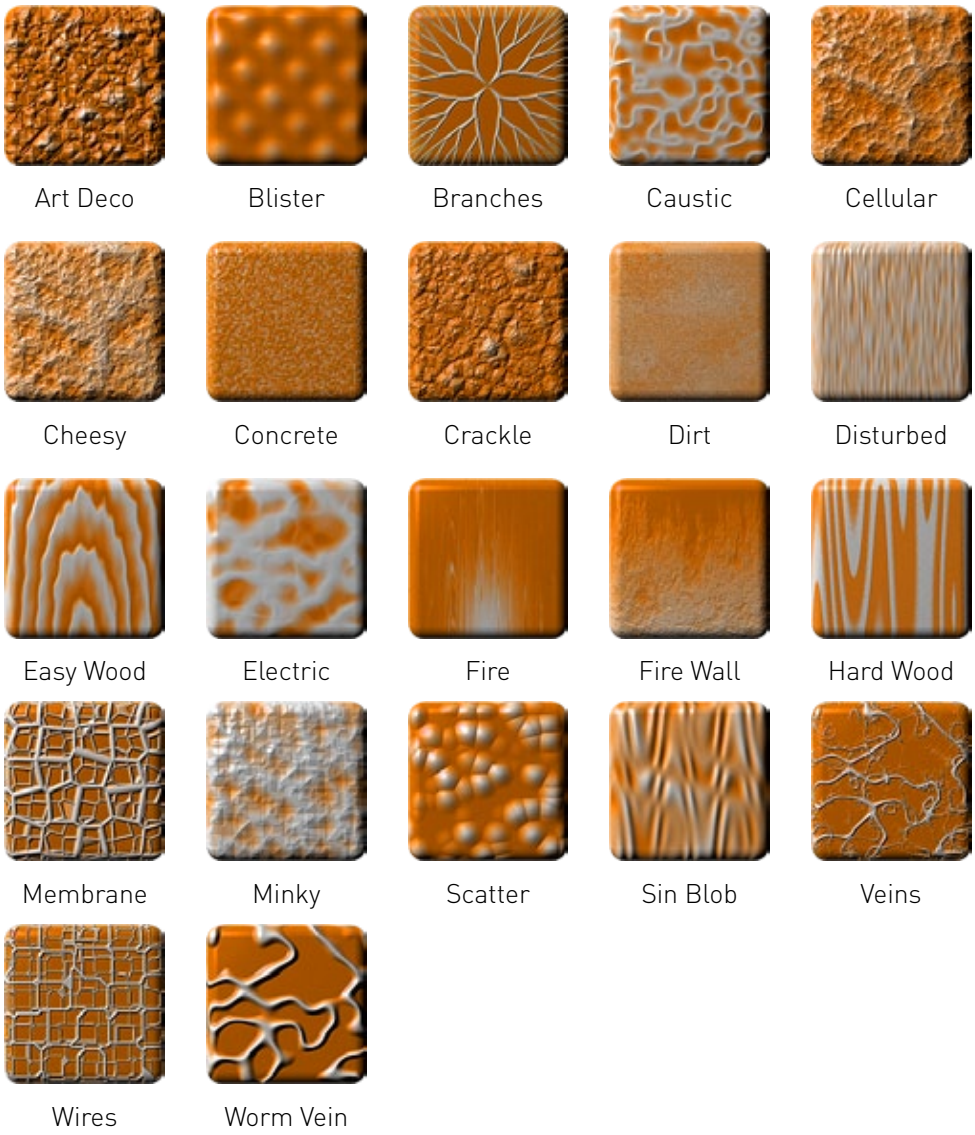
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

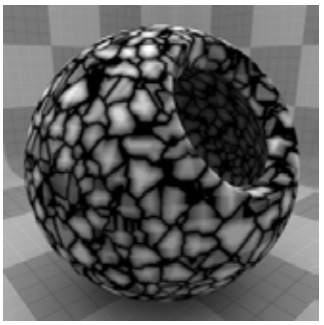
Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Organic



Art Deco



The Art Deco texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Art Deco texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader is similar to the *Crackle* shader, and provides a random crack-like pattern, muck like a dried-up river bed. The Art Deco shader differs from the Crackle shader in that distance between the cells has been changed to give them a looser, more organic feel.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

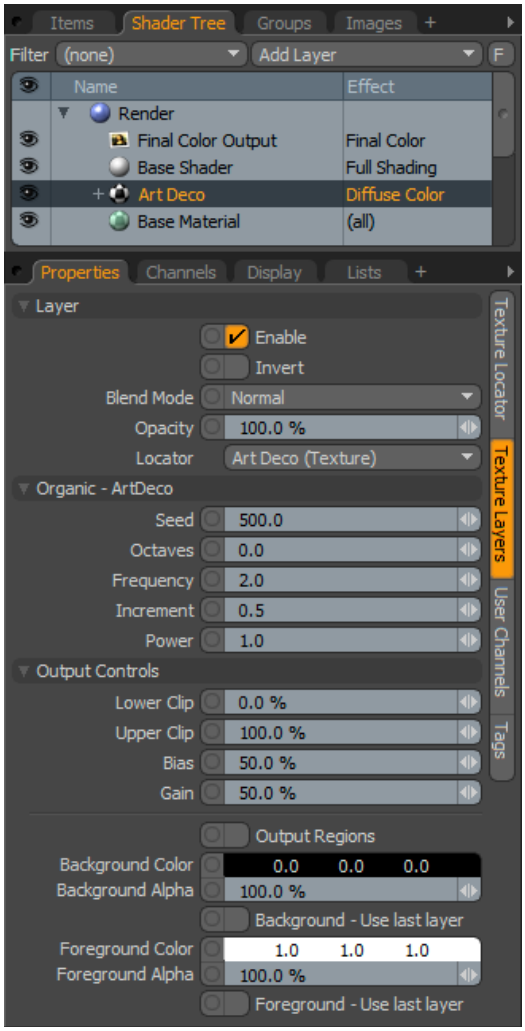
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Art Deco --

Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Power: This value controls the degree to which the edges of the texture are mixed up.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

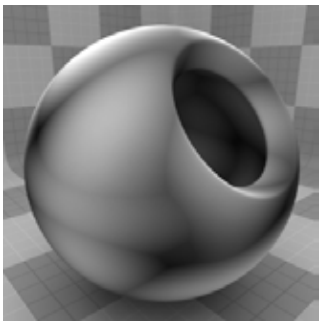
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Blister



The Blister texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Blister texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular is a pattern similar to leopard or dalmatian spots. These spots can also be applied as bumps to create a smooth *blistered* or pitted effects.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Blister --

Blister: This value defines the number of blister levels within the texture.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very

sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

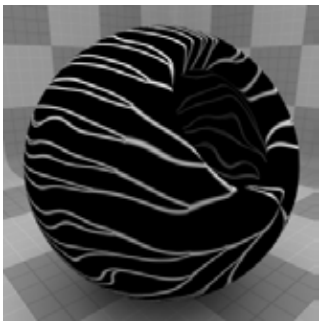
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Branches



The Branches texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Branches texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides hard-edged vein-like patterns that *branch* out from the center of the texture.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

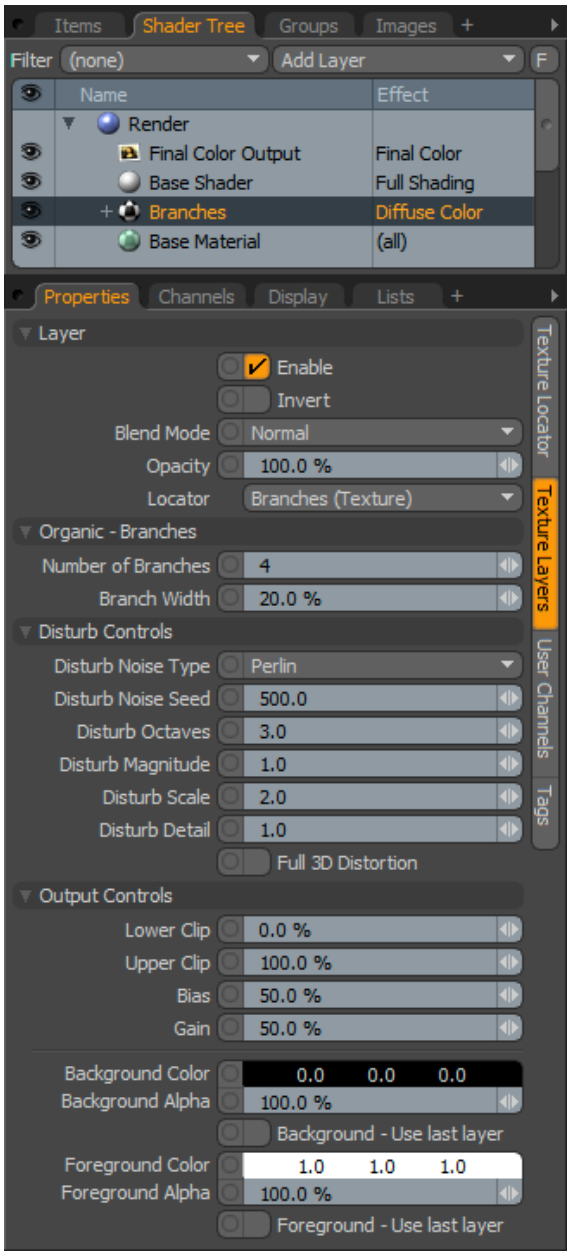
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the

layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Branches --

Number Of Branches: This value defines the number of branches that emanate from the center.

Branch Width: This value defines the width of the individual branches.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/ Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/ Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the

background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

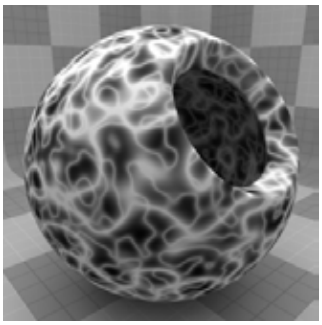
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Caustic



The Caustic texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Caustic texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides a watery, *caustic* noise, similar to how light plays benetath the surface of a swimming pool.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

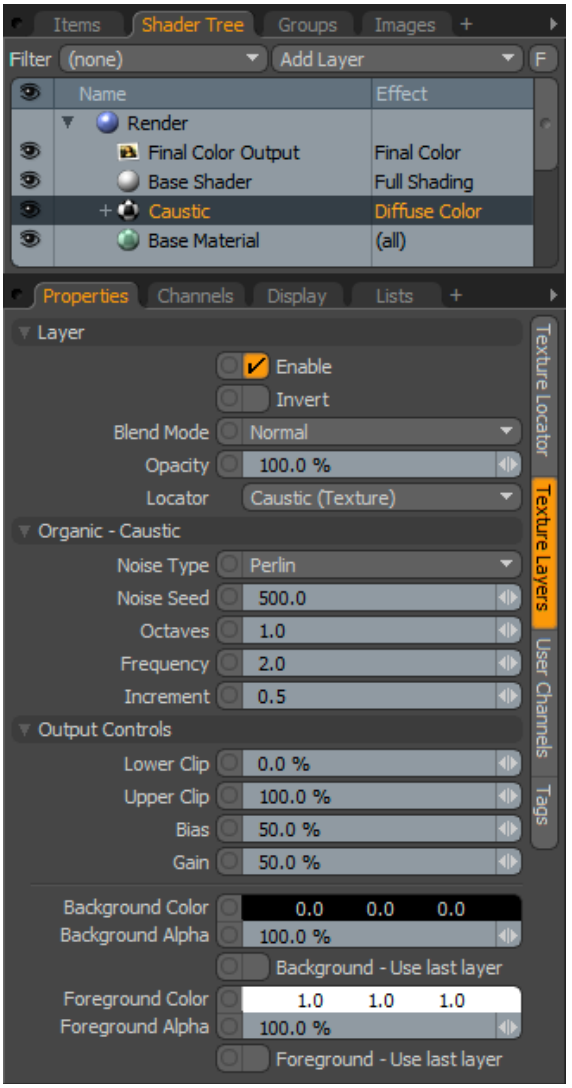
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Caustic --

Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing

this value creates more turbulence, "stirring up" the pattern.

Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

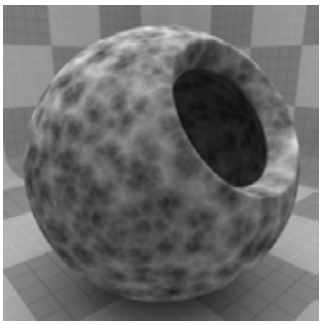
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Cellular



The Cellular texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Cellular texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides a random *cell*-like pattern, similar to clusters of varying sized circles. The texture can also be bumped and colored to provide a wide variation of organic patterns.

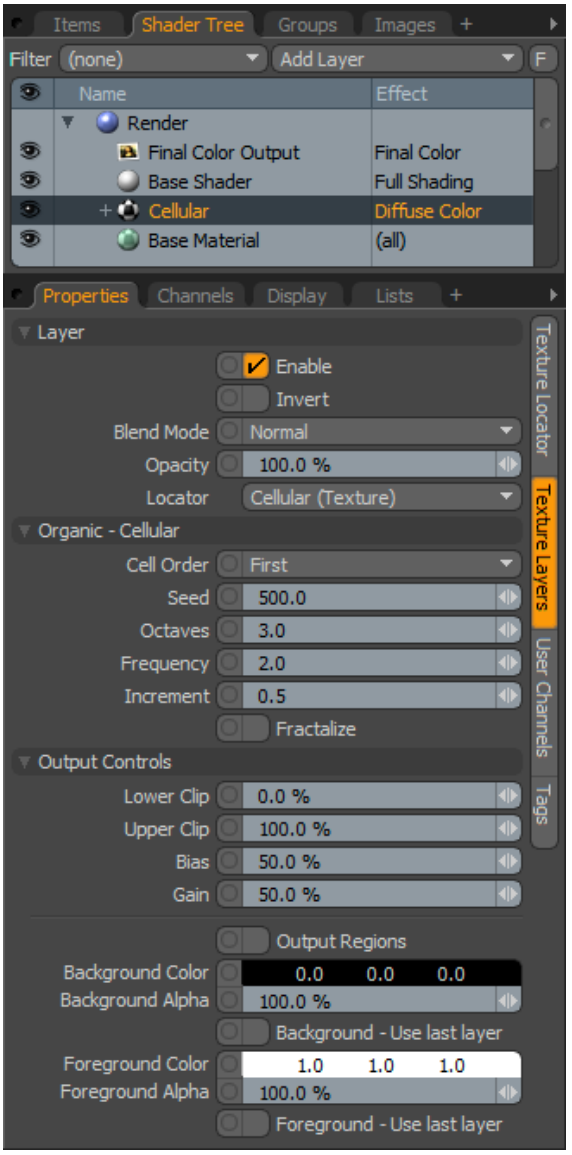
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.



Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Cellular --

Cell Order: This option determines the cell order used for texture creation-

First - Nearest Cell

Second - Next Nearest Cell

Third - Next Nearest Cell

Fourth - Furthest Cell

Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the

turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Fractalize: This option determines whether or not the texture uses the fractalize algorithm, based on Fractal Brownian Motion (FBM).

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

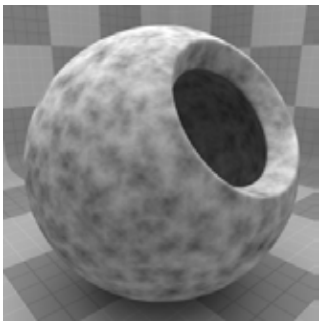
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Cheesy



The Cheesy texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Cheesy texture can be addressed by its two zones, the 'Backgrounds' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader is similar to the *Cellular* shader except that the distance between cells have been squared off. This texture can be used to create a rectangular pattern for machinery-like surfaces.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

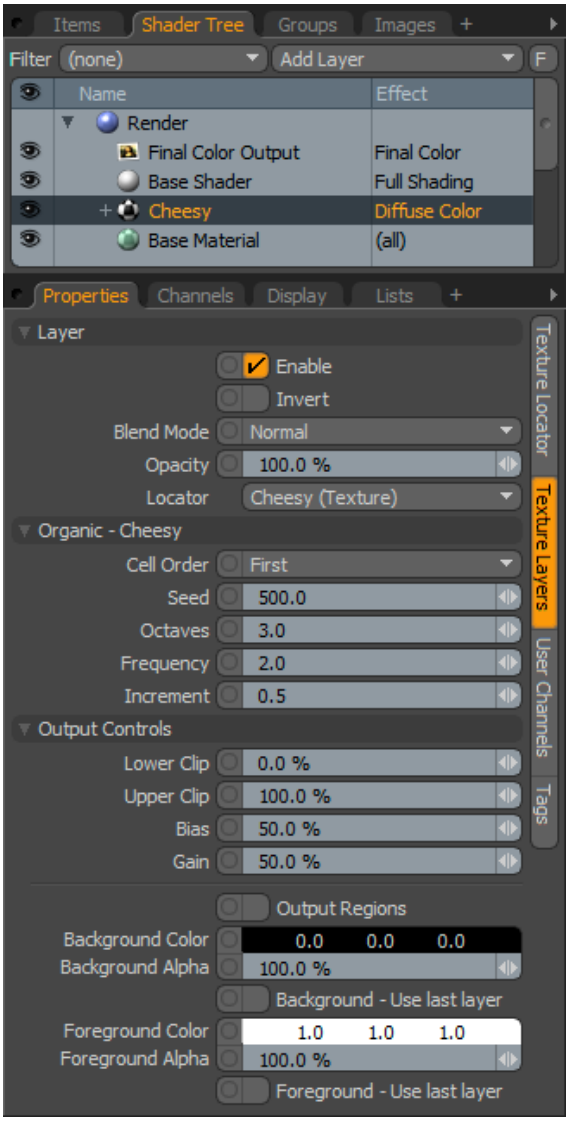
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Cheesy --

Cell Order: This option determines the cell order used for texture creation-

First - Nearest Cell

Second - Next Nearest Cell

Third - Next Nearest Cell

Fourth - Furthest Cell

Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

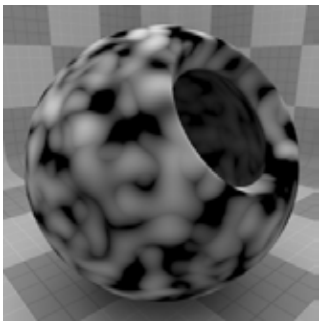
Foreground Color/Value: This value determines the Color (or Value) of the texture's

Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Concrete



The Concrete texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Concrete texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this provides a looks similar to *concrete*, with three interleaved bump map patterns.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

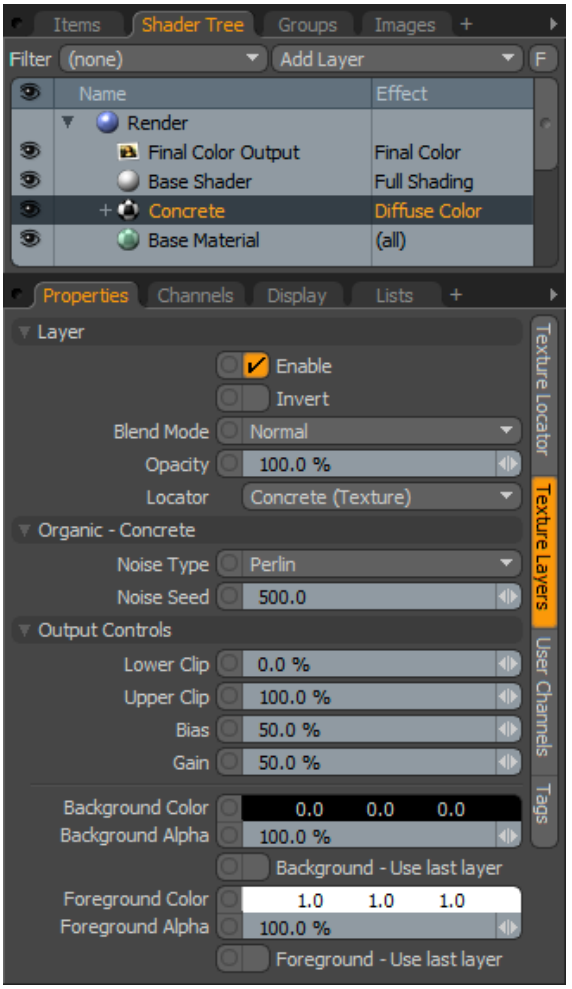
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Concrete --

Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

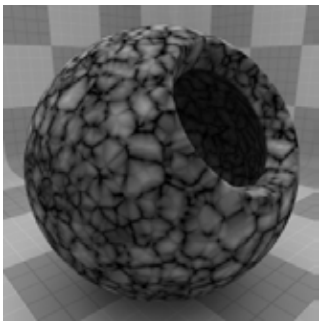
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Crackle



The Crackle texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Crackle texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides a random *crackle* pattern, similar to a dried-up river bed. The shader can also be used to create veins or random lattice-type patterns.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

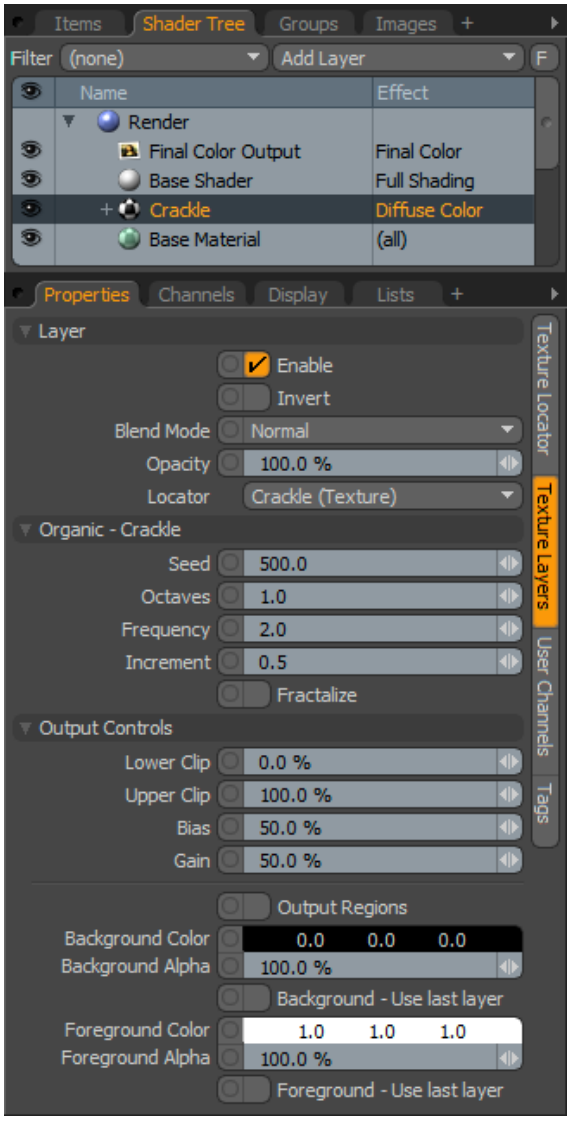
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Crackle --

Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Fractalize: This value determines whether or not the texture uses the fractalize algorithm, based on Fractal Brownian Motion (FBM).

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

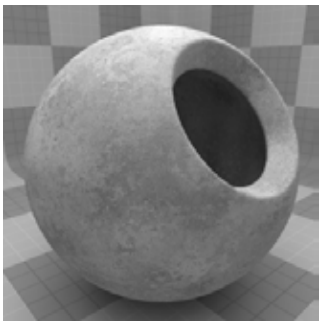
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Dirt



The Dirt texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Dirt texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a way to *dirty* up objects so they won't look quite so pristine. The dirt function can be applied in different ways, providing a splotchy, dusty look as well as a more even, scattered *dirt*.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

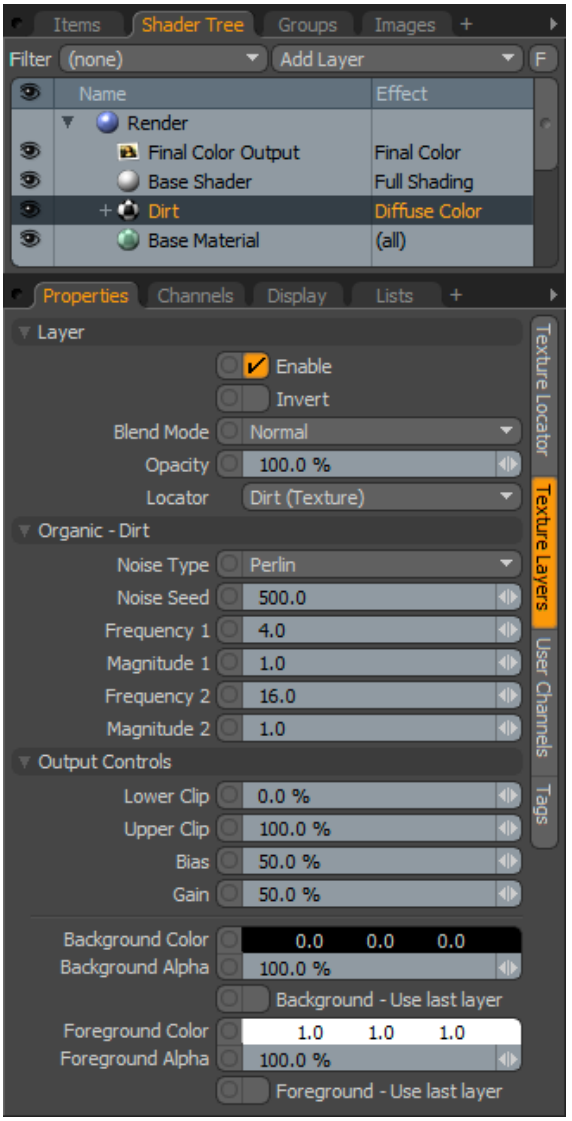
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Dirt --

Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Frequency 1: This value controls the first frequency of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Magnitude 1: This value controls the first magnitude of each iteration of the noise.

Increasing this value tightens the delineation between the background and foreground color.

Frequency 2: This value controls the second frequency of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Magnitude 2: This value controls the second magnitude of each iteration of the noise. Increasing this value tightens the delineation between the background and foreground color.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

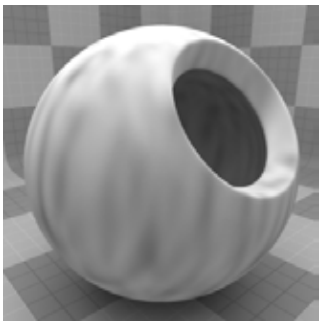
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Disturbed



The Disturbed texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Disturbed texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader produces a two wave pattern, with each wave affecting and potentially overlapping the other. This texture can be used to create a water surface or criss-cross pattern.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

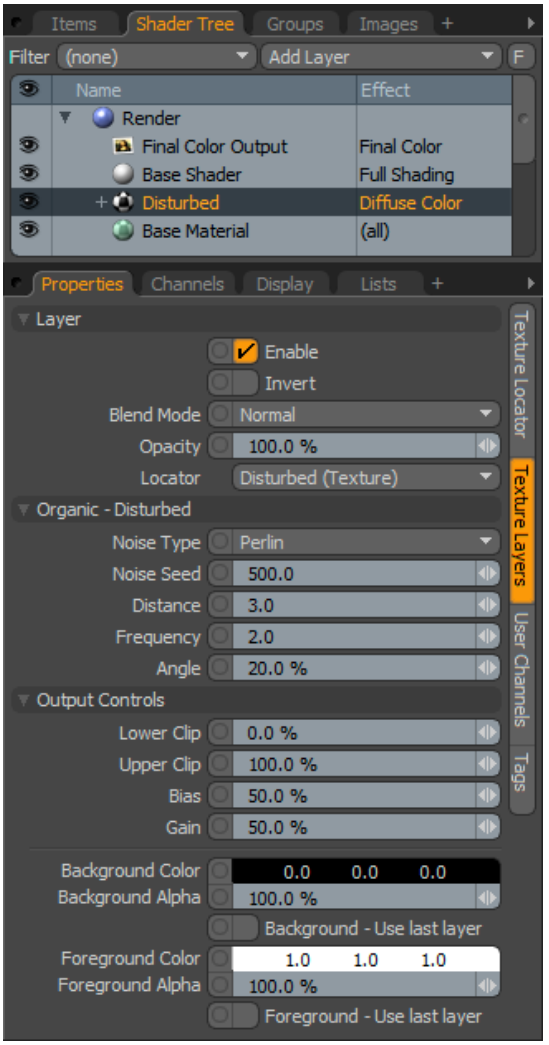
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Disturbed --

Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Distance: This value controls the distance traveled by the wave sources.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Angle: This value controls the angle between the two wave sources.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

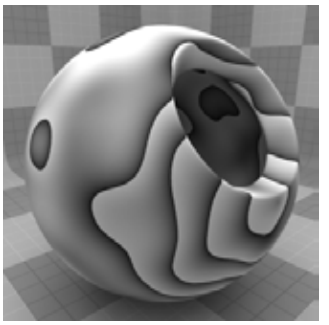
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Easy Wood



The Easy Wood texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Easy Wood texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader creates a *wood* grain pattern with concentric rings around the specified axis. A bump can also be applied to generate some relief to the wood grain.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

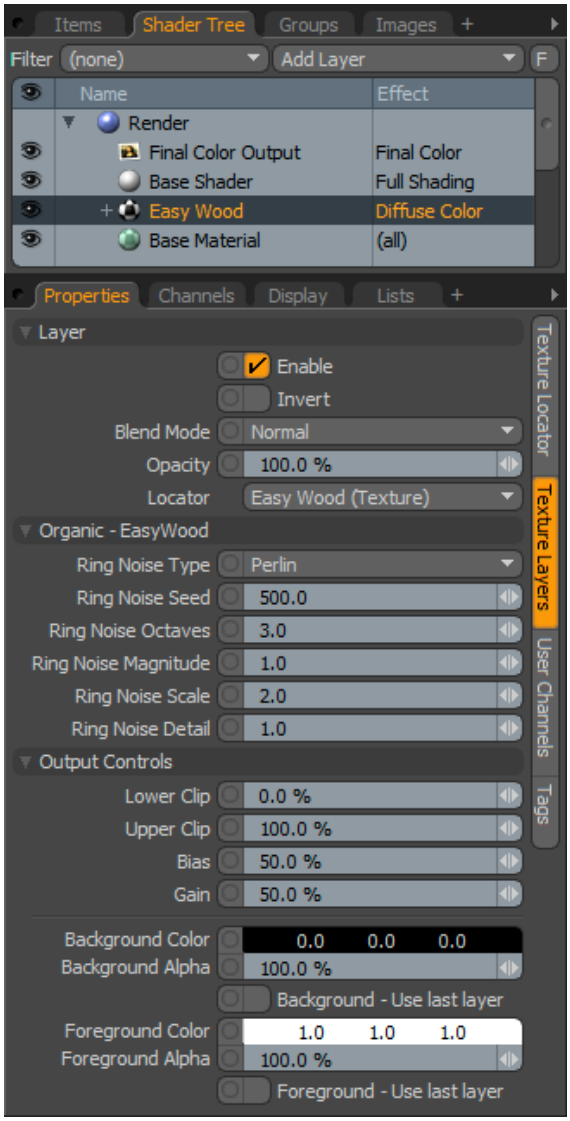
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Easy Wood --

Ring Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Ring Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Ring Noise Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Ring Noise Magnitude: This value controls the size or magnitude of each iteration of the noise.

Ring Noise Scale: This value controls the scale of each subsequent iteration of the noise.

Ring Noise Detail: This value controls the step size between each iteration of the noise.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

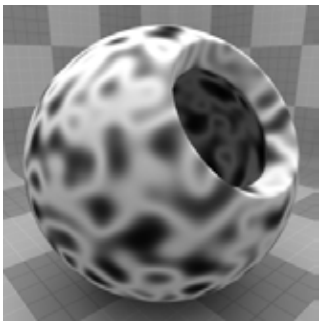
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Electric



The Electric texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Electric texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides a noisy pattern similar to large *electrical* splashes or discharges.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Electric --

Arc Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Arc Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Arc Noise Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Arc Noise Frequency: This value controls the size or magnitude of each iteration of the noise.

Arc Noise Increment: This value controls the scaling of each successive iteration.

Arc Width: This value controls the width of the electrical arc.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

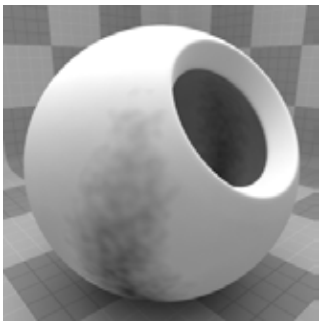
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Fire



The Fire texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Fire texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader simulates a bonfire or camp *fire* type of effect.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

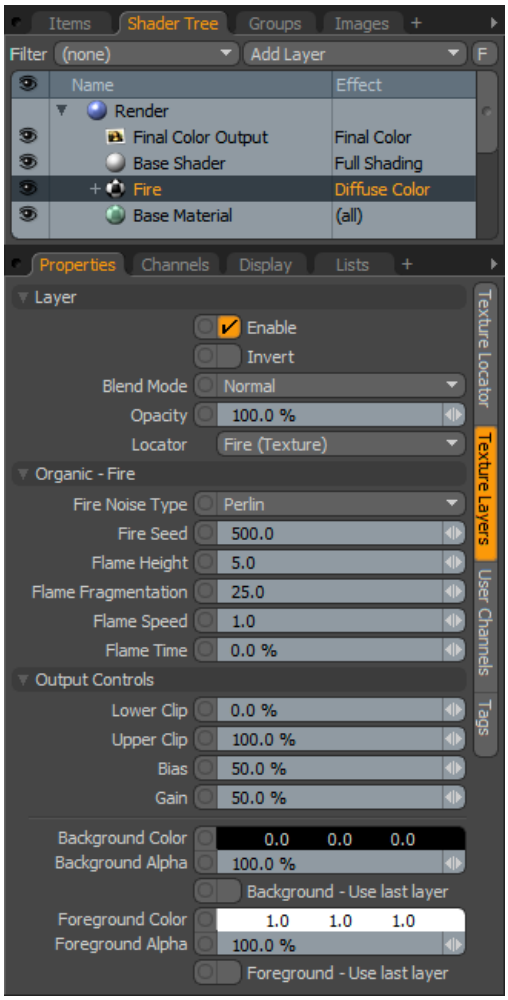
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Fire --

Fire Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Fire Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Flame Height: This value controls the height of the flame texture.

Flame Fragmentation: This value controls the fragmentation, or detail, of the flames.

Flame Speed: This value controls the speed of the flame when it is animated.

Flame Time: This value controls the overall timing of the animation, when used in conjunction the other available flame parameters.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

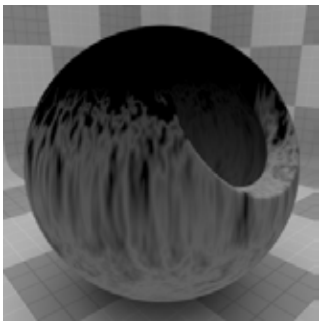
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Fire Wall



The Fire Wall texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Fire Wall texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader simulates a bonfire or camp *fire* type of effect.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

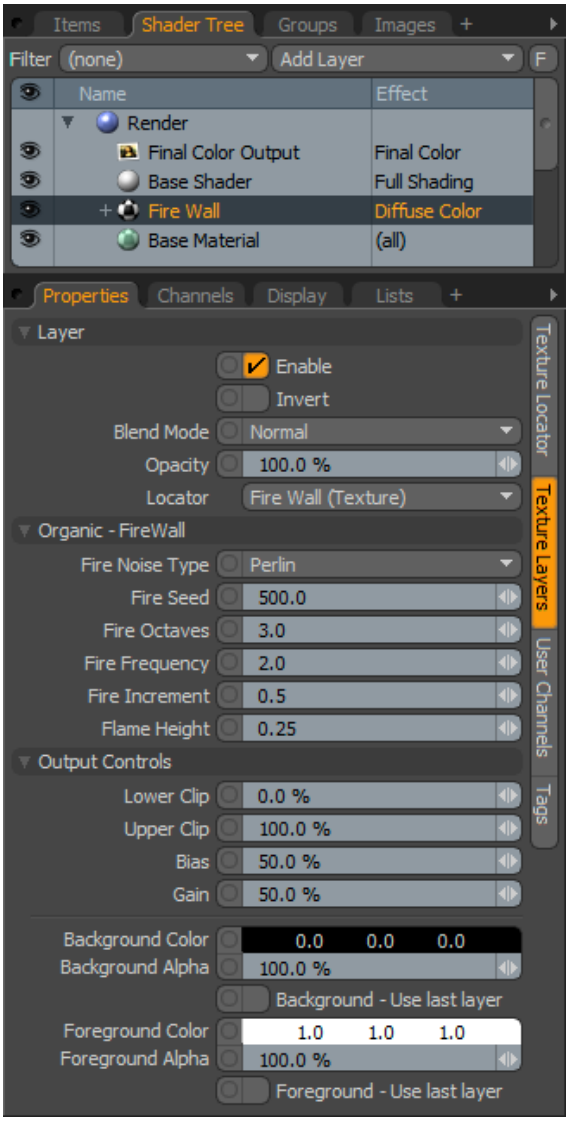
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Fire Wall --

Fire Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Fire Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Fire Octaves: This value defines the number of iterations used to create the texture. As this number is increased, the texture takes on more detail.

Fire Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Fire Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Flame Height: This value controls the height of the flame texture.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

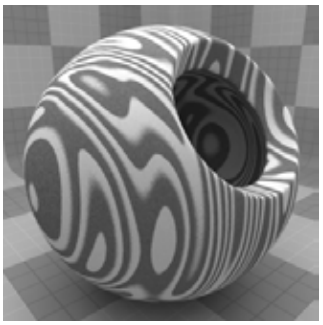
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Hard Wood



The Hard Wood texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Hard Wood texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular creates a *wood* effect. If differs from the *Easy Wood* texture in that it provides a grainier, noisier, wood pattern.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

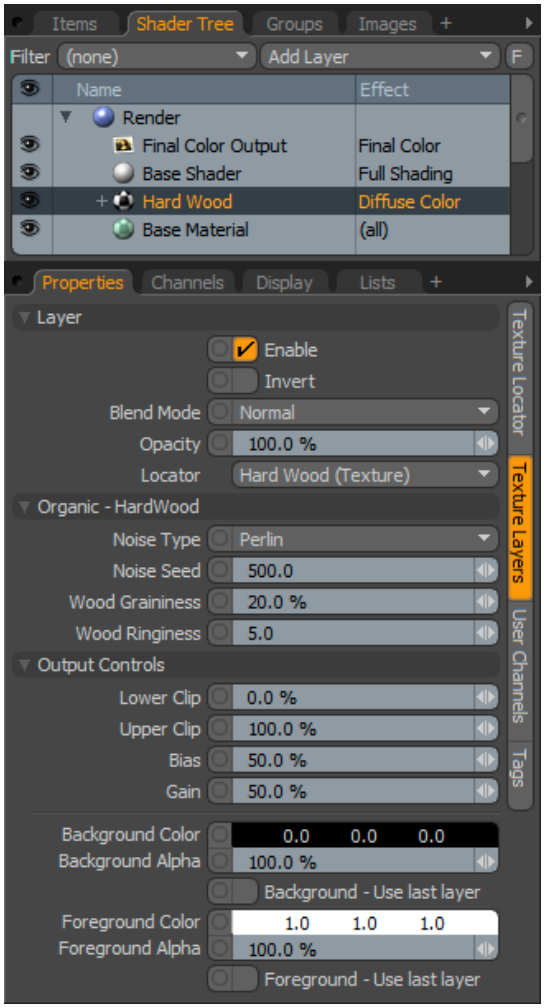
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Hard Wood --

Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Wood Graininess: This value controls the graininess of the wood rings.

Wood Ringiness: This value controls the amplitude of the wood rings.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

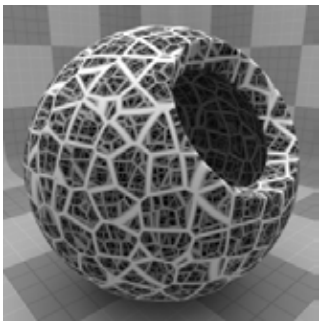
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Membrane



The Membrane texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Membrane texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader is a layered *membrane*, similar to the pith found in an Orange.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

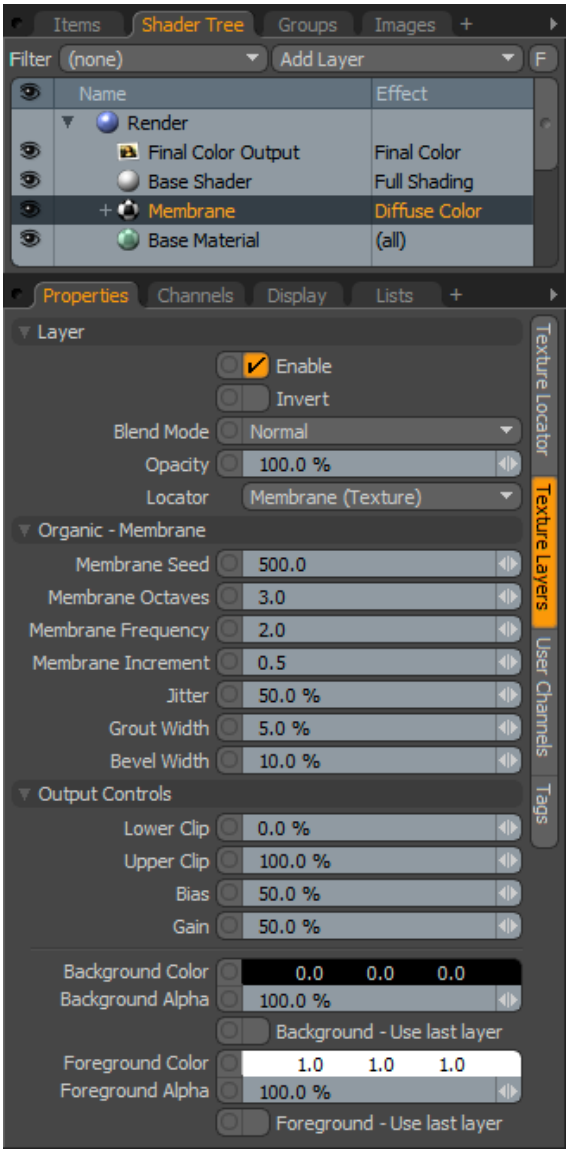
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Membrane --

Membrane Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Membrane Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Membrane Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Membrane increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Jitter: This value controls the jitter, with higher value causing more jitter in the membrane.

Grout Width: This value controls the grout width between cells.

Bevel Width: This value controls the bevel between the edges of the cells.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

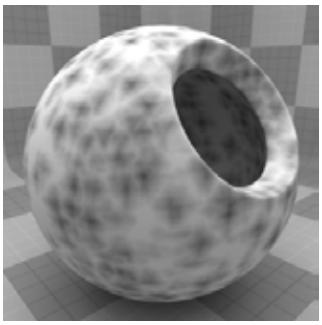
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Minky



The Minky texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Minky texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader is based on the *Cellular* shader, except that the distance between the cells has been derived in a linear manner. This creates a diamond-like surface pattern.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

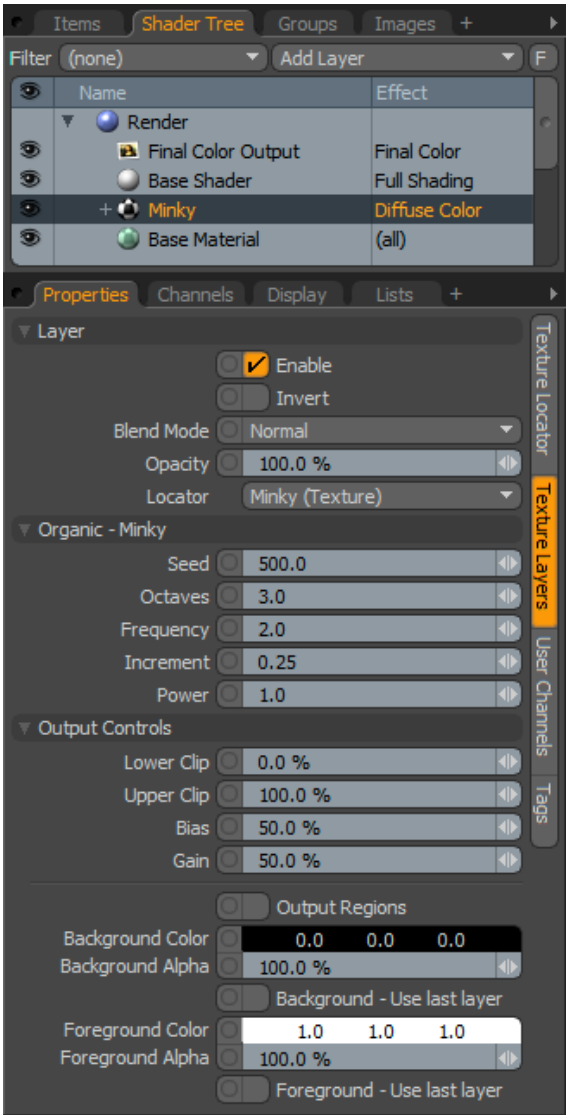
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Minky --

Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Power: This value controls the degree to which the edges of the texture are mixed up.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

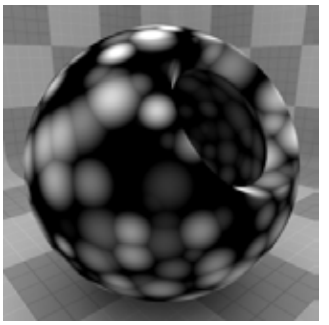
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Scatter



The Scatter texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Scatter texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a pseudo-random pattern of spheres, cubes or diamonds.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

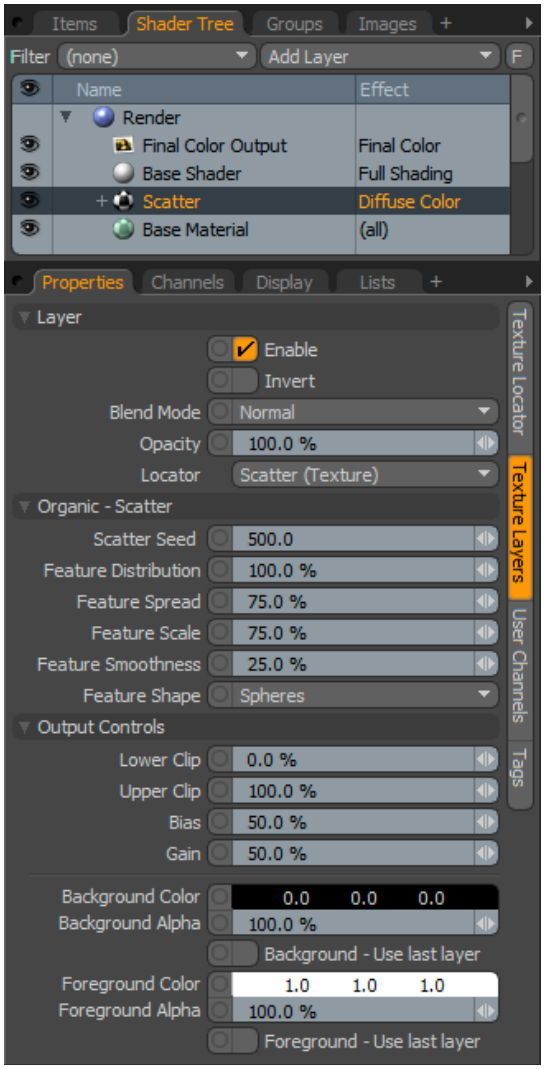
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Scatter --

Scatter Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Feature Distribution: This value defines the number of available features that are drawn in the texture. Lower values will result in less dense texture.

Feature Spread: This value controls the distance between the individual features.

Feature Scale: This value controls the scaling of the individual features. When the control is at a low value, the features will be smaller and more prevalent.

Feature Smoothness: This value controls the smoothness between the foreground and background colors of the feature.

Feature Shape: This control determines the shape of the individual features. The options are:

- Spheres
- Cubes

- Diamonds

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

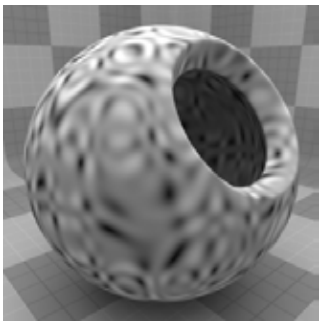
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Sin Blob



The Sin Blob texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Sin Blob texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader produces a blob-like sine wave pattern, based on a pseudo-randomizing process. A seed is provided to allow the creation of many different variations.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

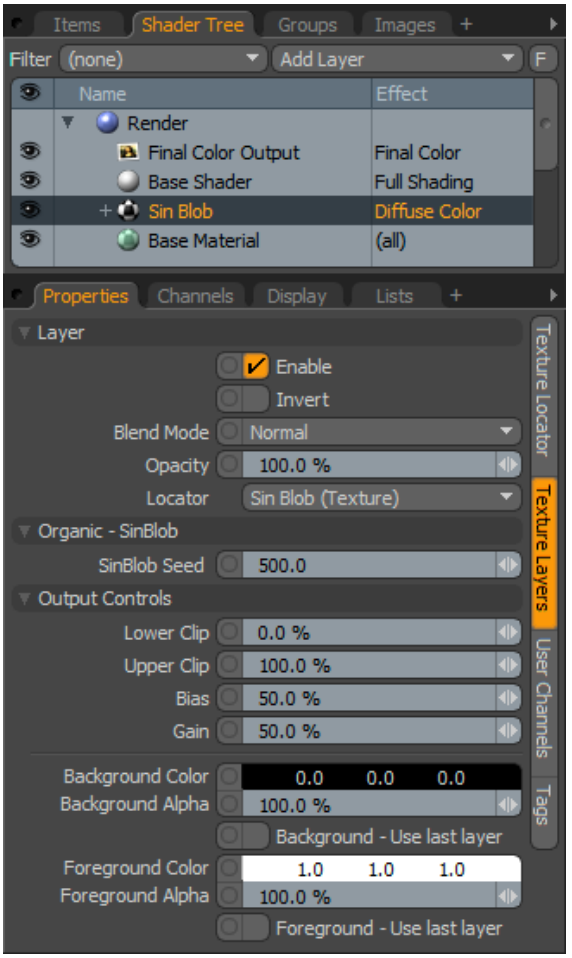
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Scatter --

SinBlob Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

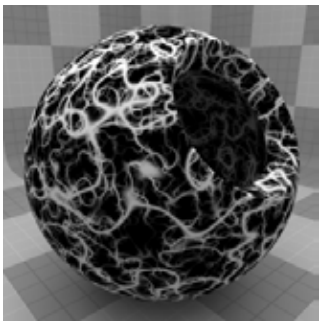
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Veins



The Veins texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Veins texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides *vein*-like patterns. This shader is ideal for texturing eyeballs or skin.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

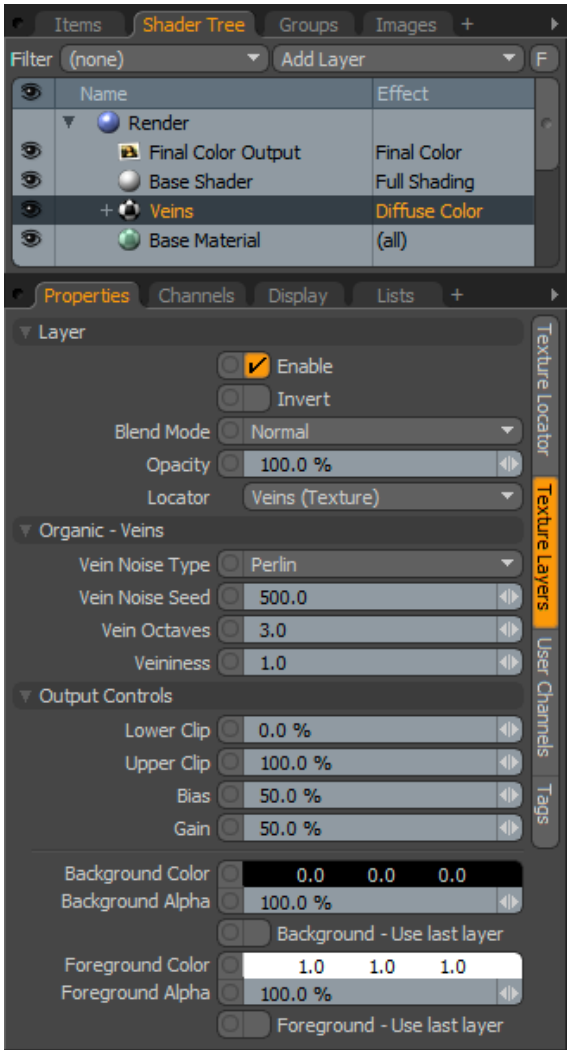
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Veins --

Vein Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Vein Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Vein Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Veininess: This value controls how thick and layered the veins will appear.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

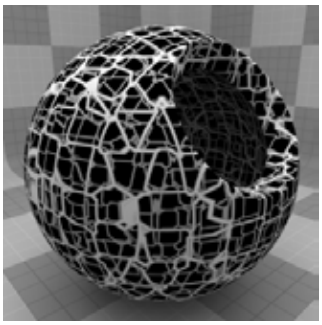
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Wires



The Wires texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Wires texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader creates a random, intersecting series of *wires*.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

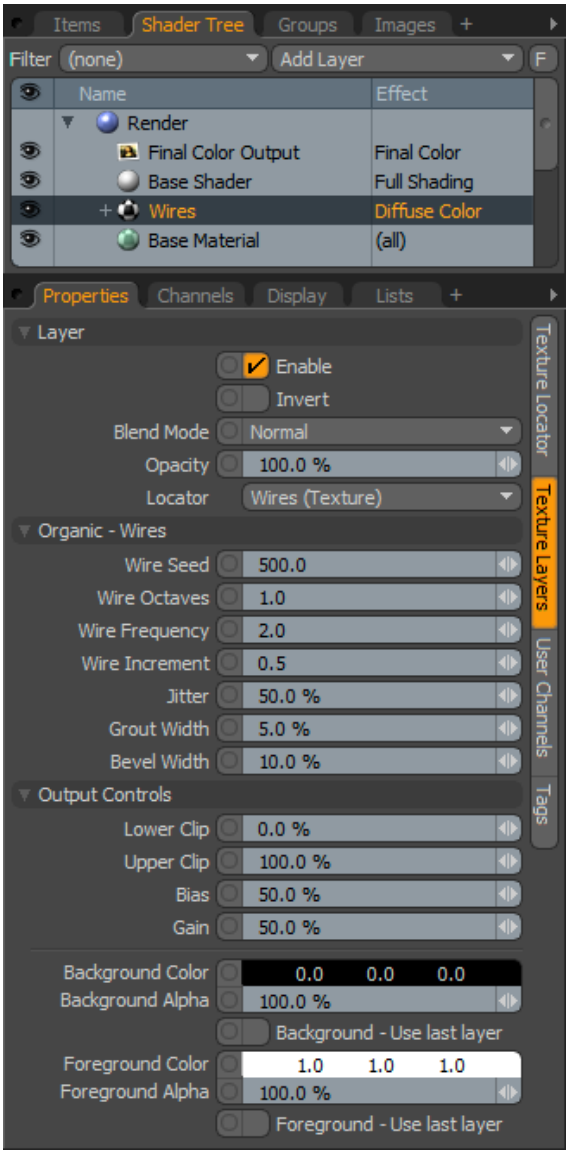
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Wires --

Wire Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Wire Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Wire Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Wire increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Jitter: This value controls the jitter, with higher values causing more jitter in the texture.

Grout Width: This value controls the grout width between wires.

Bevel Width: This value controls the bevel between the edges of the wires.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

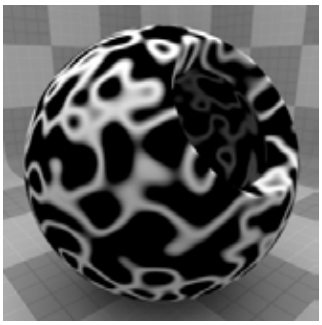
Foreground Color/Value: This value determines the Color (or Value) of the texture's

Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Worm Vein



The Worm Vein texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Worm Vein texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this shader creates a *wormy vein* type of pattern similar for veiny skin or electrical arcs.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

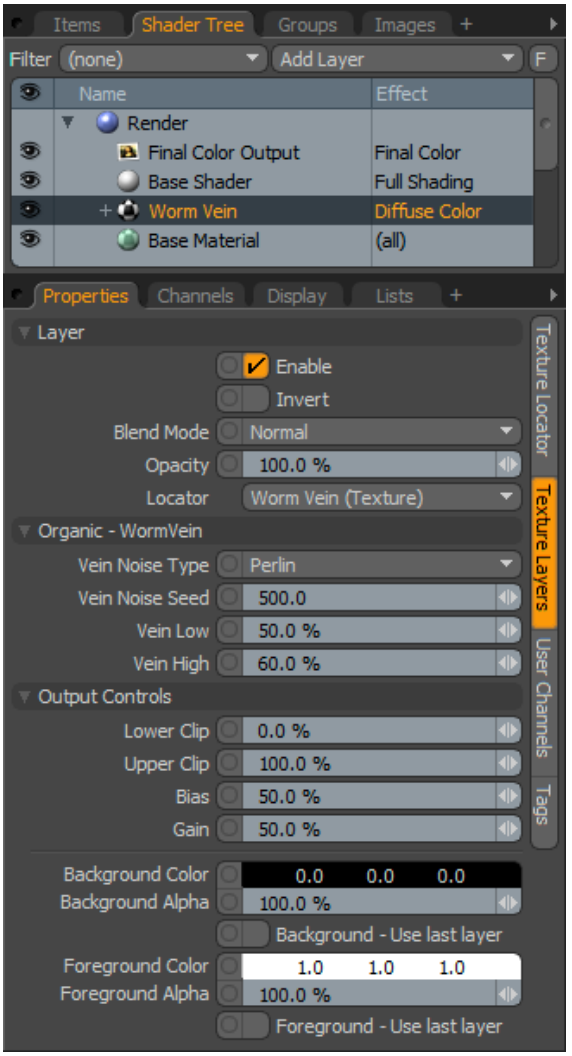
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Organic - Worm Vein --

Vein Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Vein Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Vein Low: This value controls how thick and layered the veins will appear.

Vein High: This value controls how thick and layered the veins will appear.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

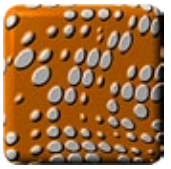
Skins

Camo

Crumpled

Dino Skin

Disease



Frog Skin

Grainy Wood

Leather

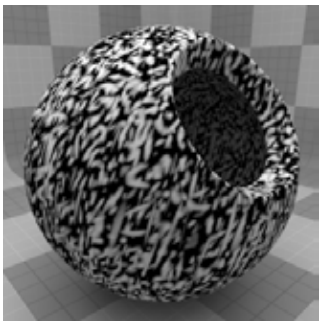
Monster

Pastella

Peened

Scratches

Camo



The Camo texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Camo texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader is a random desert *camouflage* kind of pattern.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

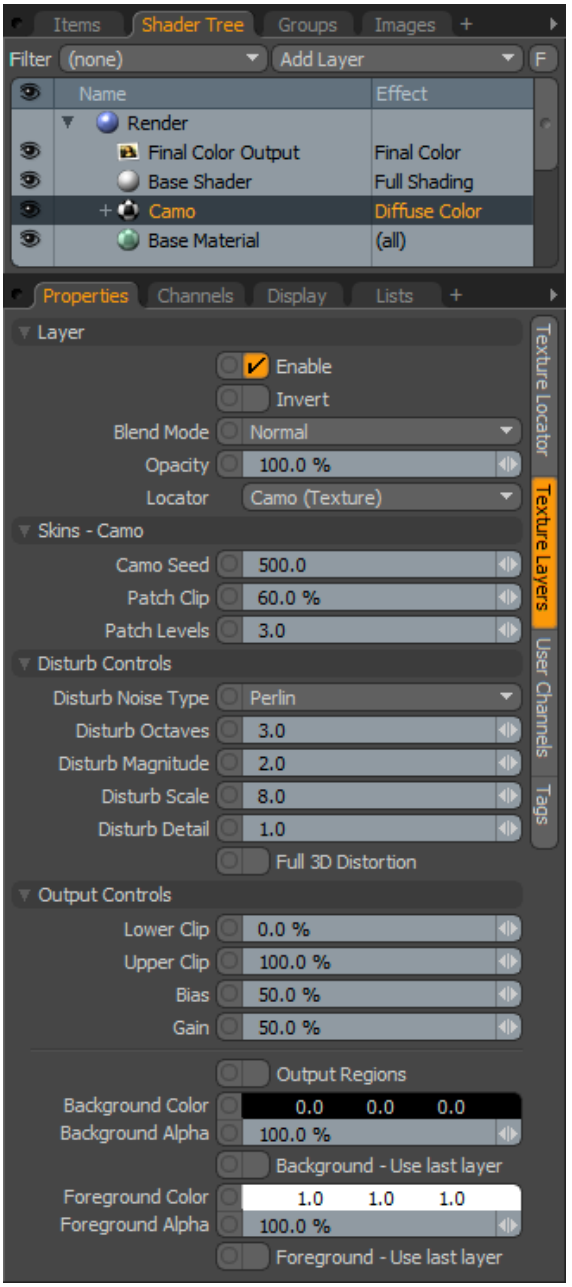
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Skins - Camo--

Camo Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Patch Clip: The 'Patch Clip' value determines where the texture shading transitions to the Background Color. The higher the value, the quicker the transition.

Patch Levels: The 'Patch Levels' value determines the number of layers of the Camo texturing. Additional levels will add additional detail, while also lightening the overall look of the texture.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

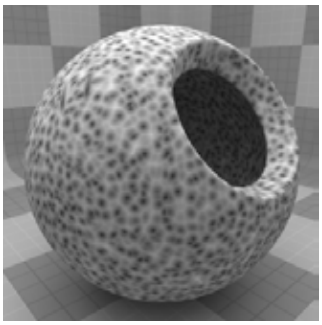
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Crumpled



The Crumpled texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Crumpled texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides an effct similar to used tinfoil, or a *crumpled*-and-flattened paper bag. By adding color, this can take on a planetary look, like the canals of Mars.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

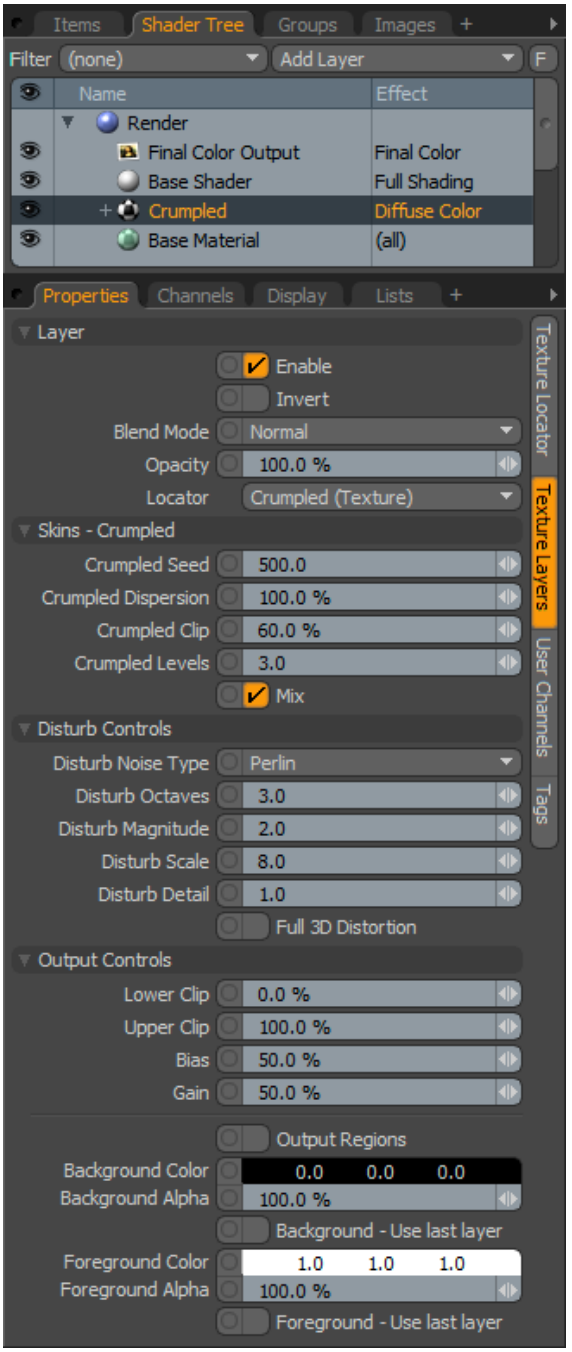
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Skins - Crumpled --

Crumpled Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Crumpled Dispersion: This value value determines how many of the crumpled sections are actually drawn.

Crumpled Clip: This value determines where the texture shading transitions to the Background Color. The higher the value, the quicker the transition.

Crumpled Levels: This value determines the number of layers of the texture. Additional levels will add additional detail, in turn also lightening the overall look of the texture.

Mix: This option, when checked, mixes the texture layers. When unchecked, the texture layers are stacked one upon the other.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

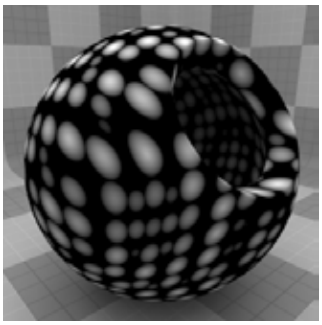
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Dino Skin



The Dino Skin texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Dino Skin texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides a scaly/bumpy/wrinkly *dinosaur skin*. The various controls can be used to make any number of alien or reptile skins.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

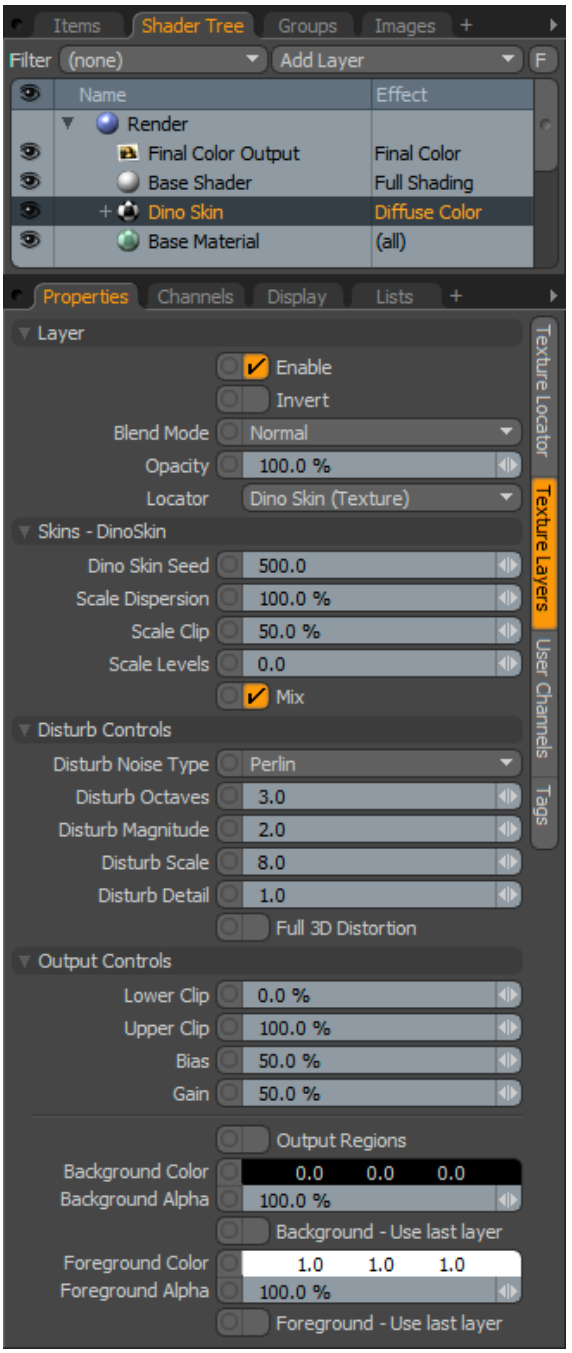
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Skins - Dino Skin --

Dino Skin Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Scale Dispersion: This value value determines how many of the bump sections are actually drawn.

Scale Clip: This value determines where the texture shading transitions to the Background Color. The higher the value, the quicker the transition.

Scale Levels: This value determines the number of layers of the texture.

Additional levels will add additional detail, in turn also lightening the overall look of the texture.

Mix: This option, when checked, mixes the texture layers. When unchecked, the texture layers are stacked one upon the other.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value,

users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

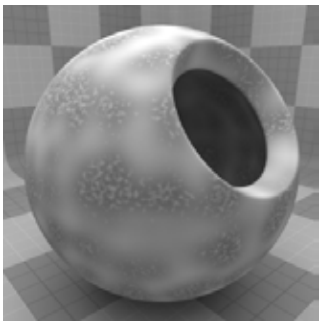
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Disease



The Disease texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Disease texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides a look similar to a skin *disease*. The controls can be used to create clumps of various sizes to make scab-like areas.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

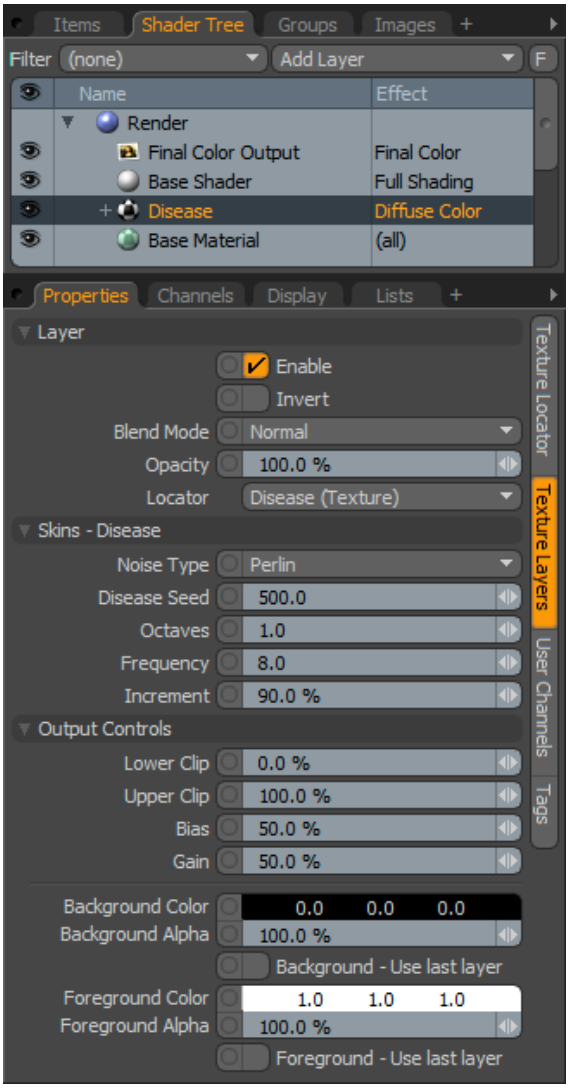
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Skins - Disease --

Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disease Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

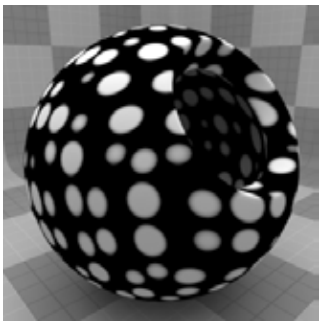
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Frog Skin



The Frog Skin texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Frog Skin texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides a speckled, bumpy, spotted pattern, similar to the skin of a *frog* or toad.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

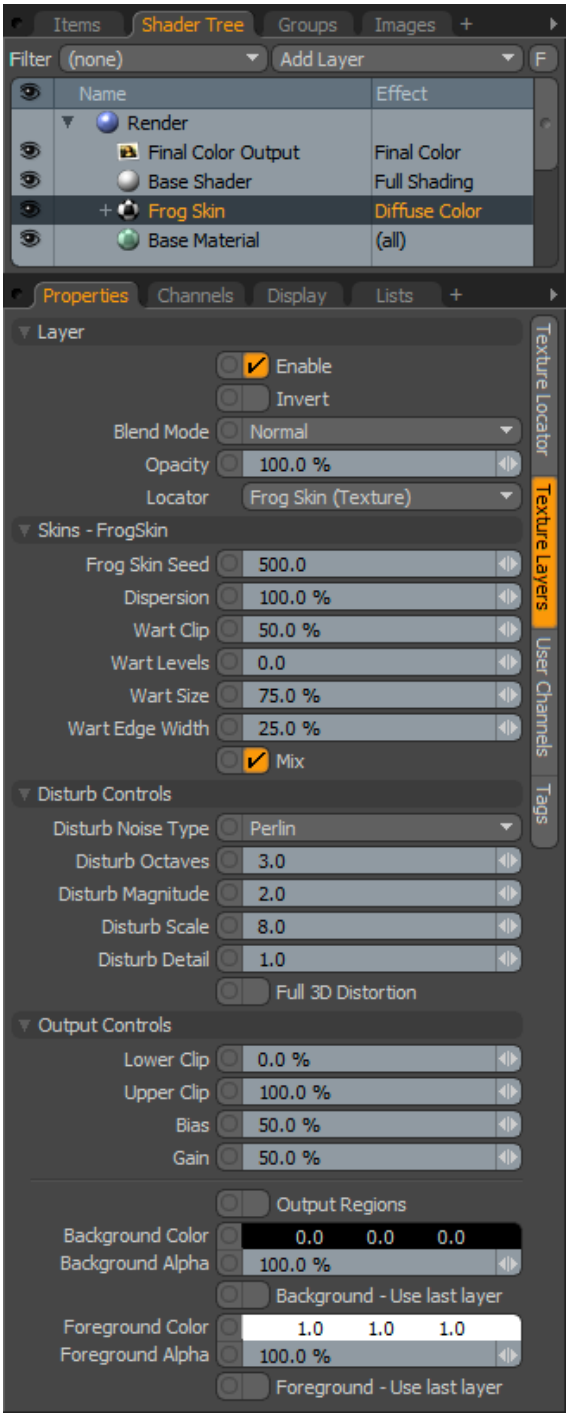
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Skins - Frog Skin --

Frog Skin Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Dispersion: This value value determines how many of the wart sections are actually drawn.

Wart Clip: This value determines where the texture shading transitions to the Background Color. The higher the value, the quicker the transition.

Wart Levels: This value determines the number of layers of the texture. Additional levels will add additional detail, in turn also lightening the overall look of the texture.

Wart Size: This value determines the overall size of the warts.

Wart Edge Width: This value determines the width of the edges of the individual warts. Higher values create a smoother transition between the foreground and background colors.

Mix: This option, when checked, mixes the texture layers. When unchecked, the texture layers are stacked one upon the other.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

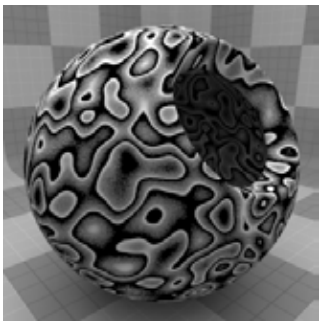
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Grainy Wood



The Grainy Wood texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Grainy Wood texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader creates a very *grainy wood* pattern. The depth of the grain can be varied along with the wood ring spacing.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

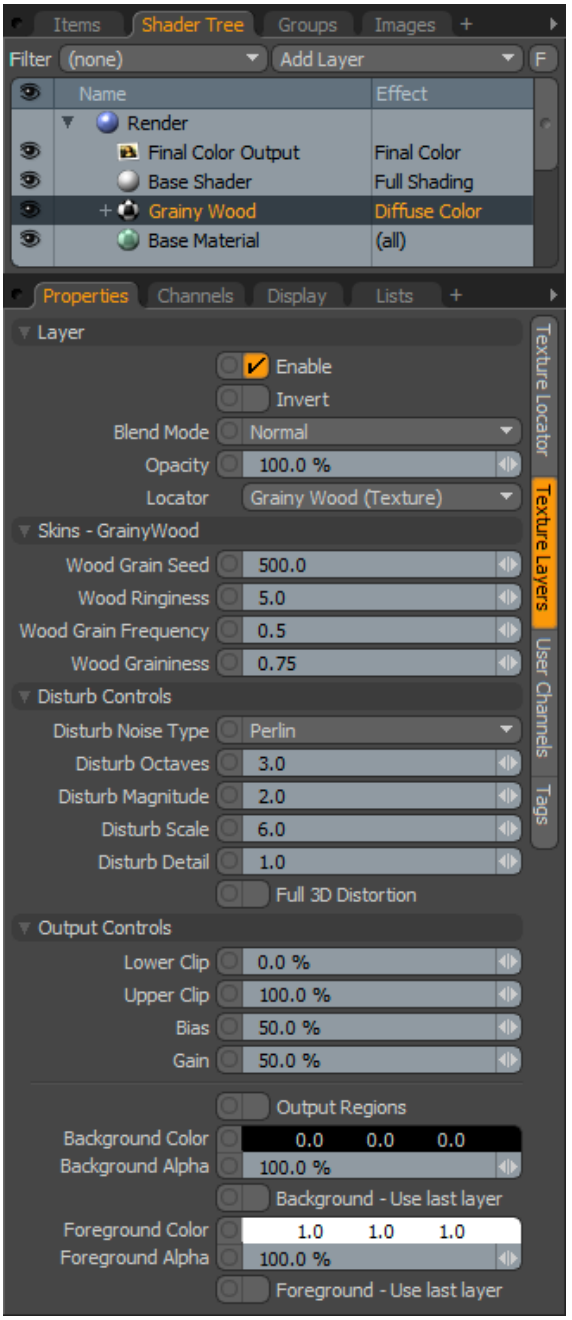
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Skins - Grainy Wood --

Wood Grain Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Wood Ringiness: This value controls the amplitude of the wood rings.

Wood Grain Frequency: This value controls the overall frequency or scale of the grain. Increasing this value creates more turbulence, "stirring up" the grain in the pattern.

Wood Graininess: This value controls the graininess of the wood rings.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value,

users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

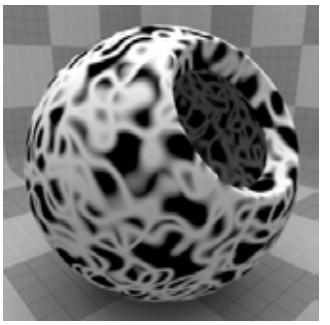
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Leather



The Leather texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Leather texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader emulates the grain of *leather* or *wrinkles*. The various controls provide multiple variations of grain width, depth and color.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

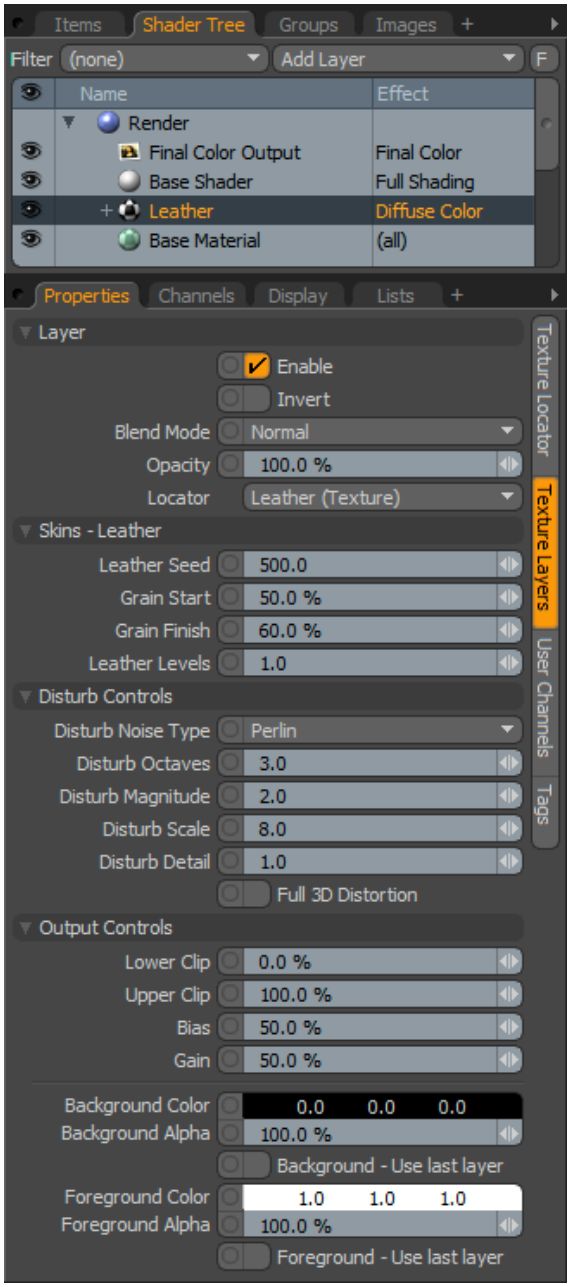
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Skins - Leather --

Leather Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Grain Start: This value controls where the valleys of the leather grain pattern start.

Grain Finish: This value controls where the valleys of the leather grain pattern finish.

Leather Levels: This value controls the number of leather patch levels.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The

noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

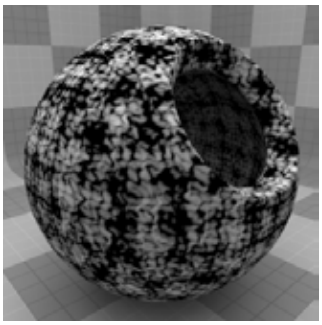
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Monster



The Monster texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Monster texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a look similar to *monster* skin. The shader is comprised of groups of bumps which can provide large scab-like areas.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

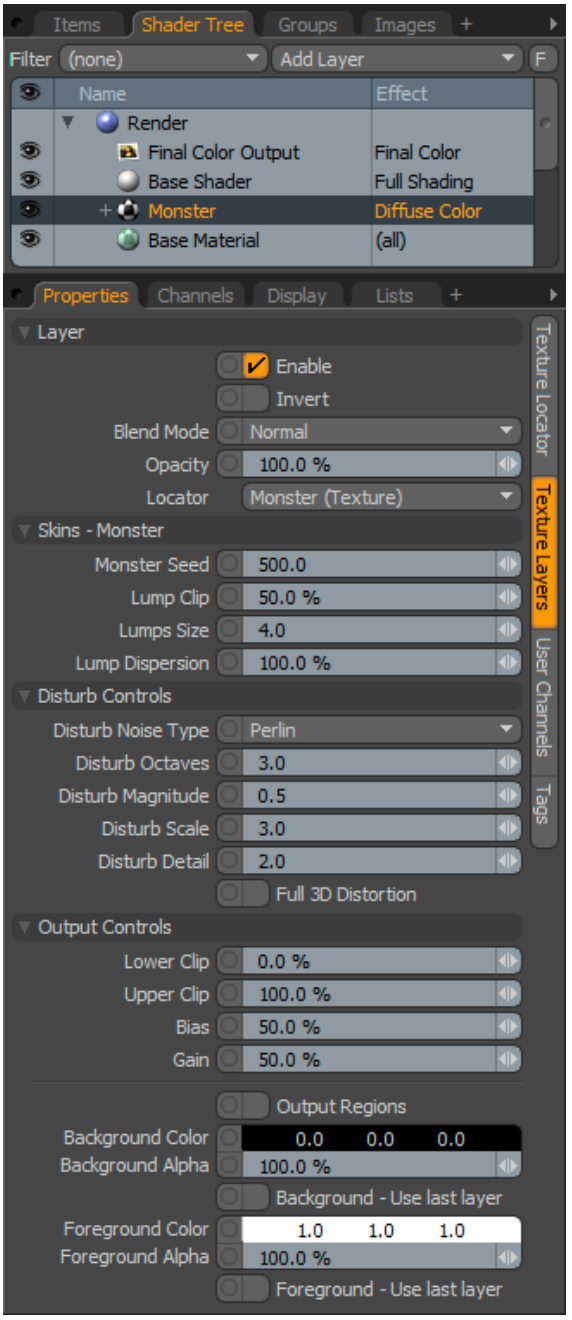
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Skins - Monster --

Monster Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Lump Clip: This value determines where the texture shading transitions to the Background Color. The higher the value, the quicker the transition.

Lump Size: This value determines the size of the smaller lumps.

Lump Dispersion: This value value determines how many of the lumps are actually drawn.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value,

users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

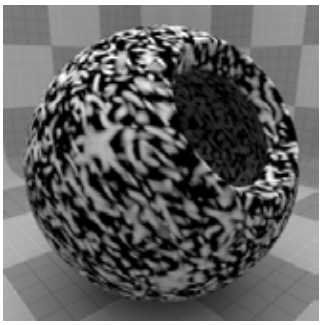
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Pastella



The Pastella texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Pastella texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader produces a pleasant, noisy sine wave effect. Utilizing the seed parameter allows for the creation of many different variations.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

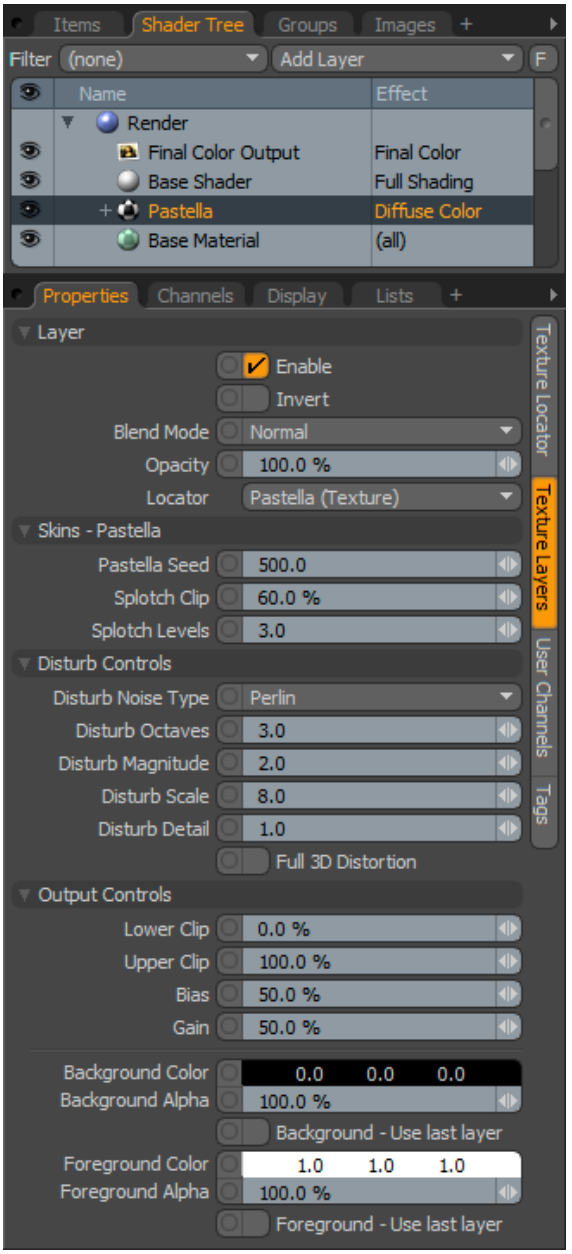
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Skins - Pastella --

Pastella Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Splotch Clip: This value determines where the texture shading transitions to the Background Color. The higher the value, the quicker the transition.

Splotch Levels: This value determines the number of layers of the texture. Additional levels will add additional detail, in turn also lightening the overall look of the texture.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value,

users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

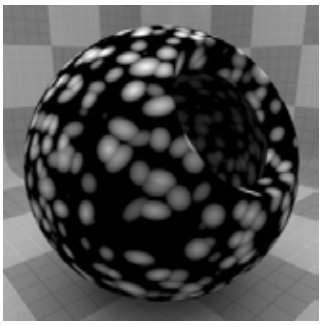
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Peened



The Peened texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Peened texture can be addressed by its two zones, the 'Backgorund' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Forground. THe look of this particular shader provide a *peened*, or hammered, looking surface, as seen on finished metals. The shader applies a series of tightly packed and overlapping spherical dents to the surface. Adding color to the dents can result in an antique finish texture.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

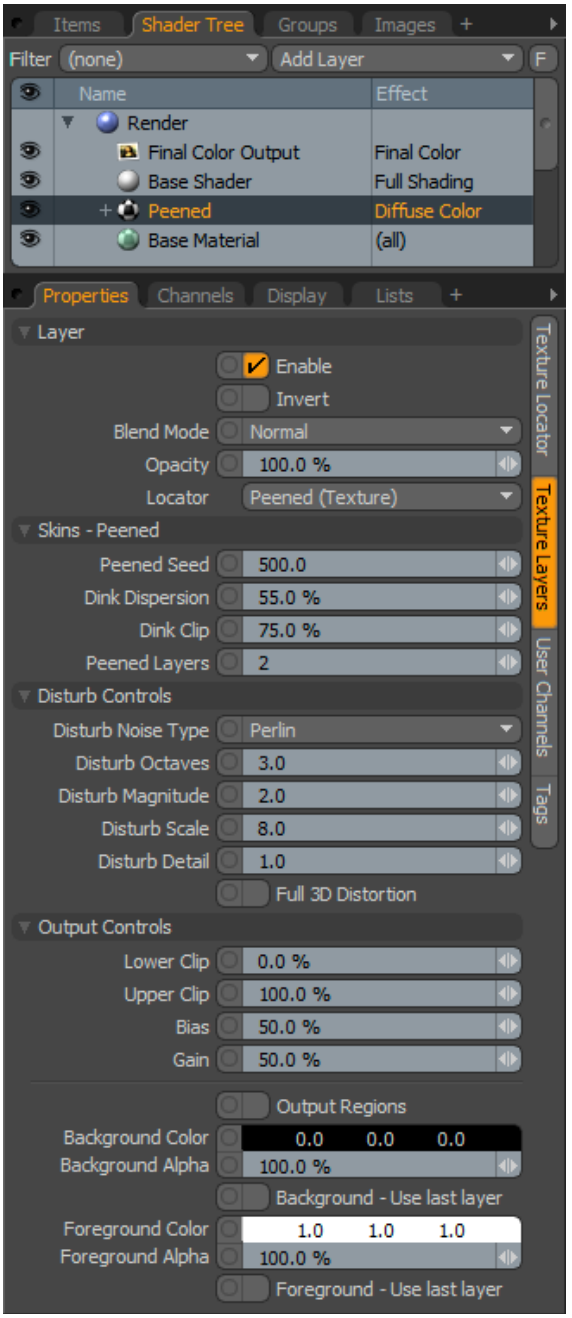
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Skins - Peened --

Peened Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Dink Dispersion: This value value determines how many of the dinks are actually drawn.

Dink Clip: This value determines where the texture shading transitions to the Background Color. The higher the value, the quicker the transition.

Peened Layers: This value determines the number of layers of the texture. Additional

detail, in turn also lightening the overall look of the texture.

Disturb Controls --

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value,

truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

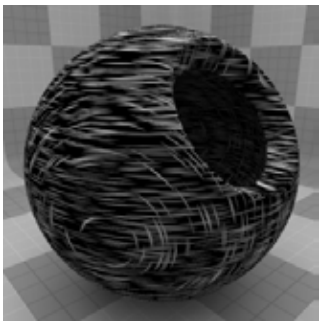
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Scratches



The Scratches texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Scratches texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader applies random *scratches*, dings, and dents to the surface of an object to provide worn-in, rough look. The shader applies several long and thin, overlapping cuts and dents in up to 16 different layers. Adding color to the scratches can provide an antique finish texture.

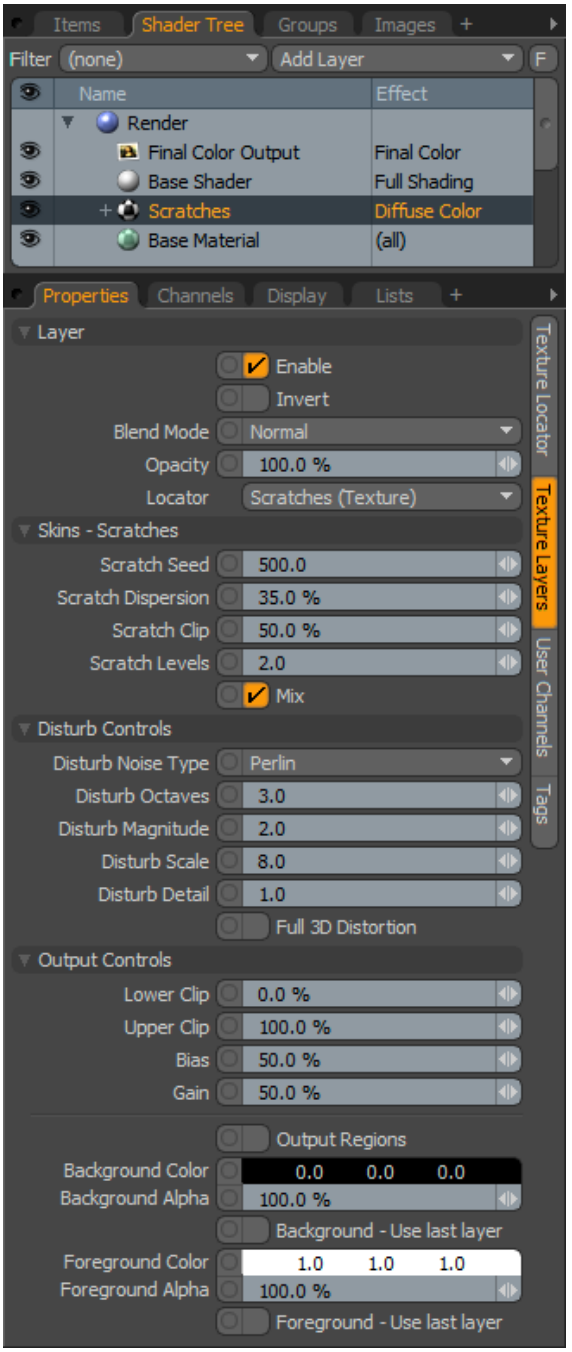
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.



Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Skins - Scratches --

Scratch Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Scratch Dispersion: This value value determines how many of the scratches are actually drawn.

Scratch Clip: This value determines where the texture shading transitions to the Background Color. The higher the value, the quicker the transition.

Scratch Levels: This value determines the number of layers of the texture. Additional levels will add additional detail, in turn also lightening the overall look of the texture.

Mix: This option, when checked, mixes the texture layers. When unchecked, the texture layers are stacked one upon the other.

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value,

users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Tiles

Basket

Bath Tile

Bricks

Checks

Cornerless

Cubes

Dash Line

Diamond Deck

Fish Scales

Hex Tile

Lattice 1

Lattice 2

Lattice 3

Mosaic

Oct Tile

Parquet

Paving

Plaid

Planks

Ribs

Rounded Tile

Shingles

Spots

Stamped

Tacos

Tartan

Tiler

Tri Checks

Tri Hexes

Tri Tile

Wall

Basket

The Basket texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Basket texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a woven pattern, the kind used to create a *basket*, with 26 reversible weave types.

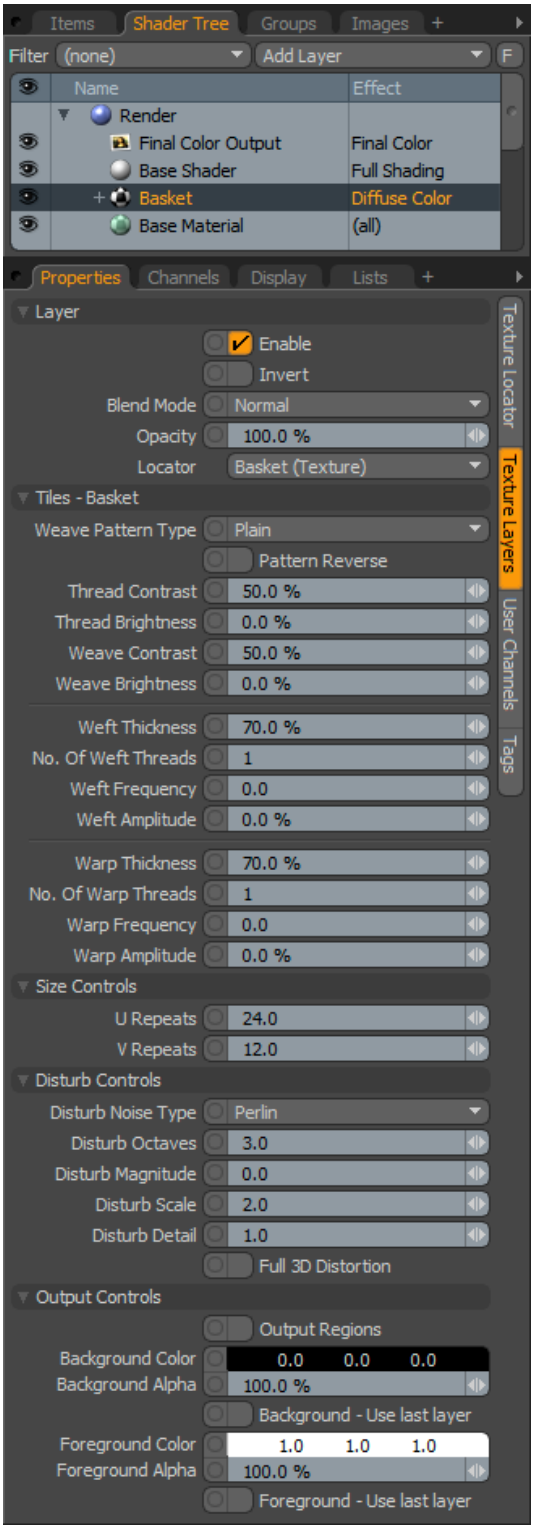
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Basket --

Weave Pattern Type: This option determines the weave pattern used in the texture. There are 26 options including:

- Plain
- Herring Bone
- Crosses
- Zig Zag

Pattern Reverse: This option, when selected, reverses the weave and weft of the pattern.

Thread Contrast: This value determines the contrast on the threads of the weave pattern.

Thread Brightness: This value determines the brightness on the threads of the weave pattern.

Weave Contrast: This value determines the contrast on the weave pattern.

Weave Brightness: This value determines the brightness on the weave pattern.

--

Weft Thickness: This value determines the width of the weft, or horizontal, threads.

Number Of Weft Threads: This value controls the number of weft threads.

Weft Frequency: This value controls the sine frequency used to disturb the weft threads.

Weft Amplitude: This value control the amplitude used to disturb the weft threads.

--

Warp Thickness: his value determines the width of the warp, or vertical, threads

Number Of Warp Threads: This value controls the number of warp threads.

Warp Frequency: This value controls the sine frequency used to disturb the warp threads.

Warp Amplitude: This value control the amplitude used to disturb the warp threads.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

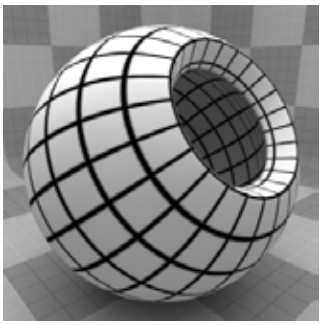
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Bath Tile



The Bath Tile texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Bath Tile texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a square tile pattern with variable grout, tiles edges and tile height, similar to to the look of *Bath Tile*. The tiles may be staggered and random tiles can be removed.

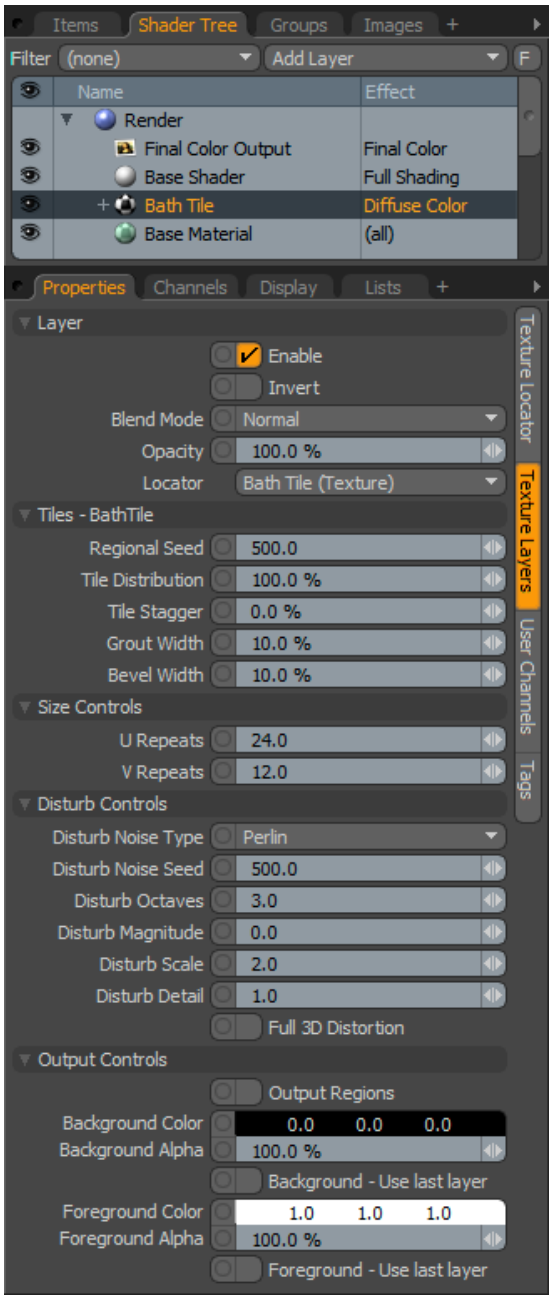
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Bath Tile --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Tile Distribution: This value determines the percentage of the surface covered by tiles.

Tile Stagger: This value determines the amount of staggering between tile rows. When the control is at a low value, the tiles line up vertically, with the tile rows becoming more staggered as the parameter is increased.

Grout Width: This value determines the amount of grout between the tiles.

Bevel Width: This value determines the width of the bevel on the tiles.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

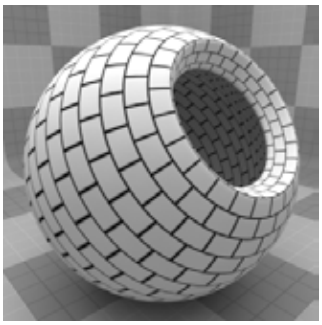
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Bricks



The Bricks texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Bricks texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a rectangular, *brick*-like pattern. The shader has the same controls as Bath Tile shader but the aspect ratio of the bricks can be modified.

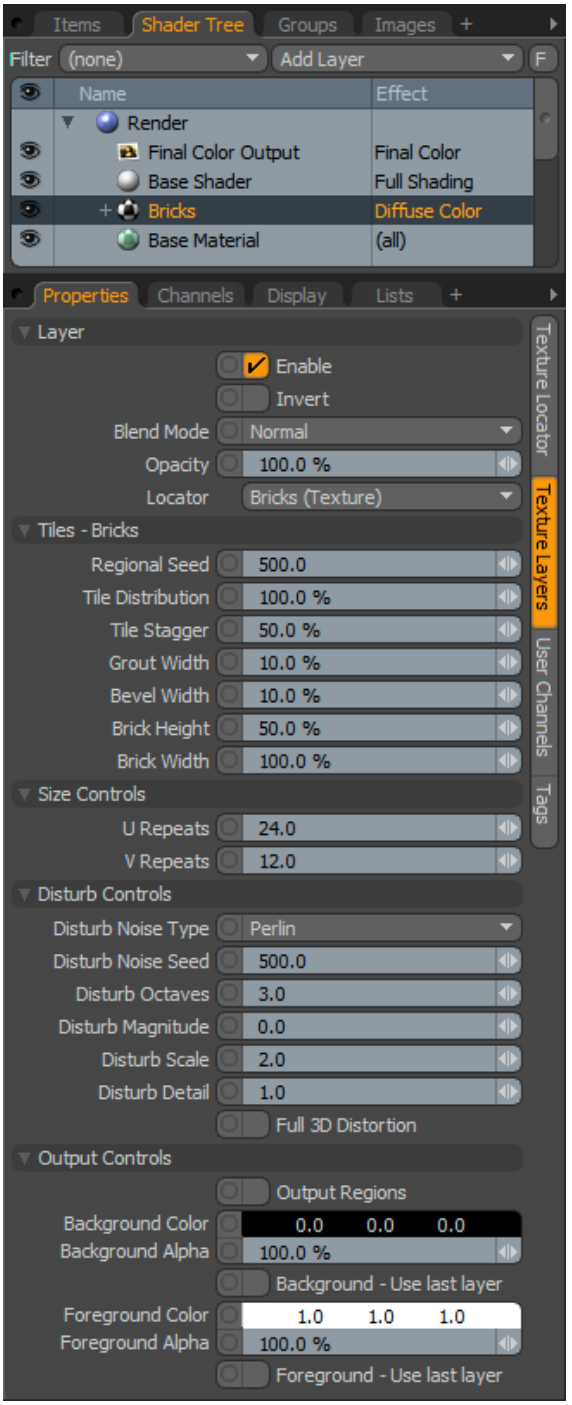
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Bricks --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Tile Distribution: This value determines the percentage of the surface covered by bricks.

Tile Stagger: This value determines the amount of staggering between brick rows. When the control is at a low value, the bricks line up vertically, with the brick rows becoming more staggered as the parameter is increased.

Grout Width: This value determines the amount of grout between the bricks.

Bevel Width: This value determines the width of the bevel on the bricks.

Brick Height: This value determines the height of the individual bricks.

Brick Width: This value determines the width of the individual bricks.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

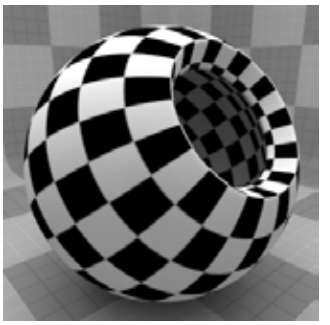
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Checks



The Checks texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Checks texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a standard *check* pattern, perfect for a chess board or marble floor-like covering.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

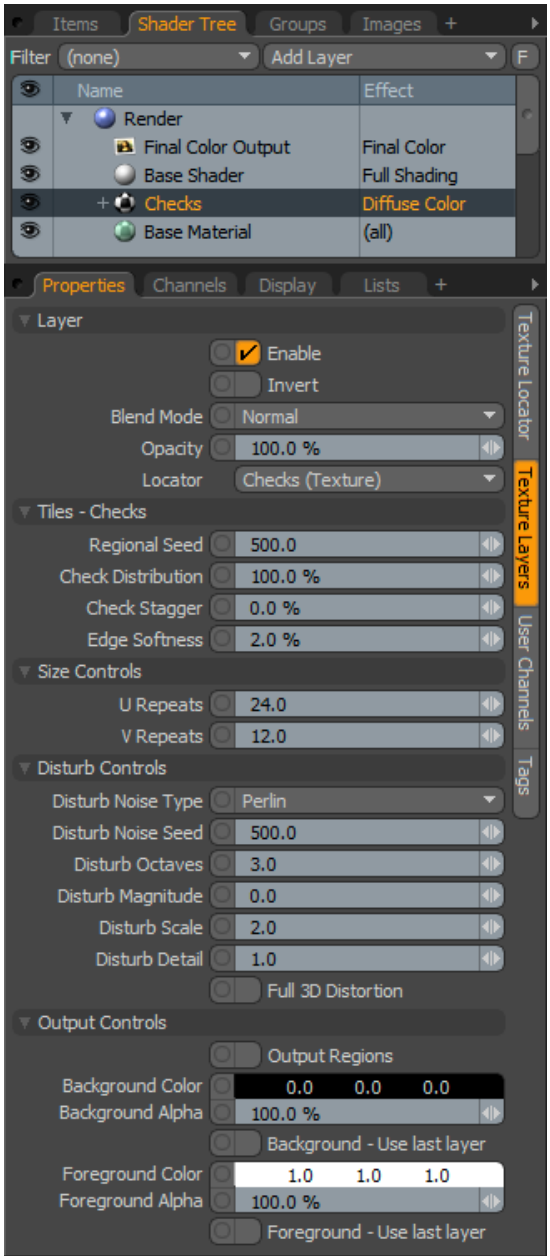
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing

user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.



Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Checks --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Check Distribution: This value determines the percentage of the surface covered by checks (tiles).

Check Stagger: This value determines the amount of staggering between check rows. When the control is at a low value, the tiles line up vertically, with the tile rows becoming more staggered as the parameter is increased.

Edge Softness: This value controls the blurring, or softness at the edges of the checks.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

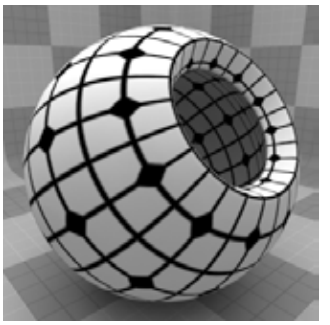
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Cornerless



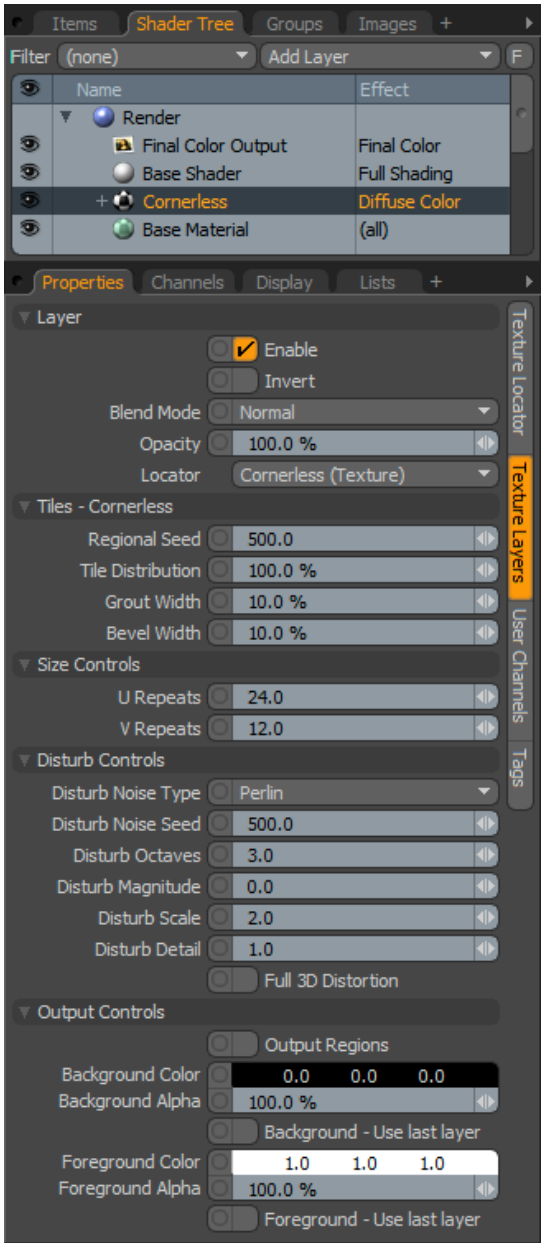
The Cornerless texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Cornerless texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a square tile pattern with variable grout, tiles edges and tile height. The tiles may be staggered and random tiles can be removed. The major difference with this tile shader is that one *corner* is missing, which, combined with the Taco shader, allows for complex tile patterns to be created.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.



Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Cornerless --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Tile Distribution: This value determines the percentage of the surface covered by tiles.

Grout Width: This value determines the amount of grout between the tiles.

Bevel Width: This value determines the width of the bevel on the tiles.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

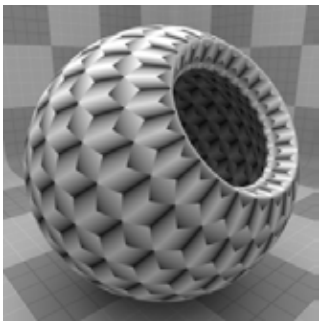
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Cubes



The Cubes texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Cubes texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader provides a tri-colored isometric *cube* pattern.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

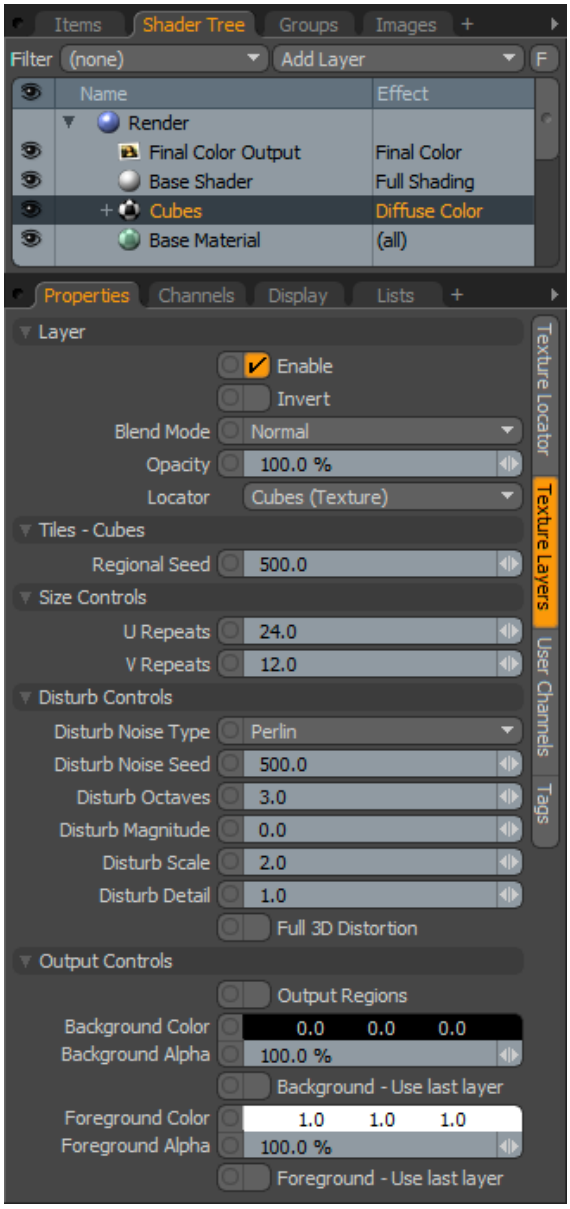
IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.



Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Cubes --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

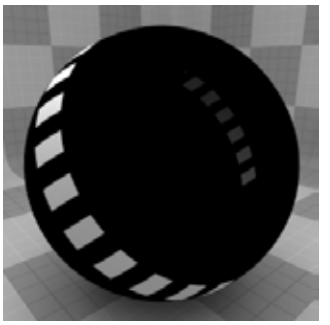
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Dash Line



The Dash Line texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Dash Line texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a *dashed line*, similar to the lane separator on highway. The layout of the dashes can altered using a binary pattern.

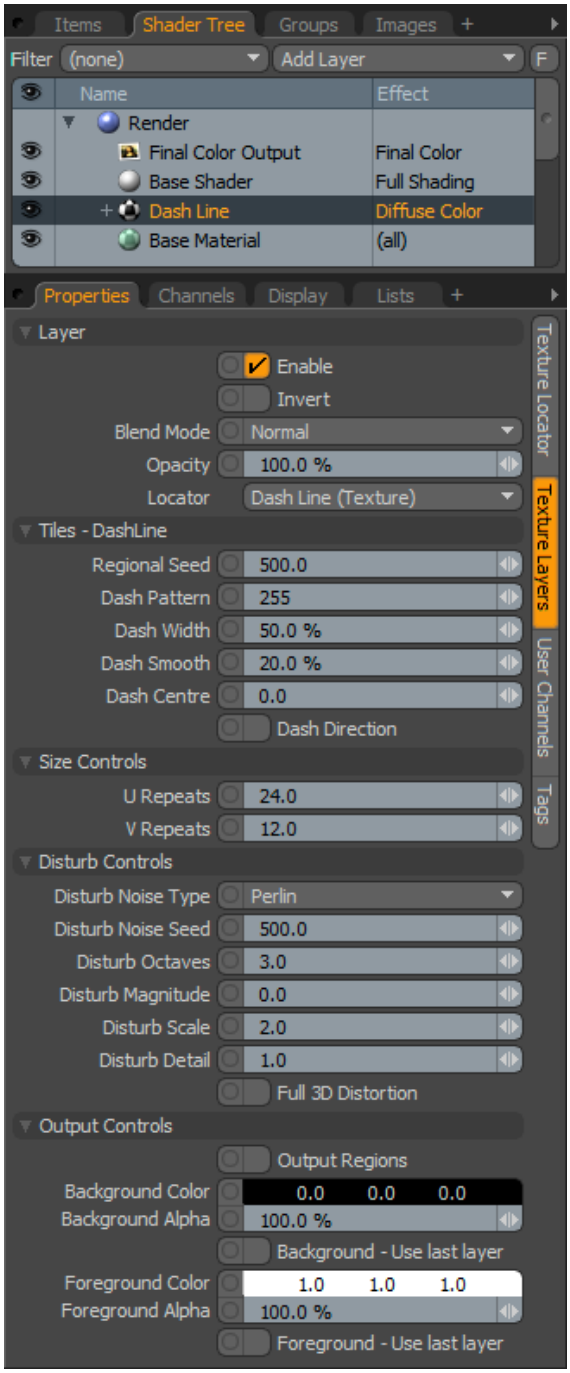
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Dash Line--

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Dash Pattern: This value determines the dash pattern for the texture.

Dash Width: This value determines the width of the dashed line.

Dash Smooth: This value determines the smoothness on the edges of the dashed line.

Dash Centre: This value determines the center point of the dashed line.

Dash Direction: This option reverses the direction of the dash.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

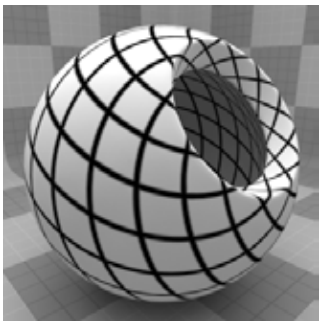
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Diamond Deck



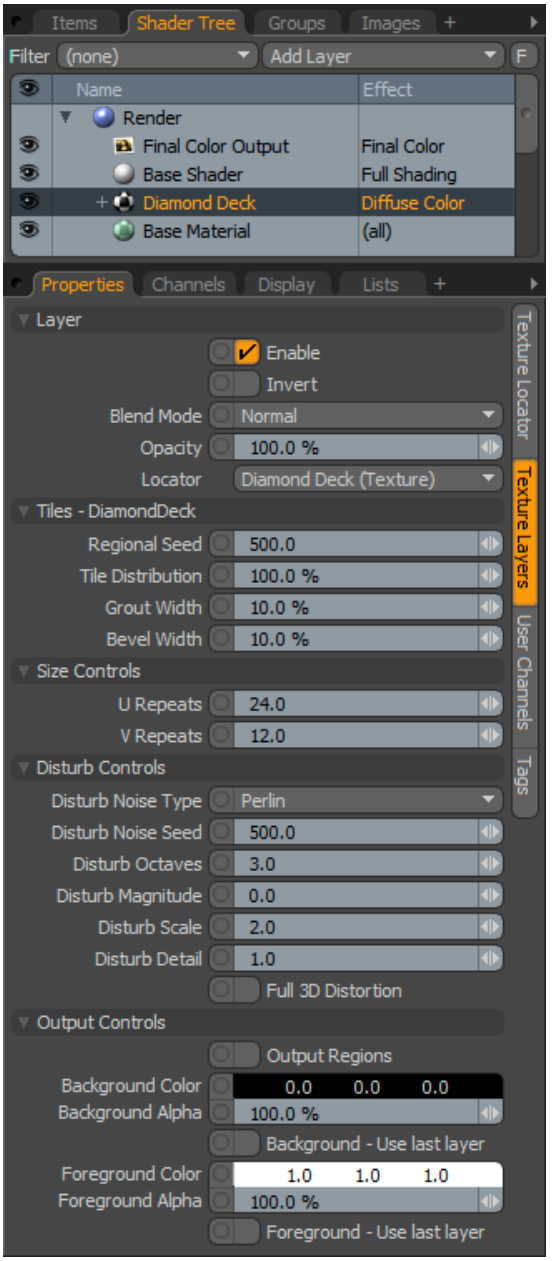
The Diamond Deck texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Diamond Deck texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a *diamond* tile pattern, similar to those found on certain types of man hole covers.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.



Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Diamond Deck --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Tile Distribution: This value determines the percentage of the surface that is covered by tiles.

Grout Width: This value determines the amount of grout between the tiles.

Bevel Width: This value determines the width of the bevel on the tiles.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

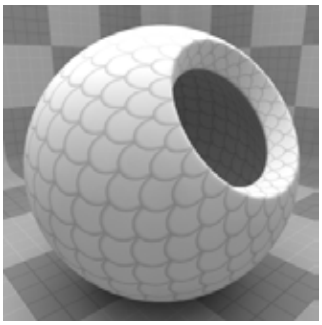
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Fish Scales



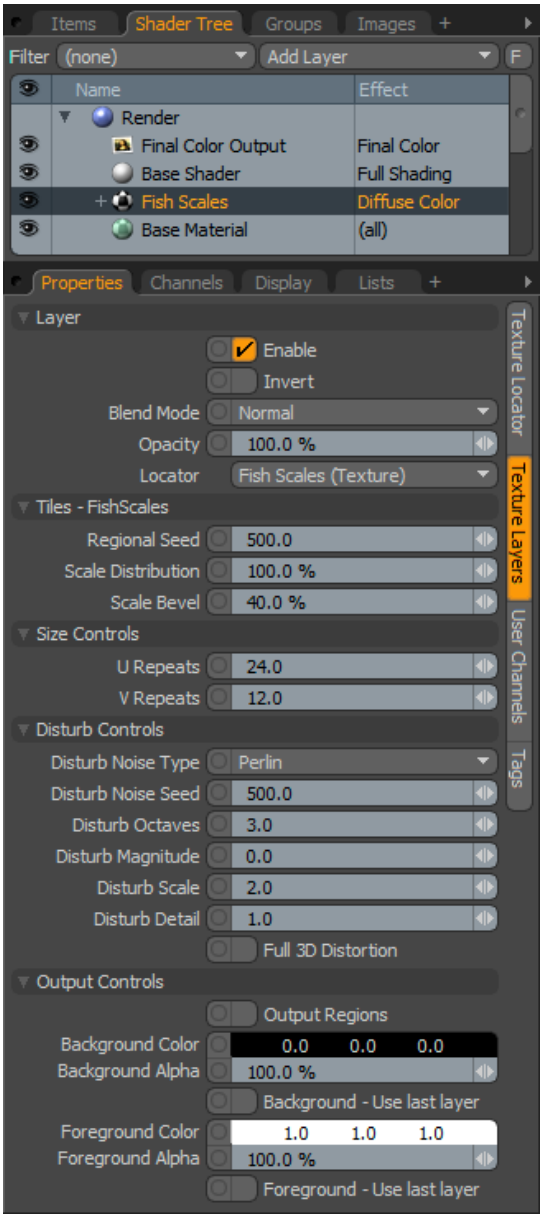
The Fish Scales texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Fish Scales texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a pattern similar to the scales on a *fish*. The radius, edges and height of the scales are all variable.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.



Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Fish Scales --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Scale Distribution: This value determines the percentage of the surface that is covered by scales.

Scale Bevel: This value determines the amount of bevel between the scales.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

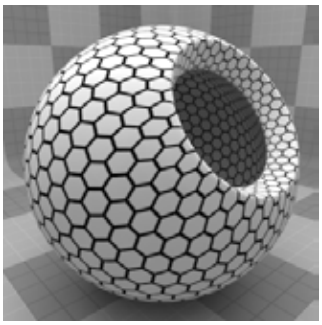
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Hexagonal Tiles



The Hexagonal Tiles texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Hexagonal Tiles texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a *hexagonal* tile pattern with variable grout, tiles edges and tile height.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

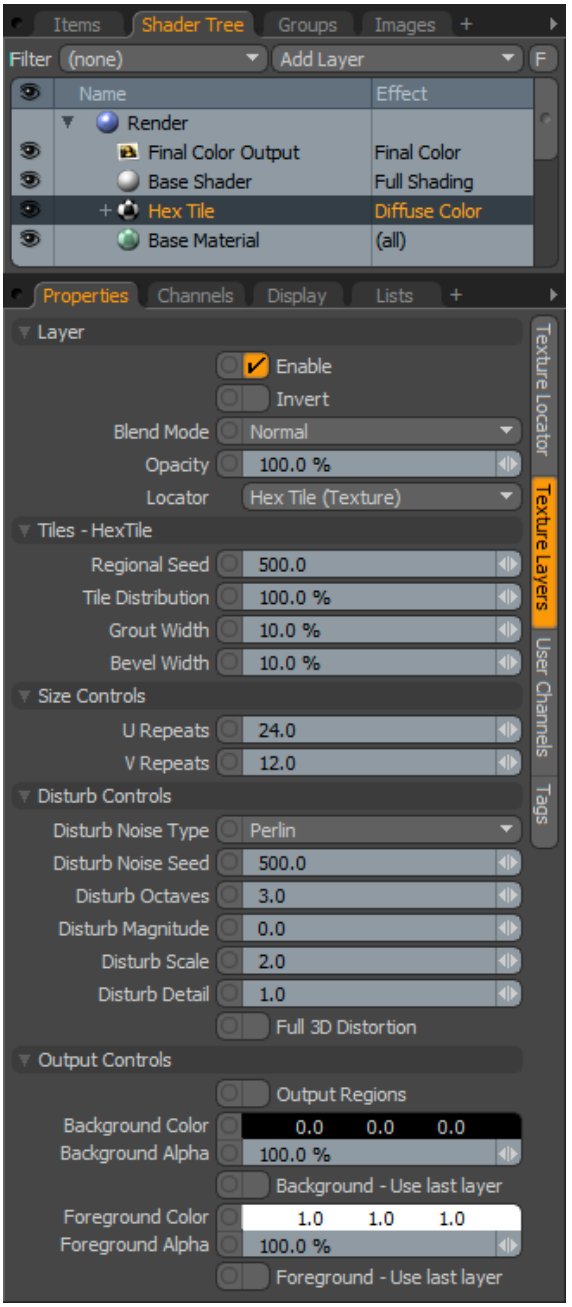
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing

user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.



Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Hex Tiles --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Tile Distribution: This value determines the percentage of the surface that is covered by tiles.

Grout Width: This value determines the amount of grout between the tiles.

Bevel Width: This value determines the width of the bevel on the tiles.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

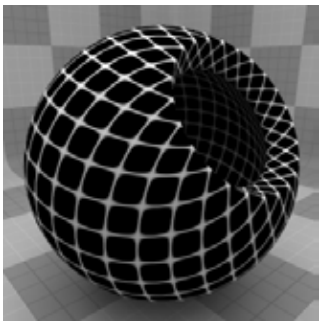
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Lattice 1



The Lattice 1 texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Lattice 1 texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides *lattice* pattern with variable thickness, which can be used to texture nets or fences.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

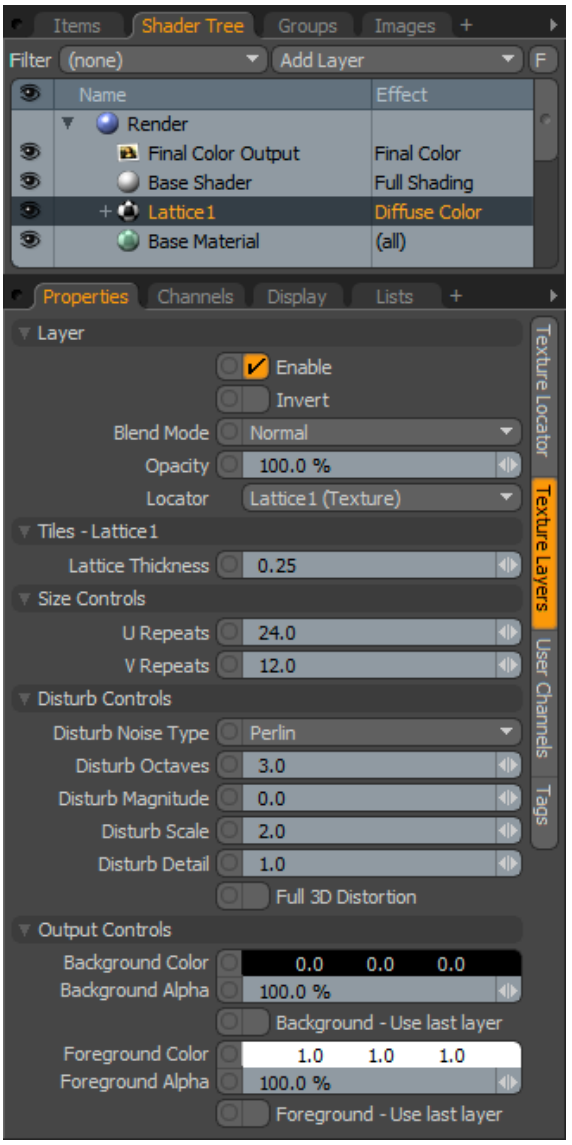
IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.



Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Lattice 1 --

Lattice Thickness: This value determines the overall thickness of the lattice.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value,

users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

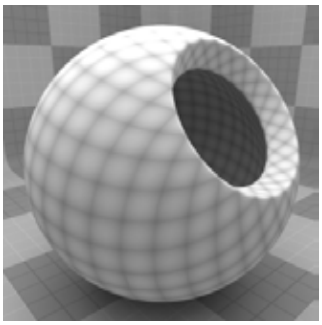
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Lattice 2



The Lattice 2 texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Lattice 2 texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides *lattice* pattern with variable thickness, which can be used to texture nets or fences.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

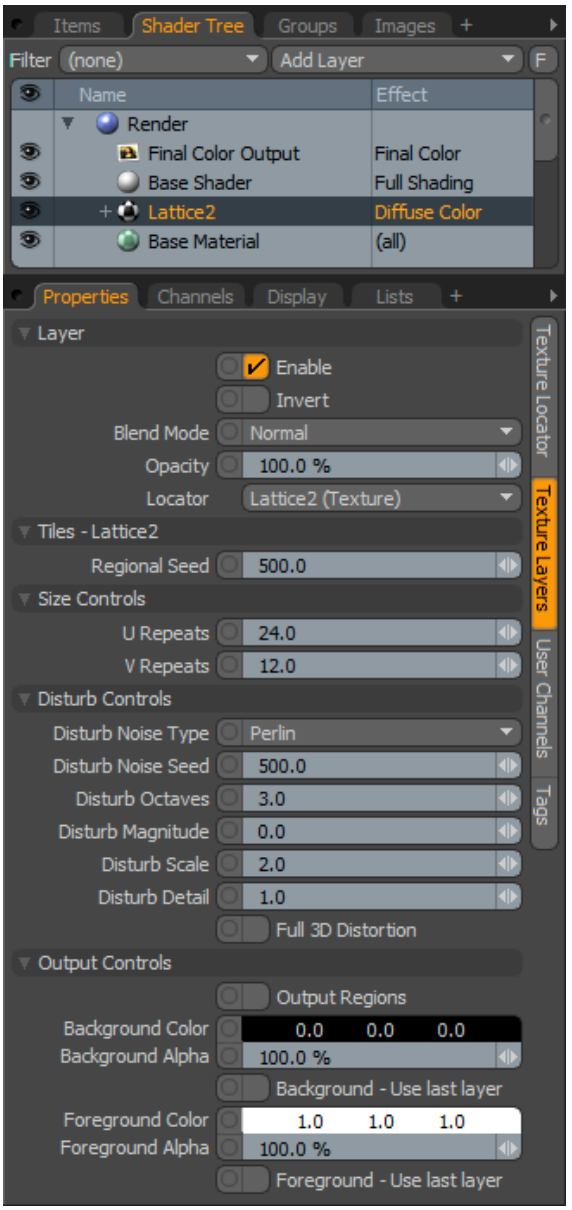
IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.



Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Lattice 2 --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

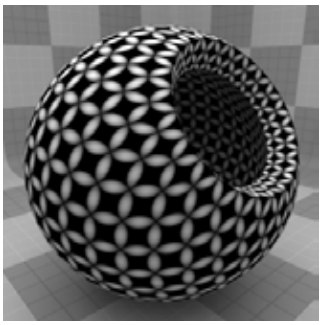
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Lattice 3



The Lattice 3 texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Lattice 3 texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides *lattice* pattern with variable thickness, which can be used to texture nets or fences.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

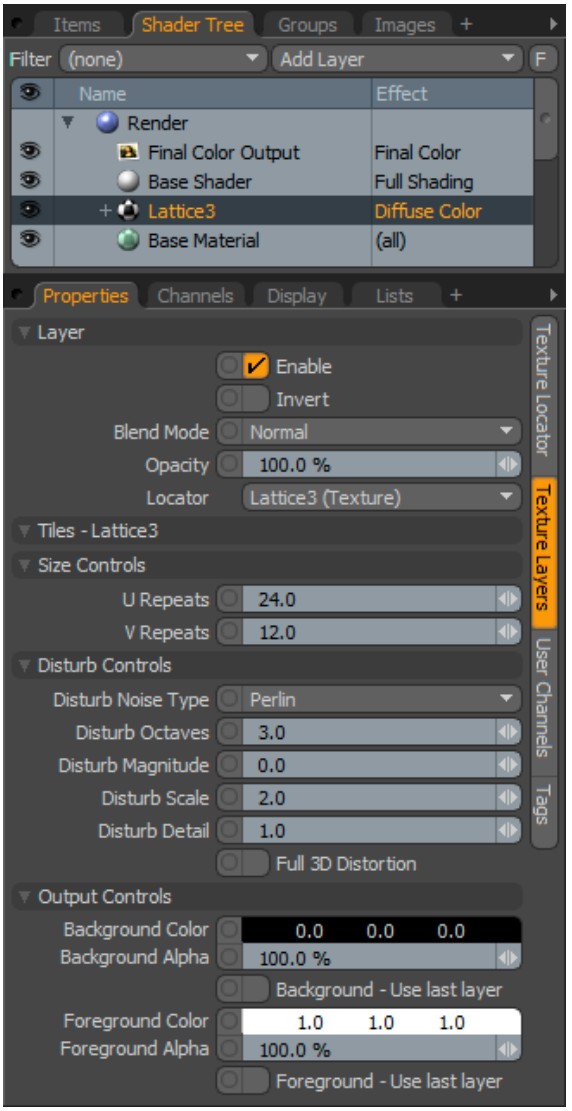
IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.



Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

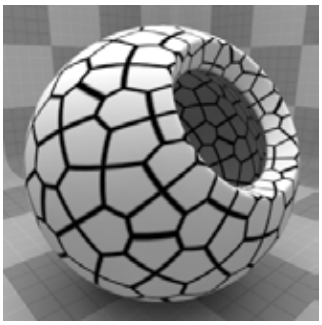
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Mosaic



The Mosaic texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Mosaic texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides *mosaic* pattern of randomly shaped tiles with variable grout, tile scale and tile height.

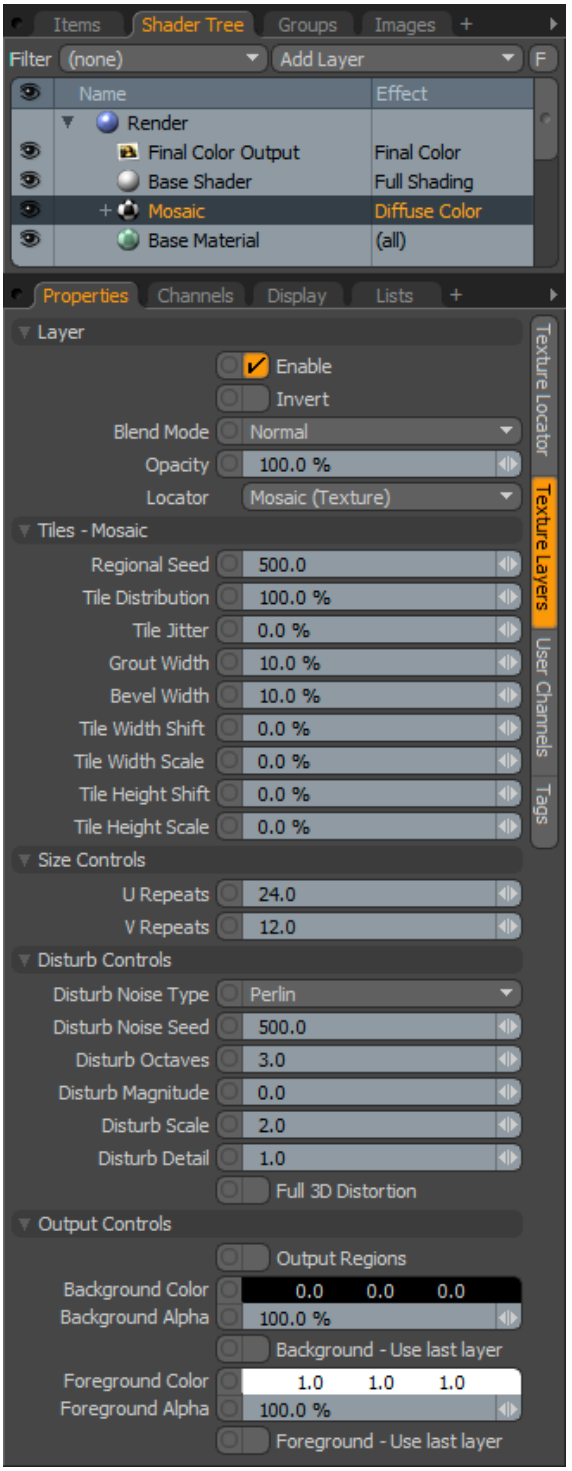
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Mosaic --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Tile Distribution: This value determines the percentage of the surface that is covered by tiles.

Tile Jitter: This value determines the amount of randomness in the tile placement. When the control is set to a low value, the tiles are bit more uniform, becoming more random as the parameter is increased.

Grout Width: This value determines the amount of grout between the tiles.

Bevel Width: This value determines the width of the bevel on the tiles.

Tile Width Shift: This value controls the horizontal shift of the tiles.

Tile Width Scale: This value controls the horizontal scale of the tiles.

Tile Height Shift: This value controls the vertical shift of the tiles.

Tile Height Scale: This value controls the vertical scale of the tiles.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

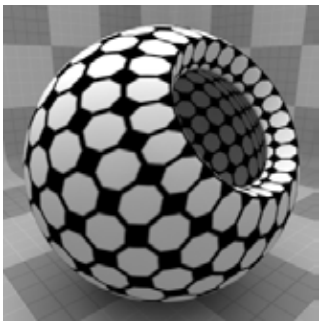
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Octagonal Tile



The Octagonal Tile texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Octagonal Tile texture can be addressed by its two zones, the 'Backgorund' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Forground. This shader provides an *octagonal* tile pattern with variable grout, tile edges and tile height.

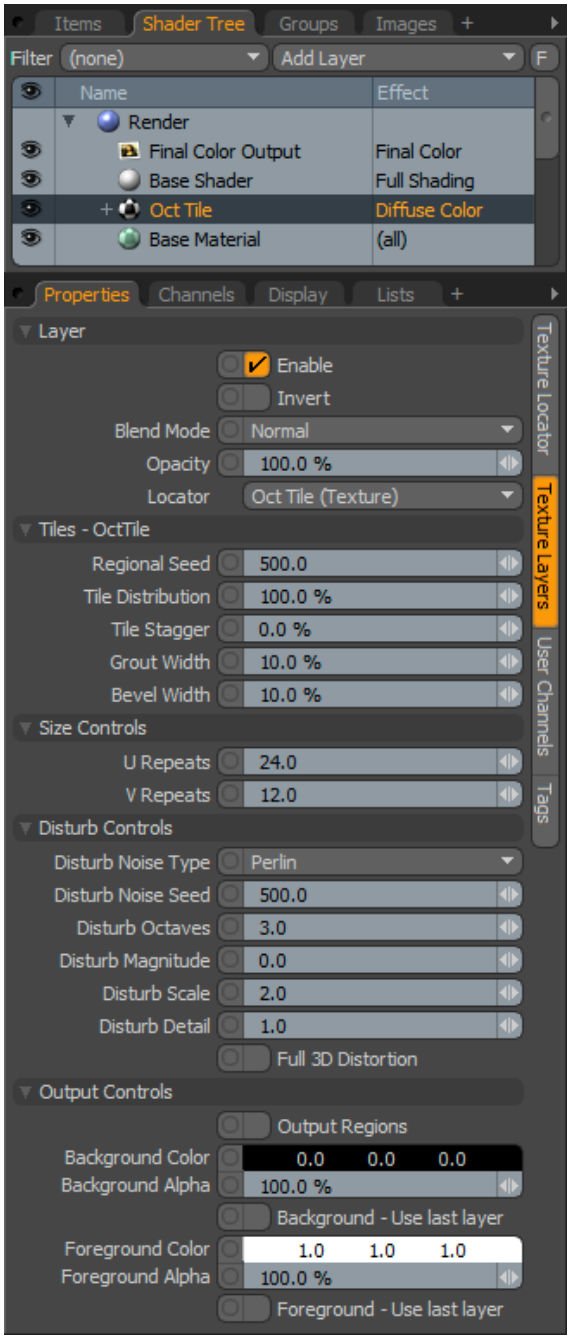
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Oct Tiles --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Tile Distribution: This value determines the percentage of the surface that is covered

by tiles.

Tile Stagger: This value determines how staggered the rows of tiles are in the texture. When the control is at a low value, the tiles line up vertically, with the tile rows becoming more staggered as the parameter is increased.

Grout Width: This value determines the amount of grout between the scales.

Bevel Width: This value determines the width of the bevel on the tiles.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the

base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Parquet

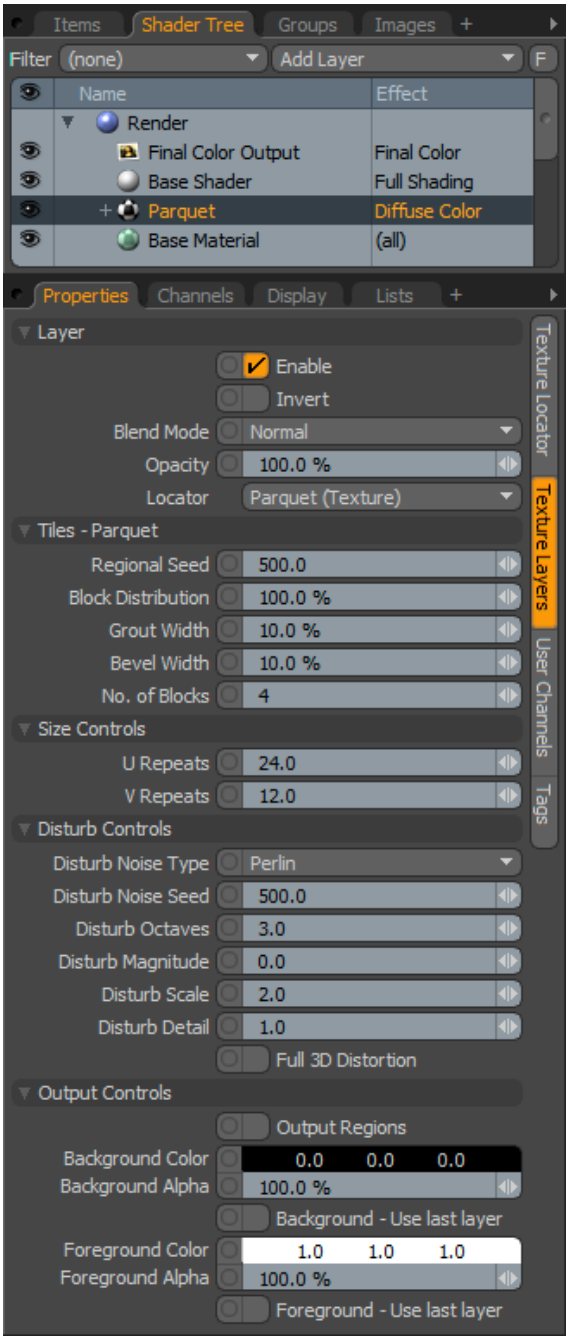
The Parquet texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Parquet texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a *parquet* or blocked-wood type texture, which can be utilized for floor coverings. The provided controls can modify the the number of blocks used, the graininess of the wood, the visibility of the wood rings and the block edges.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.



Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Parquet --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Block Distribution: This value determines the percentage of the surface that is covered by tiles.

Grout Width: This value determines the amount of grout between the tiles.

Bevel Width: This value determines the width of the bevel on the tiles.

Number Of Blocks: This value controls the number of blocks in each parquet tile.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

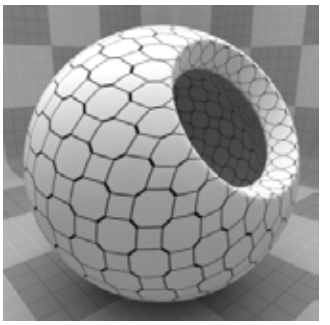
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Paving



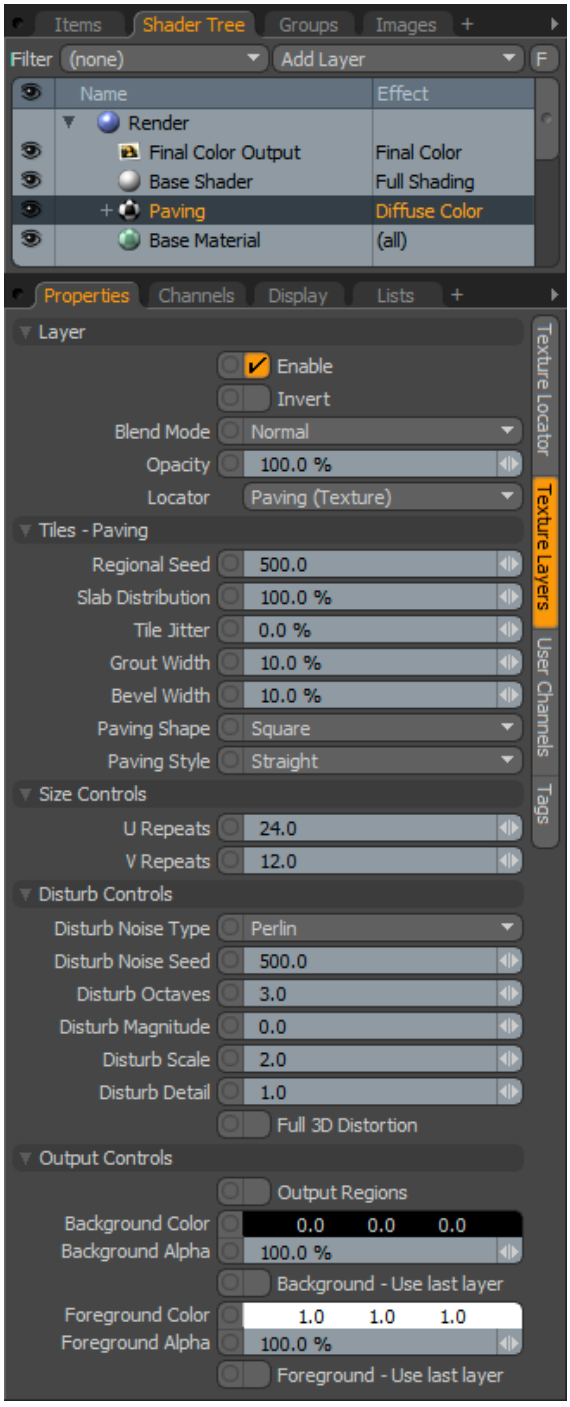
The Paving texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Paving texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a tile pattern with adjustbale shapes, variable grout, tiles edges and tile height, similar to *paving* stones. The tiles may be staggered and random tiles can be removed.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.



Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Paving --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Slab Distribution: This value determines the percentage of the surface that is covered by tiles.

Tile Jitter: This value determines how randomly placed the tiles are. When the control is at a low value, the tiles are bit more uniform, becoming more random as the parameter is increased.

Grout Width: This value determines the amount of grout between the tiles.

Bevel Width: This value determines the width of the bevel on the tiles.

Paving Shape: This option determines the shape of the paving tiles. Options are:

- Square
- Hexagon
- Octagon
- Triangle
- Wedges

Paving Style: This option determines the layout of the paving tiles. Options are:

- Straight
- Crazy
- Ornate
- Triangle
- Uneven

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

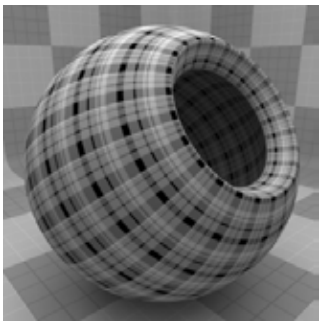
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Plaid



The Plaid texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Plaid texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a standard *plaid* effect, with an overlapping stripe pattern. This shader works best when applied to a black object.

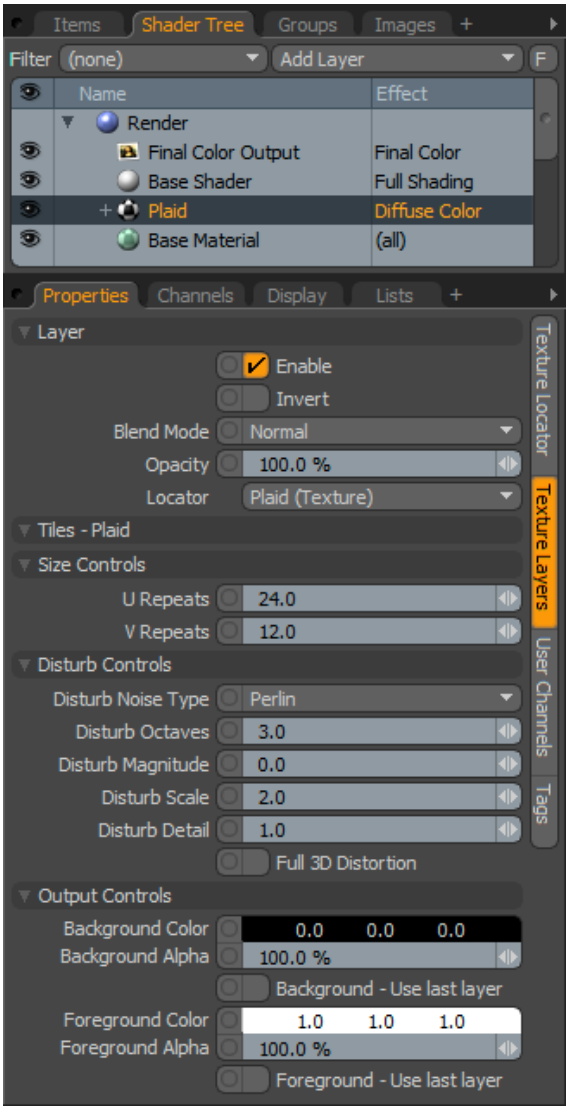
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the

background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

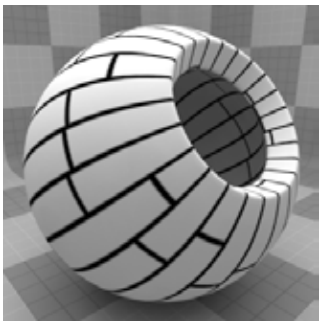
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Planks



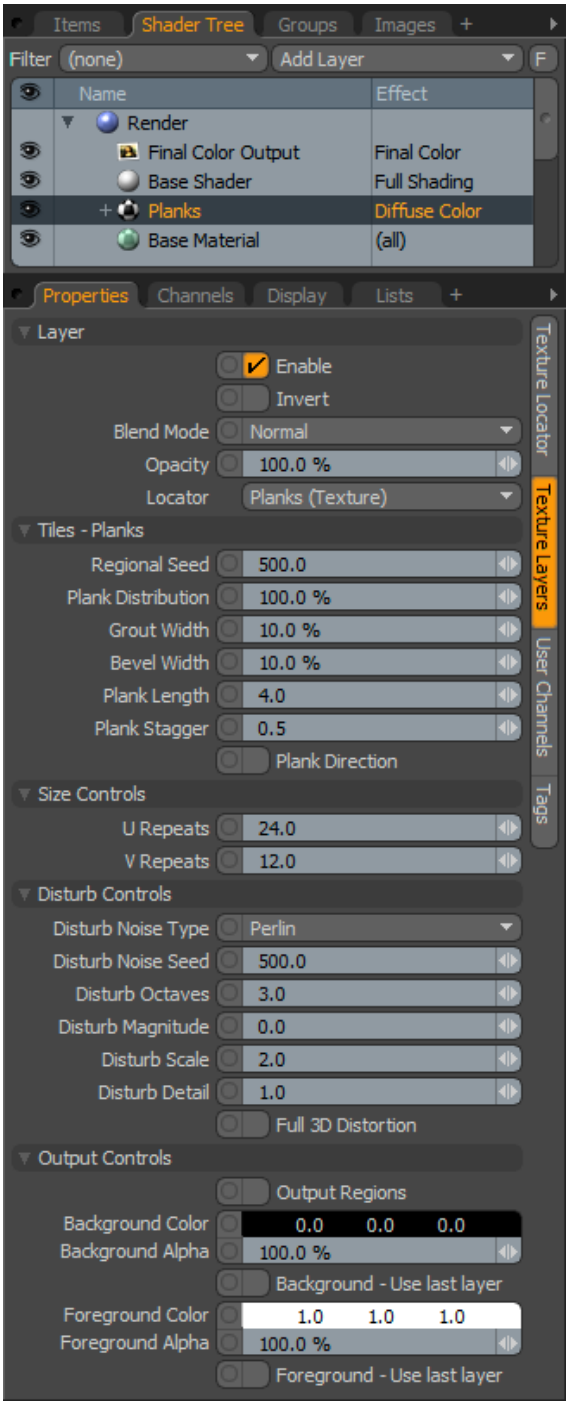
The Planks texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Planks texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a floor board, or plank, type of texture. Some of the provided controls include: the amount of stagger on each *plank*, the graininess of the wood, the visibility of the the wood rings and the plank edges.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.



Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Planks --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Plank Distribution: This value determines the percentage of the surface that is covered by planks.

Grout Width: This value determines the amount of grout between the planks.

Bevel Width: This value determines the width of the bevel on the planks.

Plank Length: This value determines the length of the individual planks.

Plank Stagger: This value controls the randomization of the planks stagger. When the control is at a low value, the planks line up horizontally, with the rows becoming more staggered as the parameter is increased.

Plank Direction: This options switches the planks orientation between vertical and horizontal.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

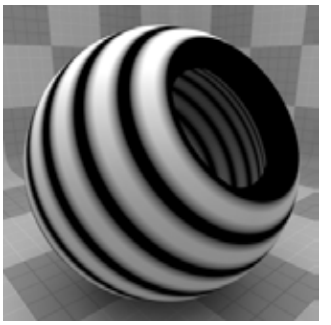
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Ribs



The Ribs texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Ribs texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a series of straight *ribs* running along an object. The number of ribs drawn can be modified, along with their location in relation to the objects center-line.

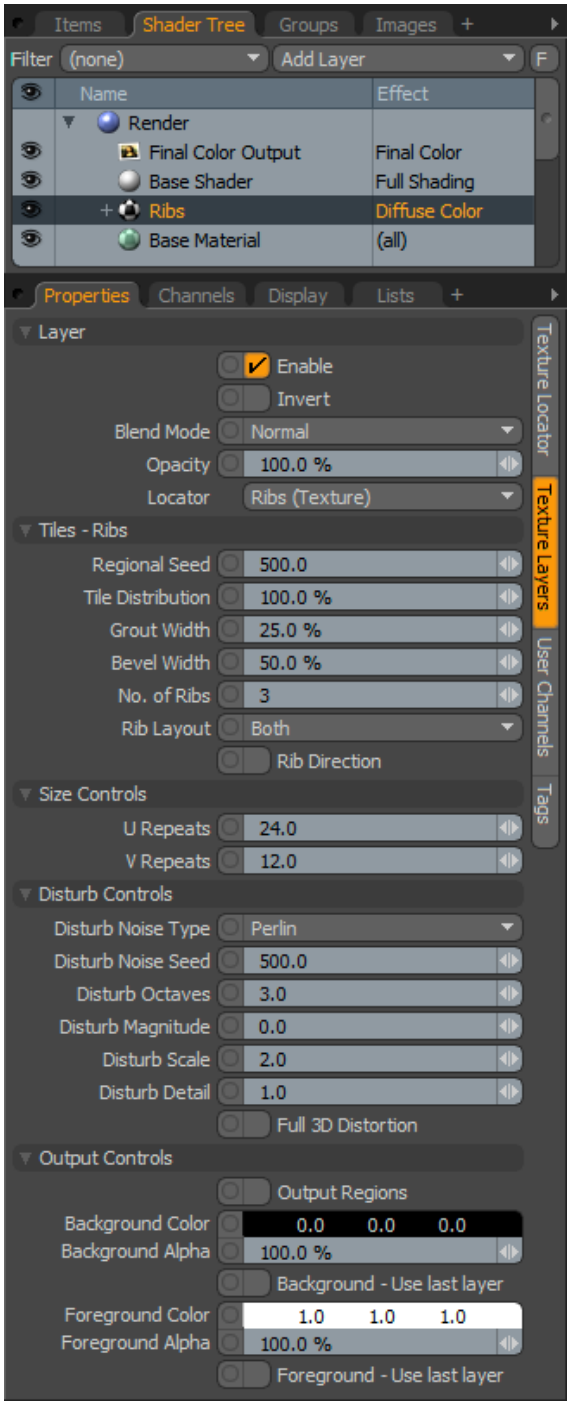
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Ribs --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Tile Distribution: This value determines the percentage of the surface that is covered by ribs.

Grout Width: This value determines the amount of grout between the ribs.

Bevel Width: This value determines the width of the bevel on the ribs.

Number of Ribs: This value determines the number of ribs drawn in the texture.

Rib Layout: This option controls which ribs are drawn in the texture. Options are:

- Both
- All Ribs
- Positive
- Negative

Rib Direction: This options switches the ribs orientation between vertical and horizontal.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

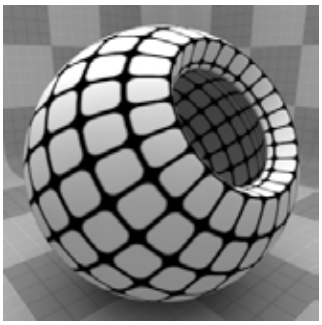
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Rounded Tile



The Rounded Tile texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Rounded Tile texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a *rounded tile* texture, similar to the Bath Tile shader, with the same controls. However, this shader also has an option to specify the amount of roundness for the corners.

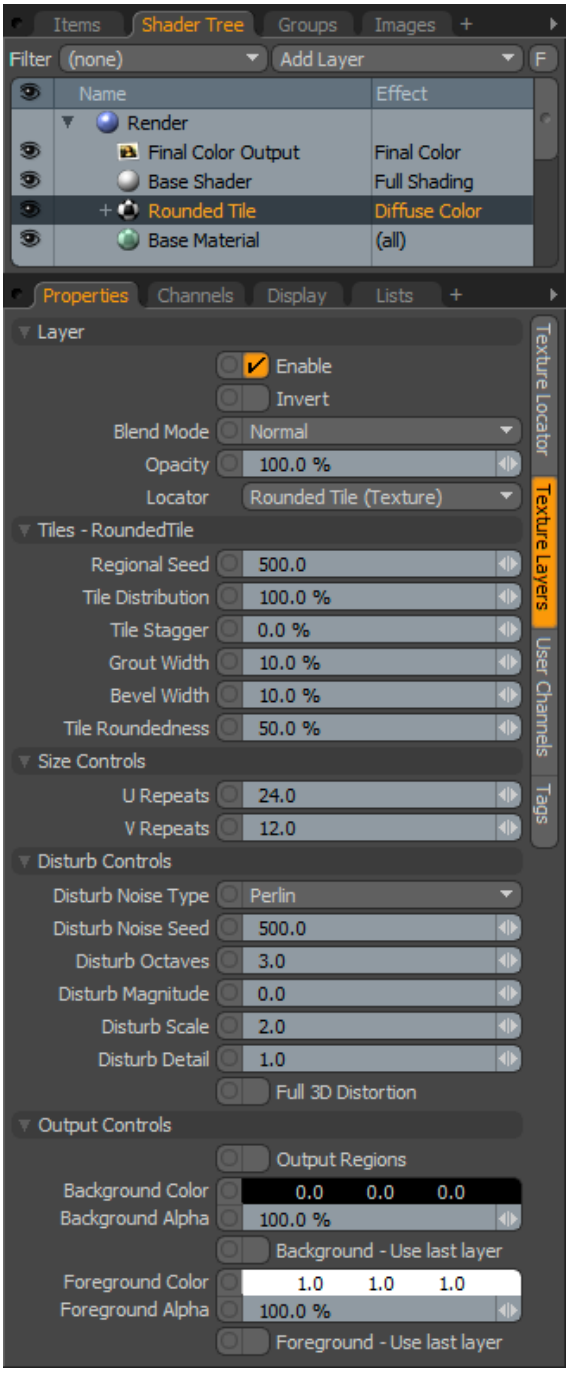
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Rounded Tile --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Tile Distribution: This value determines the percentage of the surface that is covered by tiles.

Tile Stagger: This value determines how staggered the rows of tiles are in the texture. When the control is at a low value, the tiles line up vertically, with the tile rows becoming more staggered as the parameter is increased.

Grout Width: This value determines the amount of grout between the tiles.

Bevel Width: This value determines the width of the bevel on the tiles.

Tile Roundness: This value controls the radius of each tile's corner. The width of the bevel on the tiles. When the control is at a low value, the tiles are square, becoming more rounded as the parameter is increased.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

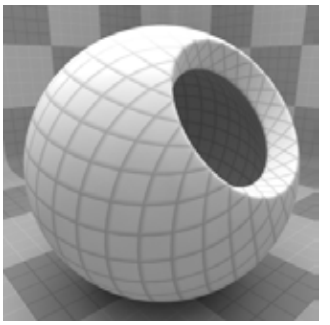
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Shingles



The Shingles texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Shingles texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a pattern similar to a *shingle* roof. The bevel width of each shingle can be varied, along with its respective height.

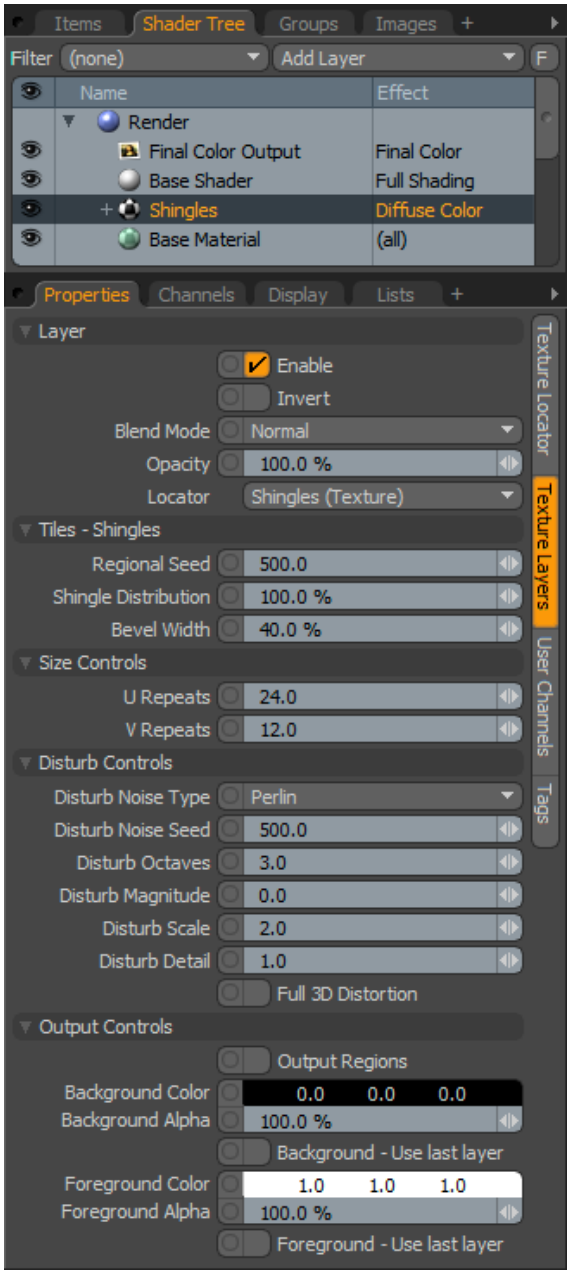
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Shingles --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Shingle Distribution: This value determines the percentage of the surface that is covered by shingles.

Bevel Width: This value determines the width of the bevel on the shingles.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

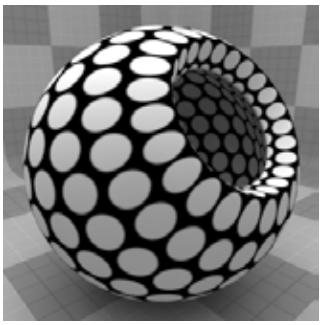
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Spots



The Spots texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Spots texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a series of *spotted*, circular tiles, which feature the same controls as the *Bath Tile* shader.

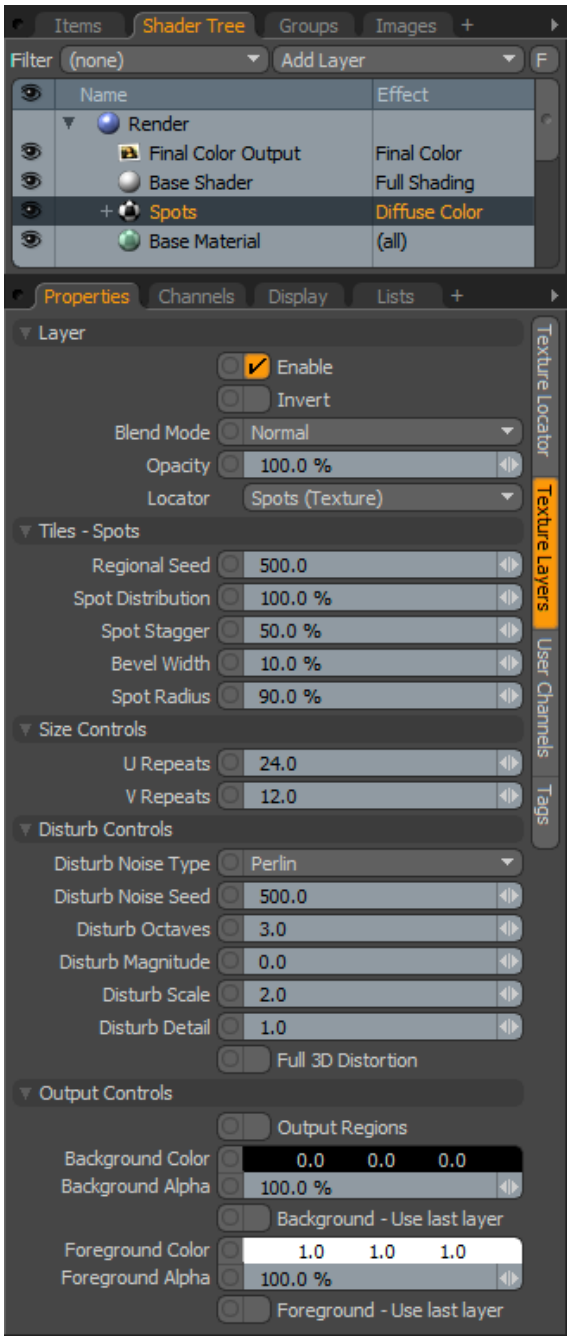
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Spots --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Spot Distribution: This value determines the percentage of the surface that is covered by spots.

Spot Stagger: This value determines how staggered the rows of spots are in the texture. When the control is at a low value, the spots line up vertically, with the spots becoming more staggered as the parameter is increased.

Bevel Width: This value determines the width of the bevel on the tiles.

Spot Radius: This value determines how much the individual tile area is filled in with the spot.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

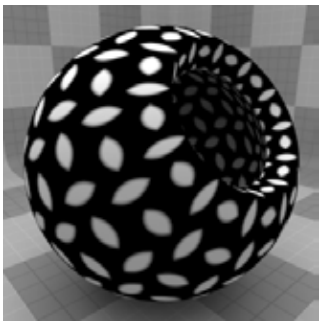
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Stamped



The Stamped texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Stamped texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a pattern similar to a *stamped* metal plate. The number of stamps per cells can be increased, along with the shape, grout width, bevel width and height.

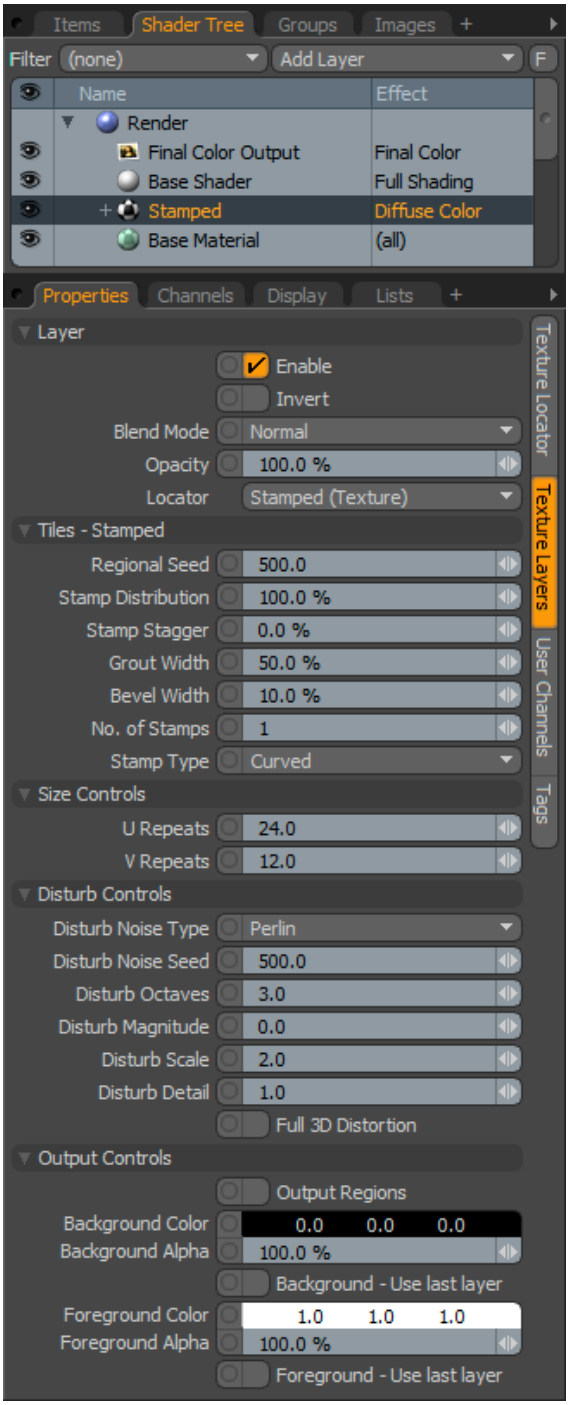
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Stamped --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Stamp Distribution: This value determines the percentage of the surface that is covered by spots.

Stamp Stagger: This value determines how staggered the rows of stamps are in the texture. When the control is at a low value, the stamps line up vertically, with the stamps becoming more staggered as the parameter is increased.

Grout Width: This value determines the amount of grout between the stamps.

Bevel Width: This value determines the width of the bevel on the stamps.

Number Of Stamps: This value controls the number of stamps in each block.

Stamp Type: This option determines the shape of the stamp used. Options are:

- Straight
- Curved
- Capsule

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

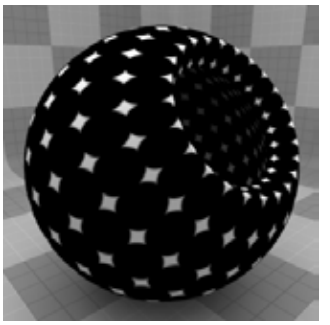
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Tacos



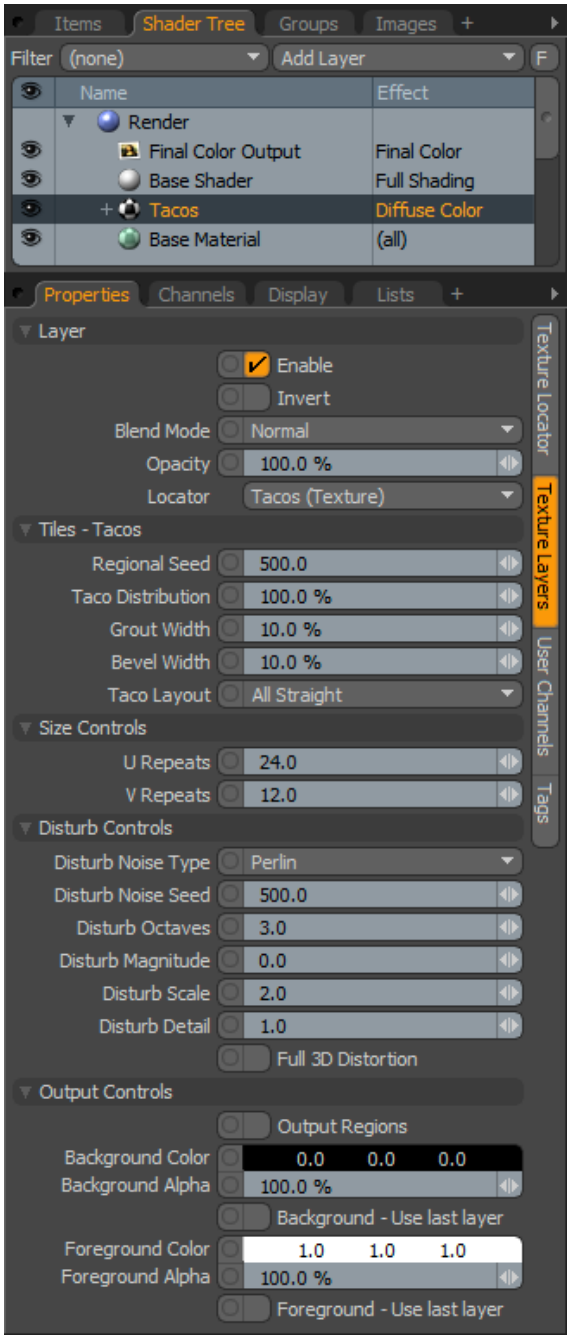
The Tacos texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Tacos texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides small, diamond-shaped tiles (tacos) utilizing the same set of controls as *Bath Tile* shader. These *tacos* can be used either independently or to fill the gaps in the *Cornerless* tile shader.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.



Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Tacos --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Taco Distribution: This value determines the percentage of the surface that is covered by tacos (diamond shapes).

Grout Width: This value determines the amount of grout between the tacos.

Bevel Width: This value determines the width of the bevel on the tacos.

Taco Layout: This option determines the layout of the taco shapes. Options are:

- All Straight
- Alternate Straight
- All Curved
- Alternate Curved

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

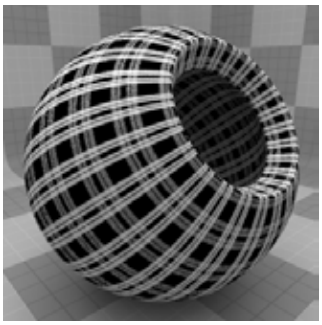
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

TarTan



The TarTan texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The TarTan texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a tartan pattern, similar to the Plaid shader, except the actual weave is visible.

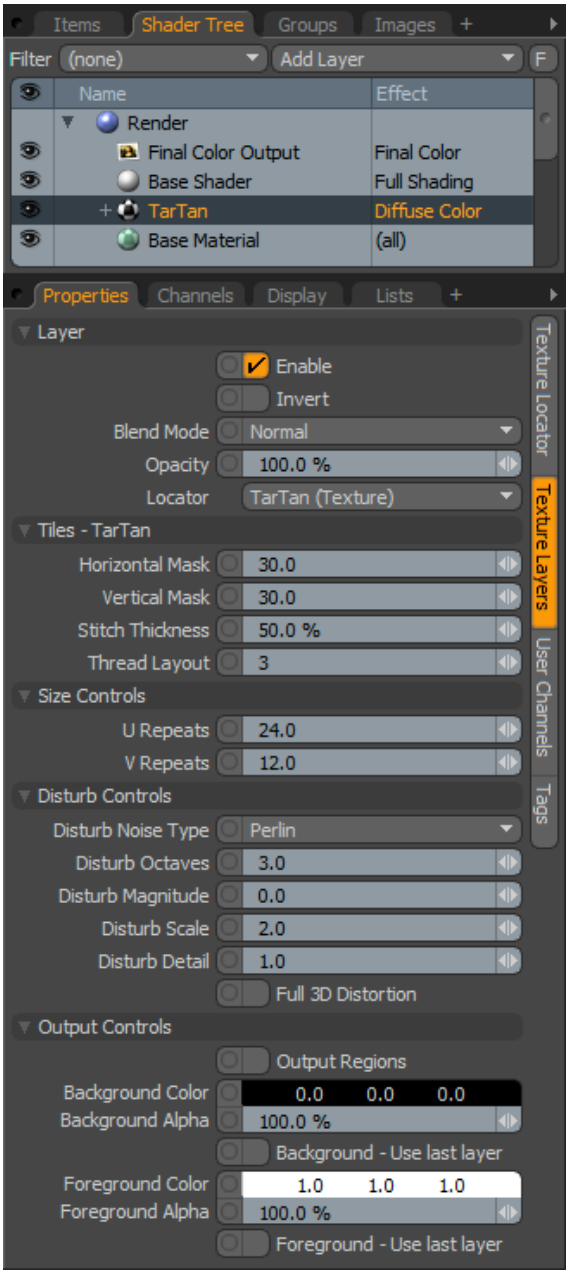
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - TarTan --

Horizontal Mask: This value controls the mask for the horizontal threads.

Vertical Mask: This value controls the mask for the vertical threads.

Stitch Thickness: This value determines the thickness of the stitching.

Thread Layout: This value controls the options for thread layout, providing five different options.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

The 'Disturb' controls are used to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

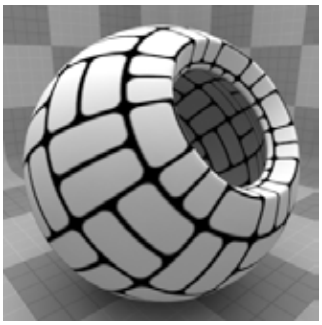
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Tiler



The Tiler texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Tiler texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a series of brick, or *tile*, patterns where the tiles have rounded corners. The available parameters provide a total of 28 different layouts.

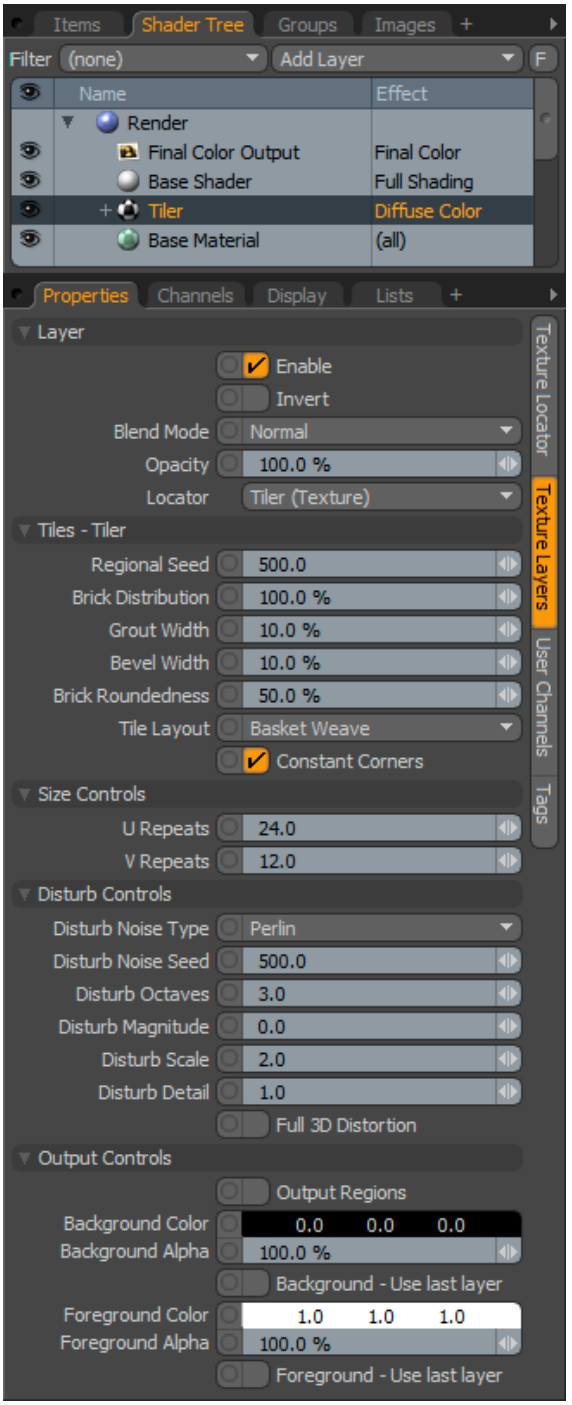
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Tiler --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Brick Distribution: This value determines the percentage of the surface that is covered by bricks.

Grout Width: This value determines the amount of grout between the tiles.

Bevel Width: This value determines the width of the bevel on the tiles.

Brick Roundness: This value determines the roundness of individual tile corners.

Tile Layout: This option determines the shape and layout of the paving tiles. There are 28 total options. Examples include:

- Basket Weave
- Windmill
- Trellis
- Cobbled
- Bricks 1

Constant Corners: This option normalizes the corners, so the larger bricks match the smaller bricks.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users

a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

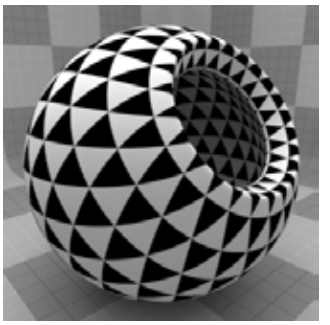
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Tri Checks



The Tri Checks texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Tri Check texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a standard *triangular check* pattern. This texture is well suited for flooring or clothing.

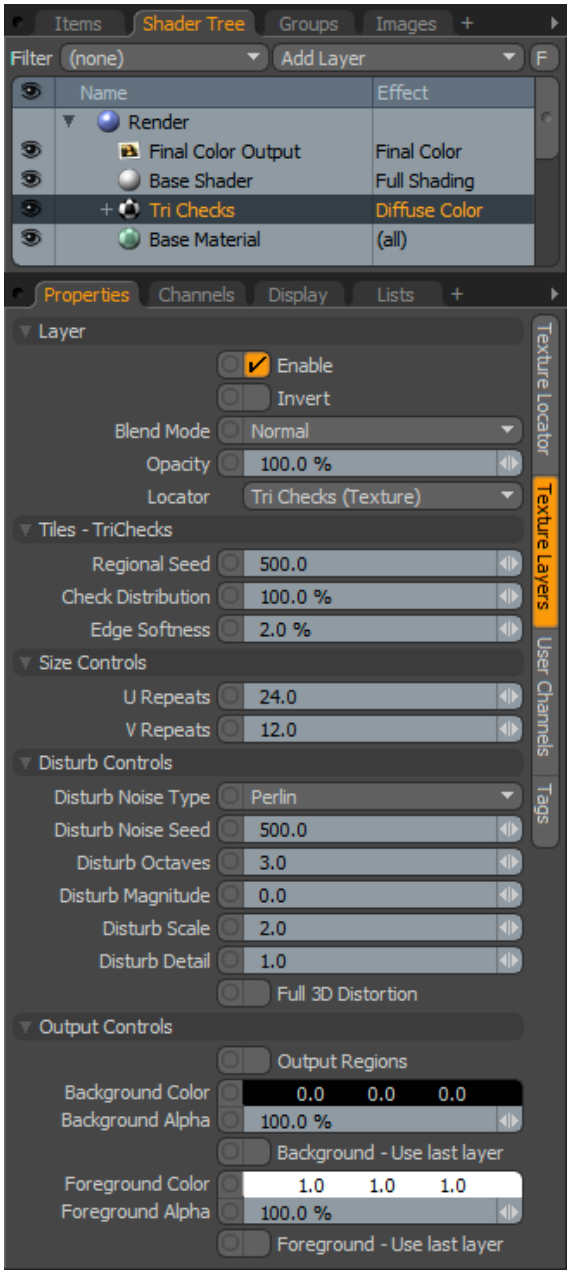
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Tri Checks--

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Check Distribution: This value determines the percentage of the surface that is covered by checks.

Edge Softness: This value controls the blurring, or softness, of the check edges.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

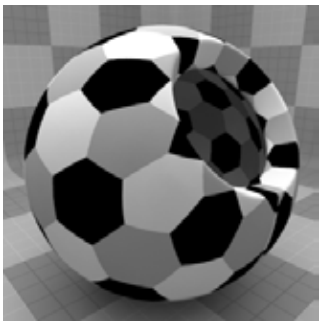
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Tri Hexes



The Tri Hexes texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Tri Hexes texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a tri-colored *hexagonal check* pattern. This texture is well suited for flooring or clothing.

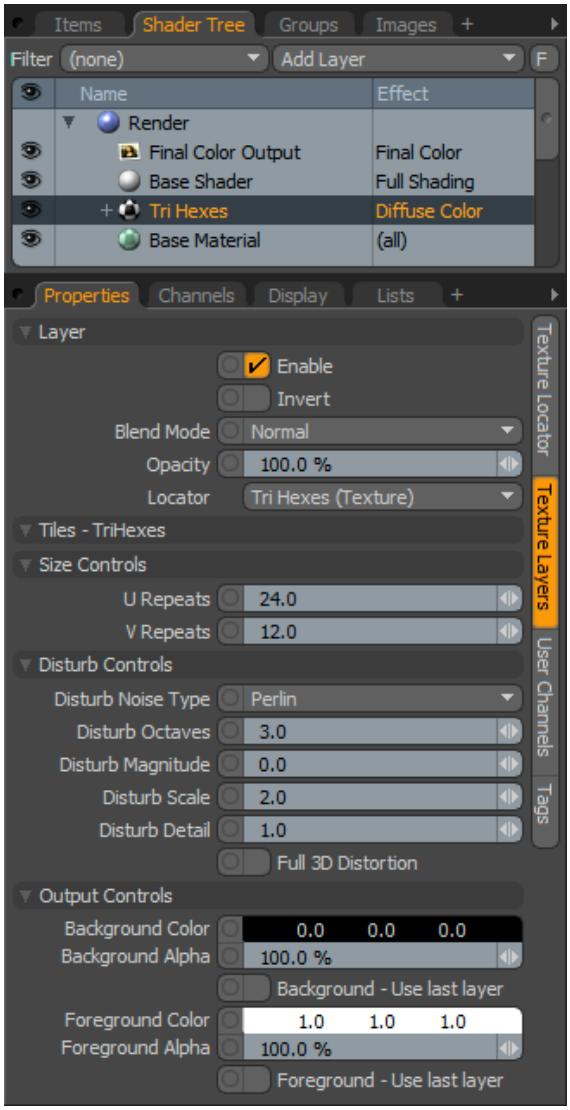
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

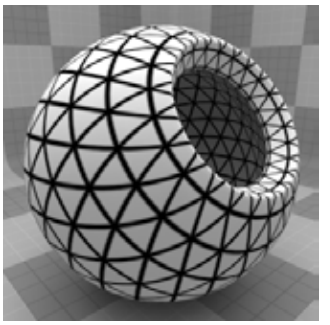
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Tri Tile



The Tri Tile texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Tri Tile texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides *triangle tile* pattern with variable grout, tiles edges and tile height. The tiles may be staggered and random tiles can be removed.

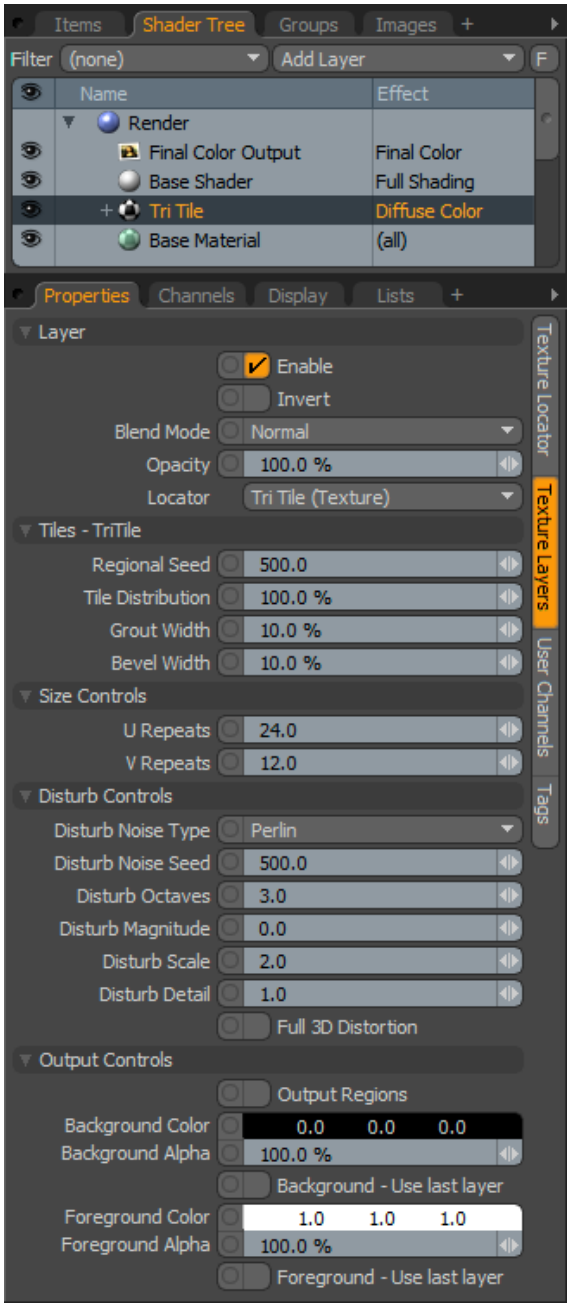
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Tri Tile--

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Tile Distribution: This value determines the percentage of the surface that is covered by tiles.

Grout Width: This value determines the amount of grout between the tiles.

Bevel Width: This value determines the width of the bevel on the tiles.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

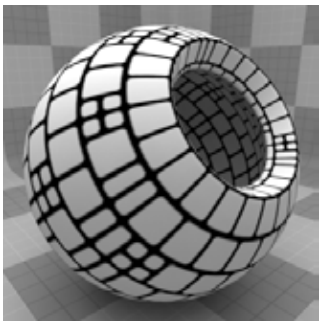
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Wall



The Wall texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Wall texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a brick *wall*-type pattern where the bricks are randomly sized. The number of patterns is adjustable, allowing for a variety of wall types.

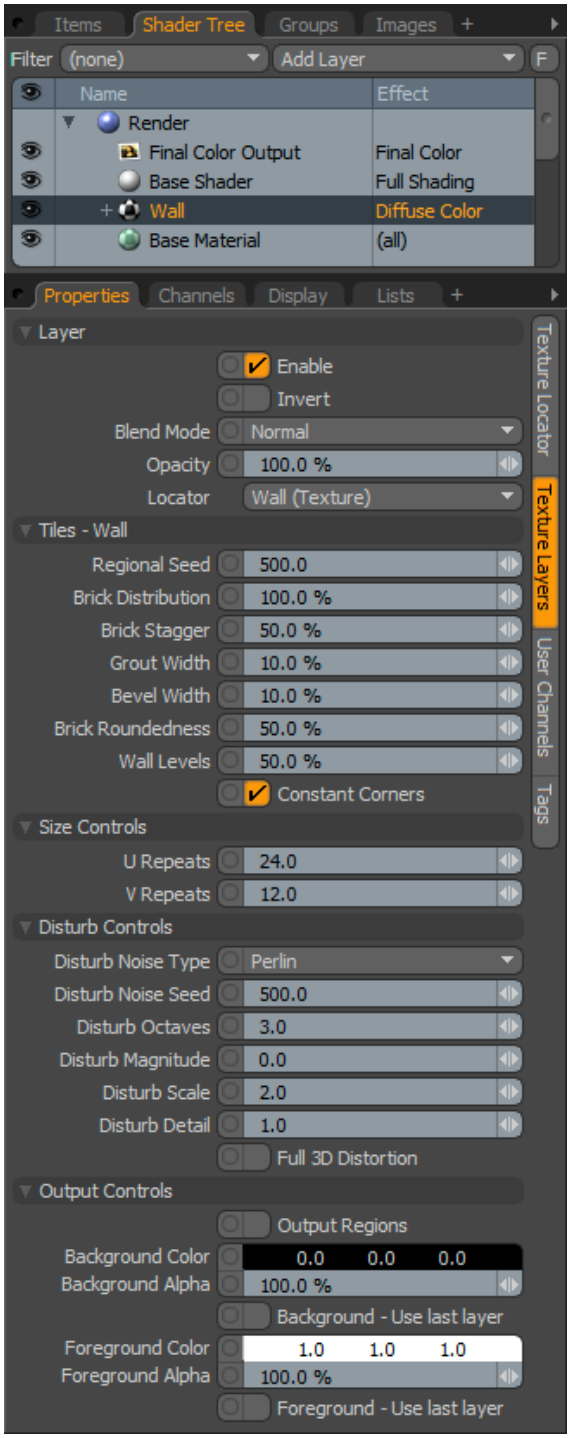
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Tiles - Wall --

Regional Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Brick Distribution: This value determines the percentage of the surface that is covered by bricks.

Brick Stagger: This value determines how staggered the rows of bricks are in the texture. When the control is at a low value, the bricks line up vertically, with the brick rows becoming more staggered as the parameter is increased.

Grout Width: This value determines the amount of grout between the tiles.

Bevel Width: This value determines the width of the bevel on the tiles.

Brick Roundness: This value determines the roundness of individual tile corners.

Wall Levels: This value determines how the bricks are sized and placed in the wall. At low values the bricks will be square in shape and set in even rows and columns. As the parameter is increased, brick sizes become variable and are placed in offset patterns.

Constant Corners: This option normalizes the corners, so the larger bricks match the smaller bricks.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Other (Display)

Counter 1

Counter 2

UVLEDs

Other (Panels)

Peel

Plates

Rivet Rust

Rivets

Rust

Smear

Other (Process)

Easy Gradient

Regional HSV

Other (Space)

Blast

Coriolis

Flare

Gas Giant

Hurricane

Nurnies

Planet

Planet Clouds

Rings

Star Field

Swirl

Terra

Windows

Other (Water)

Drip Drop

Rain

Ripples

Surf

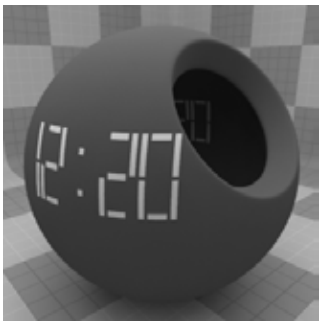
Waves

Windy Waves

Other (Waveforms)

Waveforms

Counter 1



The Counter 1 texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The shaders in the Display group differ from the other Enhance:modo textures in that they present procedurally generated text rather than traditional textures. Similar to the other shaders, Counter 1 can be addressed by its two zones, the 'Background' and 'Foreground' colors, with each zone having either a 'Value', or a 'Color' and 'Alpha'. For Counter 1, the foreground color specifically controls the display text. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader is a digital timer readout, such as those seen on a speedometer or a clock.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

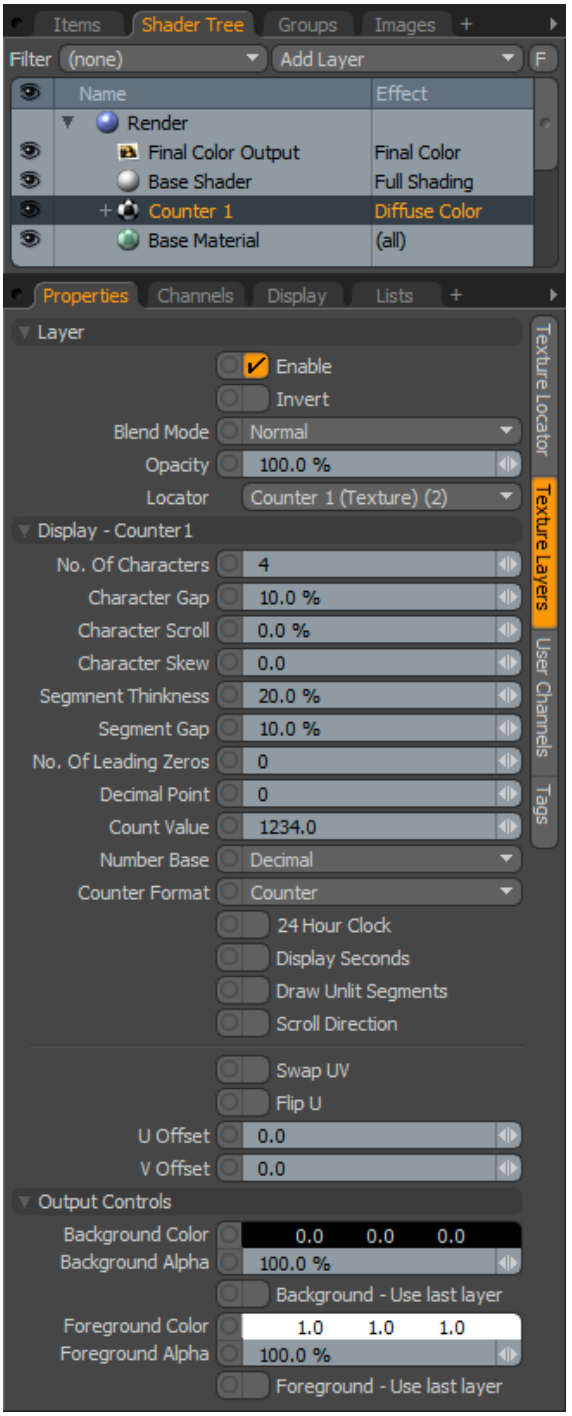
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Display - Counter 1 --

Number of Characters: This value determines the total number of characters to display.

Character Gap: This value determines the spacing between the individual characters within the overall content dimensions.

Character Scroll: This value allows the characters to scroll on a character-by-character basis. As this value is increased the characters will scroll from left to right.

Character Skew: This value applies a skew to all of the display characters. Positive values will skew the characters to the right, while negative values will skew the characters to the left.

Segment Thickness: This value

defines the thickness of the individual segments for each display character.

Segment Gap: This value determines the spacing between the individual segments for each character

No. Of Leading Zeros: This value prefixes the counter with a defined number of zeros.

Decimal Point: This value determines the number of zeros to be shown after a decimal point.

Count Value: This value determines the number that is shown in the display. In the case of time, the value is represented in seconds, then converted to hours:minutes in the display.

Number Base: This option determines the number base used to display the count value. The options are Decimal, Hexadecimal, Octal and Binary.

Counter Format: This option determines the format of the count for display. Options are Simple Count, Realtime Clock,or SMPTE time code (either SMPTE24, SMPTE25 or SMPTE30.)

24 Hour Clock: This option forces a 24 hour clock display when the Realtime Clock option is selected in the Counter Format.

Display Seconds: When this option is enabled, and the Counter Format is set to Realtime Clock, the character for seconds will be included in the display.

Draw Unlit Segments: When this option is enabled all of the unlit segments and decimals will be included in the display.

Scroll Direction: When this option is enabled, the scroll direction will switch from the default setting of left to right and scroll from right to left.

Swap UV: When this option is enabled the the UV coordinates of the display text are swapped.

Flip U: When this top is enabled it flips the U coordinates of the display text.

U Offset: This value determines U (horizontal) offset of the display text.

V Offset: This value determines V (vertical) offset of the display text.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value,

users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

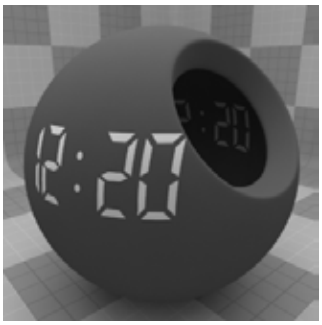
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Counter 2



The Counter 2 texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The shaders in the Display group differ from the other Enhance:modo textures in that they present procedurally generated text rather than traditional textures. Similar to the other shaders, Counter 2 can be addressed by its two zones, the 'Background' and 'Foreground' colors, with each zone having either a 'Value', or a 'Color' and 'Alpha'. For Counter 2, the foreground color specifically controls the display text. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. The look of this particular shader is a digital timer readout, such as those seen on a speedometer or clock. The difference between *Counter 1* and *Counter 2* is that Counter 2 provides a more modern display face, with beveled corners.

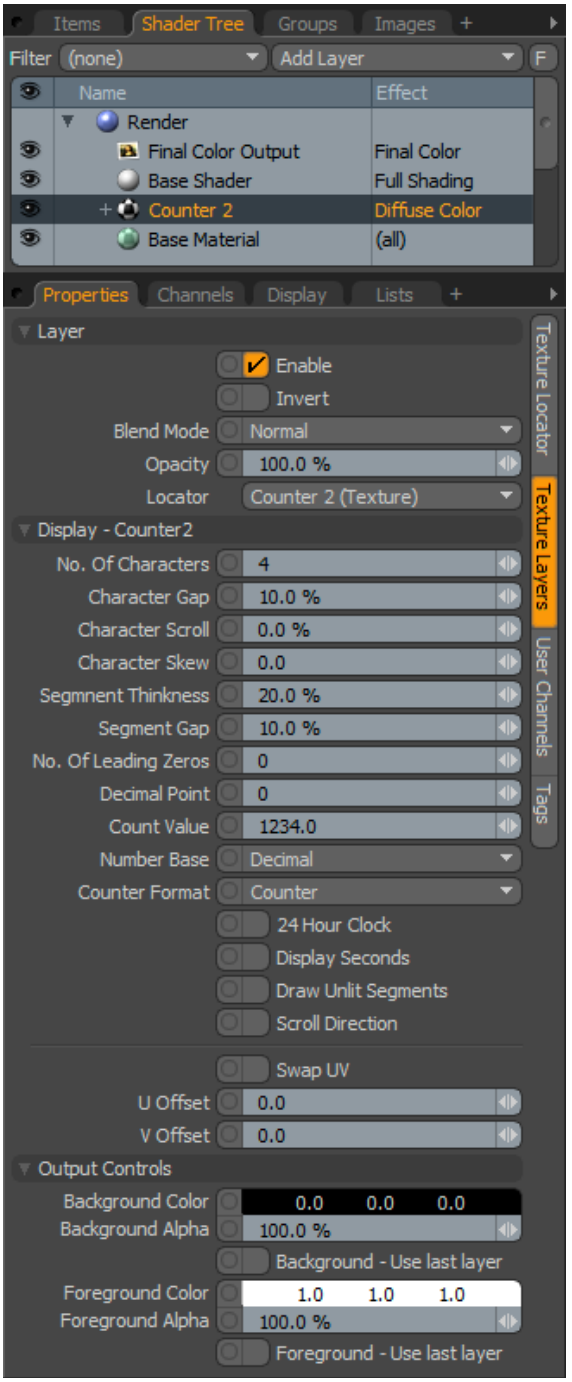
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.



Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Display - Counter 2 --

Number of Characters: This value determines the total number of characters to display.

Character Gap: This value determines the spacing between the individual characters within the overall content dimensions.

Character Scroll: This value allows the characters to scroll on a character-by-character basis. As this value is increased the characters will scroll from left to right.

Character Skew: This value applies a skew to all of the display characters. Positive values will skew the characters to the right, while negative values will skew the characters to the left.

Segment Thickness: This value defines the thickness of the individual segments for each display character.

Segment Gap: This value determines the spacing between the individual segments for each character.

No. Of Leading Zeros: This value prefixes the counter with a defined number of zeros.

Decimal Point: This value determines the number of zeros to be shown after a decimal point.

Count Value: This value determines the number that is shown in the display. In the case of time, the value is represented in seconds, then converted to hours:minutes in the display.

Number Base: This option determines the number base used to display the count value. The options are Decimal, Hexadecimal, Octal and Binary.

Counter Format: This option determines the format of the count for display. Options are Simple Count, Realtime Clock, or SMPTE time code (either SMPTE24, SMPTE25 or SMPTE30.)

24 Hour Clock: This option forces a 24 hour clock display when the Realtime Clock option is selected in the Counter Format.

Display Seconds: When this option is enabled, and the Counter Format is set to Realtime Clock, the character for seconds will be included in the display.

Draw Unlit Segments: When this option is enabled all of the unlit segments and decimals will be included in the display.

Scroll Direction: When this option is enabled, the scroll direction will switch from the default setting of left to right and scroll from right to left.

Swap UV: When this option is enabled the the UV coordinates of the display text are swapped.

U Offset: This value determines U (horizontal) offset of the display text.

V Offset: This value determines V (vertical) offset of the display text.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

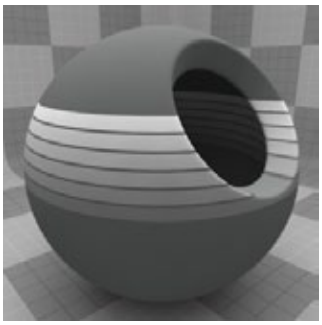
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

UV LEDs



The UVLEDs texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The shaders in the Display group differ from the other Enhance:modo textures in that they present procedurally generated text rather than traditional textures. Similar to the other shaders, UVLEDs can be addressed by its two zones, the 'Background' and 'Foreground' colors, with each zone having either a 'Value', or a 'Color' and 'Alpha'. For UVLEDs, the foreground color specifically controls the display text. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This particular shader provides a row of round or square colored LEDs. The display can be animated to provide a variety of blinking, rotating, or flashing LEDs.

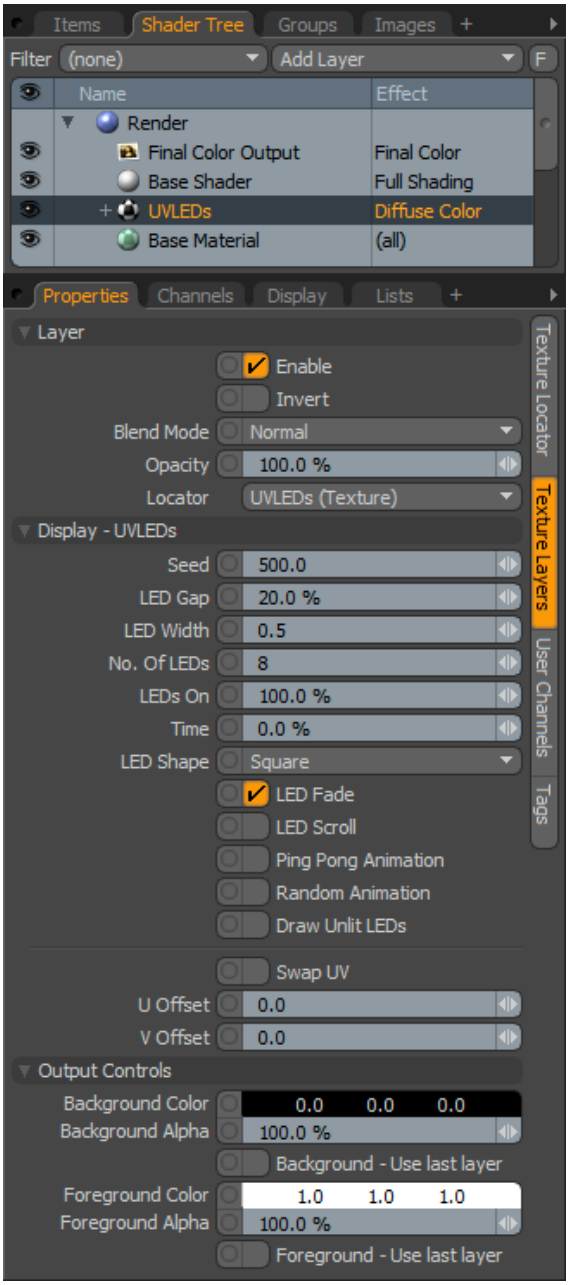
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.



Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Display - UVLEDs --

Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

LED Gap: This value determines the spacing between the individual LEDs.

LED Width: This value determines width of the individual LEDS.

Number Of LEDs: This value determines the total number of LEDs in the display.

LEDs On: This value determines the number of LEDs to be drawn, with a value of 100% showing all LEDs. This setting is useful for controlled animations.

Time: This value controls the time value of the LEDS for use with the animation controls, such as Ping Pong Animation or Random Animation.

LED Shape: This option determines the shape of the LEDs. The options are Round or Square.

LED Fade: This option determines whether the LEDs fade from beginning to end or maintain the consistent Foreground Color.

LED Scroll: This option, in conjunction with the Time value creates an animation that cycles through the LEDs from top to bottom.

Ping Pong Animation: This option, in conjunction with the Time value, creates an animation that cycles the LEDs on and off, from the bottom to the top and then back to the bottom.

Random Animation: This option, in conjunction with the Time value, creates an animation that randomly cycles through the LEDs.

Draw Unlit LEDs: When this option is enabled all of the unlit LEDs, as determined by the LED's "On" value.

--

Swap UV: When this option is enabled the the UV coordinates of the display text are swapped.

U Offset: This option determines U offset of the display text.

V Offset: This option determines V offset of the display text.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

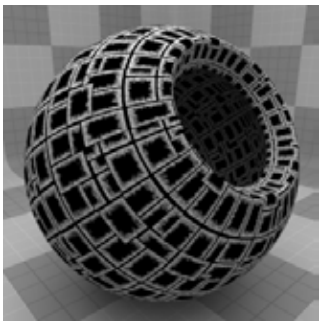
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Peel



The Peel texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Peel texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a series of rectangular panels with a *peeling* paint-like texture applied.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Note: In order for this texture to display properly, you must navigate to the Texture Locator window and set the following options:

Projection Type: Set to UV Map.

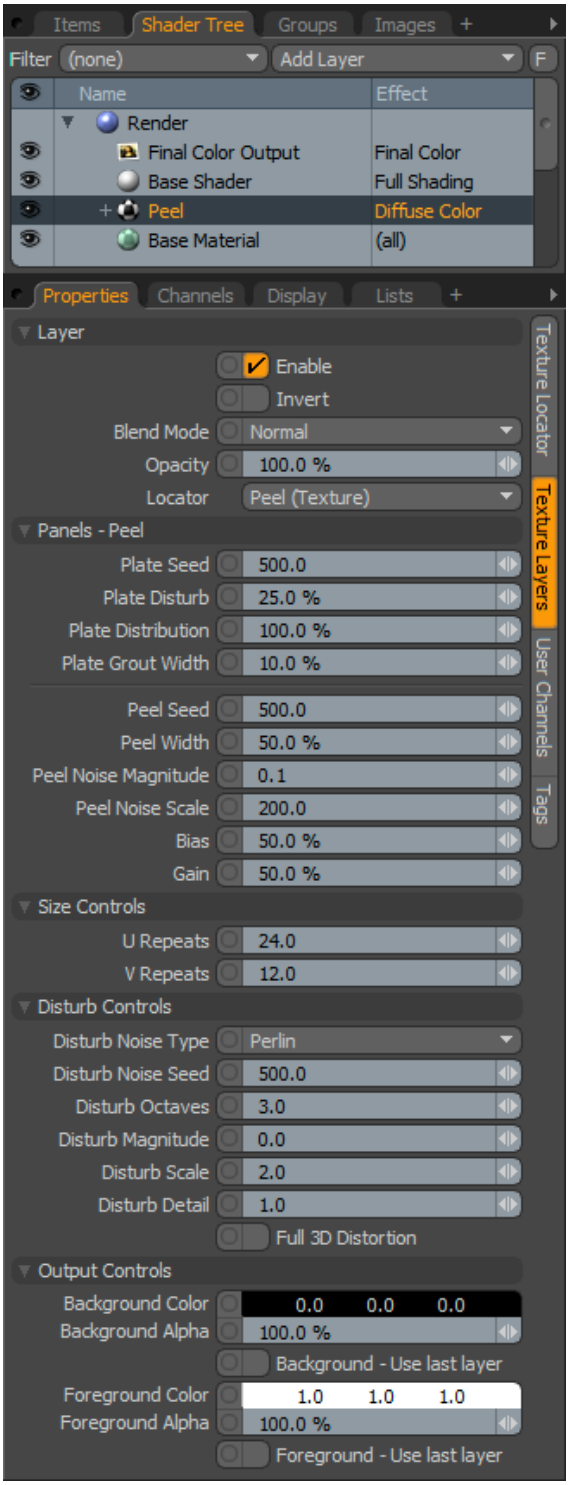
UV Map: Set to Texture.

For more information on this window, please reference the 'Texture Locator' page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Panels - Peel --

Plate Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Plate Disturb: This value determines the amount of distortion among the plates. When the control is at a low value, the plates will be very uniform, with the plate sizes having more variation as the parameter is increased.

Plate Distribution: This value determines the percentage of the surface that is covered by the plates.

Plate Grout Width: This value determines the percentage of the peel that is displayed, as defined from the outer edge of the plate.

--

Peel Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Peel Width: This value determines the strength of the distortion effect on the base procedural.

Peel Noise Magnitude: This value determines the strength of the distortion effect for the peeling.

Peel Noise Scale: This value determines the scale, or frequency, of the distortion effect for the peeling.

Bias: This value determines the brightness of the overall texture.

Gain: This value controls the gain, or contrast, of the overall texture.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very

sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

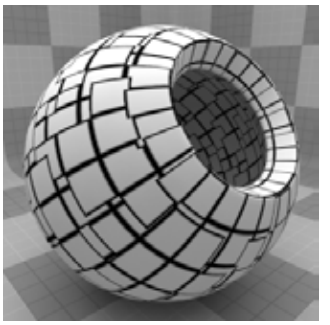
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Plates



The Plates texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Plates texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a series of rectangular *plates* which can be used to create textures such as floor tiles or metal armor.

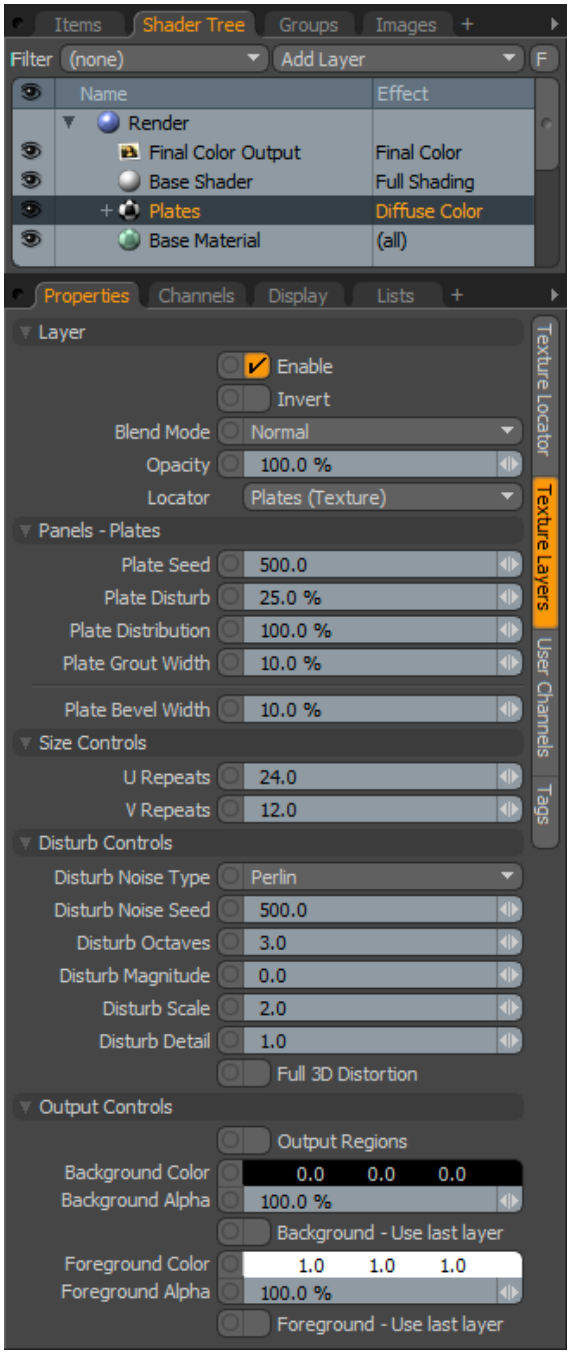
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Panels - Plates --

Plate Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Plate Disturb: This value determines the amount of distortion among the plates. When

the control is at a low value, the plates will be very uniform, with plates sizes having more variation as the parameter is increased.

Plate Distribution: This value determines the percentage of the surface that is covered by plates.

Plate Grout Width: This value determines the amount of grout between the individual plates.

--

Plate Bevel Width: This value determines the length of the bevel on the indviudal plates, as measured from the outer edge to the center of the plate.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

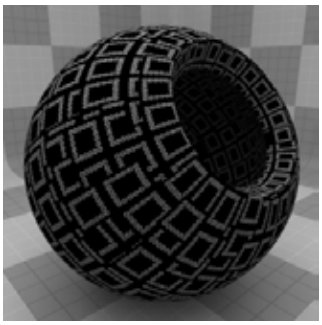
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Rivet Rust



The Rivets Rust texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Rivets Rust texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a series of rectangular plates with a border of *rusty rivets*. This shader can be used to texture an old metal ships hull or a rusty bridge tower.

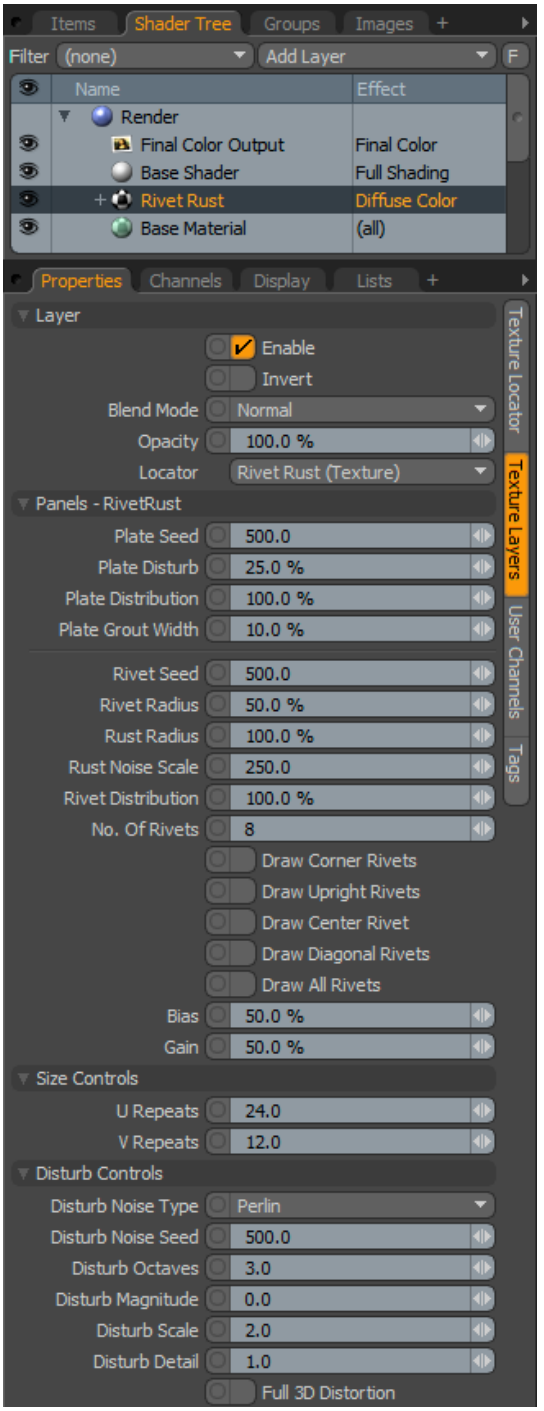
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Panels - Rivet Rust--

Plate Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Plate Disturb: This value determines the amount of distortion among the plates. When the control is at a low value, the plates will be very uniform, with plates sizes having more variation as the parameter is increased.

Plate Distribution: This value determines the percentage of the surface that is covered by plates.

Plate Grout Width: This value determines the amount of grout between the individual plates.

Rivet Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Rivet Radius: This value determines the radius of the individual rivets.

Rust Radius: This value determines the radius of the rust on the rivets.

Rust Noise Scale: This value controls the overall frequency or scale of the rust.

Rivet Distribution: This value determines the maximum amount of rivets that will be drawn.

Number Of Rivets: This value determines the number of rivets to be drawn on each edge.

Draw Corner Rivets: This option will draw the texture using only the rivets in the corner of the panels.

Draw Upright Rivets: This option will draw the texture with the rivets in a cross pattern within each panel.

Draw Center Rivets: This option will draw the texture using one rivets in the center of each panel.

Draw Diagonal Rivets: This option will draw the texture with line of diagonal rivets across each panel.

Draw All Rivets: This option will draw the texture using all available rivets in each panels.

Bias: This value determines the brightness of the overall texture.

Gain: This value controls the gain, or contrast, of the overall texture.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

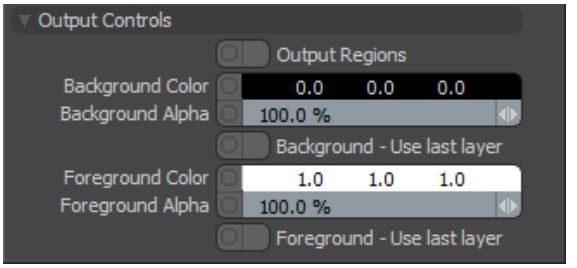
Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.



Output Controls--

Lower Clip: This value determines a clip level for the Background Color/ Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply

this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

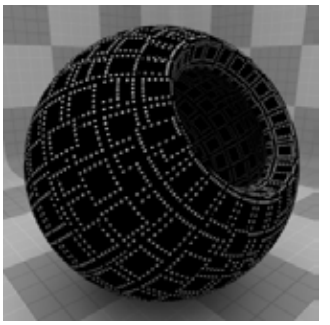
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Rivets



The Rivets texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Rivets texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a series of rectangular plates with a border of *rivets*. This shader can be used to texture a metal ships hull or a bridge tower.

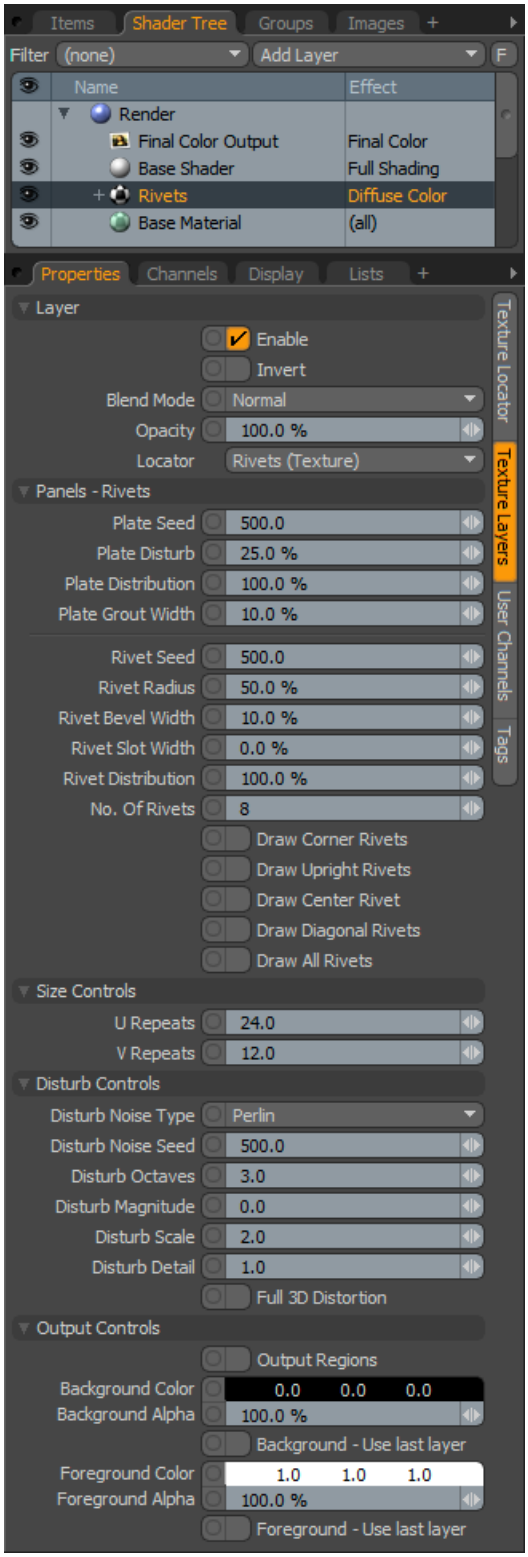
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Panels - Rivets --

Plate Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Plate Disturb: This value determines the amount of distortion among the plates. When the control is at a low value, the plates will be very uniform, with plates sizes having more variation

as the parameter is increased.

Plate Distribution: This value determines the percentage of the surface that is covered by plates.

Plate Grout Width: This value determines the amount of grout between the individual plates.

--

Rivet Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Rivet Radius: This value determines the radius of the individual rivets.

Rivet Bevel Width: This value determines the width of the bevel on the rivets.

Rivet Slot Width: This value determines the width of the slot between the rivets

Rivet Distribution: This value determines the maximum amount of rivets that will be drawn.

Number Of Rivets: This value determines the number of rivets to be drawn on each edge.

Draw Corner Rivets: This option will draw the texture using only the rivets in the corner of the panels.

Draw Upright Rivets: This option will draw the texture with the rivets in a cross pattern within each panel.

Draw Center Rivets: This option will draw the texture using one rivets in the center of each panel.

Draw Diagonal Rivets: This option will draw the texture with line of diagonal rivets across each panel.

Draw All Rivets: This option will draw the texture using all available rivets in each panels.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

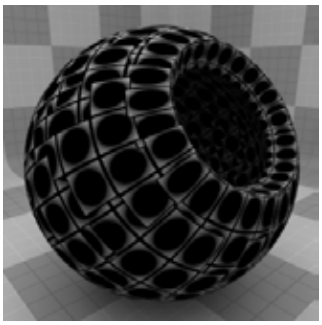
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Rust



The Rust texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Rust texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a series of rectangular plates with a internal frame of *rust*. This shader can be used to texture an old water tank or a cobblestone street.

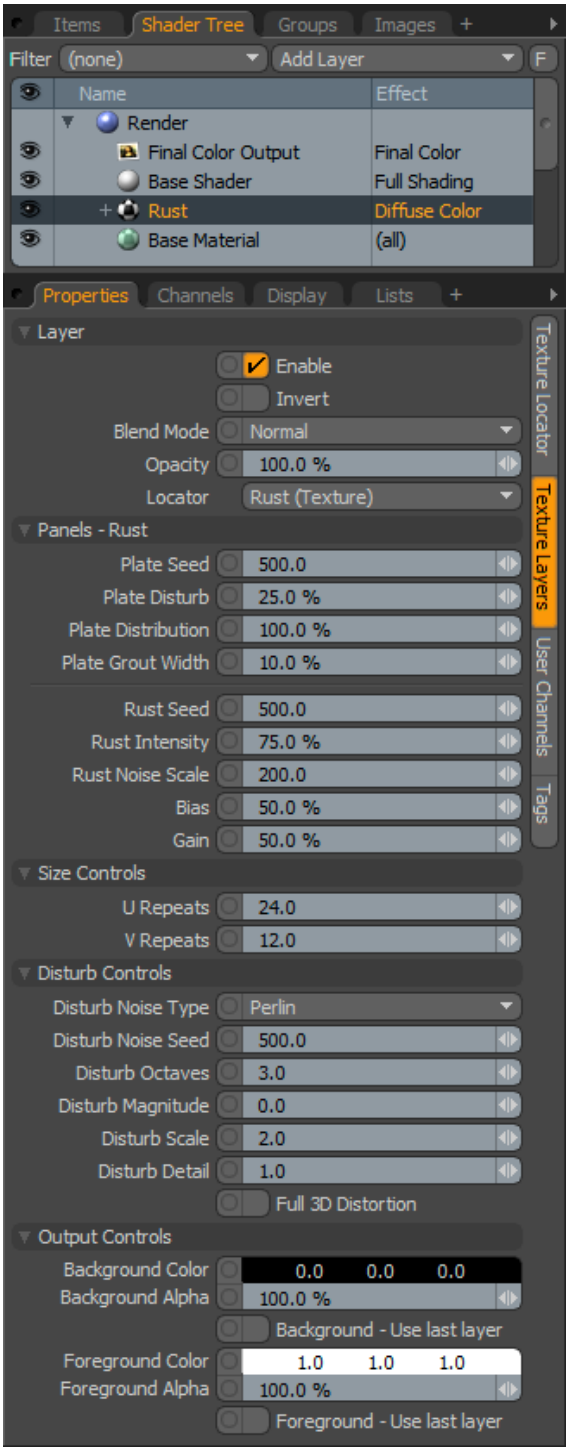
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Panels - Rust --

Plate Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Plate Disturb: This value determines the amount of distortion among the plates. When the control is at a low value, the plates will be very uniform, with plates sizes having more variation as the parameter is increased.

Plate Distribution: This value determines the percentage of the surface that is covered by plates.

Plate Grout Width: This value determines the amount of grout between the individual plates.

--

Rust Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Rust Intensity: This value determines the amount of rust inside the individual plates, as it extends out from the corners.

Rust Noise Scale: This value controls the overall frequency or scale of the rust.

Bias: This value determines the brightness of the overall texture.

Gain: This value controls the gain, or contrast, of the overall texture.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

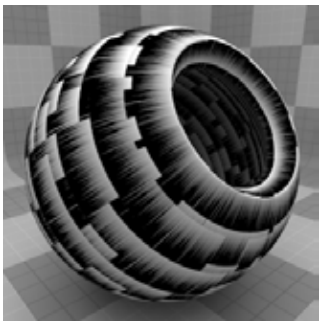
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Smear



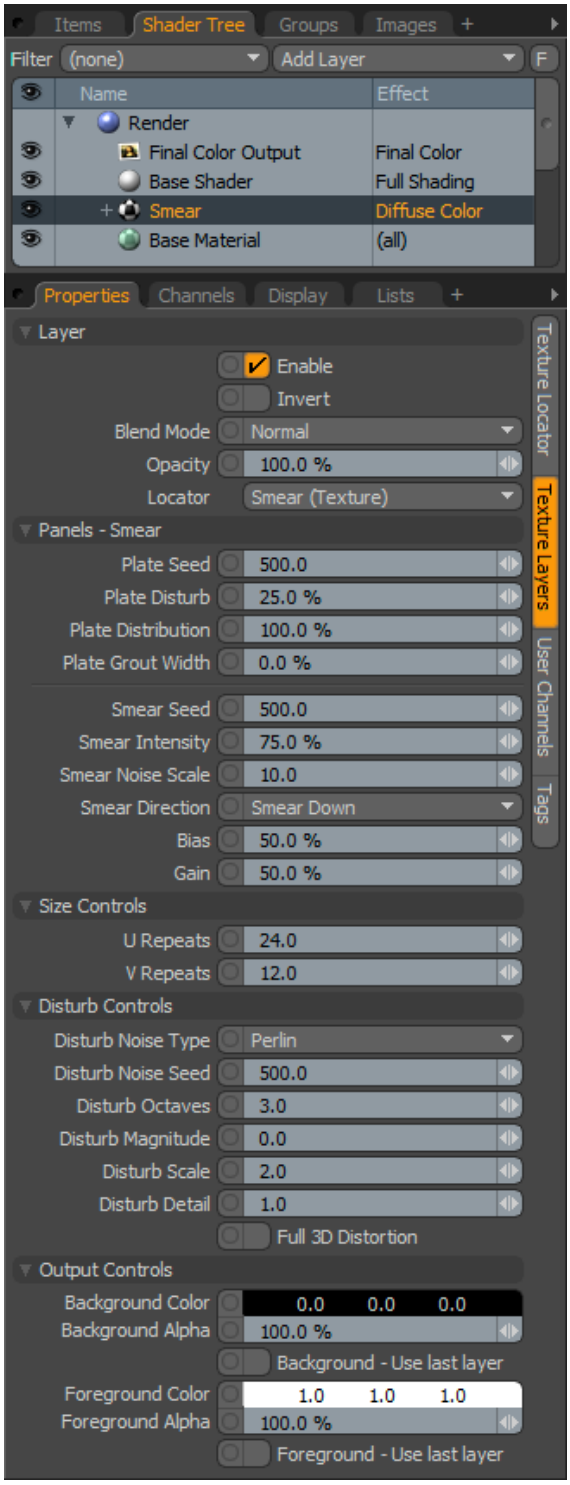
The Smear texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Smear texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a series of rectangular plates with a line of dirt or rust *smeared* in a certain direction. This shader can be used to texture a grimy cinder block wall or burned spaceship.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.



Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Panels - Smear --

Plate Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to

use the same 'Seed' value when it is necessary for items to retain the same variations.

Plate Disturb: This value determines the amount of distortion among the plates. When the control is at a low value, the plates will be very uniform, with plates sizes having more variation as the parameter is increased.

Plate Distribution: This value determines the percentage of the surface that is covered by plates.

Plate Grout Width: This value determines the amount of grout between the individual plates.

--

Smear Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Smear Intensity: This value determines the amount of smearing on the surface, as it extends out from the selected direction (Up, Down, Left, Right).

Smear Noise Scale: This value controls the overall frequency or scale of the smearing.

Smear Direction: This option determines the direction of the smearing. The options are:

- Smear Down
- Smear Up
- Smear Right
- Smear Left

Bias: This value determines the brightness of the overall texture.

Gain: This value controls the gain, or contrast, of the overall texture.

Size Controls--

U Repeats: This value determines the number of the times the pattern repeats in the U direction (horizontally).

V Repeats: This value determines the number of the times the pattern repeats in the V direction (vertically).

Disturb Controls--

Users can use the 'Disturb' controls to alter the look of the generated procedural. The noise layer will distort the base texture based on the Disturb Magnitude.

Disturb Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Disturb Octaves: Similar to the Patch Levels setting, the 'Disturb Octaves' value determines the number of layers of noise when distorting the texture, in effect producing greater detail.

Disturb Magnitude: This option determines the strength of the distortion effect on the base procedural.

Disturb Scale: This option determines the size (scale) of the distortion effect on the base procedural.

Disturb Detail: This value determines the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion: This option turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

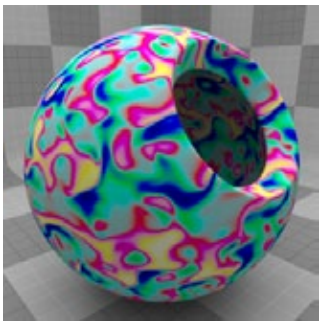
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Easy Gradient



The Easy Gradient process layer applies a user define color gradation across the grey values of any lower Shader Tree layer set to the same 'Effect' type, this makes it easy for users to colorize procedural textures that a generally defined by two colors (foreground/background). In the example above, an 'Agate' texture layer is applied, set as 'Diffuse Color' effect, with an 'Easy Gradient' layer at the default values placed directly above it in the tree also set to 'Diffuse Color'. The color gradations are applied as zones represented by the 'Blend Values'. Users can define up to 8 colors with corresponding blend zones; representing the entire grey spectrum from 0% black to 100% white.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

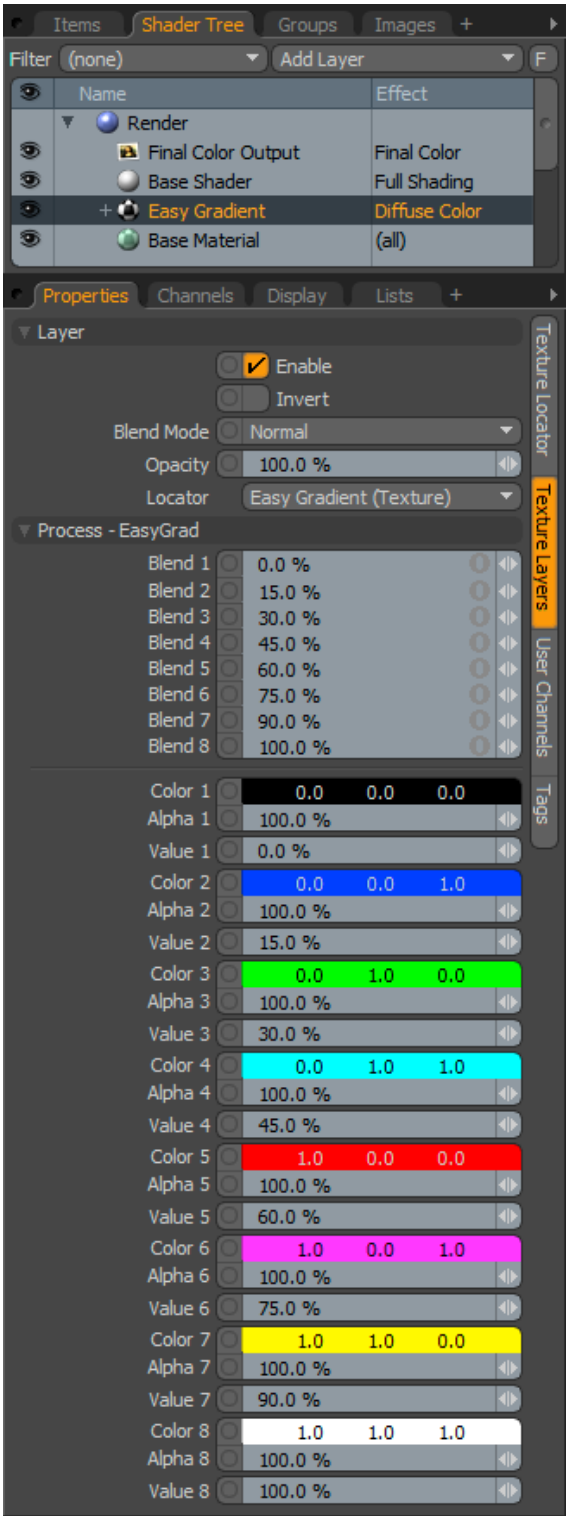
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Process - EasyGrad --

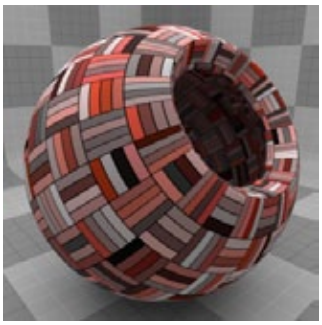
Blend 1-8: These parameters define the blend points of the gradient.

Color 1-8: These parameters define the colors of the gradient.

Alpha 1-8: These parameters define the alpha channel of the gradient.

Value 1-8: These parameters define the opacity of the gradient.

Regional HSV



The 'Regional HSV' process layer provides users a means to control variations in e:modo textures that output regions (such as the many 'Tiles' procedural layers). In order for the 'Regional HSV' layer to work properly, users must enable the 'Output Regions' option in the originating procedural texture layer. To use, define a layer that supports the 'Output Regions' options, enabling the 'Output Regions' option. Add the 'Regional HSV' layer set to the same effect type directly above the originating layer in the Shader Tree. Users may then adjust the 'Hue', 'Saturation' and 'Value' Min and Max settings to control the amount of variations in the regions.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

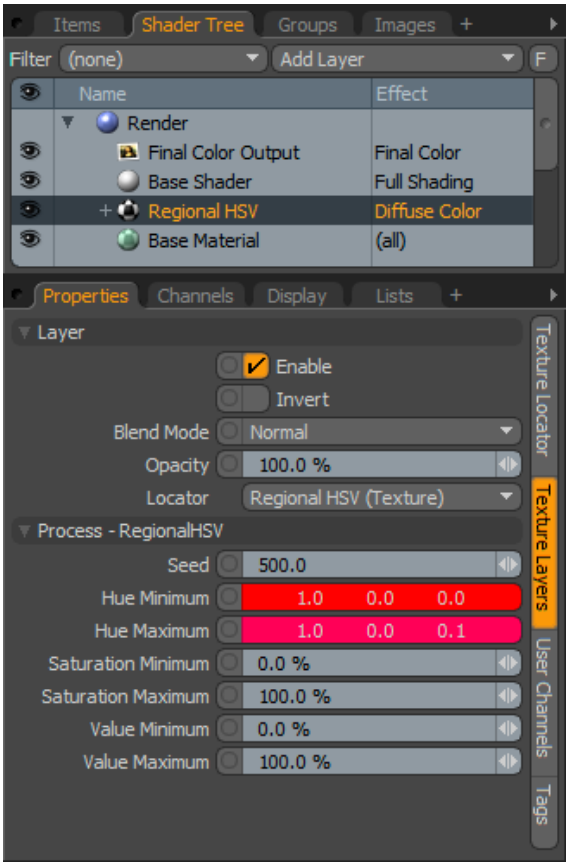
Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the

way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.



Process - Regional HSV --

Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Hue Minimum: This value determines the low hue that is used to mix the final output.

Hue Maximum: This value determines the high hue that is used to mix the final output.

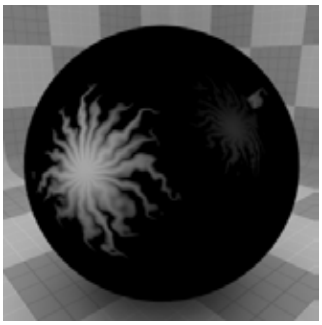
Saturation Minimum: This value determines the low saturation that is used to mix the final output.

Saturation Maximum: This value determines the high saturation that is used to mix the final output.

Value Minimum: This value determines the low intensity that is used to mix the final output.

Value Maximum: This value determines the low intensity that is used to mix the final output.

Blast



The Blast texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Blast texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a look similar to the blast marks made by a flying projectile or a futuristic phaser.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

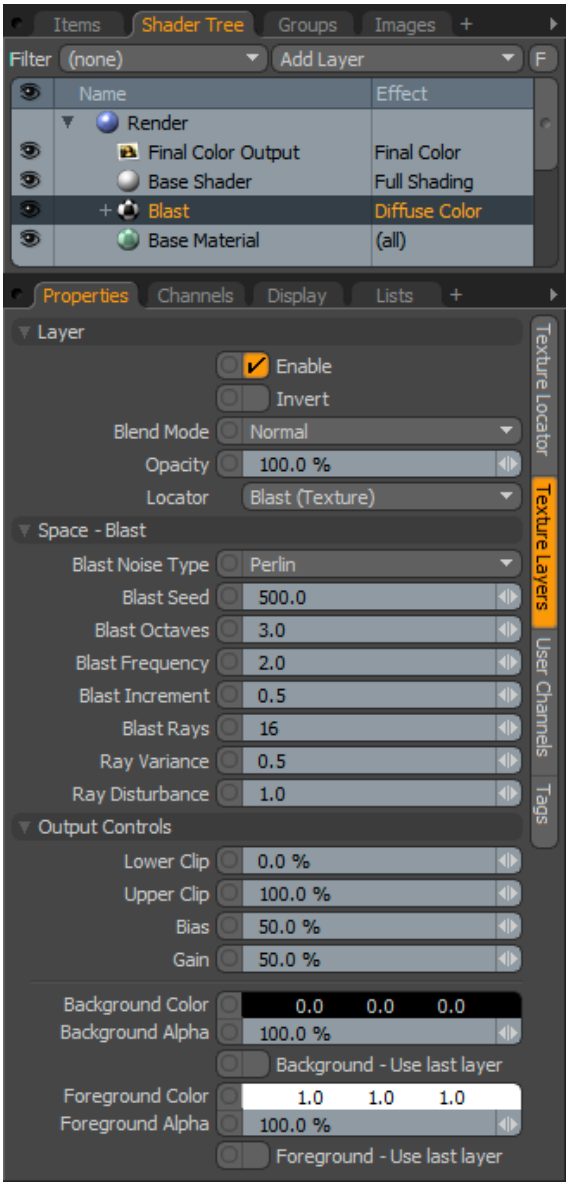
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Space - Blast --

Blast Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Blast Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Blast Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Blast Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Blast Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Blast Rays: This value controls the number of blast waves that emanate from the blast center.

Blast Variance: This value controls the variance of the blast waves as they emanate from the center. When the control is at a low value, it results in blast waves of similar distance, with the variance between the various blast waves growing as the value is increased.

Blast Disturbance: This value controls the disturbance of the blast waves as the emanate from the center. When the control is at a low value, the blast rays will be straighter, getting more disturbed as the value is increased.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

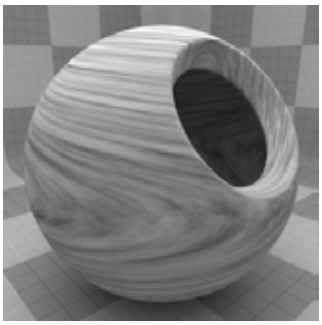
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Coriolis



The Coriolis texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Coriolis texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader applies a pattern similar to the swirling, wind swept clouds of the *Coriolis* effect. The amount of twist or swirling can be controlled, along with the detail of the clouds.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

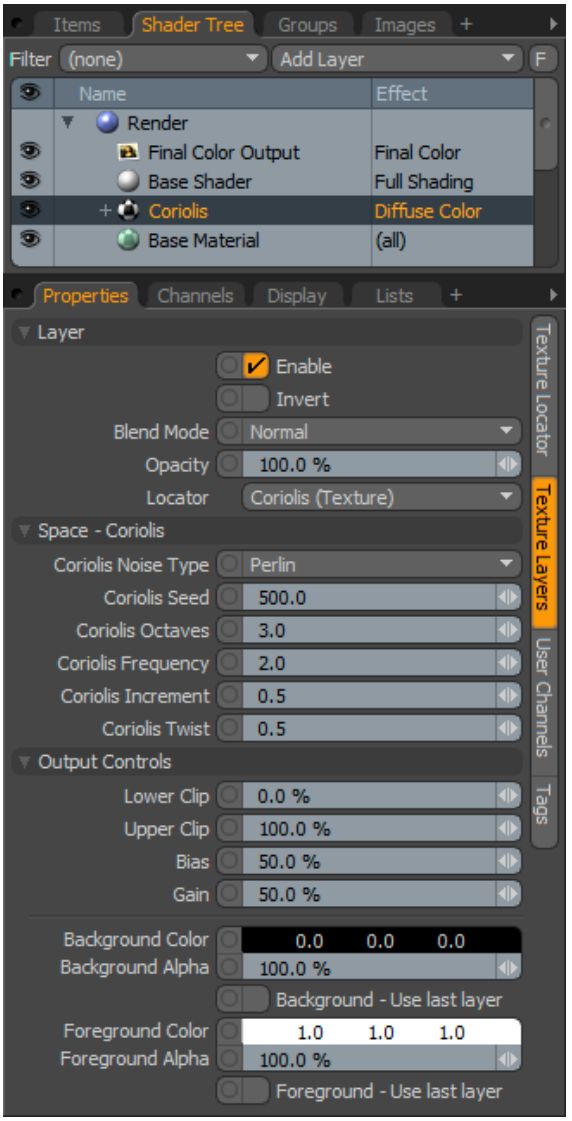
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Space - Coriolis --

Coriolis Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Coriolis Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Coriolis Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with

each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Coriolis Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Coriolis Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Coriolis Twist: This value controls the amount of Coriolis effect, or twist is applied to the surface.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

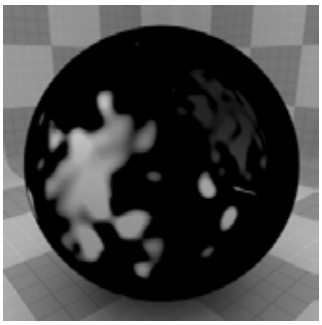
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Flare



The Flare texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Flare texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a lens *flare*-like star pattern. It can be used for highlights or glints, as well as the more traditional lens flare effects.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

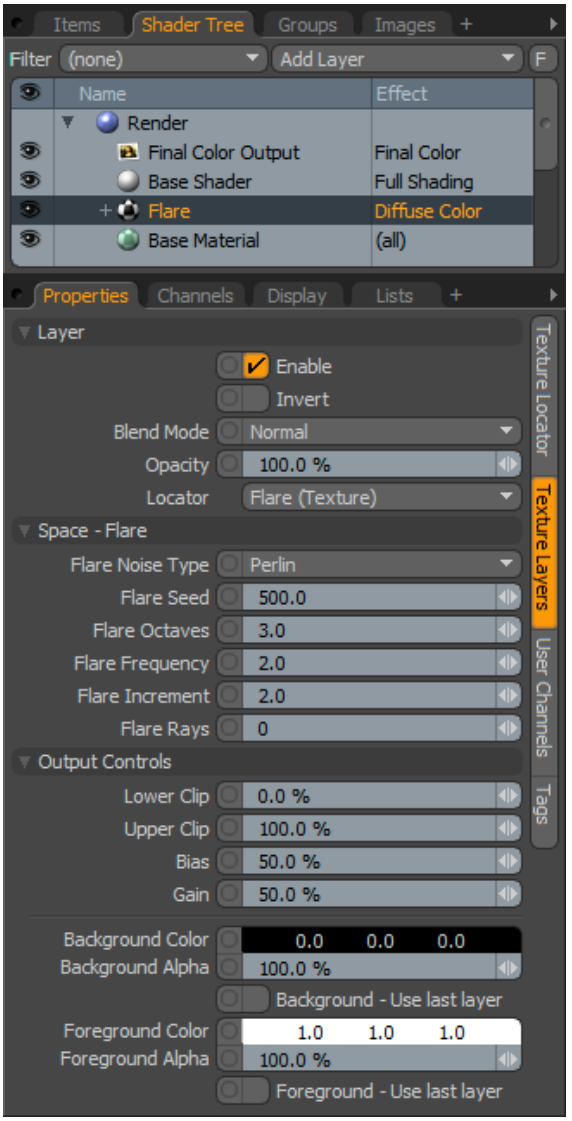
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Space - Flare --

Flare Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Flare Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Flare Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Flare Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Flare Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Flare Rays: This value controls the number of rays used for the flare pattern.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

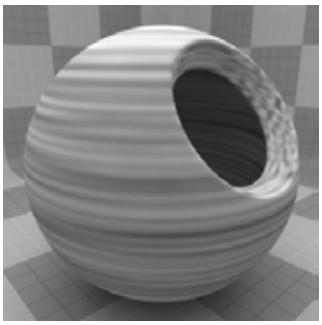
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Gas Giant



The Gas Giant texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Gas Giant texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader applies rings of color to an object, along with a noise function to disturb the banding. By selecting various base colors, you can create planets similar to the *gas giants*, Saturn, Jupiter and Uranus.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

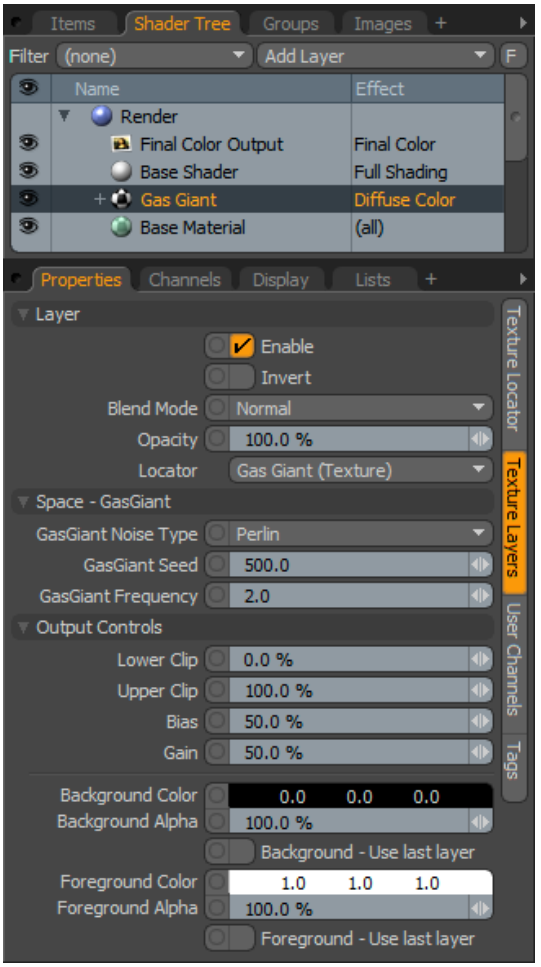
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the

layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Space - Gas Giant --

Gas Giant Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Gas Giant Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Gas Giant Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

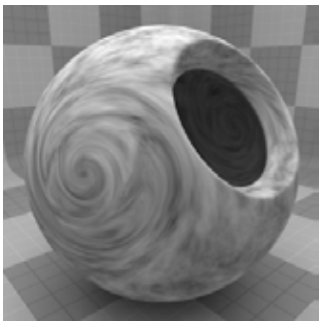
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Hurricane



The Hurricane texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Hurricane texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader applies a pattern similar to the swirling clouds present in a hurricane. The amount of twist or swirling can be controlled, along with the size of the eye of the *hurricane* and detail of the clouds.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

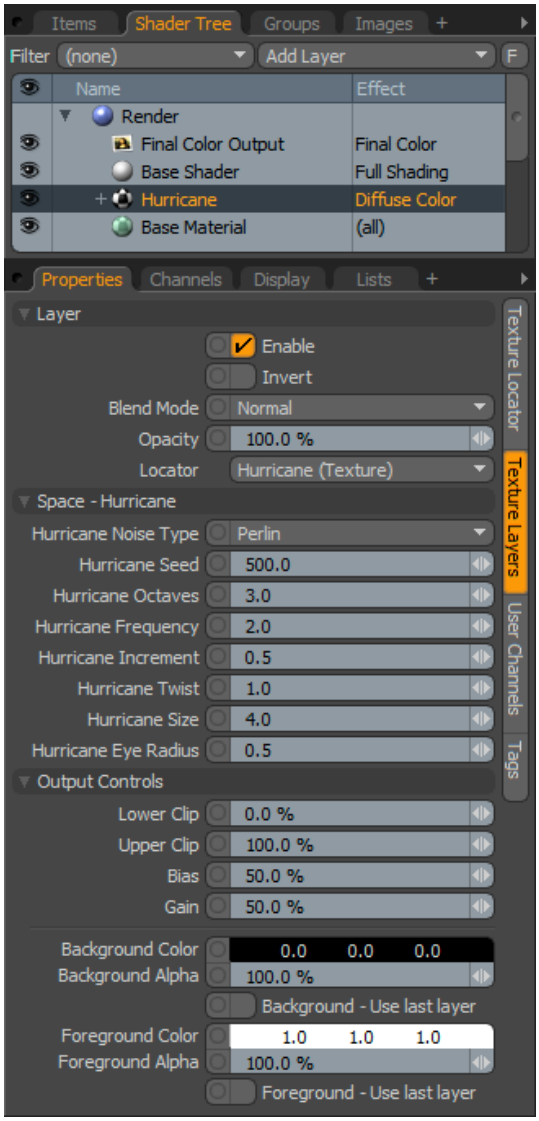
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Space - Hurricane --

Hurricane Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Hurricane Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Hurricane Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Hurricane Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Hurricane Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Hurricane Twist: This value controls the amount of twist applied to the surface.

Hurricane Size: This value controls the overall size of the hurricane.

Hurricane Eye Radius: This value controls the size of the eye of the hurricane.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

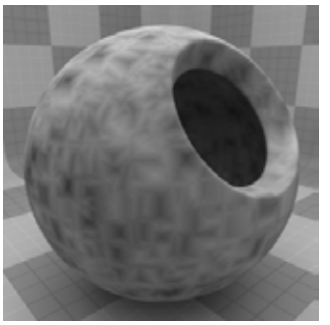
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Nurnies



The Nurnies texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Nurnies texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a random tile type of pattern, making it useful for adding detail to a surface, such as the hull of a space ship.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

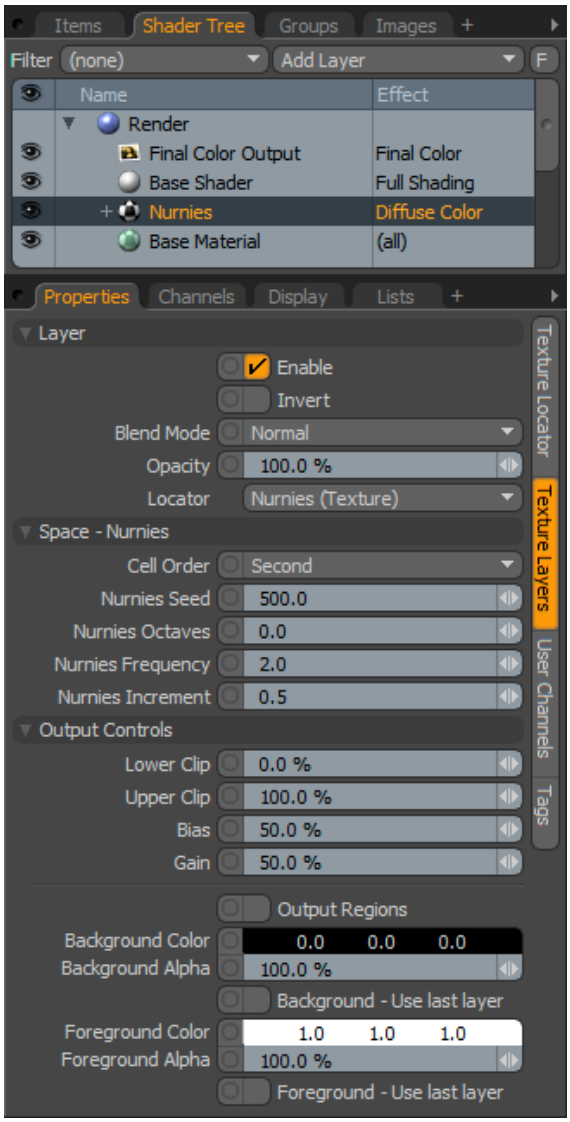
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Space - Nurnies --

Cell Order: This option determines the cell order used for texture creation-

First - Nearest Cell

Second - Next Nearest Cell

Third - Next Nearest Cell

Fourth - Furthest Cell

Nurnies Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Nurnies Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Nurnies Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Nurnies Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

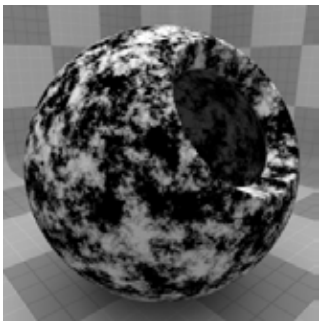
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Planet



The Planet texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Planet texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a pattern similar to the atmosphere of an Earth-like *planet*.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

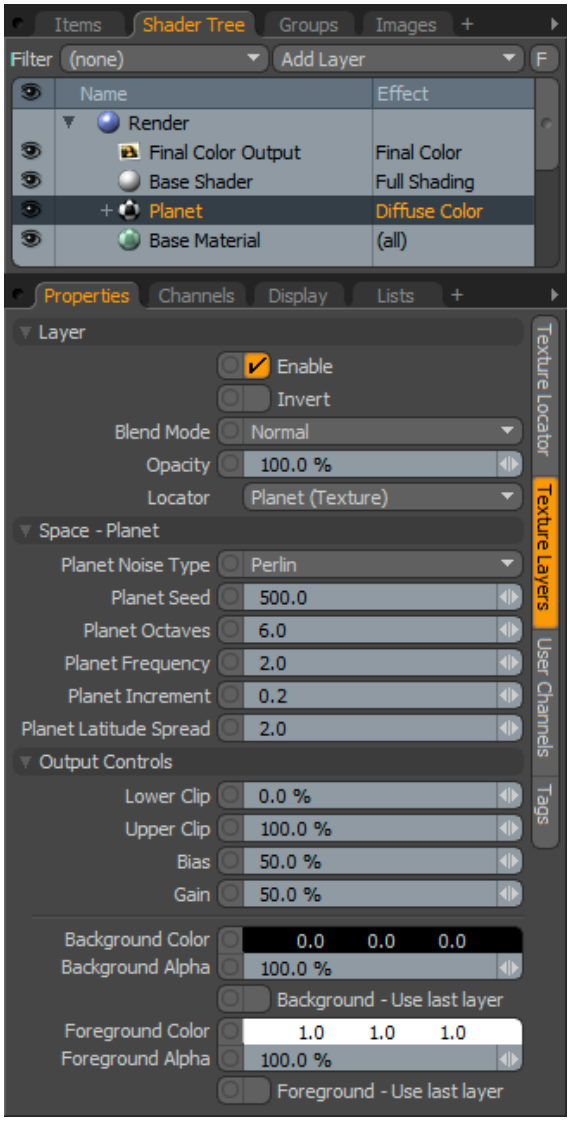
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Space - Planet --

Planet Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Planet Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Planet Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Planet Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Planet Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Planet Latitude Spread: This value controls the scaling between each of the polar regions.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

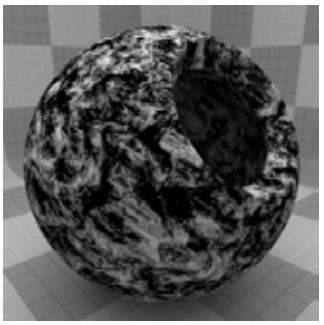
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Planet Clouds



The Planet Clouds texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Planet Clouds texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides a pattern similar to the atmosphere of an Earth-like *planet*. It differs from the *Planet* shader by provding more of a swirling cloud effect.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

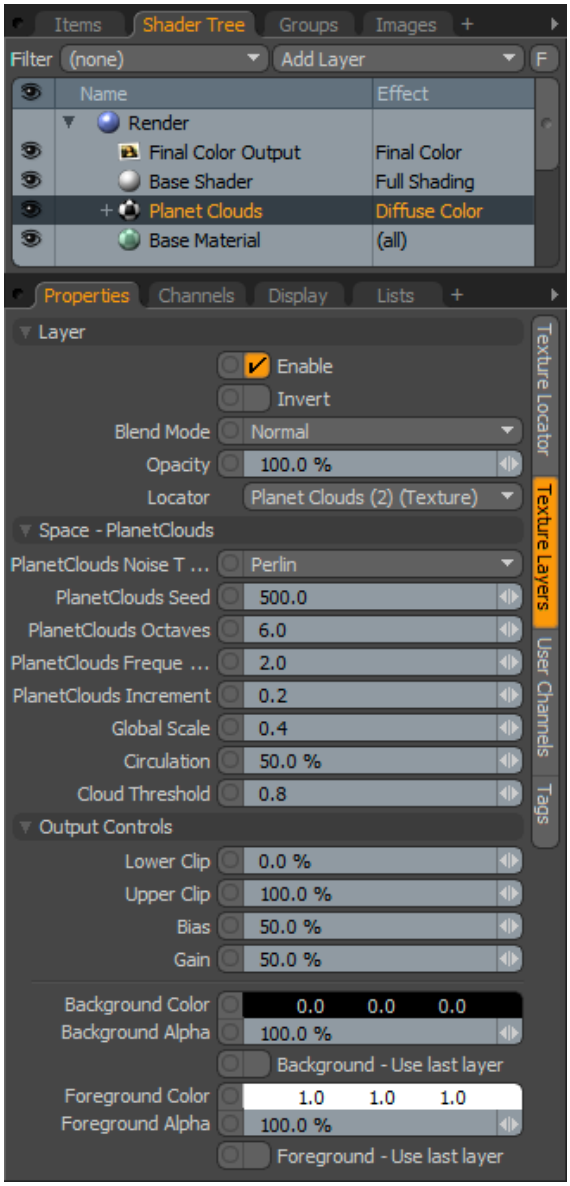
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Space - Planet Clouds --

Planet Clouds Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Planet Clouds Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Planet Clouds Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low

values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Planet Clouds Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Planet Clouds Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Global Scale: This value controls how much the global circulation disturbs the clouds. When the control is at a low value, the clouds are long and sweeping, becoming increasingly broken up and small as the parameter is increased.

Circulation: This value controls the global circulation of the pattern. Increasing this value causes the clouds have more swirl.

Cloud Threshold: This value determines where the texture shading transitions to the Background Color. The higher the value, the smoother the transition.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users

a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

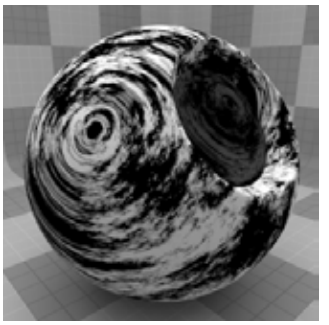
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Rings



The Rings texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Rings texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a noisy ring pattern, similar to the *rings* around Saturn. A minimum radius for the rings can be set, along with the ring scaling and noise amount.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

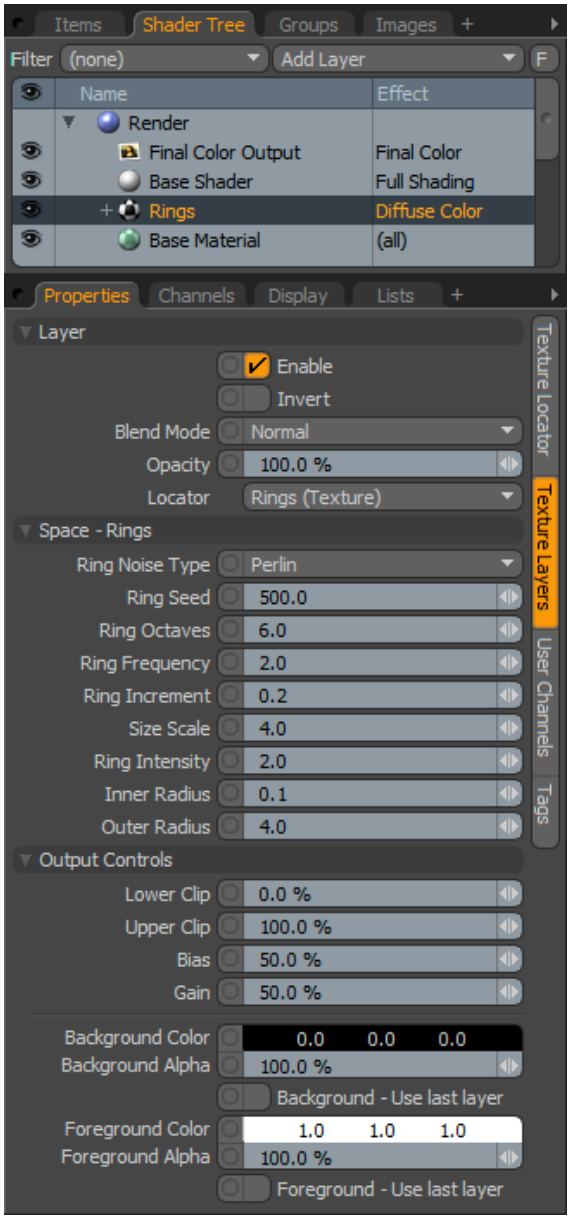
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Space - Rings --

Ring Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Ring Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Ring Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low

values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Ring Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Ring Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Size Scale: This value controls the scaling of actual rings.

Ring Intensity: This value controls the intensity of the rings as they transition to the Background Color. The higher the value, the smoother the transition.

Inner Radius: This value sets the minimum radius for the inner rings.

Outer Radius: This value sets the maximum radius for the outer rings.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

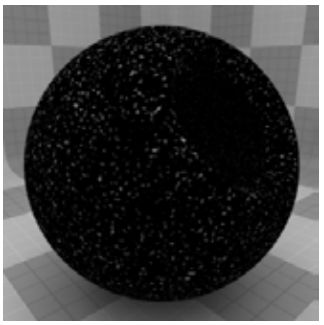
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Star Field



The Star Field texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Star Field texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a *star field*, as the kind seen in field on a starry night or at a planetarium.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

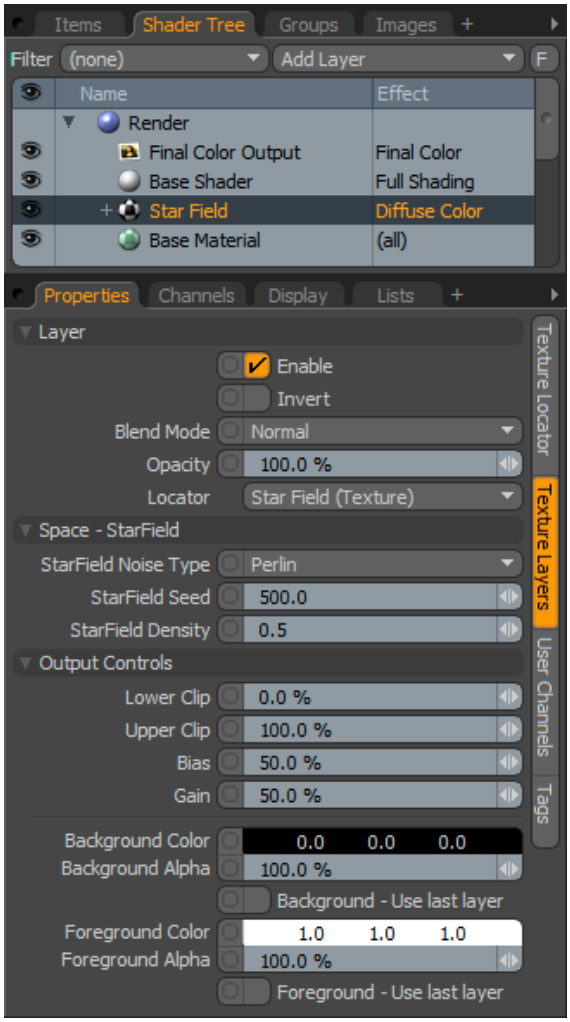
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Space - Star Field --

Star Field Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Star Field Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Star Field Density: This value controls the amount of stars present in the star field. Increasing this value creates a more star-filled sky.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value,

truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

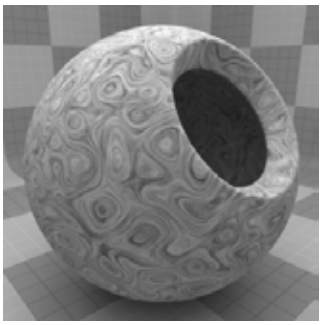
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Swirl



The Swirl texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Swirl texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader is similar to the FBM (Fractal Brownian Motion) noise pattern, except that it has an additional *'swirl'* parameter. This allows the texture to be highly stirred, ideal for creating smoke, fog or planets much as Jupiter.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

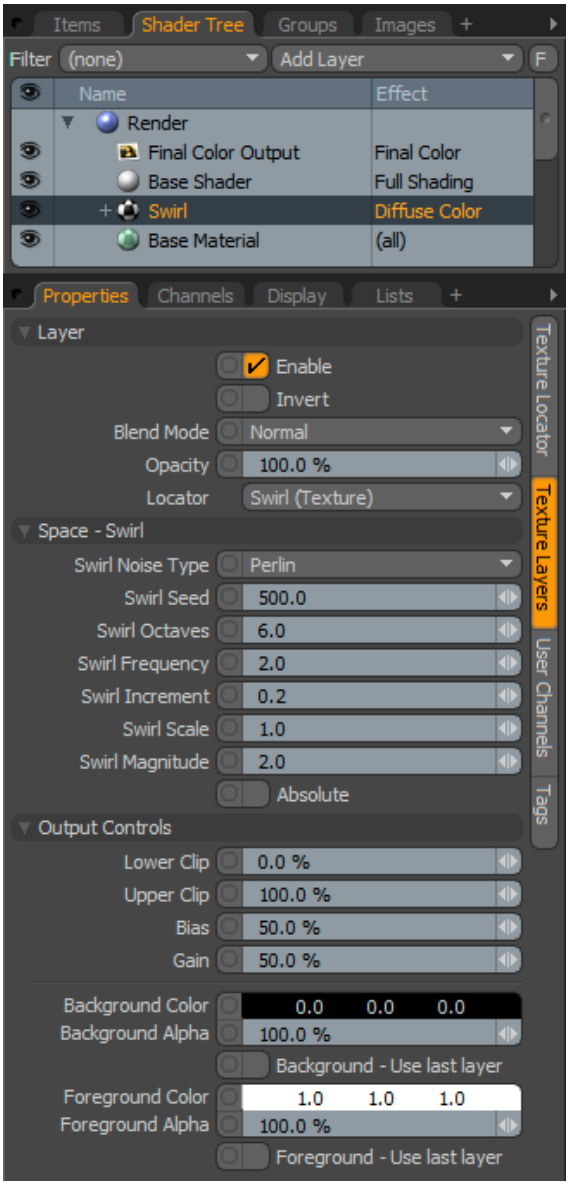
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Space - Swirl --

Swirl Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Swirl Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Swirl Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each

summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Swirl Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Swirl Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Swirl Scale: This value controls the scaling of swirls. When the control is at a low value, the swirls will be loose, becoming increasingly tighter as the parameter is increased.

Swirl Magnitude: This value controls how much the swirls affect the overall texture. The higher the value, the more prominent the swirls.

Absolute: This option determines whether or not the Absolute value for the noise is used, providing a stronger, more defined variation on the default texture.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users

a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

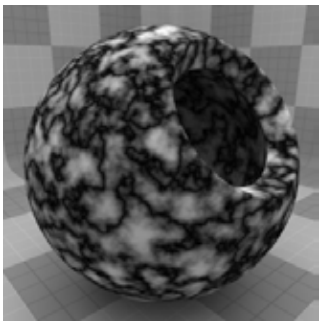
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Terra



The Terra texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Terra texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader provides two sets of noisy bumps with independent height controls, ideal for creatin the hills and valleys of a landscape surface.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

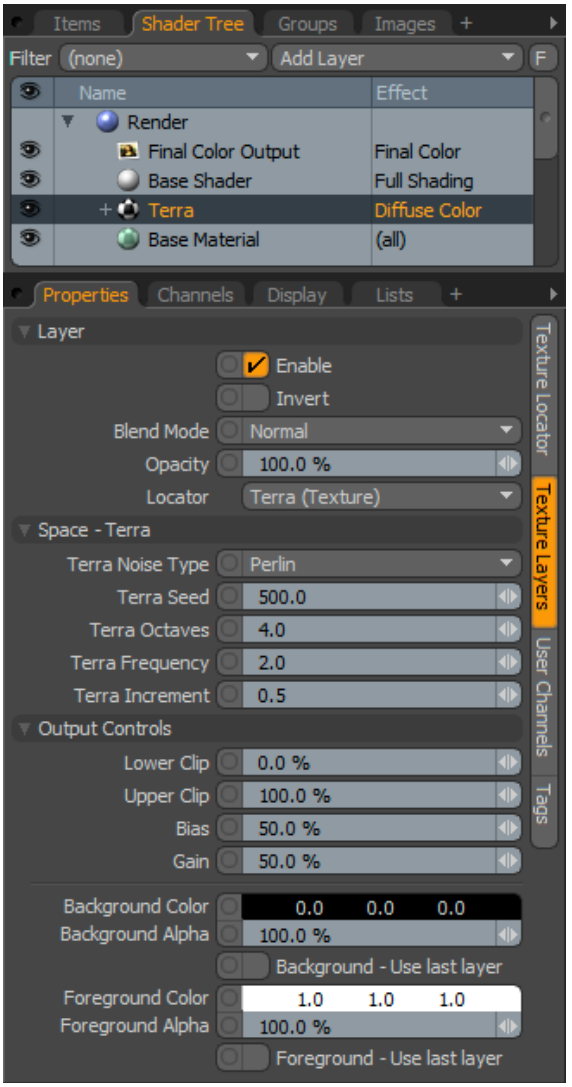
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Space - Terra --

Terra Noise Type: This option determines the look of the distortion by providing several Noise function types.

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Terra Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Terra Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Terra Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Terra Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

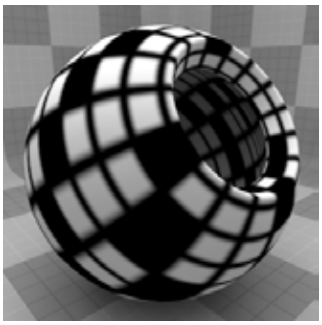
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Windows



The Windows texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Windows texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a series of rectangular *windows* which can be smoothed and turned on or off. There is also an option to illuminate windows to create banks of lit and unlit lights.

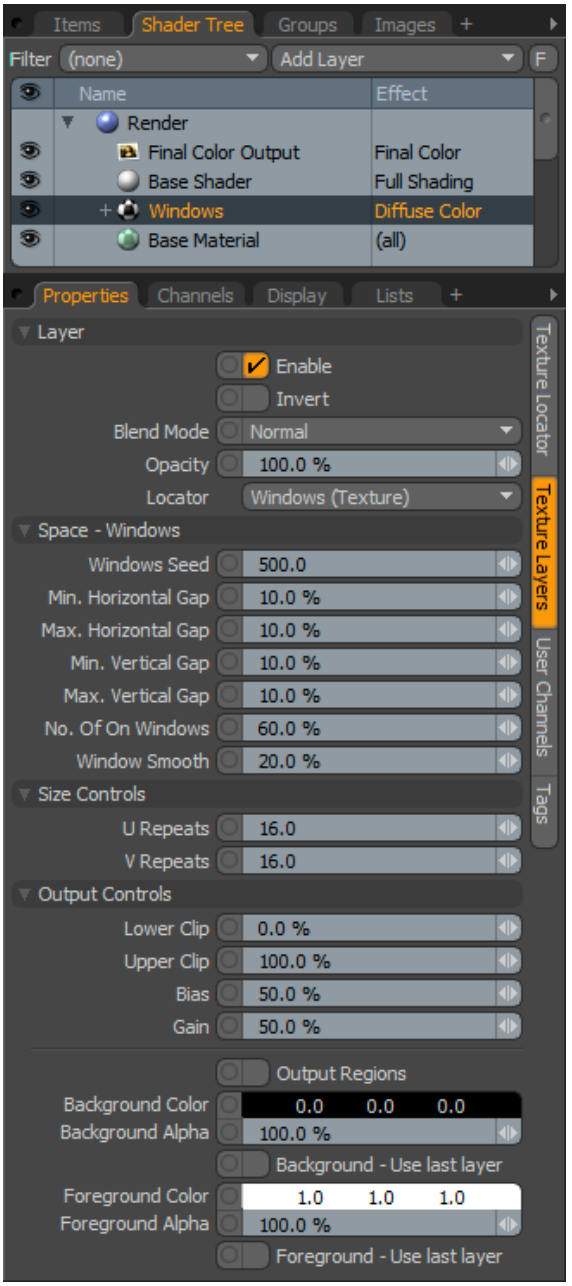
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

IMPORTANT: Due to the nature of this procedural, the default 'Solid' projection method produces no result. When applied, users will need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator press the small ' + ' icon preceding the layer name in the Shader Tree and LMB+click on the layer once revealed. In the Properties panel change the 'Projection Type' to a UV Map (*highly recommended*) or choose one of the geometric projection, such as 'Planar', 'Cylindrical', 'Spherical', etc. and size the texture accordingly.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.



Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Space - Windows --

Windows Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Minimum Horizontal Gap: This value sets the minimum distance between the windows in the horizontal direction.

Maximum Horizontal Gap: This value sets the maximum distance between the windows in the horizontal direction.

Minimum Vertical Gap: This value sets the minimum distance between the windows in the vertical direction.

Maximum Vertical Gap: This value sets the maximum distance between the windows in the vertical direction.

Number Of On Windows: This value controls the number of windows that are 'on' and therefore visible.

Window Smooth: This value controls the edge smoothness of the windows. When the control is at a low value, the transition between the windows and background color will be very strong, becoming smoother as the parameter is increased.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

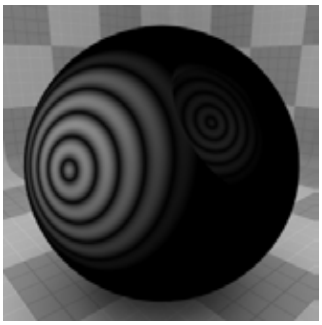
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Drip Drop



The Drip Drop texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Drip Drop texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a pattern of concentric circles, similar to the ripple effect when a pebble is dropped into water. All of the parameters are keyed to the time function, allowing for easy animation.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

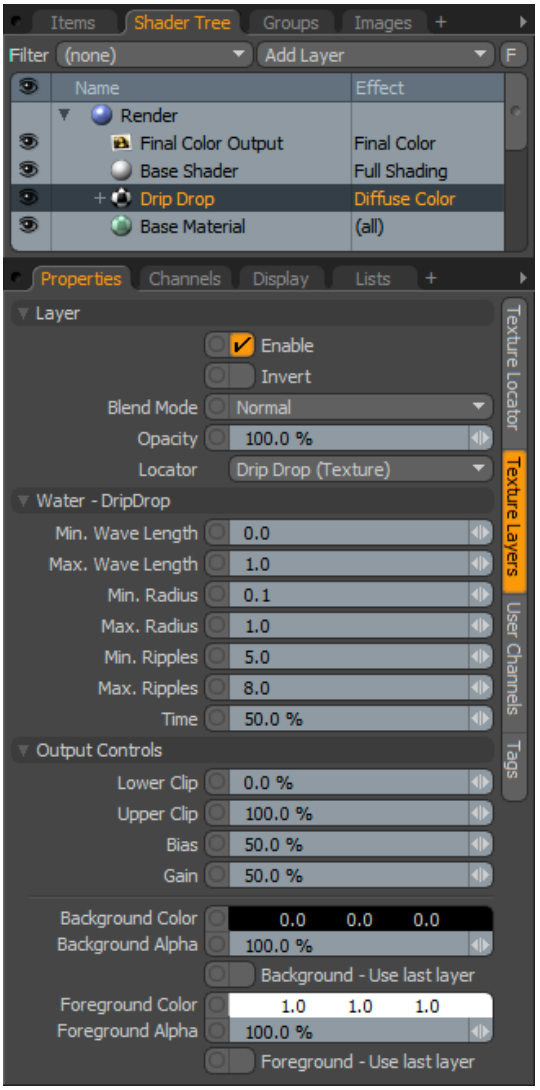
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Water - Drip Drop --

Minimum Wave Length: This value determines the wave length of the ripples at the start of the animation (when the time parameter is set to 0%).

Maximum Wave Length: This value determines the end wave length of the ripples at the end of the animation (when the time parameter is set to 100%).

Minimum Radius: This value determines the radius of the ripples at the start of the animation (when the time parameter is set to 0%).

Maximum Radius: This value determines the radius of the ripples at the end of the animation (when the time parameter is set to 100%).

Minimum Ripples: This value determines the number or ripples at the start of the animation (when the time parameter is set to 0%).

Maximum Ripples: This value determines the number or ripples at the end o the animation (when the time parameter is set to 100%).

Time: This value controls the animation progress of the other settings through time. The animation morphs from the beginning of the water drop (0%), when the concentric circles start, to the end (100%), when the circles have been exhausted.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

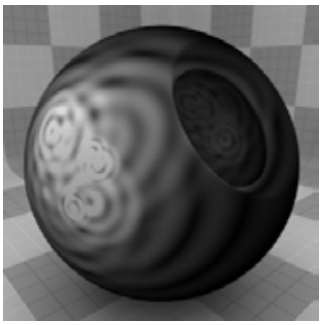
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Rain



The Rain texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Rain texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a series random concentric circles, similar to when *rain* drops fall into a body of water. All of the parameters are keyed to the time function, allowing for easy animation.

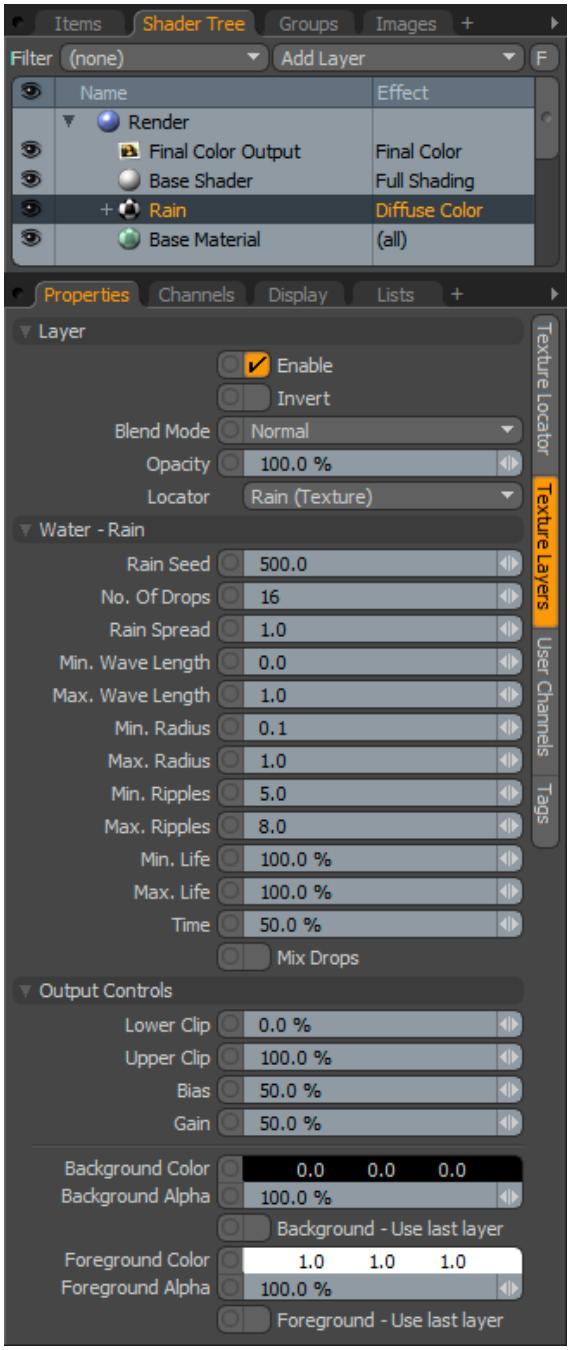
For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.



Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Water - Rain --

Rain Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Number Of Drops: This value determines the number of rain drops in the pattern.

Rain Spread: This value determines the spread of the rain drops, with the drops moving farther from one another as the parameter is increased.

Minimum Wave Length: This value determines the wave length of the ripples at the start of the animation (when the time parameter is set to 0%).

Maximum Wave Length: This value determines the end wave length of the ripples at the end of the animation (when the time parameter is set to 100%).

Minimum Radius: This value determines the radius of the ripples at the start of the animation (when the time parameter is set to 0%).

Maximum Radius: This value determines the radius of the ripples at the end of the animation (when the time parameter is set to 100%).

Minimum Ripples: This value determines the number or ripples at the start of the animation (when the time parameter is set to 0%).

Maximum Ripples: This value determines the number or ripples at the end o the animation (when the time parameter is set to 100%).

Minimum Life: This value determines the minimum life expectancy of the rain drops.

Maximum Radius: This value determines the maximum life expectancy of the rain drops.

Time: This value controls the animation progress of the other settings through time. The animation morphs through multiple batches of rain drops as the timer goes from 0 to 100%.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very

sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

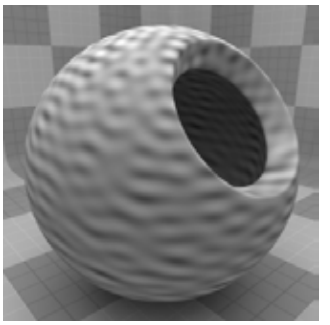
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Ripples



The Ripples texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Ripples texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a pattern based on ever-expanding *ripples*, similar to the water surface disturbed by wind.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

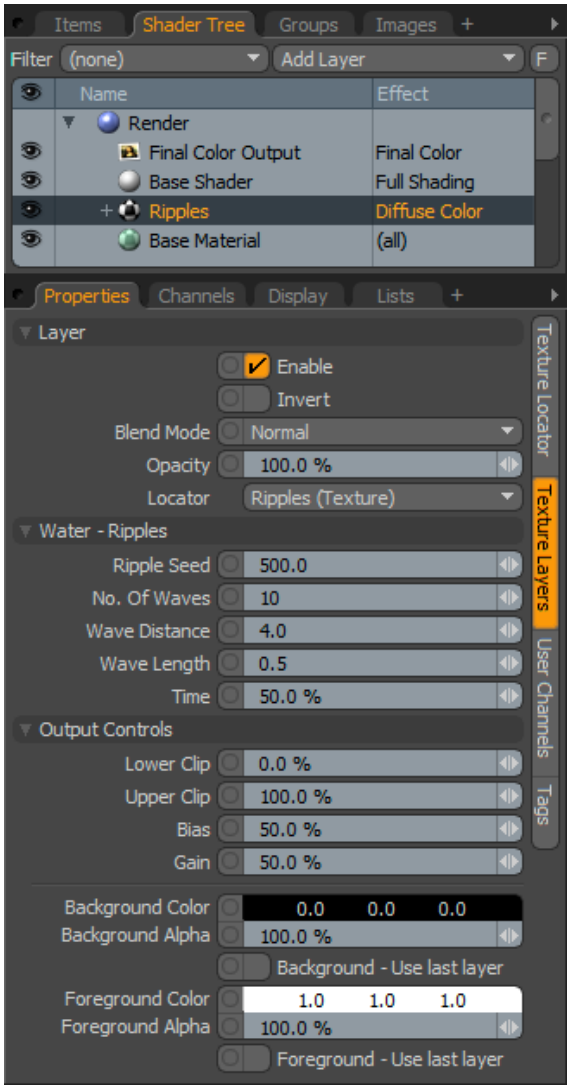
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Water - Ripples --

Ripple Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Number Of Waves: This value determines the number of waves in the pattern.

Wave Distance: This value determines the spread between the ripples.

Wave Length: This value determines the wave length of the ripples throughout the animation.

Time: This value controls the animation progress of the other settings through time. The animation morphs through multiple waves as the timer goes from 0 to 100%.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value,

users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

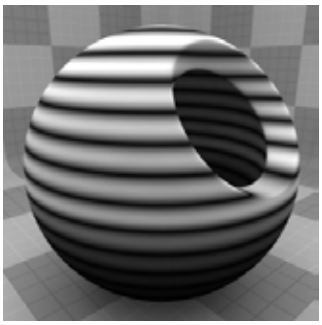
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Surf



The Surf texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Surf texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a standard wave pattern emanating from the texture's center. The amplitude of the wave decreases as the wave moves away from the wave source.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

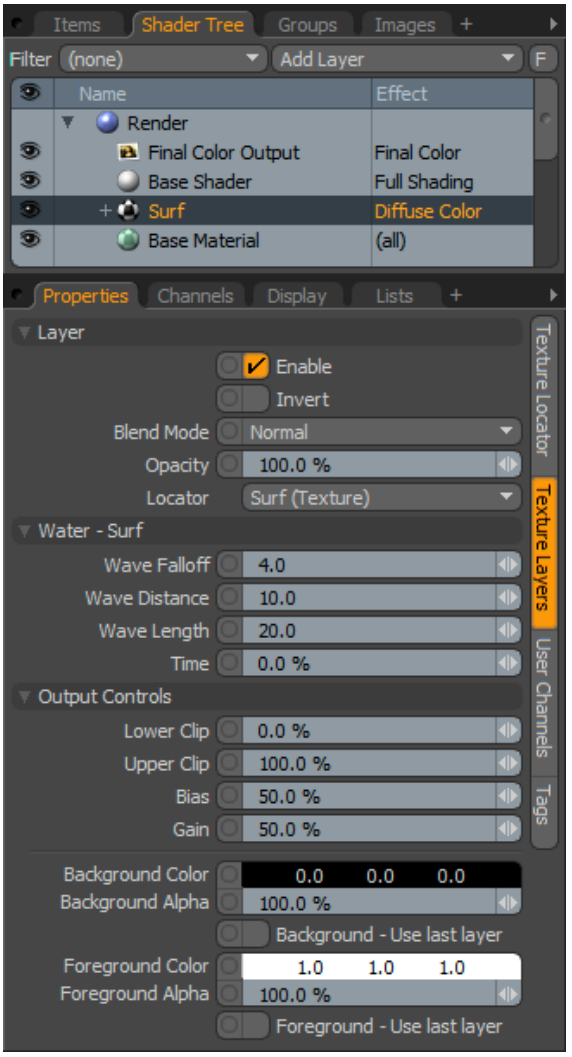
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Water - Surf --

Wave Falloff: The value controls the falloff, or fade, of the waves as they emanate from the center. When the control is at a low value, the foreground and background transition will be similar across the texture. At the value is increased the transition between colors becomes smoother at the opposite end off the wave source.

Wave Distance: This value controls the number of waves drawn on the surface.

Wave Length: This value determines the distance moved by the waves.

Time: This value controls the animation progress of the other settings through time. The animation continually moves the waves downward drops as the timer goes from 0 to 100%.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value,

truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

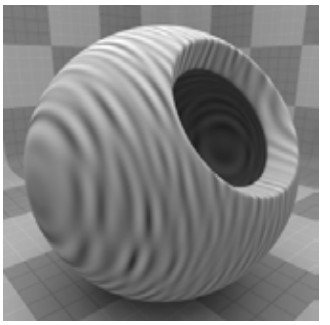
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Waves



The Waves texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Waves texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a loose, random *wave* pattern emanating from the texture's center. The amplitude of the wave decreases as the wave moves away from the wave source.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

Layer--

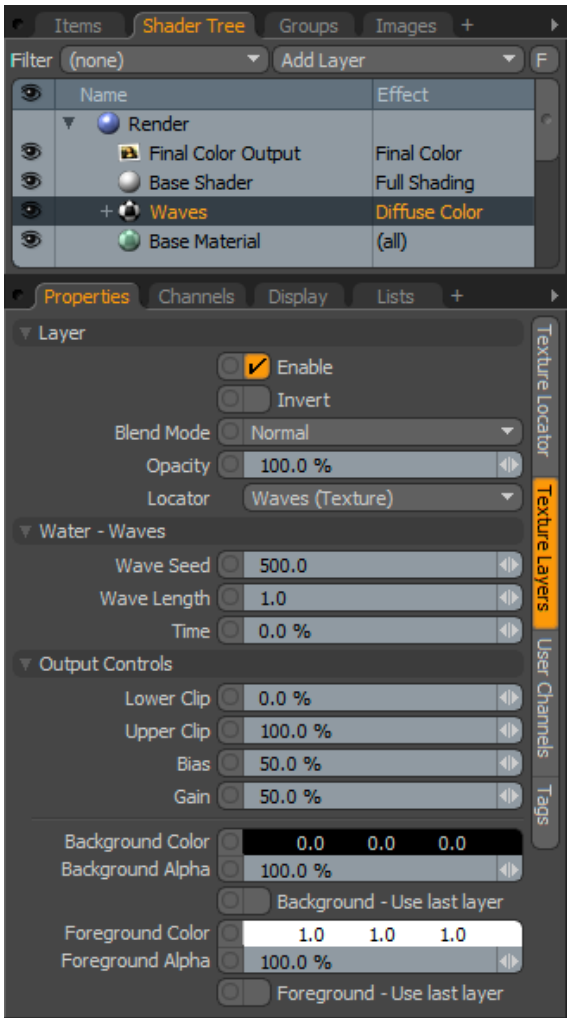
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Water - Waves --

Wave Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Wave Length: This value determines the distance moved by the waves.

Time: This value controls the animation progress of the other settings through time. The animation continually moves the waves downward drops as the timer goes from 0 to 100%.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value,

users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

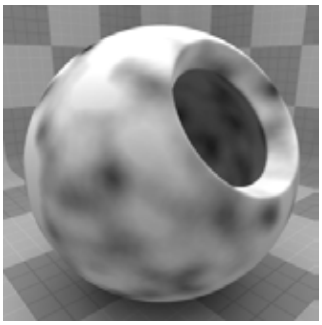
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Windy Waves



The Windy Waves texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The Windy Waves texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. The texture will modulate from one zone to the other based on the users settings; each zone can have either a 'Value', or a 'Color' and 'Alpha'. The applied Zone is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a 'Displacement', the 'Value' settings would be utilized, whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Background and Foreground. This shader creates a very loose, random wave pattern, similar to the deep ocean.

For information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation.

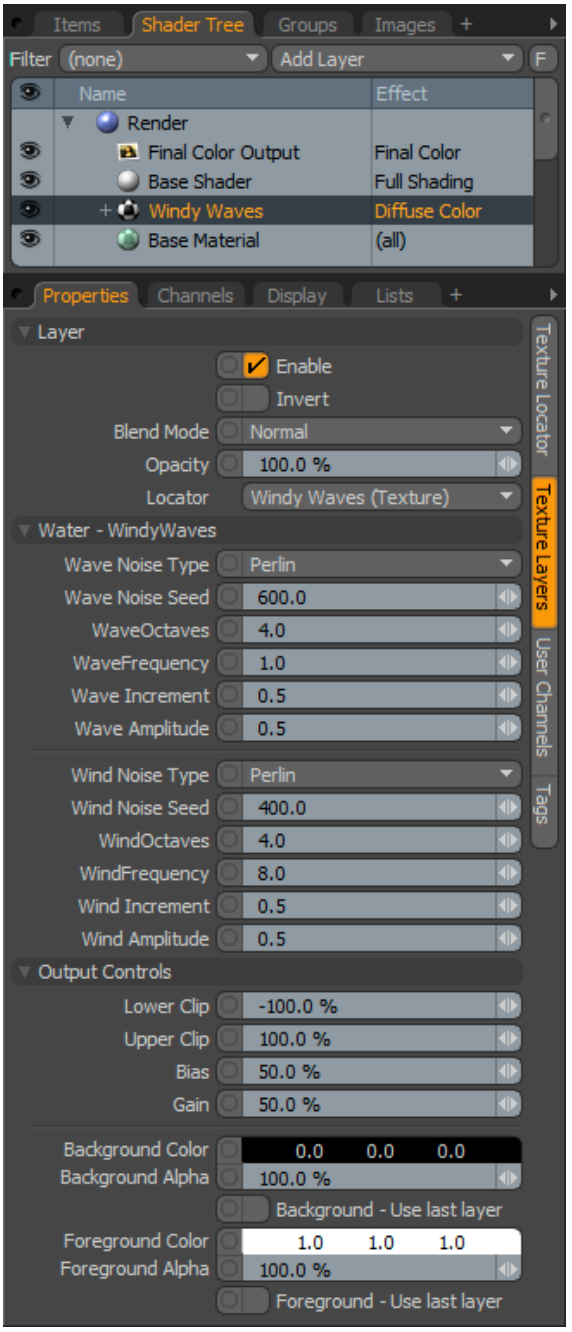
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Water - Windy Waves --

Wave Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Wave Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Wave Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each

summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Wave Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Wave Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Wave Amplitude: This value determines the wave height.

Wind Noise Type: This option determines the look of the texture distortion, with several Noise function types provided:

Perlin-

Enhanced Perlin-

Gradient-

Value-

Gradient Value-

Impulse-

Lattice-

Bubble-

Wind Noise Seed: The 'Seed' value is the initial number used when generating the procedural values. Different 'Seed' values will produce different random variations and can be useful in changing the texture result; however, users will want to use the same 'Seed' value when it is necessary for items to retain the same variations.

Wind Octaves: This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence will be smoother, increasing in detail as the number of octaves is increased.

Wind Frequency: This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, "stirring up" the pattern.

Wind Increment: This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence will be tight and crinkled, becoming increasingly smoother as the parameter is increased.

Wind Amplitude: This value determines the amount of wind.

Output Controls--

Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.

Bias: Increasing this value will cause the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect, whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions: When the 'Output Regions' options is enabled, the procedural texture will output random grey shades per region rather than outlines for tiles providing users a means to add random variety to the procedurally created texture. Users can further control the amount of variation using the 'Regional HSV' process layer.

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

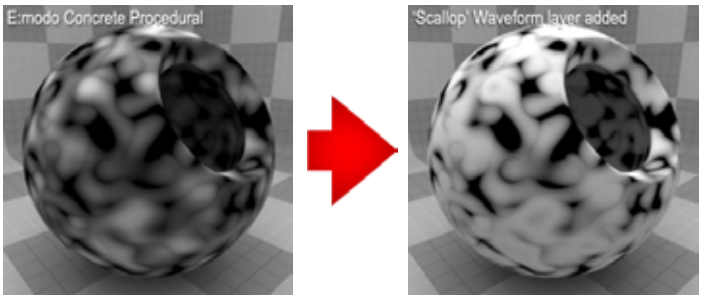
Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

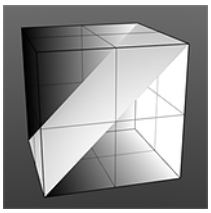
Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

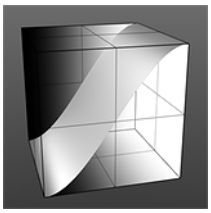
Waveforms

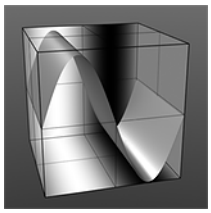


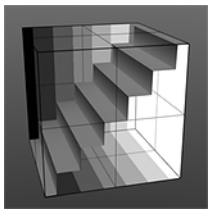
The Enhance:modo 'Waveform' layers are a special kind of Shader tree layer that produce no results by themselves, as they are meant to modify any like texture layer *below* them in the Shader Tree hierarchy. What this means is any texture layer, for example a 'Diffuse' layer would only be affected by a Waveform layer with its effect set to 'Diffuse' above it; in order to affect a 'Displacement' layer, the Waveforms layer effect will need to be changed to 'Displacement' as well. If there are multiple layers blended together of the same type, the waveform will only affect the nearest visible layer below it. Each Waveform type produces its own independent result.

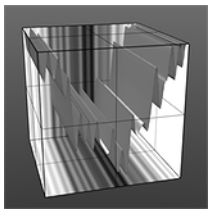
If you are familiar with the 'Curves' functionality of Photoshop, then it will be much easier to understand the 'Waveform' function, as it is a means to easily manipulate the values of the texture layer, just like curves affect the values in an image. If you were to make a graph of the input values of a surface versus the output values, the direct correlation between the two would be a straight line. As adjustments are made to the graph, the values mapped on the graph are adjusted. These adjusted values can adjust the shades of gray across a procedural texture or the shape of a displacement map. Consider this visual example below of a displaced plane on the far left. A gradient ramp has been applied to a flat plane and offset the same height as its width producing a ramp at a 45 degree angle. When a Waveform layer is applied to the gradient, it clearly demonstrates how the Waveform layer affect the values across the gradient ramp, illustrating the shaping ability that the Waveform function applies.


Displaced Plane

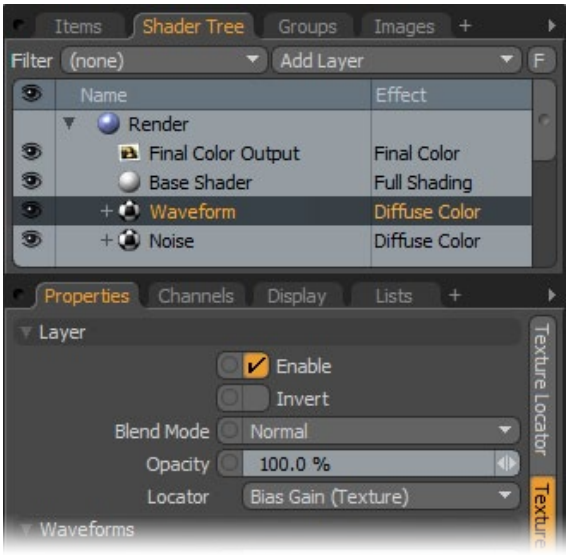

S-Curve Waveform


Sine Waveform


Stairs Waveform


Noise Waveform

To work with a Waveform layer, simply select the Material Group item in the Shader Tree that you wish place the texture into, and use the 'Add Layer' option, choosing one of the available Waveform options from the popup menu - "Add Layer > Enhance:modo Textures > Waveforms". For additional information regarding adding and working with Shader Tree Items Layers please reference the the Shader Tree page of the documentation. Each Waveform option produces its own unique results (with its name suggesting the shaping result of the layer). Users can further modify the attributes of the layer in the associated Properties panel adjusting from the various settings covered below.



Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When unchecked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this value will increasingly reveal lower layers in the Shader tree if present, or dim the effect of the layer itself on the surface.

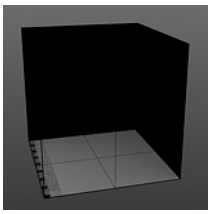
Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (the way it is applied) to the surface. The 'Locator' option sets that association. While users can choose alternate locators, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator. Since the Waveform layers affect the lower layers gray shades, the settings of the Locator, in this case, are unnecessary.

Waveforms--

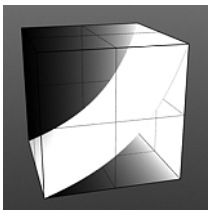
The following section covers options specific to each type of Waveform.

Waveforms - Bias Gain --

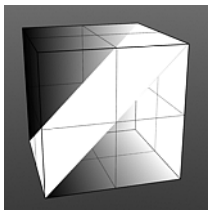
Bias: The Bias setting affects the attenuation (gradations) between the primary and secondary colors in a colored texture, or the gray values in a gray scale texture. Increasing this value will cause the texture to favor the primary color or value over the secondary whereas decreasing the value causes the secondary color or value to be favored.



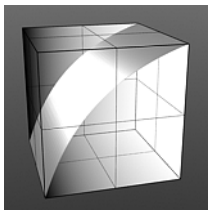
Bias 0%



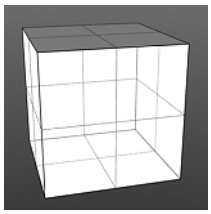
Bias 25%



Bias 50%

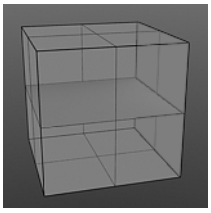


Bias 75%

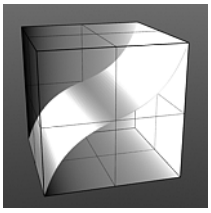


Bias 100%

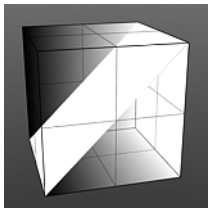
Gain: The Gain setting is similar to a contrast control that effects the falloff of the gradient ramp between the primary Color/Value and secondary Color/Value. Setting the Gain to 100% will create a very sharp contrast whereas setting the value to 0% would reduce the contrast between the two to almost imperceptible amount.



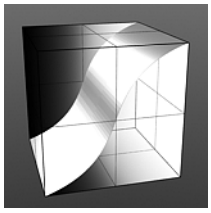
Gain 0%



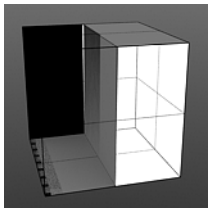
Gain 25%



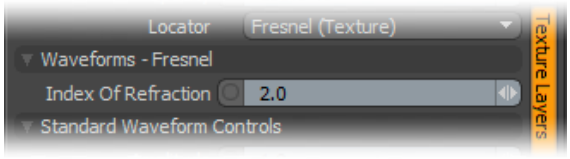
Gain 50%



Gain 75%



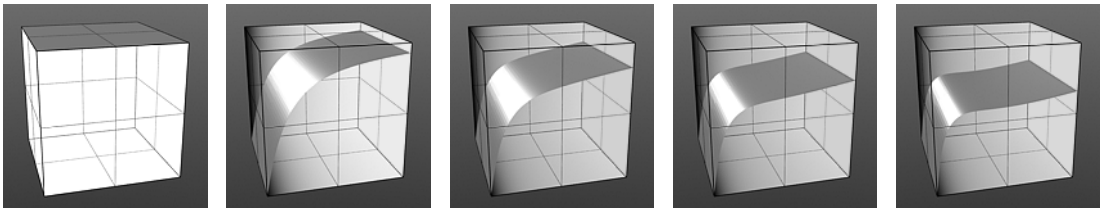
Gain 100%



Waveforms - Fresnel --

Index of Refraction: The Fresnel (pronounced Fruh-nell) effect is a phenomenon that describe particular refractive and reflective properties of a surface when the viewers angle relative to the surface changes. That particular rate at which the change happens is expressed in the Fresnel function and

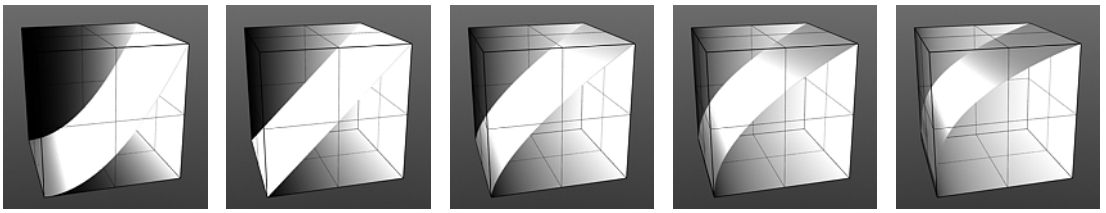
when applied to a gradient, such as one applied to the incidence angle of a surface, will produce the appropriate falloff rate. The Index of Refraction option will attenuate the Fresnel falloff rate based on the users input.



IOR 1.0 IOR 2.0 IOR 3.0 IOR 4.0 IOR 5.0

Waveforms - Gamma --

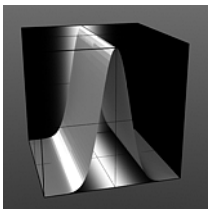
Gamma Amount: Gamma is a nonlinear adjustment function for manipulating the values across a spectrum. Like the Fresnel 'Index of Refraction' option above, the particular curve is expressed as a mathematical function. When applied to a texture layer, adjustments to the mid-tone value (those between black and white favoring the darker side) will match any gamma adjustment values applied to image map layers directly. A value of 1 will produce no results, as the output will match that of the input. Values lower than 1 will darken the gradient, while values larger than 1 will lighten it.



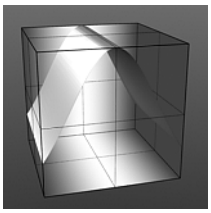
Gamma 0.5 Gamma 1.0 Gamma 1.5 Gamma 2.0 Gamma 3.0

Waveforms - Gaussian --

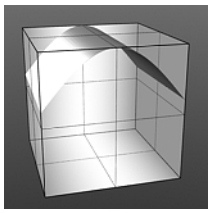
Gaussian Spread: A Gaussian distribution is a specific mathematical function that determines the spread of a number of values across a spectrum. When applied to a graduated texture layer, it will adjust the values based on this formula.



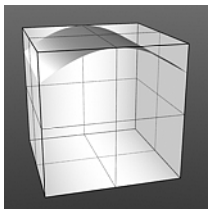
Spread 0.1



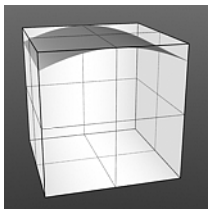
Spread 0.3



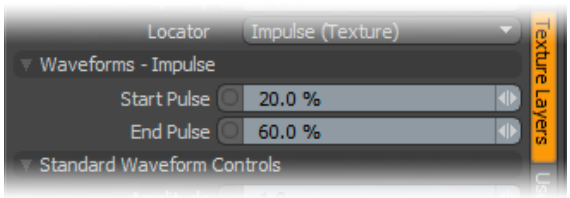
Spread 0.5



Spread 0.75

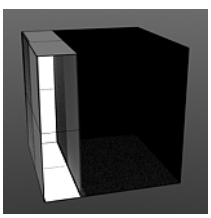


Spread 1.0

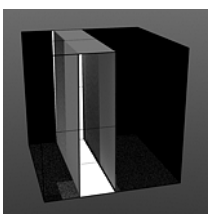


Waveforms - Impulse --

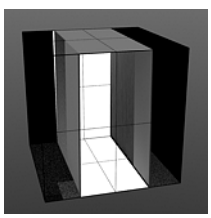
Start/End Pulse: The Impulse function will clamp the values of the gradient at the user defined 'Start' and 'End' positions and maximize all values in between.



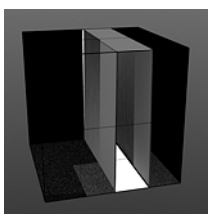
Start 0/End 25



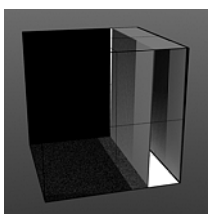
Start 25/End 50



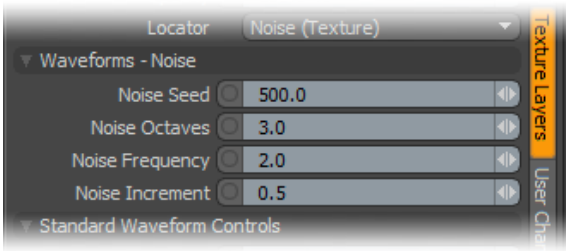
Start 25/End 75



Start 50/End 75



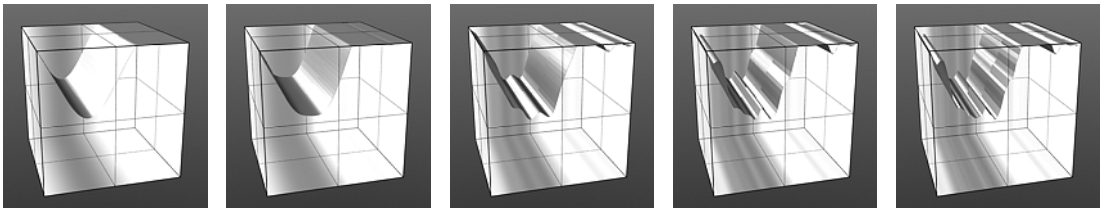
Start 75/End 100



Waveforms - Noise --

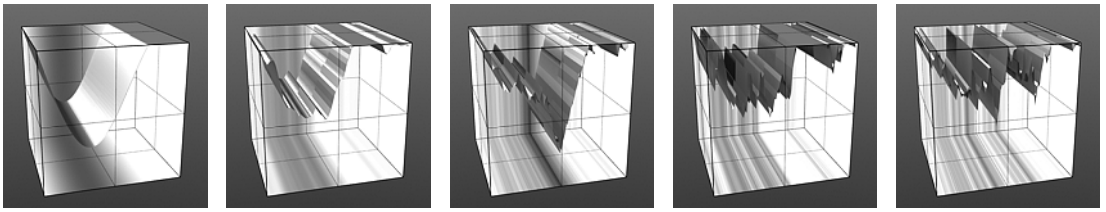
Noise Seed: The 'Seed' value is an initial value used when determining the noise that helps to randomize the generated. Different Seed values will produce dramatically or subtly different results.

Noise Octaves: The 'Noise Octaves' option determines the number of *layers* of noise that are combined to produce the final result. Greater numbers of Octaves will produce more randomized (detailed) results.



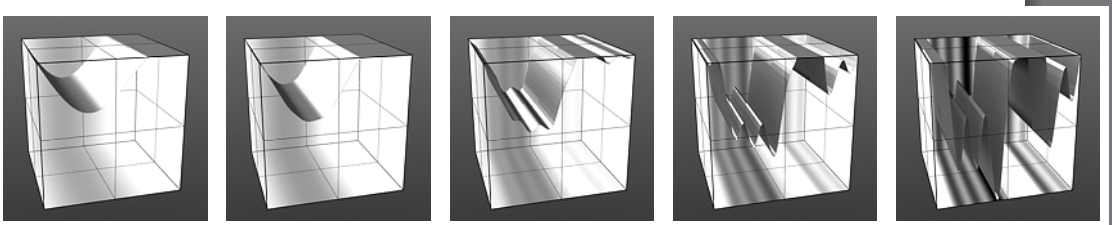
Octaves 1 | Octaves 2 | Octaves 3 | Octaves 4 | Octaves 5

Noise Frequency: The Frequency option determines the number of occurrences of the noise function when applied to the surface.

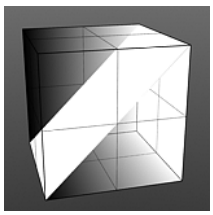


Frequency 1 | Frequency 2 | Frequency 3 | Frequency 4 | Frequency 5

Noise Increment: The Increment option, similar to frequency determines the number of noise bands across each frequency occurrence.

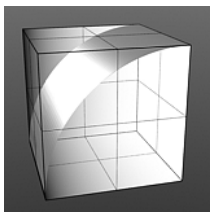


Increment 0.0 Increment 0.25 Increment 0.5 Increment 0.75 Increment 1.0



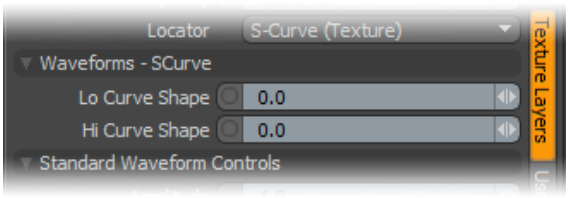
Waveforms - Ramp --

(Ramp has no user controlled settings)



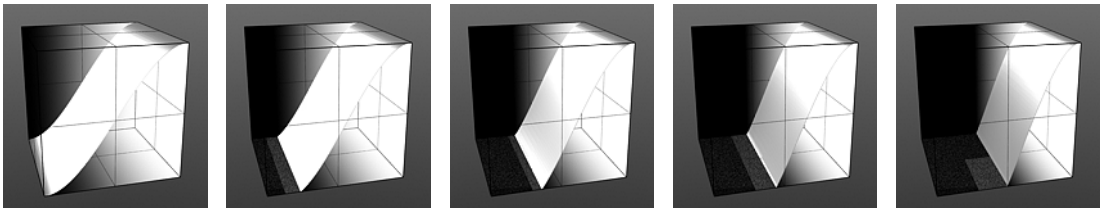
Waveforms - Rounded --

(Rounded has no user controlled settings)



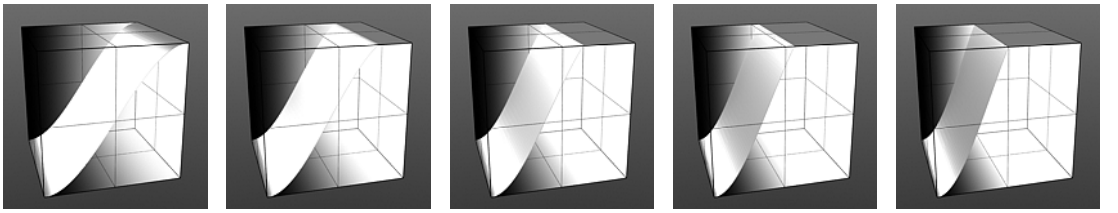
Waveforms - S-Curve --

Lo Curve Shape: The S-Curve waveform function produces a smoothed curve across the graduated spectrum. The 'Lo Curve Shape' setting determines the low cutoff area for the gradient.

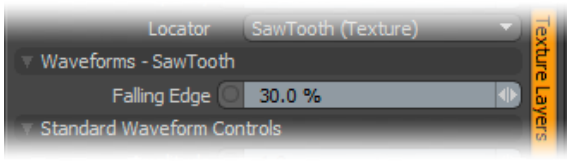


Low Curve 0 | Low Curve 1 | Low Curve 2 | Low Curve 3 | Low Curve 4

Hi Curve Shape: The S-Curve waveform function produces a smoothed curve across the graduated spectrum. The 'Hi Curve Shape' setting determines the high cutoff area for the gradient.

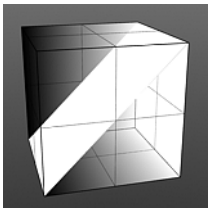


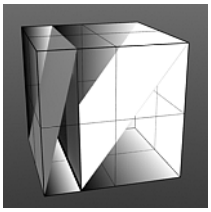
Hi Curve 0 | Hi Curve 1 | Hi Curve 2 | Hi Curve 3 | Hi Curve 4



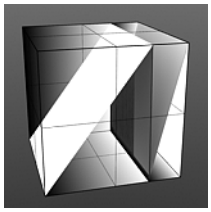
Waveforms - Sawtooth --

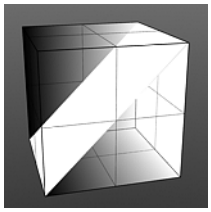
Falling Edge: The Sawtooth function produces a sudden minimum value drop in the graduated spectrum, with the position of the drop determined by the 'Falling Edge' option.

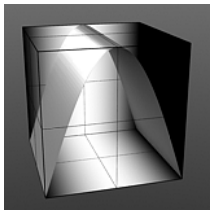

Falling Edge 0


Falling Edge 25

Falling Edge 50

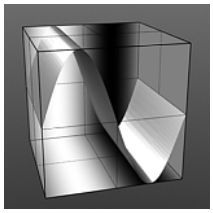

Falling Edge 75


Falling Edge 100



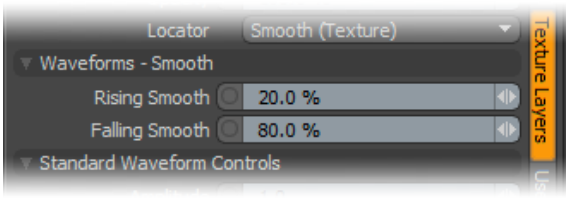
Waveforms - Scallop --

(Scallop has no user controlled settings)



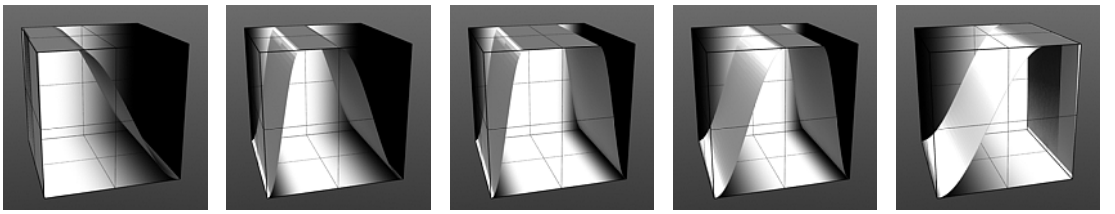
Waveforms - Sine --

(Sine has no user controlled settings)

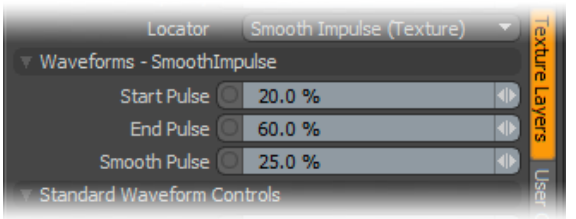


Waveforms - Smooth --

Rising/Falling Smooth: The Smooth function produces a result similar to what the Impulse function provides, by maximizing values between a low and high end, but between the minimum value to the Rise and the Fall to the maximum produces a smooth s shaped curve.

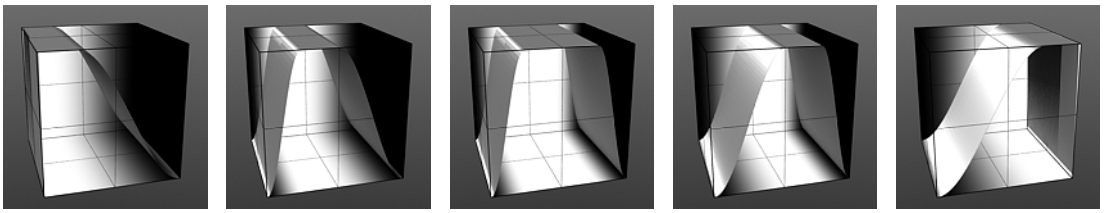


Rise 0/Fall 25 | Rise 25/Fall 50 | Rise 25/Fall 75 | Rise 50/Fall 75 | Rise 75/Fall 100



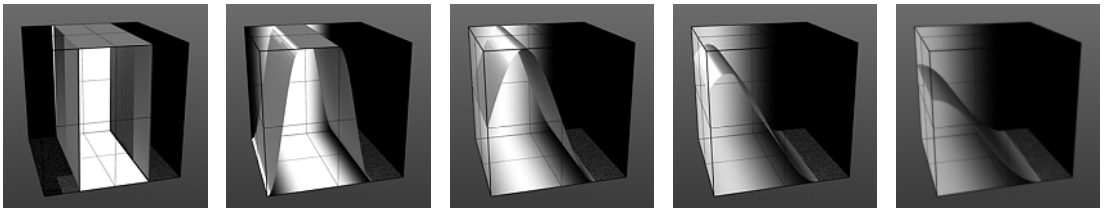
Waveforms - Smooth Impulse--

Start/End Pulse: The Smooth Impulse function is a cross between the Impulse function and the smooth function but allows the user to control the amount of smoothing of the result using the 'Smooth Pulse' option.

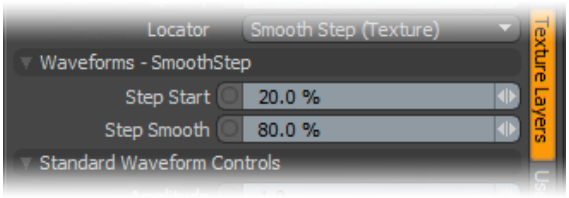


Start 0/End 25 | Start 25/End 50 | Start 25/End 75 | Start 50/End 75 | Start 75/End 100

Smooth Pulse: Determines the amount of smoothing applied to the Impulse function.

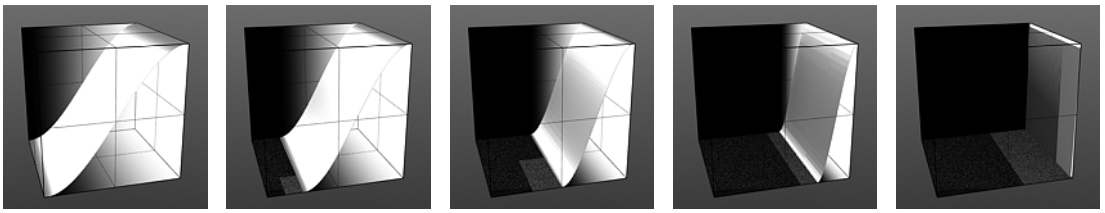


Smooth 0% Smooth 25% Smooth 50% Smooth 75% Smooth 100%



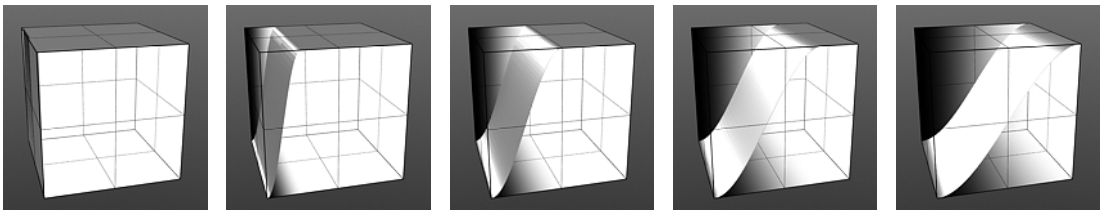
Waveforms - Smooth Step--

Step Start: Determines the start position for the smoothed step ramp.

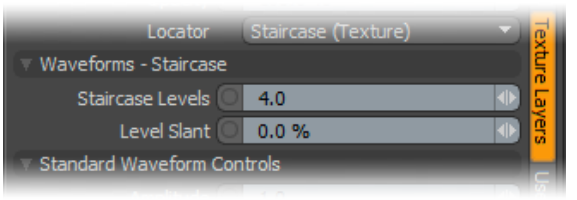


Start 0% Start 25% Start 50% Start 75% Start 100%

Step Smooth: Determines the amount of smoothing applied to the step position.

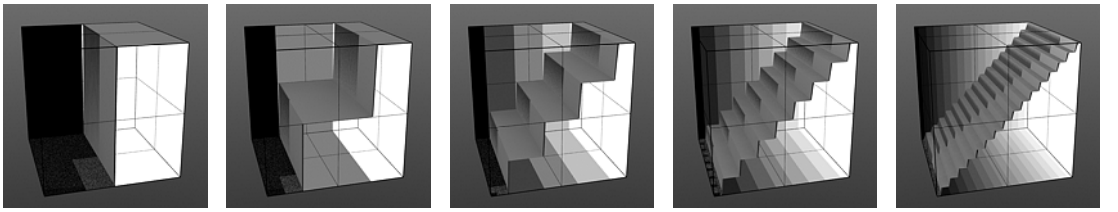


Smooth 0% Smooth 25% Smooth 50% Smooth 75% Smooth 100%



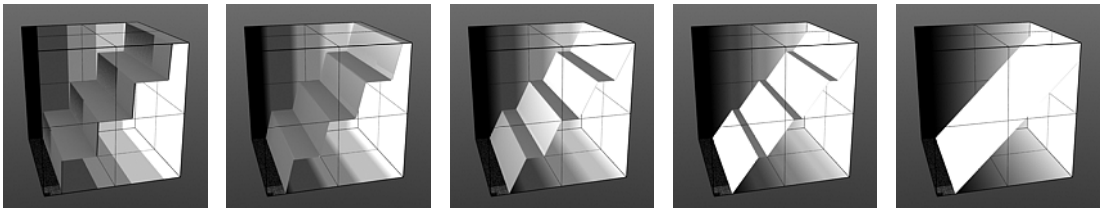
Waveforms - Staircase --

Staircase Levels: The Staircase function breaks the smooth gradations into specific discreet levels, producing a posterized effect. The number of discreet levels is determined by this option.

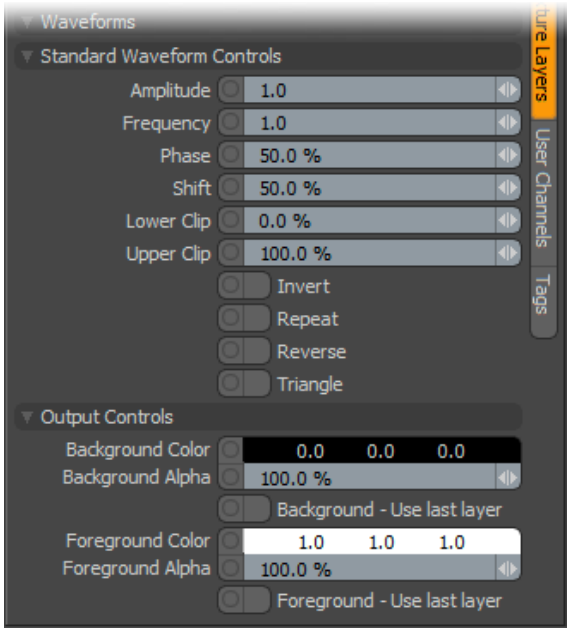


Levels 1 Levels 2 Levels 4 Levels 8 Levels 16

Level Slant: The 'Level Slant' option provides a way to interpolate between the step levels producing a less abrupt change from one step level to the next.

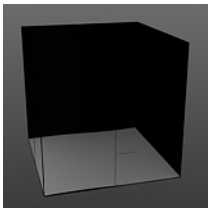


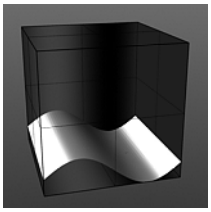
Slant 0% Slant 25% Slant 50% Slant 75% Slant 100%

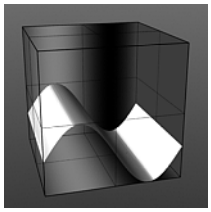


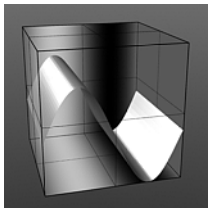
Standard Waveform Controls--

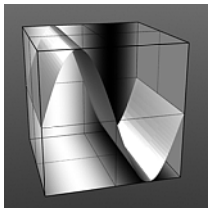
Amplitude: This value controls the maximum value of the Waveform scaling all other values in relation to the maximum.


Amplitude 0%

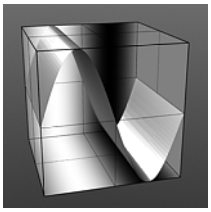

Amplitude 25%

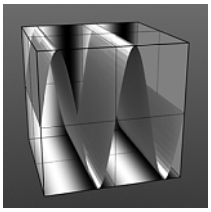

Amplitude 50%

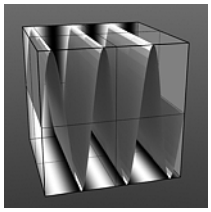

Amplitude 75%

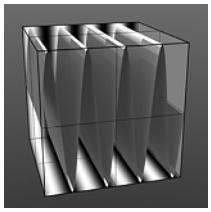

Amplitude 100%

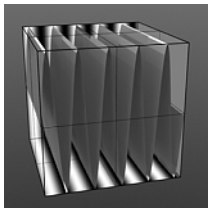
Frequency: This value controls the number of waveform occurrences in a single cycle. Waveform will only cycle multiple times when the 'Repeat' option is *enabled*.


Frequency 1

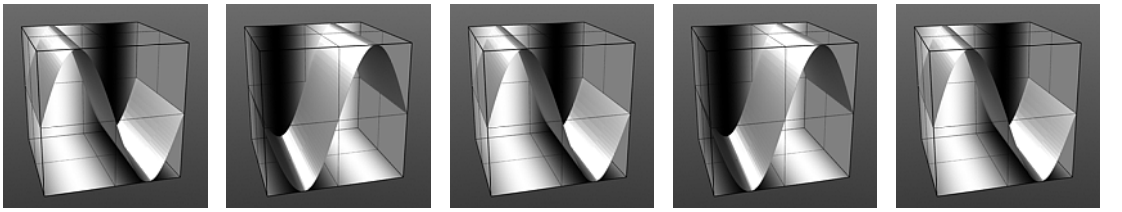

Frequency 2


Frequency 3


Frequency 4

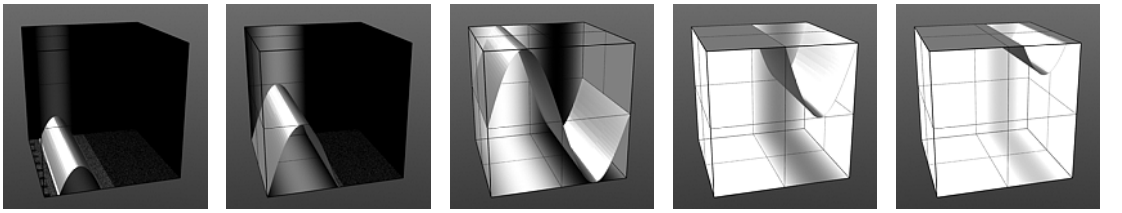

Frequency 5

Phase: This value moves the waveform forwards and backwards.



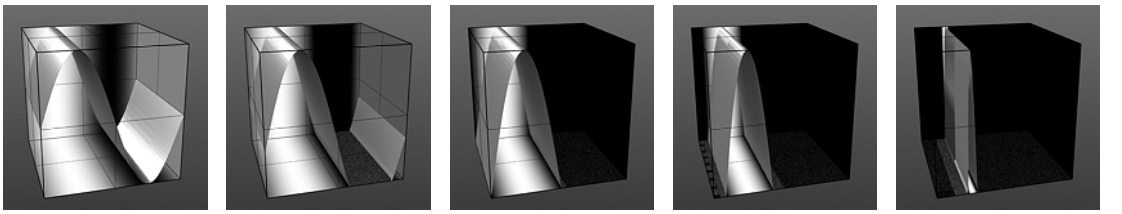
Phase 0% Phase 25% Phase 50% Phase 75% Phase 100%

Shift: This value moves the waveform up and down.



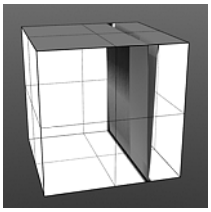
Shift 10% Shift 25% Shift 50% Shift 75% Shift 90%

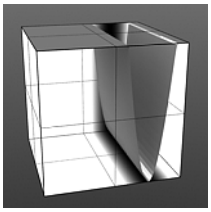
Lower Clip: This value determines a clip level for the Background Color/Value, truncating values beyond the defined setting and scaling those that remain. Combined with the Upper Clip value, users can apply this option to extend or contract the total range of values for the texture.

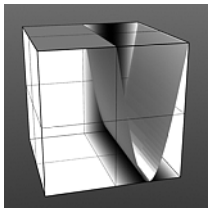


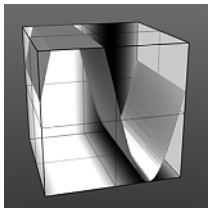
Low Clip 0% Low Clip 25% Low Clip 50% Low Clip 75% Low Clip 100%

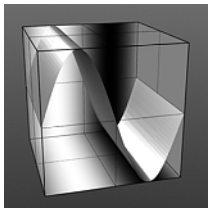
Upper Clip: This value determines a clip level for the Foreground Color/Value, truncating values beyond the defined setting and scaling those that remain. Combined with the Lower Clip value, users can apply this option to extend or contract the total range of values for the texture.


Upper Clip 0%


Upper Clip 25%


Upper Clip 50%


Upper Clip 75%


Upper Clip 100%

Invert: This option flips the waveform vertically.

Repeat: This option repeats the waveform, useful when the frequency option is set beyond a value of 1.

Reverse: This option flips the waveform horizontally.

Triangle: This options allows the waveform to be mirrored on it's own axis.

Output Controls--

Background Color/Value: This value determines the Color (or Value) of the texture's Background area, which will ramp toward the Foreground Color/Value.

Background Alpha: This value determines the Alpha transparency of the Background Color.

Background - Use Last Layer: When this option is enabled the Background Color area will be completely transparent, revealing the shading results of any lower layers.

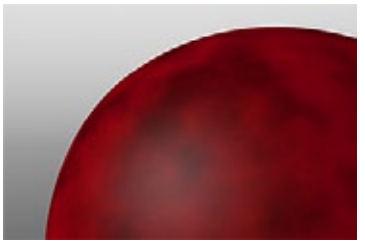
Foreground Color/Value: This value determines the Color (or Value) of the texture's Foreground area, which will ramp toward the Background Color/Value.

Foreground Alpha: This value determines the Alpha transparency of the Foreground Color.

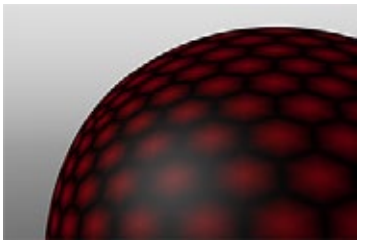
Foreground - Use Last Layer: When this option is enabled the Foreground Color area will be completely transparent, revealing the shading results of any lower layers.

Waveforms in Use

Here is a collection of rendered images using the Waveform functionality.

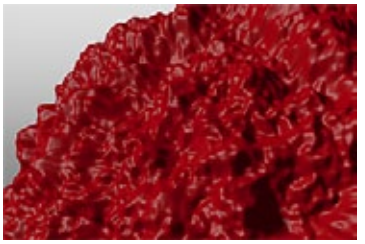


Noise as Diffuse Color Base

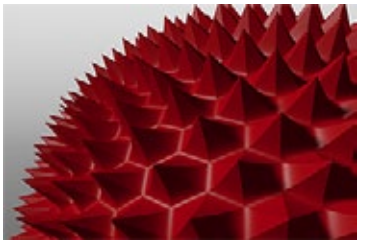


Grid as Diffuse Color Base

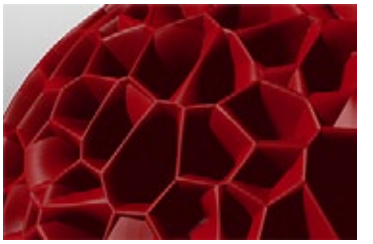
Cellular as Diffuse Color Base



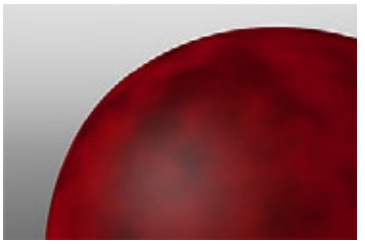
Noise as Displacement Base



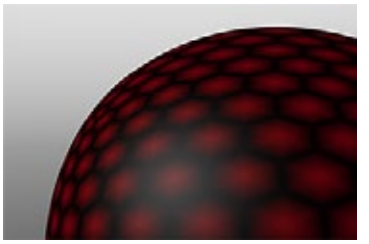
Grid as Displacement Base



Cellular as Displacement Base

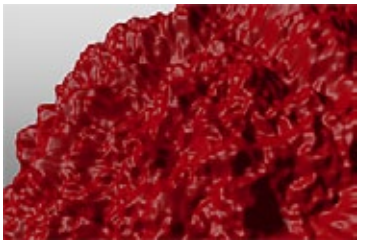


Noise as Diffuse Color Base

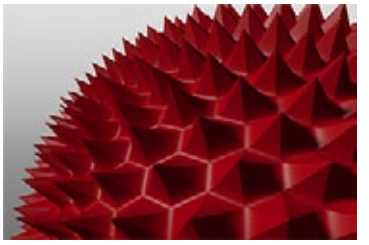


Grid as Diffuse Color Base

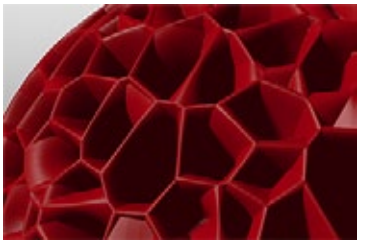
Cellular as Diffuse Color Base



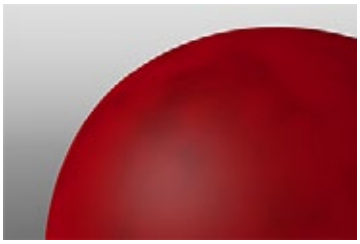
Noise as Displacement Base



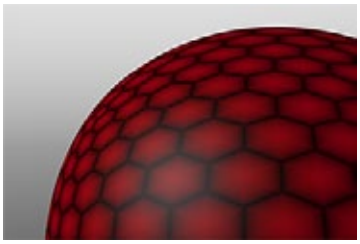
Grid as Displacement Base



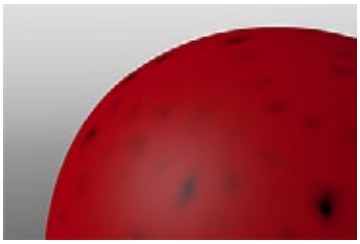
Cellular as Displacement Base



Rounded Waveform Applied to Base

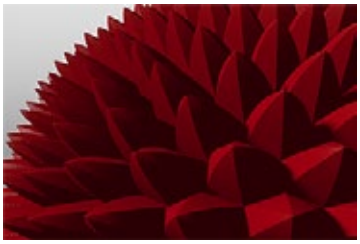


Rounded Waveform Applied to Base



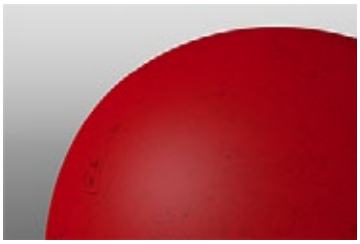
Rounded Waveform Applied to Base

Rounded Waveform Applied to Base

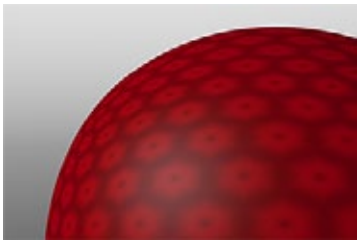


Rounded Waveform Applied to Base

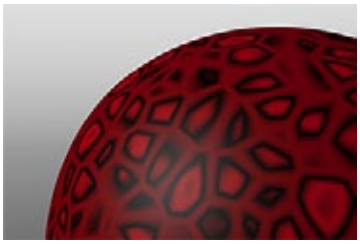
Rounded Waveform Applied to Base



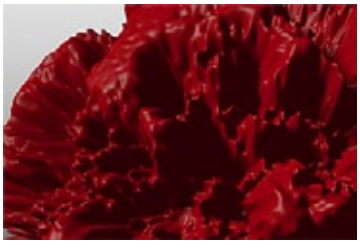
Sine Waveform Applied to Base



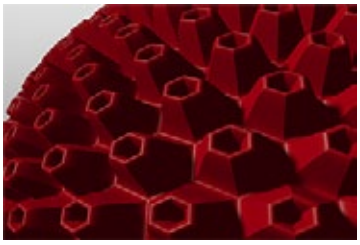
Sine Waveform Applied to Base



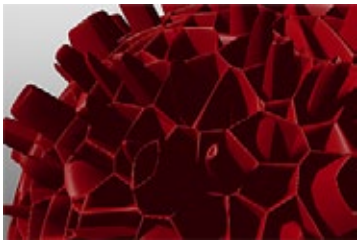
Sine Waveform Applied to Base



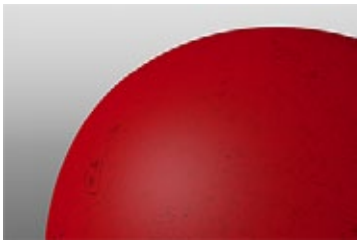
Sine Waveform Applied to Base



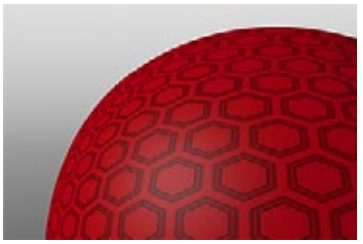
Sine Waveform Applied to Base



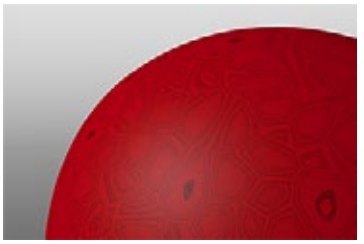
Sine Waveform Applied to Base



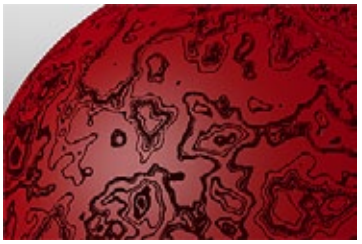
Noise Waveform Applied to Base



Noise Waveform Applied to Base



Noise Waveform Applied to Base

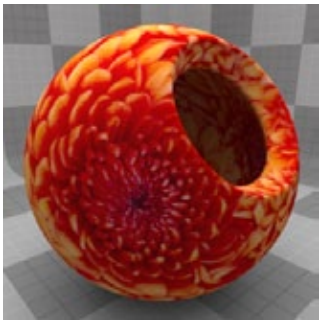


Noise Waveform Applied to Base

Noise Waveform Applied to Base

Noise Waveform Applied to Base

Image Map



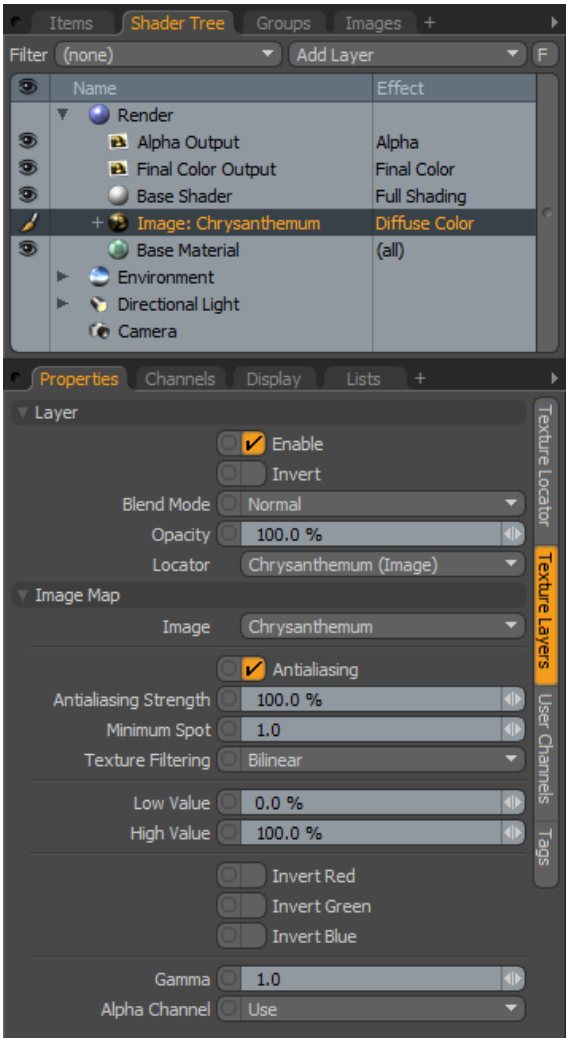
The 'Image Map' item could be considered the most important texture layer as it allows users to apply bitmap images to surfaces in modo. When originating from a photographic source, bitmaps are a means of literally getting the richness of the real world into modo, producing effects that would be difficult, if not impossible to produce otherwise. Want a surface to look like asphalt or dirt? Take a photograph and apply it as an Image Map; photographic textures are essential to photo real rendering.

Images created in editing/painting and illustration/typographic programs are extremely useful as well; label, posters, and other signage are great examples of this application. Image maps shouldn't be strictly limited to representing textured surfaces either, as they are also useful in many other cases; they can represent multiple complex procedural layers baked together, reducing the calculation overhead. They can be displacement maps created by modo or other packages that apply high frequency details to lower resolution subdivision surface models, providing a better level of interactivity to items among other benefits. Volumes could literally be written about the utility of image maps (and they have been!). Searching the internet for tutorials on 'Texture Mapping' will uncover untold numbers of websites dedicated to this very topic.

Adding the Image Map can be done through menu bar command 'Texture > Add Shader Tree Layer > Image Map' or using the 'Add Layer' function at the top of the full 'Shader Tree' viewport. The Image Map item contains all the attributes directly related to modo's usage of the bitmap and are available in the 'Properties' viewport when the item layer is selected in the Shader Tree. Scale, Position, Projection and other mapping specific settings are available in the associated 'Texture Locator'.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.



Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (way the texture is applied) to the surface. The 'Locator' option sets that association. Users can choose alternate locators, however, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Image Map--

Image: This pop-up allows you to select or load the specific image to be applied. Additionally, there are options for creating new blank images which can be useful for painting, sculpting and image baking.

Antialiasing: This check box toggles the 'Antialiasing' function for the selected image layer, a means of filtering the texture to produce smoother rendered results. Antialiasing samples the images pixels in texture space (as opposed to screen or render space) and averages them based on spot size and strength. When 'enabled', the 'AntiAliasing Strength' and 'Minimum Spot' options become available.

Antialiasing Strength: Active only when Antialiasing is enabled, this setting allows you to control the amount of antialiasing on the image. Setting this value above 100% will create a blurring effect.

Minimum Spot: Let's start by saying you probably don't need to adjust this value. The default of 1 means that the spot size is accurate to the sub pixel level. Increasing the minimum spot size is mostly useful for blurring an image or reducing details when using an image for an environment map. Now to explain what it actually does.

The "Spot" size relates to the spot created on the surface of a mesh when intersected by a ray. Rays are actually conical; think of the beam of light from a flashlight. The further the ray's origin is from the surface that it intersects the larger the spot will be. Further, if the ray hits the surface at an glancing angle it can be even larger. When there is an image applied to the surface in question the render engine needs to know how many 2D texture pixels should be sampled within that spot. This becomes more important the closer the camera is to the surface. If the minimum spot value is 1 or smaller the spot size is not clamped and will accurately sample the texture image at the sub pixel level if necessary. When the spot size is set to a value of 2 or higher the number of pixels sampled will always be greater than 1. This will result in a slight blurring of the image.

Texture Filtering: When spot size is set to 1 and the camera is so close to a surface that it samples the image at the sub-pixel level (i.e. an image pixel is bigger than a rendered pixel), Texture filtering will average across those samples. The result is a smooth interpolation of pixels rather than seeing sharp angular pixel boundaries. Users can choose from several options--

Nearest-- Disables filtering of image, actual pixel of image will be sampled.

Bilinear-- Samples neighboring pixels vertically and horizontally, producing smooth (some might say blurry) results.

Bicubic-- Samples all adjacent pixels providing the smoothest result.

Low/High Value: The Low/High values provide a means for users to re-map the total range of tones in an image, especially useful controlling value setting. For instance, if you are using an image as a displacement map you might want the darkest pixels to push the surface in and the bright pixels to push them out. By default the pixels of an image go from 0 to 100% which means that none of the pixels would yield a value to push the pixels inward. By setting the value range from -100% to 100% the image is effectively remapped so that the black pixels create a negative displacement and the white pixels a positive displacement with mid-tones being "neutral". Also could be useful in extending the range of standard (low dynamic range) images, making it possible to use standard non-HDR images to light a scene (although the results won't

look as good as using a real HDRI)

Low Value-- Determines the effective value of pixels that have a true value of 0,0,0 (r,g,b)

High Value-- Determines the effective value of pixels that have a true value of 255,255,255 (r,g,b)

Invert Red, Green, Blue: These three check boxes allow users to invert any of the three color channels of an image map individually. While users won't likely use this option often, especially on color images, it can be helpful to save normal maps where certain channels are mapped differently (inverted?) to what modo expects them to be.

Gamma: When applying images to a surface, they may appear in the rendered frame as being too light or washed out. This is because modo always renders internally at gamma 1.0 (linearly) and then applies the user specified output gamma when displaying or saving the rendered result. Bitmapped images typically have automatic gamma applied to them in an image editor or digital camera (otherwise, they would look too dark to most users); because of this, rendering with modo's default gamma value (2.2) will make images appear light or washed out as they have received double gamma. Therefore it is important to de-gamma the images. This can be done by inverting the gamma amount here, simply by dividing the render outputs Gamma amount (whatever it may be, 2.2 by default) by the image maps items 1.0 value. This can be done directly in the value input field using mathematical shorthand, users would type "1.0/2.2" in the dialog, pressing enter will return the value "0.455" (modo does the math for you!). Now the rendered image will produce the correct result.

Alpha Channel: This popup allows users to choose how modo will handle alpha channel data (if present) contained in the image file.

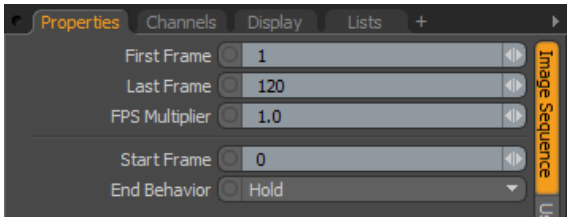
Use Alpha-- Masks the color pixels based on the embedded alpha channel data.

Ignore-- Discards the embedded Alpha channel.

Alpha Only-- Ignores any RGB information in the file treating it strictly as an single (alpha) channel image.

Image Sequence

Additional options appear when users apply an Image Sequence as a texture layer. To access these options, select the 'Image Sequence' item, then select the 'Image Sequence' subtab from the Shader Tree properties panel.



First/Last Frame: The 'First Frame' and 'Last Frame' options determine the range from the sequence to display/render in modo. Frames outside of this range will be ignored/ truncated.

FPS Multiplier: This value determines the speed at which the sequence plays in relation to the FPS rendered in the modo scene allowing users to adjust the timing of frames. So a 30fps image sequence applied to a 30fps would play at 1:1 frames when the value is 1.0 (One sequence frame for every rendered frame). A 2.0 value would be twice as fast as the scene, effectively skipping every other frame, conversely a 0.5 value would play back half as fast doubling the frames. Note that this speed affects the 'Start Frame' value below. A 'Start Frame' of 48 would actually start at 24 when the speed was 2.0.

Start Frame: The 'Start Frame' value determines the first rendered frame in modo in which to begin the playback of the image sequence allowing users to delay the start of the sequence. Frames rendered prior to this value would hold the 'First Frame' as specified above essentially duplicating the 'First Frame' in subsequent rendered frames until reaching the 'Start Frame' value.

End Behavior: These options determine how modo handles the image sequence when the 'Last Frame' value is reached-

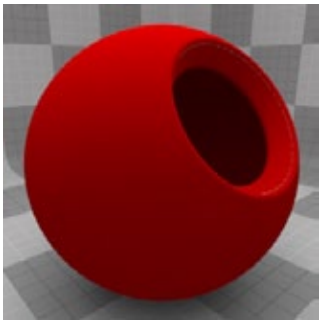
Hold- Duplicates the 'Last Frame' on all subsequent rendered frames.

Repeat- Loops the sequence between the 'First' and 'Last Frame' values.

Mirror- Ping-Pongs the sequence forwards and backwards between the 'First' and 'Last Frame' values.

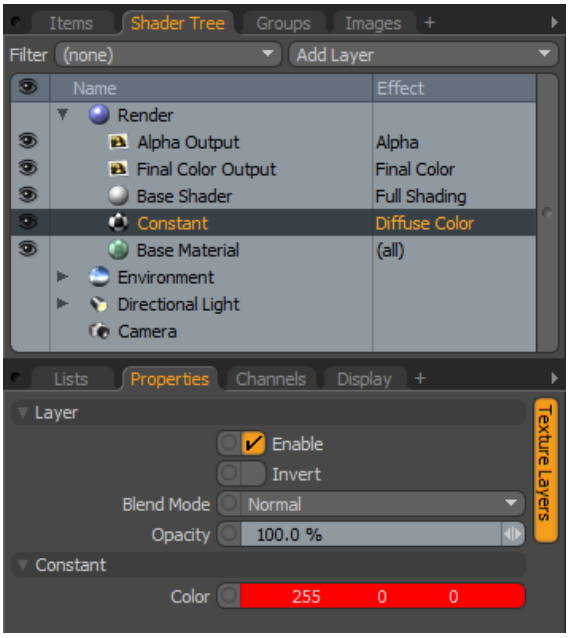
TIP: modo will automatically use the first available or last available frame when corresponding frames don't exist in the specified sequence. Take, for example, a sequence from frame 45 to 100 applied as a background plate. If the 'First' and 'Last Frame' values are 1 and 100 respectively, and the Start Frame value is 0, when rendering out the sequence, frame 45 would be duplicated across rendered frames 1-44 as no frames existed in the specified sequence - essentially the same behavior as the 'First /Last option. Interim missing frames will be duplicated until a proper sequence frame is encountered.

Constant



The Constant texture simply generates a constant Value or Color layer. This can be useful for setting a single color or value to a specific Effect without the need to create an entire Material. For instance, with a Constant texture set to Diffuse Color and applied inside an Item or Polygon mask the masked elements could use all the Material settings from the last Material beneath the mask but have its own Diffuse Color setting.

For information regarding adding and working with Shader Tree Items Layers, please reference the the Shader Tree page of the documentation.



Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

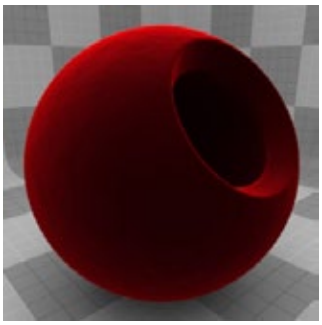
Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Constant--

Color: Determines the 'Color' or 'Value' of the layer depending on the 'Effect'.

Gradient



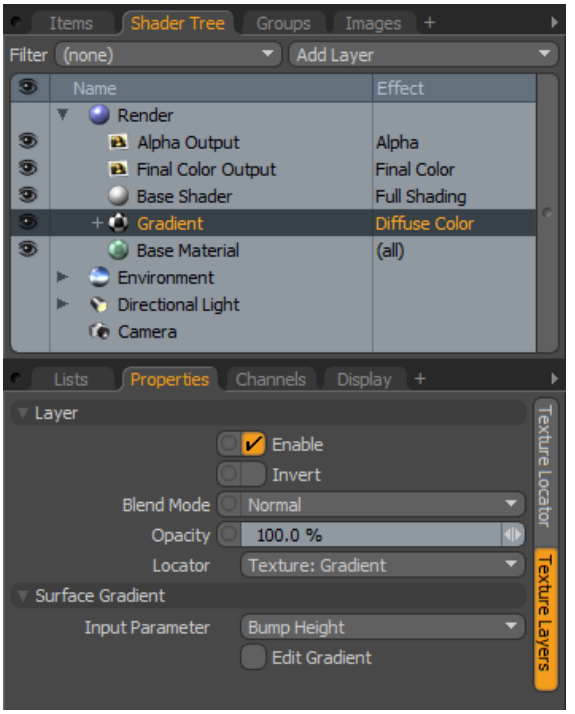
The Gradient texture layer is an incredibly powerful asset in the shading arsenal. Gradients provide a method for creating parametric materials. Put simply, gradients remap colors or value based input parameters the user defines and produces an output based on the values in the gradient itself. For instance, with a gradient texture you could tell a surface to ramp from purple to light blue based on the incidence angle to the camera or purple to green based on the slope of the surface. You can even control the number of clones in a Replicator based on height and angle of a surface. Gradients are also useful in controlling the color of fur, and Replicators. With so many options, it can feel a little overwhelming; the following samples should help ease any confusion regarding a particular gradients use.

Gradients are added to the shader tree where it is indicated the 'Effect' it will have on the surface, such as Transparency or Diffuse Color. Next, users will need to select the 'Input Parameter' on the Gradient textures properties panel, this defines how the gradient modulates the surface it is affecting. By clicking the 'Edit Gradient' check box users can open the Gradient Editor where the actual gradient is defined. Its usage is covered below. For additional information on adding and using layers, please reference the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When *disabled* (unchecked), the layer has no effect

on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.



Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Gradient layers always have an associated 'Texture Locator' that is automatically created in the 'Item List'. While they typically define the mapping of the texture (way the texture is applied) to the surface, for gradients, the mapping is largely defined by the 'Input Parameter' options. For gradients, the Locator option is necessary to select the appropriate locator that will control certain input parameter effects, such as 'Distance to Locator' and 'Locator Incidence'. Within the Texture Locator options, users may also specify UV maps and others useful options for other parameters.

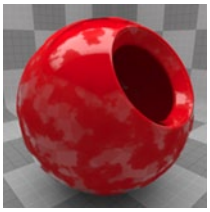
Surface Gradient--

Input Parameters: There are many different input parameters types, select from one of the various categories to see examples-

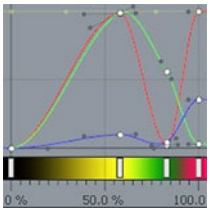
Input Parameter - Basic

Clearcoat Amount-- When the Input Parameter is set to 'Clearcoat Amount', the gradient will key off of any Clearcoat texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the

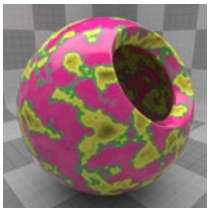
corresponding single gradient color/value.



Clearcoat



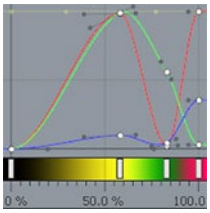
Gradient



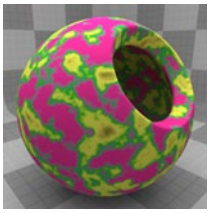
Clearcoat Gradient

Diffuse Amount-- When the Input Parameter is set to 'Diffuse Amount', the gradient will key off of any Diffuse Amount texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Diffuse Roughness

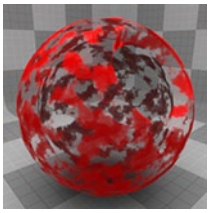


Gradient

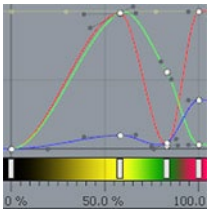


Diff. Roughness Gradient

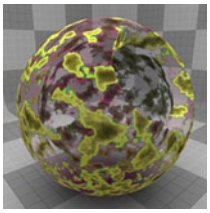
Dissolve-- When the Input Parameter is set to 'Dissolve', the gradient will key off of any Dissolve texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.



Dissolve



Gradient

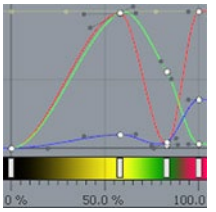


Dissolve Gradient

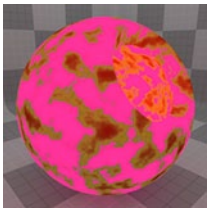
Luminous Amount-- When the Input Parameter is set to 'Luminous Amount', the gradient will key off of any Luminous Amount texture layer present on the surface.

Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Luminous Amount



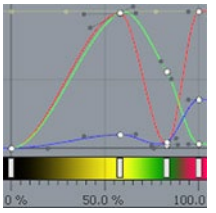
Gradient



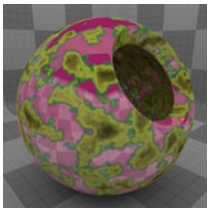
Luminous Amt. Gradient

Reflection Amount-- When the Input Parameter is set to 'Reflection Amount', the gradient will key off of any Reflection Amount texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Clearcoat



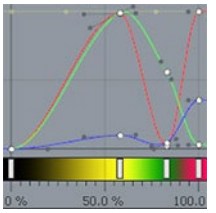
Gradient



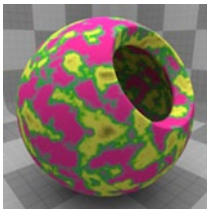
Clearcoat Gradient

Reflection Fresnel-- When the Input Parameter is set to 'Reflection Fresnel', the gradient will key off of any Fresnel texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Clearcoat

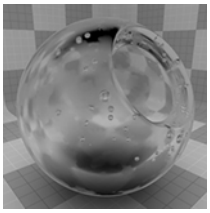


Gradient

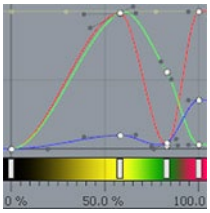


Clearcoat Gradient

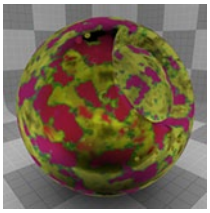
Refraction Roughness-- When the Input Parameter is set to 'Refraction Roughness', the gradient will key off of any Refraction Roughness texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.



Refraction Roughness

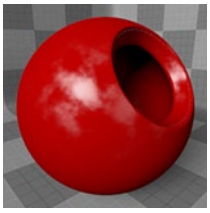


Gradient

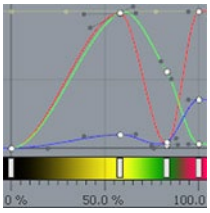


Clearcoat Gradient

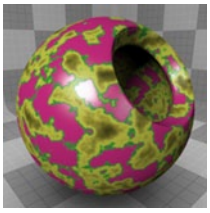
Roughness-- When the Input Parameter is set to 'Roughness', the gradient will key off of any Roughness texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.



Roughness



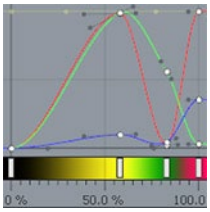
Gradient



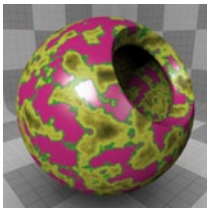
Roughness Gradient

Specular Amount-- When the Input Parameter is set to 'Specular', the gradient will key off of any Specular Amount texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Specular Amount



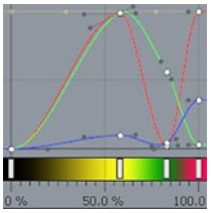
Gradient



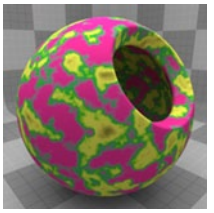
Specular Gradient

Specular Fresnel-- When the Input Parameter is set to 'Specular Fresnel', the gradient will key off of any Specular Fresnel texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Specular Fresnel



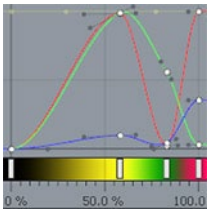
Gradient



Spec. Fres. Gradient

Subsurface Amount-- When the Input Parameter is set to 'Subsurface Amount', the gradient will key off of any Subsurface Amount texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

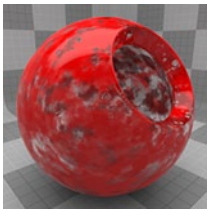
Subsurface Amount



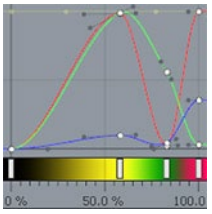
Gradient

Subsurface Gradient

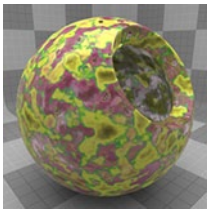
Transparent Amount-- When the Input Parameter is set to 'Transparent Amount', the gradient will key off of any Transparent Amount texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.



Subsurface Amount



Gradient



Subsurface Gradient

Input Parameter - Custom Material

Cel Edge Width-- When the Input Parameter is set to 'Cel Edge Width', as related to a Cel Edge Material, the gradient will key off of any Cel Edge Width texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Halftone Edge Width-- When the Input Parameter is set to 'Halftone Edge Width', as related to a Halftone Material, the gradient will key off of any Halftone Edge Width texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Halftone Tiles-- When the Input Parameter is set to 'Halftone Tiles', as related to a Halftone Material, the gradient will key off of any Halftone Tiles texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Skin Epidermis Diffuse Amount-- When the Input Parameter is set to 'Skin Epidermis Diffuse Amount', as related to a Skin Material, the gradient will key off of any Skin Epidermis Diffuse Amount texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Edit Gradient: Check this box to open the gradient editor. This is where the actual gradient is created (see below).

Input Parameter - Fur

Fur Bump-- When the Input Parameter is set to 'Fur Bump', as related to a Fur Material, the gradient will key off of any Fur Bump texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Fur Clump Density-- When the Input Parameter is set to 'Fur Clump Density', as related to a Fur Material, the gradient will key off of any Clump Density texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Fur Clumps-- When the Input Parameter is set to 'Fur Clumps', as related to a Fur Material, the gradient will key off of any Fur Clumps texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Fur Curls-- When the Input Parameter is set to 'Fur Curls', as related to a Fur Material, the gradient will key off of any Fur Curls texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

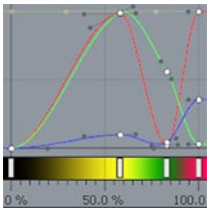
Fur Density-- When the Input Parameter is set to 'Fur Density', as related to a Fur Material, the gradient will key off of any Fur Density texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Fur Flex-- When the Input Parameter is set to 'Fur Flex', as related to a Fur Material, the gradient will key off of any Fur Flex texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

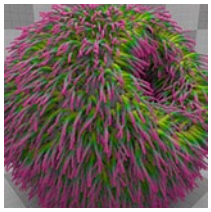
Fur Growth Jitter-- When the Input Parameter is set to 'Fur Growth Jitter', as related to a Fur Material, the gradient will key off of any Growth Jitter texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Fur Length-- When the Input Parameter is set to 'Fur Length', as related to a Fur Material, the gradient will key off of any Fur Length texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Fur Parametric Length-- When the Input Parameter is set to 'Fur Parametric Length', as related to a Fur Material, the gradient will apply its colors/values along the length of individual fur fibers, 0% at the root, to 100% at the tip.



Gradient



Fur Length

Fur Root Bend-- When the Input Parameter is set to 'Fur Root Bend', as related to a Fur Material, the gradient will key off of any Root Bend texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Fur Root Frizz-- When the Input Parameter is set to 'Fur Root Frizz', as related to a Fur Material, the gradient will key off of any Root Frizz texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Fur Root Kink-- When the Input Parameter is set to 'Fur Root Kink', as related to a Fur Material, the gradient will key off of any Root Kink texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Fur Stray Hairs-- When the Input Parameter is set to 'Fur Stray Hairs', as related to a Fur Material, the gradient will key off of any Stray Hairs texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Fur Tip Frizz-- When the Input Parameter is set to 'Fur Tip Frizz', as related to a Fur Material, the gradient will key off of any Tip Frizz texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

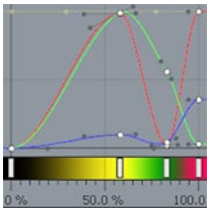
Fur Tip Kink-- When the Input Parameter is set to 'Fur Tip Kink', as related to a Fur Material, the gradient will key off of any Tip Kink texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a

procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

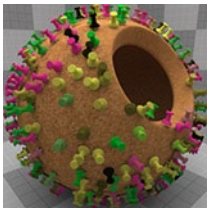
Input Parameter - Particle Effects

Particle Age-- When the Input Parameter is set to 'Particle Age', the gradient will key off Particle Age values present in an imported RealFlow particle animation providing additional controls over the animation of Blob, Sprite and Volume items.

Particle ID-- When Fur fibers and Surface Generator replicas are created, each is assigned a random Particle ID value between 0 and 1. When a Gradient is applied with the Particle ID input parameter, each fiber or replica will pull a value from the gradient based on that randomly generated ID offering users the ability to create controlled randomness. Now generating 'salt and pepper' fibers for a Fur Material is as easy as pie.



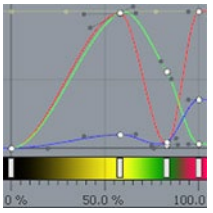
Gradient


Replicator Particle ID

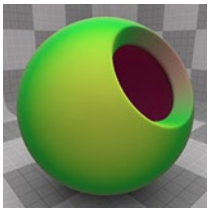
Fur Particle ID

Input Parameter - Sample Parameters

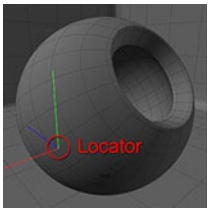
Distance to Camera-- Modulates the value as a function of the physical distance to the camera measured from its center point. This opens a lot of interesting possibilities, such as adding fog like effects or fading luminosity as the surface gets further from the camera. Keep in mind a gradients layer isn't strictly limited to affecting Diffuse Color.



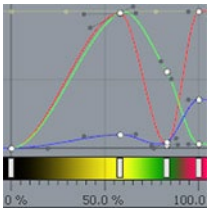
Gradient


Distance to Camera

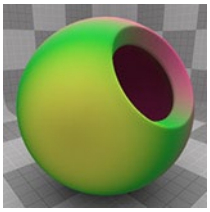
Distance to Locator-- Very similar to the Distance to Camera parameter, the Distance to Locator modulates the gradient based on the position of the items texture locator, specified by the 'Locator' option above.



Item Locator Position

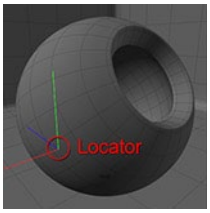


Gradient

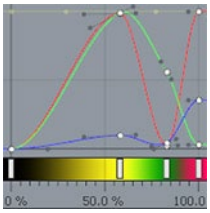


Dist. to Locator

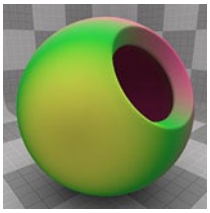
Screen R/X/Y-- Very similar to the Distance to Camera parameter, the Distance to Locator modulates the gradient based on the position of the items texture locator, specified by the 'Locator' option above.



Item Locator Position

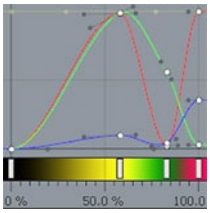


Gradient

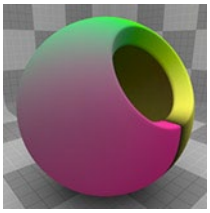


Dist. to Locator

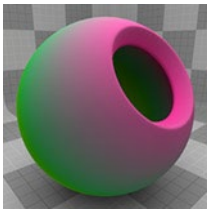
Texture U/V-- The Gradient will read the texture coordinates from the UV map and modulate the Value based on the 'U' (horizontal) or 'V' (vertical) value of the map.



Gradient



Texture 'U'



Texture 'V'

Texture Value-- This setting will apply the Gradient across the values of any texture layer directly beneath it regardless of that texture's 'Effect', 'Blending' or 'Opacity' settings.

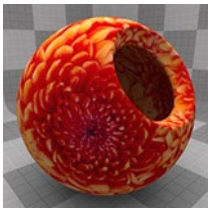
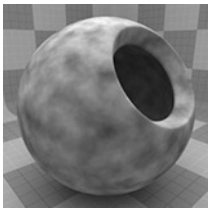
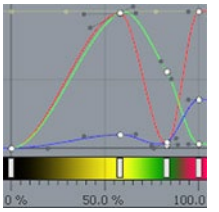


Image Map Texture


Procedural Texture



Gradient

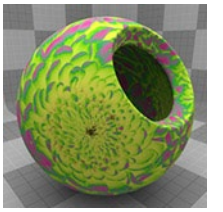
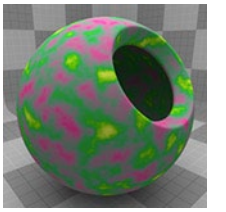
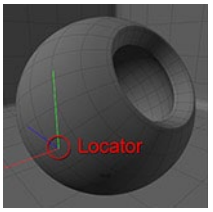


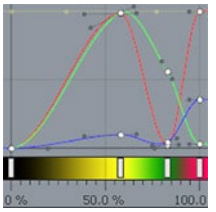
Image Map Grad. result


Procedural Grad. result

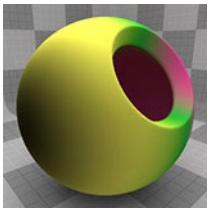
XYZ Distance to Locator-- Creates a distance based linear gradient that aligns itself along either the X, Y or Z axes as specified, with the gradients origin controlled by the position of the items texture locator, specified by the 'Locator' option above.

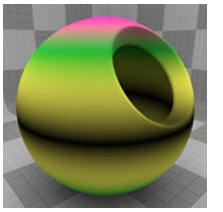


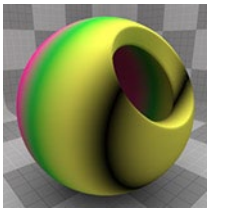
tem Locator Position



Gradient


X Distance


Y Distance


Z Distance

Input Parameter - Shader Control

Driver A, B, C, D-- The 'Driver' settings will read any Texture item or combination thereof, whose 'Effect' is set to the same Driver letter (A,B, C or D). This allows users to have texture layers that can drive Gradients very easily. It also allows users the ability to remap the tones of a texture layers by using the gradient. Users should note that when using colored bitmap textures to drive the gradient, the brightness, or luminosity of the image will be the actual driving property. Users should also note that the gradient needs to be above the driving texture in the shader tree and that the letter value of the texture items 'Effect' needs to match the gradients 'Input Parameter' value (for example, both would need to be set to "Driver A' for it to work).

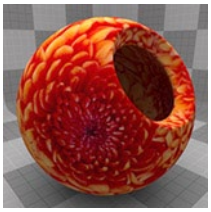
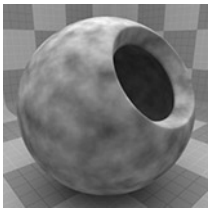
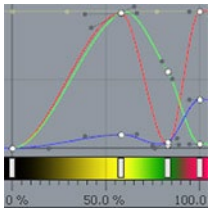


Image Map Driver



Procedural Driver



Gradient

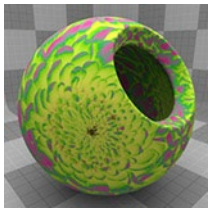
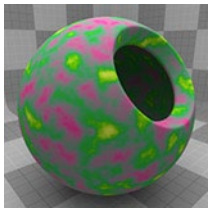
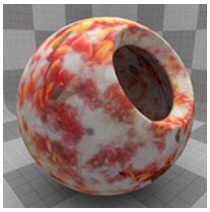


Image Map Grad. result

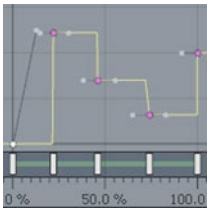


Procedural Grad. result

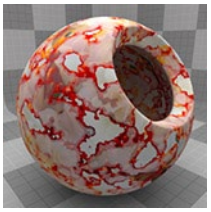
Group Mask-- When the Gradient layer items 'Input Parameter' is set to 'Group Mask' and the 'Effect' is set to the same, a gradient can modulate any texture layers themselves set as 'Group Mask (within the the same group, of course). When the 'Effect' is set to anything else, the Gradient is driven by the Group Mask items, much like the Driver input parameter, overriding the effect of the mask on the group.



Base Group Mask

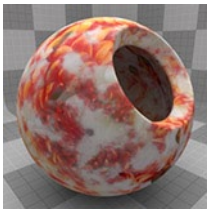


Stepped Gradient

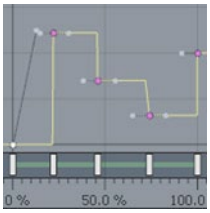


Modulated Mask

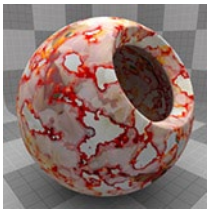
Layer Mask-- The 'Layer Mask' will work identically to the 'Group Mask input parameter, but where a Group Mask affects an entire group, a layer mask only affects a single layer. Setting the Gradient layers 'Effect' to 'Layer Mask', the gradient will modulate the texture layer directly below when set as 'Layer Mask'. When Gradients 'Effect' is set to anything else, the Gradient is driven by the Layer Mask items, much like the Driver input parameter, overriding the effect of the mask.



Base Layer Mask



Stepped Gradient



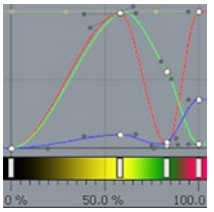
Modulated Mask

Input Parameter - Special Effects

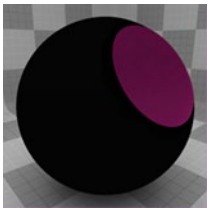
Stencil-- When the Input Parameter is set to 'Stencil', the gradient will key off of any Texture layers with their effect designated as Stencil. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Input Parameter - Surface Parameters

Back Facing-- When set to 'Back Facing' the gradient will color both the front and rear sides of a polygon. The Color or Value at the 0% position shades the front of the polygon, while the Color or Value at the 100% position shades the the opposite side of the polygon. Users will first need to enable the 'Double Sided' option in the Material item to see the results of this Gradient Parameter.

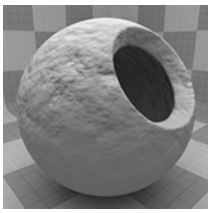


Gradient

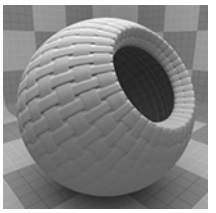


Back Facing

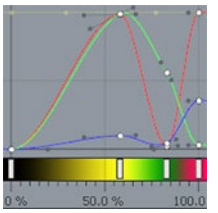
Bump Height-- When set to 'Bump Height' the Gradient is driven by any Bump Map textures that are effecting the same material and the input value becomes a percentage where 0% is the lowest area of the bump texture and 100% is the highest peak. If there are no texture layers driving the bump map effect the Gradient will be disabled.



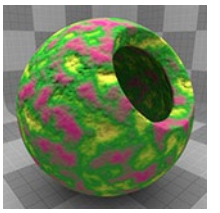
Noise Bump Map



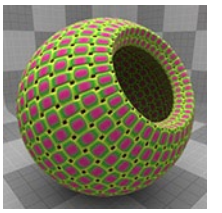
Weave Bump Map



Gradient

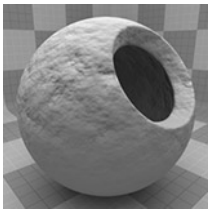


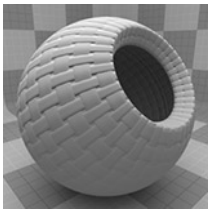
Bump Height Gradient

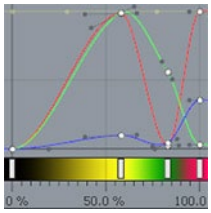


Bump Height Gradient

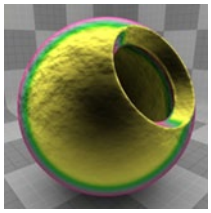
Bumpless Incident-- When set to 'Bump Height' the Gradient is driven by any Bump Map textures that are effecting the same material and the input value becomes a percentage where 0% is the lowest area of the bump texture and 100% is the highest peak. If there are no texture layers driving the bump map effect the Gradient will be disabled.

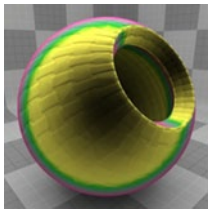

Noise Bump Map


Weave Bump Map

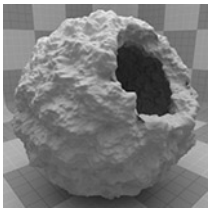


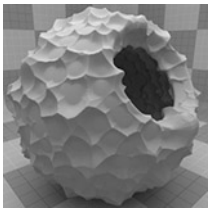
Gradient

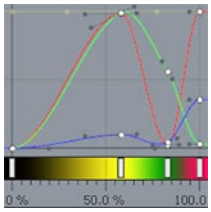

Bumpless Incidence


Bumpless Incidence

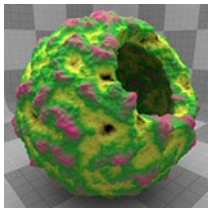
Displacement Height-- Similar to Bump Height, this input parameter is driven by a Displacement texture layers directly below the Gradient in the Shader Tree and maps itself along the the displacement height amount. 0% at the lowest displacement area, and 100% at the maximum displacement amount. Negative gradient values may also be necessary for remapped displacement values (i.e. changing the low/high values in the texture item to -100%, 100% respectively).

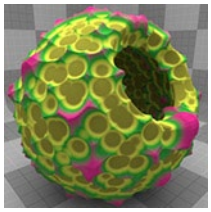

Displacement Map


Displacement Map

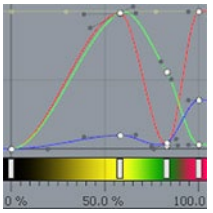


Gradient

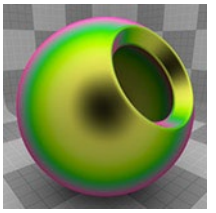

Disp. Height Gradient

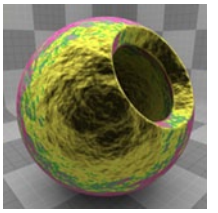

Disp. Height Gradient

Incidence Angle-- This is the angle between the camera and the surface being evaluated. Surfaces that face the camera head on have an incidence angle of 0° (0%) while the edges that face away (perpendicular) to the camera have a value of 90°(100%). This setting is often used to simulate the Fresnel effect as well as simulate the look of velvet surfaces.

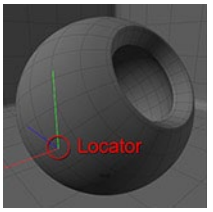


Gradient

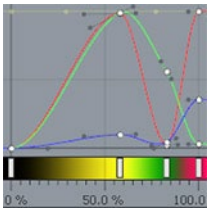

Incidence


Incidence w. Bump

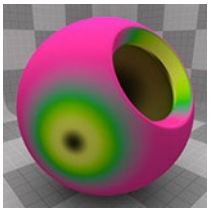
Locator Incidence-- This parameter is identical in effect to incidence angle but can be controlled by a locators position instead of fixed to the cameras point of view.



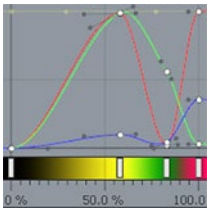
Item Locator Position



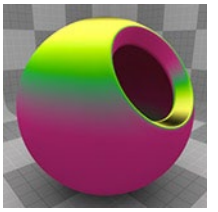
Gradient


Locator Incidence

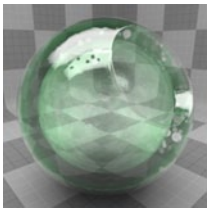
Slope-- This is simply the angle of the surface normal as compared to the World axes. 0° straight up is 0%, and 90° perpendicular is 100%



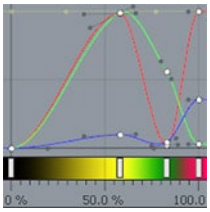
Gradient


Slope

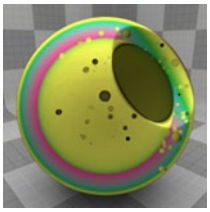
Thickness-- Determines how thick or deep an area is and applies the gradient based on the measured depth. The 'Thickness' is calculated by a ray being fired inward from the first surface, until the next outward facing surface is encountered (the backside) or another surface is encountered in the case of surface intersections.



Transparent Surface



Stepped Gradient



Thickness Gradient

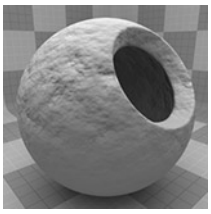
Input Parameter - Surface Particle Generation

Surface Particle Density-- When the Input Parameter is set to 'Surface Particle Density', the gradient will key off of the Surface Particle Density texture layer present on the surface. Keep in mind the gradient maps across multitudes values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

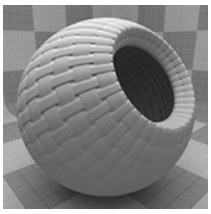
Surface Particle Size-- When the Input Parameter is set to 'Surface Particle Size', the gradient will key off of the Surface Particle Size texture layer present on the surface. Keep in mind the gradient maps across values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Input Parameter - Surface Shading

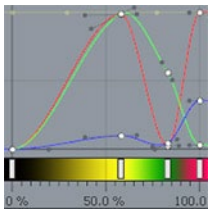
Bump-- When set to 'Bump' the Gradient is driven by any Bump Map textures that are effecting the same material and the input value becomes a percentage where 0% is the lowest area of the bump texture and 100% is the highest peak. If there are no texture layers driving the bump map effect the Gradient will be disabled.



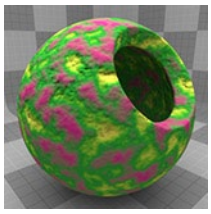
Noise Bump Map



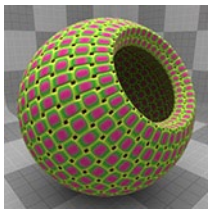
Weave Bump Map



Gradient

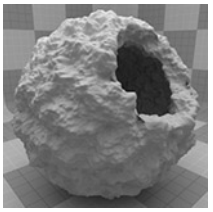


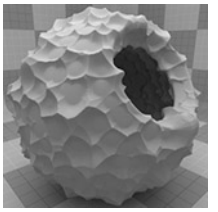
Bump Height Gradient

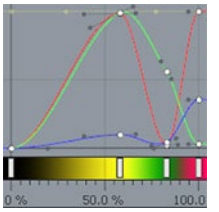


Bump Height Gradient

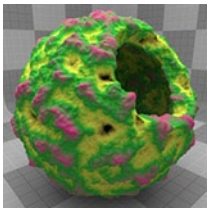
Displacement-- Similar to Bump Height, this input parameter is driven by a Displacement texture layers directly below the Gradient in the Shader Tree and maps itself along the the displacement height amount. 0% at the lowest displacement area, and 100% at the maximum displacement amount. Negative gradient values may also be necessary for remapped displacement values (i.e. changing the low/high values in the texture item to -100%, 100% respectively).

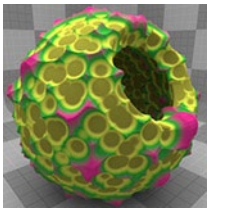

Displacement Map


Displacement Map



Gradient


Disp. Height Gradient


Disp. Height Gradient

Input Parameter - Volumetric Effects

Volumetric Absorption Amount-- When the Input Parameter is set to 'Volumetric Absorption Amount', as related to a Volume Item layer, the gradient will key off of any Volumetric Absorption Amount texture layer present. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Volumetric Density-- When the Input Parameter is set to 'Volumetric Density', as related to a Volume Item layer, the gradient will key off of any Volumetric Density texture layer present. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Volumetric Level-- When the Input Parameter is set to 'Volumetric Level', as related to a Volume Item layer, the gradient will key off of any Volumetric Level texture layer present. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Volumetric Scattering Amount-- When the Input Parameter is set to 'Volumetric Scattering Amount', as related to a Volume Item layer, the gradient will key off of any Volumetric Scattering Amount texture layer present. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient will only register the corresponding single gradient color/value.

Input Parameter - Other

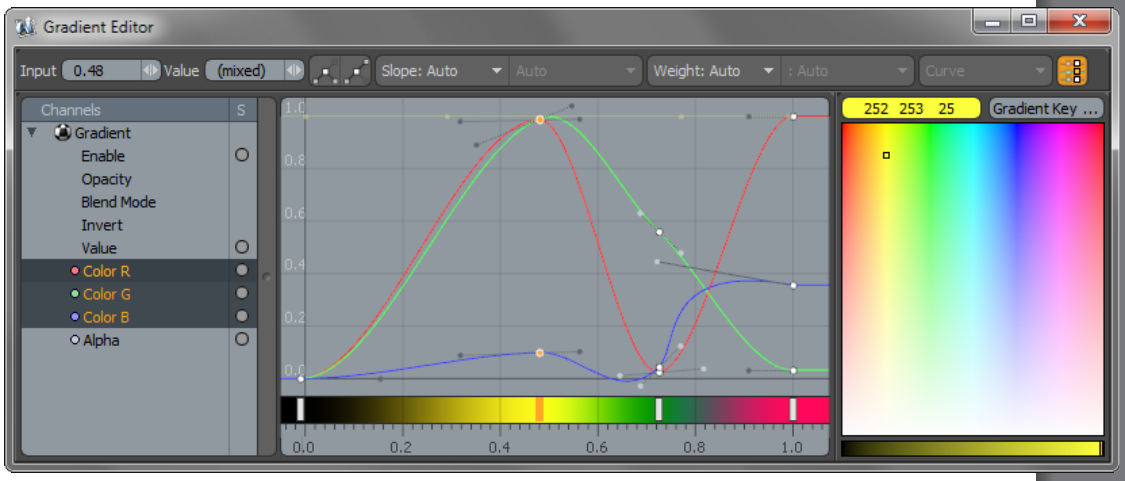
ClipDist-- Description and sample images forthcoming.

particelVelocity-- Description and samples forthcoming.

textureParticleID-- Description and samples forthcoming.

Edit Gradient: Check this box to open the gradient editor.

Gradient Editor



The Gradient Editor is where users define the actual Gradient values. Though visually and functionally similar to the Graph Editor, the Gradient Editor assigns values over a range instead of over time as the Graph Editor does. Users will typically only be concerned with the R,G & B color, Alpha and Value channels. Modifying other attributes reverts the bar to a timeline where values are modified over time.

The window itself is divided into several sections.

Across the top are the numeric input text fields followed by key controllers. These allow users to apply specific values to a key as well as control the slope of incoming and outgoing curve handles. The left side of the window displays the channels available for editing. modo will generally select the appropriate channel based on the layers 'Effect' setting, but user can select 'Value' or 'RGB Colors' based on the layers intended use. The center window is where users define the keys and modify their positions. New keys are added by MMB+clicking in the graph area at the position where the key should be created. Once set, users can modify the key LMB+clicking in the color palette on the right side. Positions of keys can be modified interactively by simply clicking on the individual handle bases (the white dots along the curves) and dragging them around in

the viewport. Key positions can also be modified by dragging the handles that appear in the gradient bar below the graph area.

Channels--

From within the channel section, users can select the value they wish to modulate. For diffuse color this would be the 'Color R,G and B channels, for specular amount, this would be the 'Value' channel. Once selected, a line representing the value appears in the graphing window.

Setting Keys--

Keys can be added to the currently selected curve(s) by MMB+clicking either on the curve itself or anywhere on the graph area background.

Setting Colors for Keys--

The color of selected keys can be set or adjusted by selecting the color key and either LMB+clicking and dragging on the Color Picker viewport or from the RMB context menu. Selecting 'Key Color' from this menu will open the Color Picker from which a color may be chosen. In both cases additional keys may be created on associated curves at the key Input value.

Deleting Keys--

Keys can be deleted from curves in two ways, to delete only the currently selected keys press the 'Delete' key. To delete the currently selected keys and keys at the same Input value on associated curves select 'Delete Key' from the RMB context menu.

Selection--

LMB - Selects a key under the cursor, holding the Shift key while selecting adds to the selection, and holding the Control key removes a key from the selection set. Clicking on the background deselects all keys.

LMB & Drag on the background starts box selection, drag out a box to enclose the keys you want to select and release the mouse button. All keys within the box will become selected, any outside the box will be deselected. Holding down the Shift key adds the keys within the box to the selection while the Control key removes them.

LMB click on a curve in the background (dimmed) will select the curve. Curves can be added to the selection by holding down the Shift key and clicking on a curve and can be removed from the selection by using the Control key.

Curves may also be selected from the menu at the top left of the viewport, this menu lists all the curves associated with the currently selected Gradient item(s).

Editing--

LMB & Drag on a key selects the key if it's not already selected and allows quick editing of both Input & Output values at the same time. If there are several keys already selected and you click drag on one of them then all the selected keys will be modified.

MMB & Drag on a key works in a similar way except that editing is constrained to the Output value. Similarly RMB & Drag constrains the edits just to the Input Parameter value.

MMB & Drag vertically on the background adjusts the Output value of all selected keys.

RMB & Drag horizontally on the background adjusts the Input value of all selected keys.

Control-MMB & Drag on the background provides constrained editing of Input & Output values for selected keys based on the initial direction of movement.

LMB Double click on a key selects all the keys on the curve.

Context Menu--

The Gradient Editor context menu is accessed by RMB click & release. If the cursor is over an un-selected key when the menu is opened the key will be added to the selection. Operations performed from the context menu will affect all the currently selected keys.

Navigation--

Panning & Zooming behavior in the Gradient Editor is similar to the 3D viewports with a few additions:

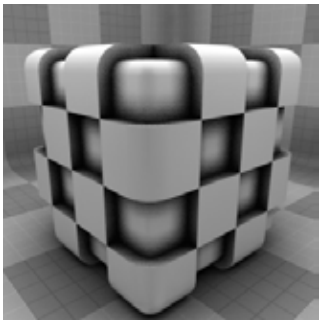
Alt-MMB Drag adjusts the start value of the Input (horizontal) range displayed in the editor, while Alt-RMB Drag adjusts the end value of the Input Range.

Alt-Shift-MMB zooms the Output Range displayed in the editor, while Alt-Shift-RMB zooms the Input Range. In both cases the zoom is centered on the initial cursor location.

Alt-Control-RMB provides box zooming.

Mouse Wheel - Zooms in and out around the cursor location.

Occlusion



Occlusion is a technique for calculating the shading of surfaces based on how occluded it is, depending on the settings, any receding or recessed areas would be shaded darker (or lighter). modo also has a 'Render Output' called 'Ambient Occlusion', that produces similar results, but does not take into account surface attributes. Occlusion is a simple and fast way to simulate Global Illumination without the overhead of calculating any scene lighting, as it only takes into account surfaces and how they adjoin to each other. Occlusion makes its calculations by firing a number of somewhat random rays and calculating shading for a pixel based on how many rays intersect with neighboring surfaces. There are a number of settings for users to control just how the occlusion layer shades the surface, producing a wide variety of effects. It's also useful to change the layer 'Effect' where the contributions of other layers can be controlled by the occlusion layer (masking of a noise layer just in the crevices anyone?).

For information regarding adding and working with Shader Tree Items Layers, please reference the the Shader Tree page of the documentation.

Layer--

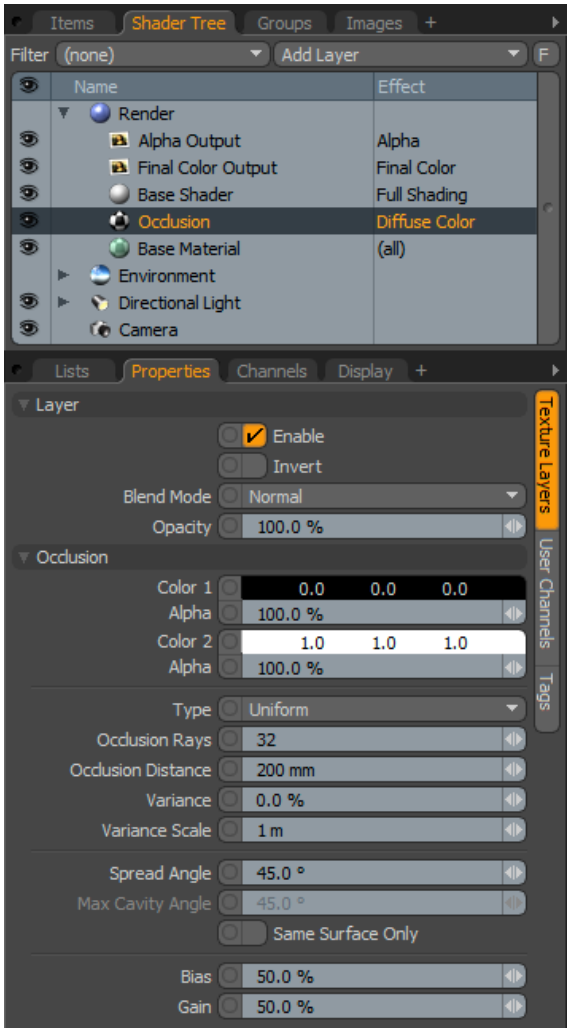
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the

layer itself on the surface.



Occlusion--

Color/Value 1: Determines the color of the shading in the occluded areas. This color will ramp toward 'Color 2'.

Alpha: Value determines the level of transparency for 'Color 1'.

Color/Value 2: Determines the color of the shading in the non-occluded areas. This color will ramp toward 'Color 1'.

Alpha: value determines the level of transparency for 'Color 2'.

Type: Determines the occlusion calculation method, in essence controlling the way rays are fired from the surface at each pixel location. Each type produces different results.

Uniform- Fires rays in all directions in the hemisphere.

Up- Looks for occlusions above the shading point.

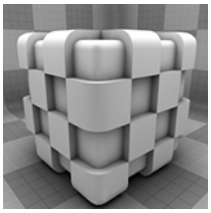
Down Slope- Looks for occlusions above the shading point and in the surface plane.

Reflection- Looks for occlusions in the direction of reflection. This is determined by the cameras position (when rendering) where a ray is fired and when it hits a surface, it bounces off at the angle of incidence and if it hits another surface after bouncing, its considered occluded. Useful for faking soft reflection among other uses.

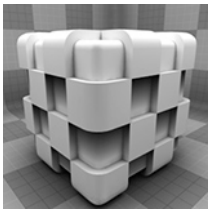
Concavity- Occludes where the surface is concave, as determined by the 'Max Cavity Angle' value.

Convexity- Occludes where the surface is convex, as determined by the 'Max Cavity Angle' value.

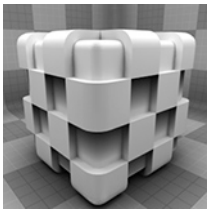
Concavity & Convexity- Occludes in both the convex and concave areas of a surface, as determined by the 'Max Cavity Angle' value.



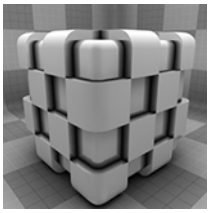
Uniform



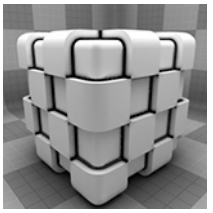
Up



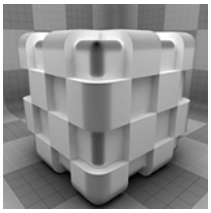
Down Slope



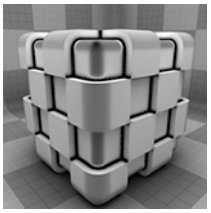
Reflection



Concavity

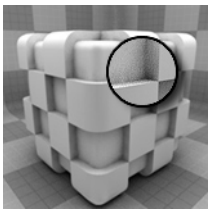


Convexity

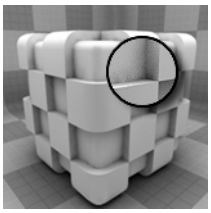


Concavity & Convexity

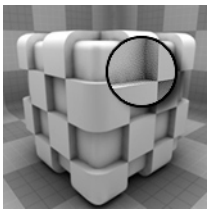
Occlusion Rays: The 'Occlusion Rays' setting specifies the number of semi-random rays modo shoots out from a surface when shading a pixel for Ambient Occlusion. Generally, the higher the number of rays, the more accurate (and smoother) the Occlusion calculations will be, low numbers introduce noise as variations in neighboring pixels become more evident.



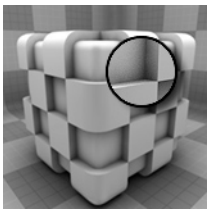
Occlusion Rays 32



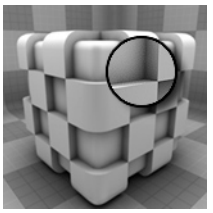
Occlusion Rays 64



Occlusion Rays 128

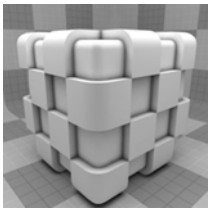


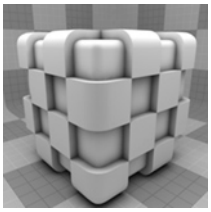
Occlusion Rays 256

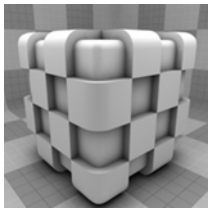


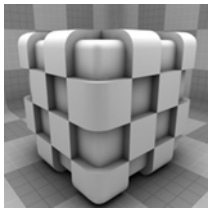
Occlusion Rays 512

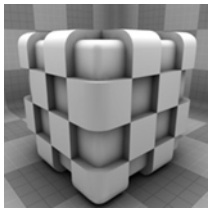
Occlusion Distance: The 'Occlusion Distance' settings specifies the length of the rays that are fired from a surface. If you have a complex object with overlapping elements, limiting the ray range is a good way to control to what degree surfaces are shaded, it also helps to push the shading further into or away from recessed areas.


Distance 5mm

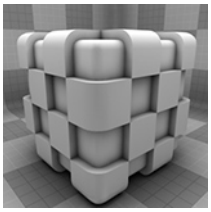

Distance 10mm

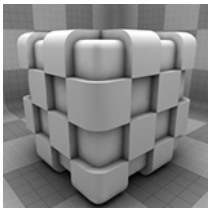

Distance 20mm

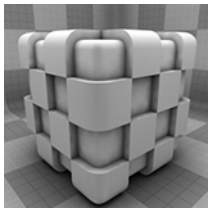

Distance 40mm

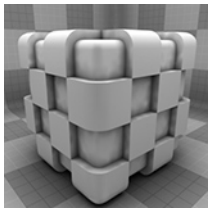

Distance 80mm

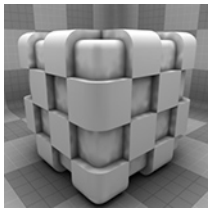
Variance: Adds a perlin noise function modulating the 'Occlusion Distance', meant to vary the strength of the Occlusion effect. Value determines the amplitude of the function where greater values produce stronger, more obvious results (try values above 100%).


Variance 0%

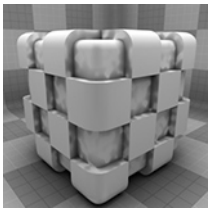

Variance 25%

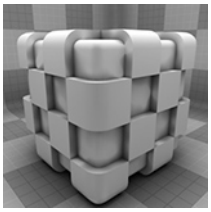

Variance 50%

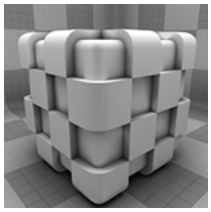

Variance 75%

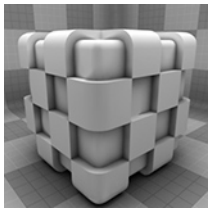

Variance 100%

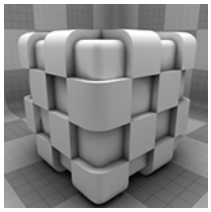
Variance Scale: This value determines the scale (size) of the noise function applied to the variance operation.


Scale 5mm


Scale 10mm


Scale 15mm


Scale 20mm

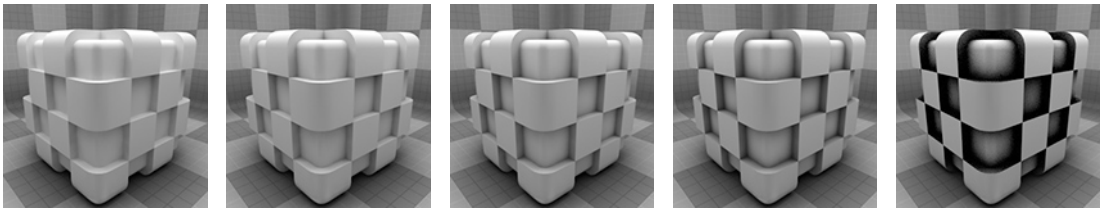

Scale 25mm

Spread Angle: Determines the maximum amount of spread for each shading position in the direction determined by the 'Type' setting. At each shading position, a number of rays (determined by the 'Occlusion Rays' option) are fired in a semi-random pattern, starting in the normal direction from the shading surface and angling out the amount determined here (half each side away from the *normal*), so at 180°'Spread Angle' a full hemisphere of rays would be fired from each shading position. The wider the 'Spread Angle, the more likely a recessed area will be considered *occluded*, also a higher number of rays will be necessary to produce a smooth result.

Max Cavity Angle: Used with only the Concavity/Convexity options, this value determines the angle of difference between adjoining surfaces that determines is surface is 'Concave' or 'Convex'.

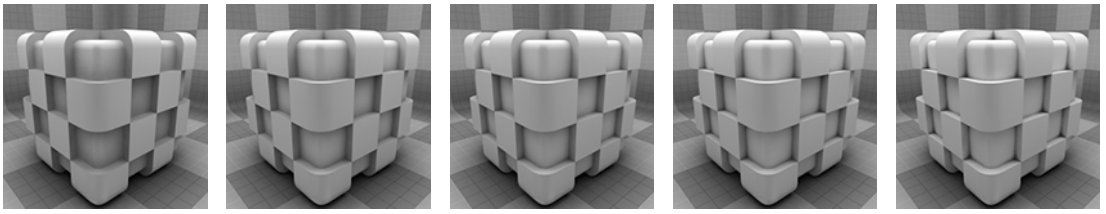
Same Surface Only: When this option is enabled, Occlusion will only be calculated for the surface that the shader is applied to, ignoring all other surrounding polygons. When *disabled* (the default), surrounding polygons *are* taken into account in shading calculations.

Bias: Increasing this value will cause the texture to favor the primary color or value over the secondary whereas decreasing the value causes the secondary color or value to be favored.



Bias 0% Bias 25% Bias 50% Bias 75% Bias 100%

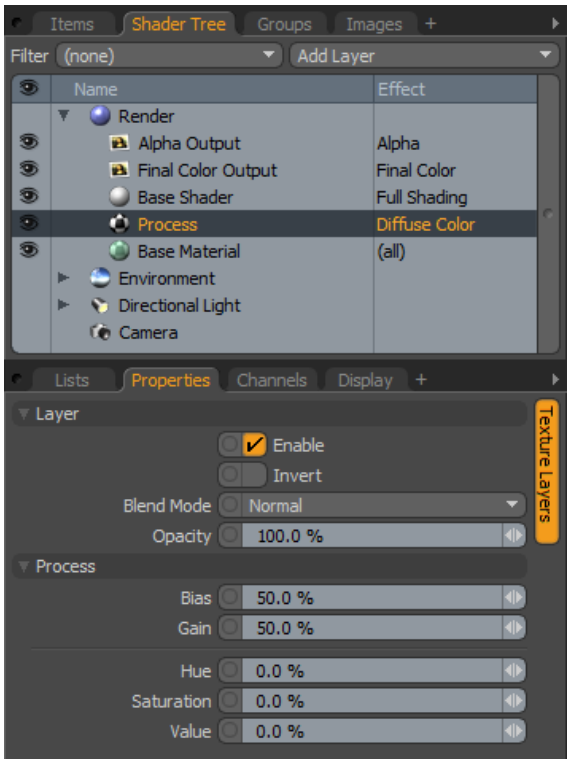
Gain: The Gain setting is similar to a contrast control that effects the falloff of the gradient ramp between the Color/Value 1 and Color/Value 2. Setting the Gain to 100% will create a very sharp contrast whereas setting the value to 0% would reduce the contrast between the two to almost imperceptible amount.



Gain 0% Gain 25% Gain 50% Gain 75% Gain 100%

Process

The process layer offer some basic image adjustment options that affect any texture layer directly below it in the shader tree, including image map, procedural and gradient texture layers.



Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Process--

Bias: The 'Bias' amount, when adjusting a texture layer with 2 colors/values, works identically to the texture layers 'Bias' function where high values will favor secondary colors or value settings, and lower values will favor primary color or value settings. When adjusting full color RGB texture layers, the Bias will adjust the color spectrum based on its luminosity or brightness amount.

Gain: The 'Gain' setting produces similar results to adjusting gamma in a texture layer having its greatest affect on the contrast of the layer. Higher values increase contrast

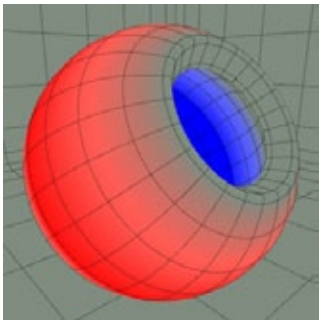
while lower values decrease contrast.

Hue: The 'Hue' setting manipulates the *hue* or *color* of the texture layer.

Saturation: The 'Saturation' setting adjust the colors apparent *saturation* or *brilliance*. A setting of -100% would turn a colored texture layer to shades of gray (grayscale).

Value: The 'Value' setting adjusts the *brightness* or *luminosity* of the texture layer.

Weight Map Texture



The 'Weight Map' texture uses a vertex Weight Map to blend between two values or colors (with alpha). The Value or Color 1 setting will appear anywhere the vertex map value is '0'. The Value or Color 2 setting will appear anywhere the vertex map value is '1'. There will be a blend between the primary and secondary value or color settings where the vertex map moves from 0 to 100. The gradient can be modified by the 'Bias' and 'Gain' controls. This texture is especially useful as a layer or group mask.

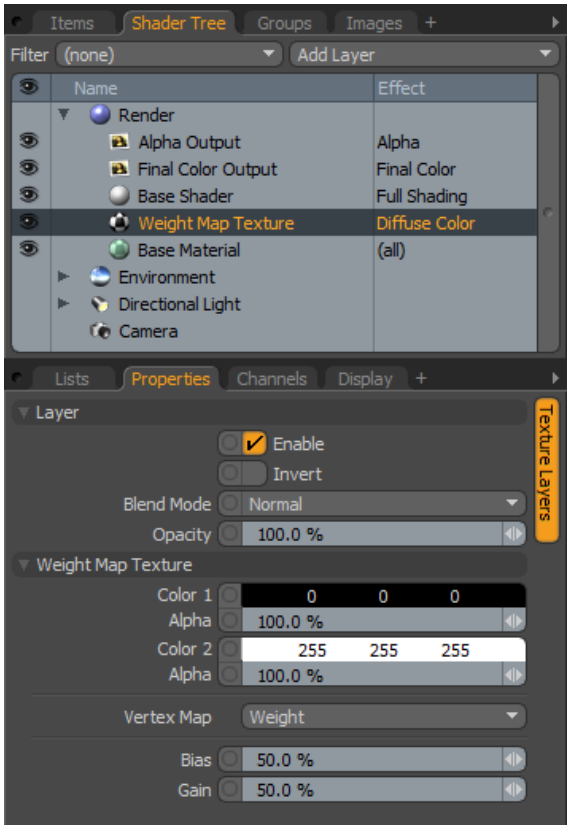
For information regarding adding and working with Shader Tree Items Layers, please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the weight values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.



Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Weight Map Texture--

Color 1: Determines the color of the texture for Weight values of 0, ramping toward Color 2 based on the weight map itself. This setting only applies to surfaces that require a color value. The 'Alpha' value determines the level of transparency for the Color 1 zone.

Value 1: Determines the value of the texture for Weight values of 0, ramping toward Value 2 based on the weight map itself. This control is only active when the texture layers Effect is set to a non-color attribute.

Color 2: Determines the color of the texture for Weight values of '1'. This setting only applies to surfaces that require a color value. The 'Alpha' value determines the level of transparency for the Color 2 zone.

Value 2: Determines the value of the texture for Weight values of 1. This control is only active when the texture layers Effect is set to a non-color attribute.

Vertex Map: Users may use this drop down menu to choose an existing Weight Map or Vertex Color Map.

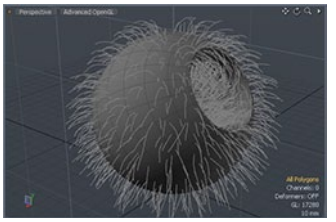
Bias: The bias control will cause the texture to favor either the primary or secondary value or color. By increasing the bias more of the color/value1 will appear while decreasing the number will force more of the color/value 2 into the gradient blend.

Gain: The Gain setting is similar to a contrast control that effects the falloff of the gradient ramp between the Color/Value 1 and Color/Value 2. Setting the Gain to 100% will create a very sharp contrast whereas setting the value to 0% would reduce the contrast between the two to almost imperceptible amount.

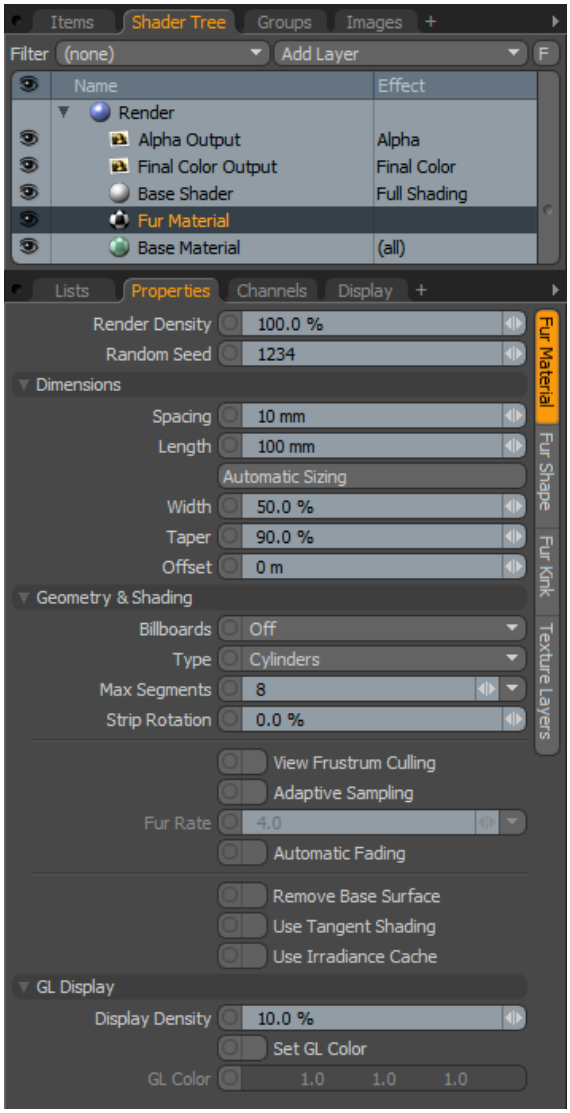
Fur Material

Endless fields of well manicured lawns, small woodland creatures, shag carpeting and werewolves - what do these all have in common? Why of course they are all possible with modo's fur material. This flexible texture layer covers surfaces with all manner of fuzzy, frazzled, fluffy, fibrous fur. The various functions provided should allow for a great range of creative options, including the ability to recreate tame or wild lawns, all manner of fibrous carpeting, little fuzzy animals, maybe even cover a conifer with pine needles. Best part is, modo still renders all of these fibers with amazing speed, even using global illumination.

For information regarding adding and working with Shader Tree Items Layers, please reference the the Shader Tree page of the documentation. Additionally, there is a 'Working with Fur' page that covers additional information regarding the use of fur in modo.



Once a fur layer is added, the first thing users may notice are the lines extending from the object or surface. They may be long or short depending on the objects scale, don't worry though as they can be adjusted later. Users can't select these lines, they're only meant to represent the fibers now growing from the surface, providing a convenient preview of how fibers will look when rendered. As users make changes in the material properties, the guides update in real time in the viewport. There are a number of options available in the properties panel that are broken into two tabs, the 'Fur Material', where you specify the settings for the fibers density, size and shape as well as how they render, and second tab, the 'Fur Shape' controls where you adjust the fibers themselves, if they are curly or straight, wavy or limp. There are also options to control your fur with guides that can be styled with modo's painting tools (covered on the 'Working with Fur' page).



Fur Material

Render Density: The 'Render Density settings allows the user to easily adjust the density or overall number of fibers on the surface, increasing this value will increase the number of fibers creating a thicker matte, while lower values will thin the number of overall fibers. This only affects the number of fibers that are rendered. Keep in mind that increased numbers of fibers will require more memory to calculate and take longer to render.

Random Seed: The 'Random Seed' value is an initial setting used when generating random numbers. Different values will produce different random variations, however you will want to use the same 'Seed' value when it is necessary that items retain the same variations.

Dimensions--

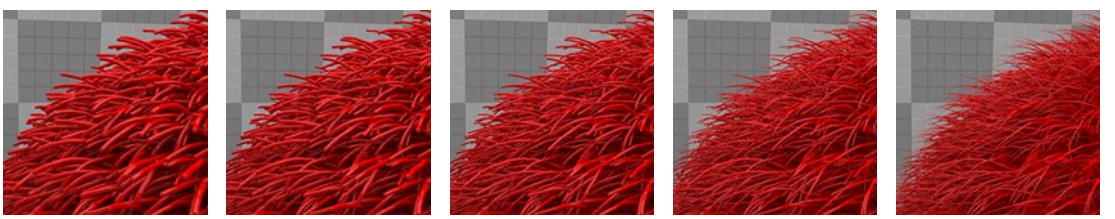
Spacing: The 'Spacing' value sets the average space between individual fibers. Smaller values will result in thicker, more dense fur, while larger values result in thinner fur. This setting works in tandem with the Width percentage in specifying the thickness of the fiber itself.

Length: The 'Length' value sets the overall length of the fibers from their root to their tip.

Automatic Sizing: The 'Automatic Sizing' command will set an initial value for your furs spacing, and length based on the size of the fur surfaces bounding box. Calculated as the spacing of 0.5% percent of the surface bounding box maximum extent, (the maximum length of the bounding box diagonals). The fur length is then 10 times that value. This should only be considered a starting point.

Width: The 'Width' amount sets the width or thickness of the individual fibers at their root. It is a percentage of the Spacing value, So lets say if you have a spacing of 10mm and the Width is set to 10%, the fibers would then be 1 mm wide at the root; 50% would make them 5 mm and so on.

Taper: The 'Taper' value allows the user to narrow the width of individual fibers over their length. A value of 0% would produce fibers that were equal in thickness at the root and tip, while a value 100% would produce tips with no thickness effectively making each fiber a long cone.



Fur Taper 0% Fur Taper 25% Fur Taper 50% Fur Taper 75% Fur Taper 100%

Offset: The 'Offset' value pushes the root of the fibers away from the surface the specified distance. Fibers will retain their overall length.

Geometry and Shading--

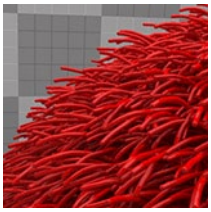
Billboards: The 'Billboards' option provides users an easy way to set up billboard polygons, distributing flat simple polygon planes across a suface. When images are mapped to the planes, it can easily be made to look more complex than their shape would suggest. Generally, the Billboard setting is set to 'Off' and has no effect on the fibers. When set to the other options, their setting override to 'Type', 'Taper' and other settings providing one-click setup of these popular functions.

Off- Disables the billboard option.

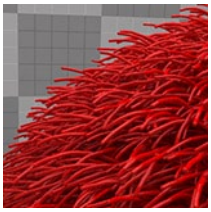
Trees- Replaces each fiber with a flat, upright simple polygon plane that always faces the current render camera.

Leaves- Replaces each fiber with a flat simple polygon plane.

Type: The 'Type' drop down allows the user to specify the method modo uses to generate the fur fibers. 'Cylinders' are like tiny tubes, great for fur and hair, while 'Strips' are flat planes, these are great for making grass among other thing.

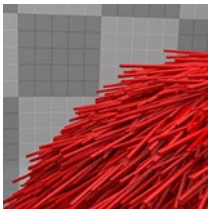


Type Cylinder

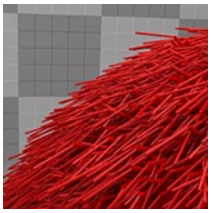


Type Strip

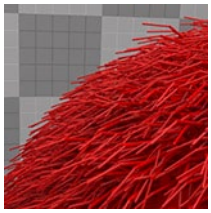
Max Segments: Just like how a curved surface requires more polygons to produce a smooth effect, increasing the 'Max Segments' will produce smoother curves to generated fibers, but setting this value too high can result in an increase in memory usage and processing time required to calculate the fur layer. You can get away with using fewer segments on fibers that bends only slightly, however long or curly hair will likely require more segments to produce a smooth result.



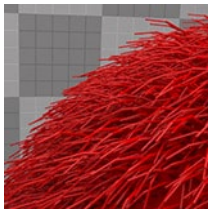
Segments 1



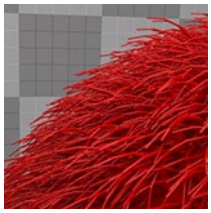
Segments 2



Segments 3

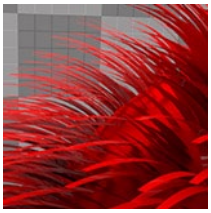


Segments 4

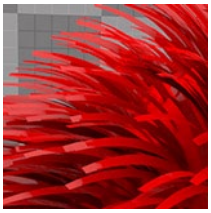


Segments 5

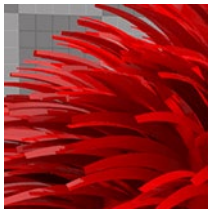
Strip Rotation: This option is only effective when 'Type > Strip' is selected, as the 'Strip Rotation' value will randomly rotate individual fibers at their root. Increasing the value will increase the amount of rotation applied, up to 180 degrees at 100%. You can also change the direction using 'Jitter Direction' in the Fur Shape tab, the difference is that Strip Rotation will retain the same bend direction while Direction Jitter does not.



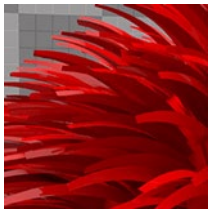
Strip Rotation 0%



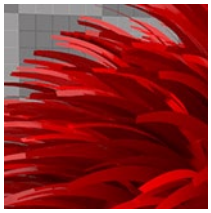
Strip Rotation 25%



Strip Rotation 50%



Strip Rotation 75%



Strip Rotation 100%

View Frustrum Culling: Enabling 'View Frustrum Culling' completely removes fur

outside the view of the camera. Overall this results in faster fur generation time and reduced memory usage. It is especially usefull on large landscapes covered with grass where a great deal of the fibers fall outside of the cameras view.

Adaptive Sampling: Enabling 'Adaptive Sampling' will decrease the density of a fur layer as it recedes away from the camera as based on the 'Fur Rate', reducing the overall memory required to calculate. To avoid popping as fibers disappear, enable 'Automatic Fading'.

Fur Rate: The 'Fur Rate' specifies the threshold that 'Adaptive Sampling' uses to reduce fibers, calculated as an average distance, in pixels, between two fibers. Adaptive Sampling and Fur Rate are best used on scenes like grassy fields or any situation where large amounts of fibers recede into the distance

Automatic Fading: 'Automatic Fading' works with 'Adaptive Sampling' by fading fibers in and out when they are created\destroyed by 'Adaptive Sampling'. This is ideal for animated sequences, the caveat being that render times will increase as semi transparent geometry takes longer to render.

Remove Base Surface: Fur takes it's color and shading information from the material of the surface the fur grows from. Therefore you end up with a green polygon with green fur growing from it. 'Remove Base Shader' will make that base polygon invisible, like setting the Dissolve Amount to 100%. You more than likely want the grass shading to be different than that of the surface it's attached to, That can be accomplished pretty easily too, look for more detail on that in the 'Working with Fur' page.

Use Tangent Shading: When 'Use Tangent Shading' is enabled, the specular shading normal is rotated toward the camera creating highlights that run parallel to the fur direction providing an overall more realistic 'hair' type highlight.

Use Irradiance Cache: While the Irradiance Caching options for rendering are quite fast, when rendering fur they can be come quite slow, so by default modo now uses the the Monte Carlo method of Calculating Global Illumination on just Fur overriding the GI settings of the 'Render' item. In some special cases, users may wish to use Irradiance Caching instead, enable this option to do so.

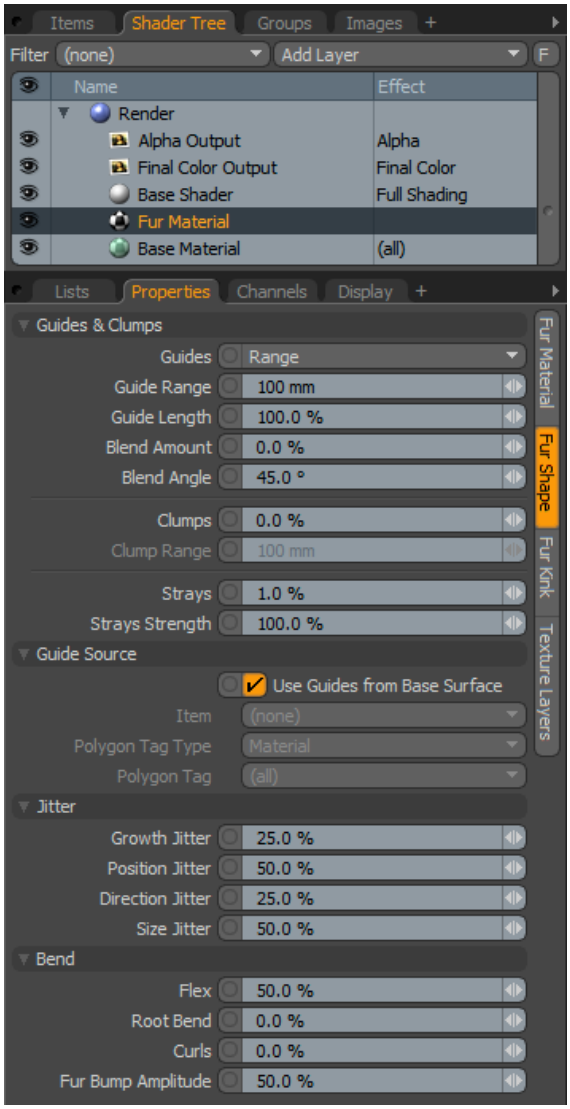
GL Display--

Display Density: In order to speed up the onscreen display, you may wish to view a fewer number of fibers than what will render. The 'Display Density' only affects the number of preview fibers displayed in the 3D viewport.

Set GL Color: When this toggle is enabled, users can use the 'GL Color' option to set the color for the display of the guide hairs visible in the 3D GL viewport. This

option is useful in helping to differentiate fibers from other scene elements visually differentiating them from the rest of the scene.

Fur Shape



Guides and Clumps--

Guides: Guides for Fur can be created a number of ways. There's more on that on the 'Working With Fur' page as well as on the 'Styling Fur' page of the Sculpting and Painting section. The 'Guides' drop down menu allows the user to choose how guides will affect the fur material--

None-- Guides have no affect on the fur texture.

Clump-- Guides will be used for clumping. Based on the 'Clumps' percentage, fiber tips will clump toward the guides.

Direction-- Guides root direction will set the fiber root direction.

Direction+Length-- Guides root direction will set the fiber root direction and the Fur Length setting works with guide length acting as a multiplier to define the final length. In other words, the length parameter still controls the maximum length, but the guides can control overall variations in length; shorted guides equal shorter fur.

Shape-- Fur will follow the overall shape of the guides.

Range-- With Range selected, the guides basically control fur growth directly. Working with Guide Range and Length, and the Blend settings, you can control the fur for long hair, specialized styling and specific fur rendering effects, such as braids.

Guide Range: When the Guide type is set to 'Range', the 'Guide Range' value sets the effective area around the root of the guide where fibers will be generated. When 'Guide Ranges' overlap, fibers will blend fibers between the two guides based on the blend amount. There is also a small amount of falloff around the guide range border.

Guide Length: The 'Guide Length' setting allows the user to adjust the length of fur generated by guides without the need to edit the guides themselves. A value of 100% would equal the overall guide length, while lower values will reduce the length the appropriate percentage amount.

Blend Amount: When the Guide Range setting creates areas of overlapping fibers, the 'Blend Amount' controls to what degree fibers generated between guides will blend between the two guides positions.

Blend Angle: The 'Blend Angle' sets the maximum angle between two adjoining guides that will create a blend of fibers.

Clumps: When the tips of fibers gather in small groups, it is referred to as clumping. With the guide type set to 'Clump', the greater the value set for Clumps, the more the tips of fibers will tend toward the guides. When there are no guides, the 'Clump Range' setting will determine the area of fibers that will clump together based on the 'Clumps' setting. Higher values will bring the tips into tighter groups while lower values will have less of an effect.

Fur Clumps 0% | Fur Clumps 25% | Fur Clumps 50% | Fur Clumps 75% | Fur Clumps 100%

Clump Range: The 'Clump Range' sets the average range or area of fibers that will gather together based on the 'Clumps' setting.

Strays: The 'Strays' option randomizes a user specified percentage of fibers to produce a more realistic, random look to fur generated using guides. Also used on fur with 'Clumping', otherwise this setting produces no results. Users can specify the number of fibers affected as a percentage of all fibers, defining the number of fur fibers that will be affected by the 'Strays Strength' value.

Strays Strength: The Stray Strength values determines the amount of randomization

applied to the fibers chosen as strays defined as a perlin noise function that deviates the fiber along its length away from the guide. In the case of 'Clumps' this value determines the strength a fiber strays away from the clump group.

Guide Source--

Guides meant to control styling (both combing and length) of fur can be specified by polygon tag and by object. This makes it possible to separate the guides from the surface on which hairs are grown, so users can have a mesh with just the guides controlling the hairs growing on another mesh. Users can also tag the guides for each fur layer which should allow more sophisticated hair styling. Tags can be applied to guides by selecting the curves in 'Polygons' mode and applying the tags same as applying a standard polygon tag.

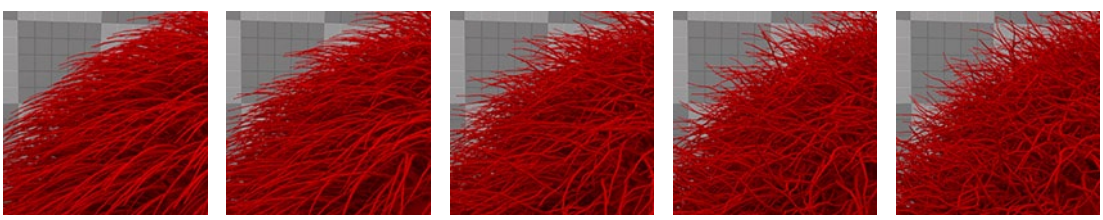
Use Guides From Base Surface: When this option is enabled, modo will use the guides associated with the surface from which the fur is growing (where guides and polygons are sharing a material tag). When disabled, users can specify specifically tagged guides to control the styling the fur layer. When this option is disabled (toggled off), the next three option become enabled.

Item: When 'Use Guides form Base Surface" is disabled, users can define a guide layer by selecting an 'Item' (layer) that contains guides.

Polygon Tag Type: When 'Use Guides form Base Surface" is disabled, users may select guides based on modo's various poygon tags. Working in the same way users can define surface masks based on modo's polygon tags covered in the 'Material Group' items documentation. Creation of tags is covered in the 'Defining Polygon Tags' page. Once a tag type is determined, useres can use the 'Polygon Tag' option to set the particular defined tag name.

Jitter--

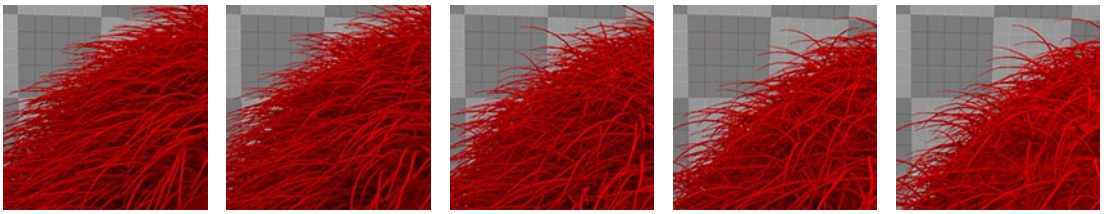
Growth Jitter: The 'Growth Jitter' setting will increase the randomness of a fiber along its length. A setting of 0% will produce straight fibers, while increasing values will produce increasing jagged and random fibers.



Growth Jitter 0% | Growth Jitter 25% | Growth Jitter 50% | Growth Jitter 75% | Growth Jitter 100%

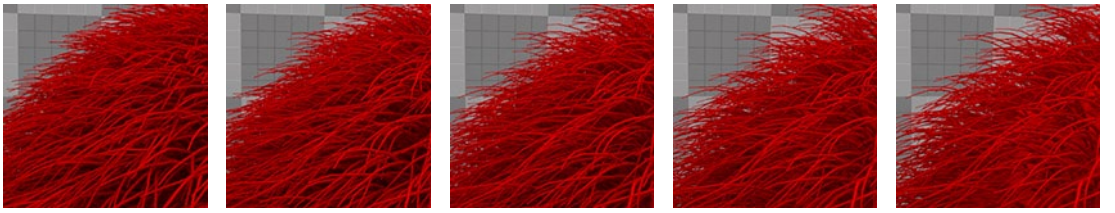
Position Jitter: The 'Position Jitter' setting will increase the randomness of the root positions of fibers. When creating multi-layered fur, subsequent fiber layer will grow from identical positions. If this is not the intended result, you can increase the "Position Jitter' to eliminate the overlap.

Direction Jitter: The 'Direction Jitter' setting will randomize the angle of rotation for the root of a fiber. Similar to 'Strip Rotation', however 'Direction Jitter' will also randomize a fibers bend direction. A setting of 0% will produce fibers in a uniform direction, values above 0% will increase the randomness of rotations for each fiber.



Direction Jitter 0% | Direction Jitter 25% | Direction Jitter 50% | Direction Jitter 75% | Direction Jitter 100%

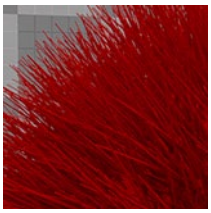
Size Jitter: The 'Size Jitter' setting will randomize the overall scale of each fiber. A setting of 0% will produce fibers of equal size, values above 0% will increase the the random scaling of fibers.

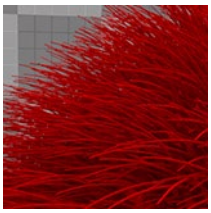


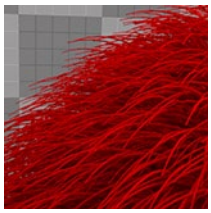
Size Jitter 0% | Size Jitter 25% | Size Jitter 50% | Size Jitter 75% | Size Jitter 100%

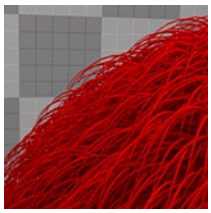
Bend--

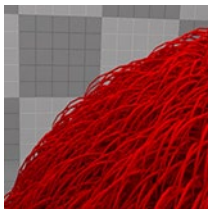
Flex: The 'Flex' setting adds bending or drooping to individual fibers as if affected by gravity. For fibers growing straight up, like grass, a value of 0% would produce perfectly straight fibers, while a setting of 100% would produce a fiber tip that is bent a full 180° from its root. Fibers growing downward would be less affected by this setting. In order to produce smooth bend results, it may be necessary to increase the number of 'Segments'.


Flex 0%

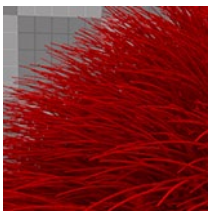

Flex 25%

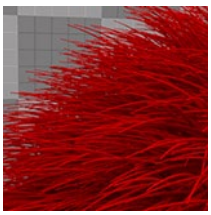

Flex 50%

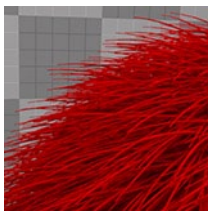

Flex 75%

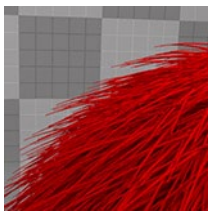

Flex 100%

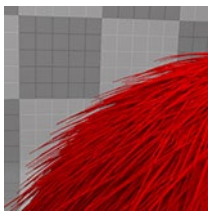
Root Bend: With a "Root Bend' value of 0%, the root direction for a fiber is based on the direction of the corresponding smoothed polygon normal, simply stated this means whatever direction a polygon is facing, the fiber will grow from that surface in the same direction. Increasing this value will rotate the root, altering the growth direction for the fiber. A setting of 100% would produce a fiber that grows perpendicular to the normal, a rotation of 90 degrees.


Root Bend 0%

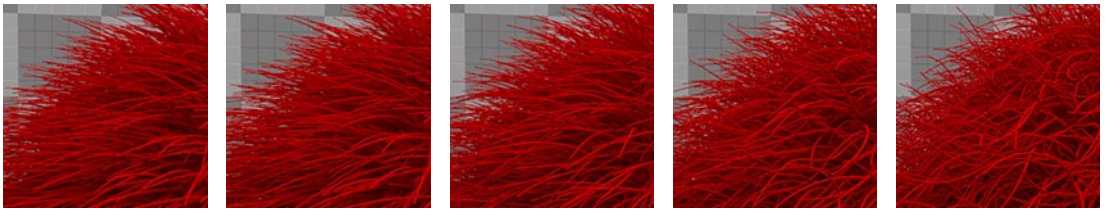

Root Bend 25%


Root Bend 50%


Root Bend 75%


Root Bend 100%

Curls: Increasing the 'Curls' setting will add corkscrew type swirl to fibers. Small values will appear as if fiber are slightly wavy, while greater values will increase the effect creating tight overlapping swirls.



Curls 0% Curls 25% Curls 50% Curls 75% Curls 100%

Fur Bump Amplitude: The 'Fur Bump Amplitude' setting controls the strength of influence a texture layer assigned to the layer effect 'Fur Bump' has over the 'Root Bend' and 'Flex' direction of the fur, effectively controlling the growth direction. Similar in function to the Fur Direction controller used for combing fur, however a Fur Bump layer can be an arbitrary image map or procedural texture, controlling the direction much like a bump map perturbs the surface rays simulating bumpy surfaces.

Fur Kink

Kink--

Based on a noise modifier, the Kink effect will apply a chaotic twisting to the individual fibers, making them look what would be described as 'Tangled'. The detail introduced is dependant on the number of segments in the fiber, higher segments will produce more detailed results, but also require additional memory and time to render.

Root Kink: The 'Root Kink' value determines the chaotic twisting to the fiber fading from the Tip and increasing in strength toward the Root. Being at the base this will affect the fibers direction, much like jitter does. (Samples Below)

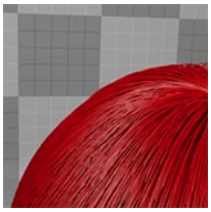
Tip Kink: The 'Tip Kink' value determines the chaotic twisting to the fiber fading from the Root and increasing in strength toward the Tip. (Samples Below)

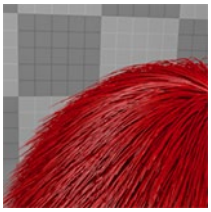
Kink Scale: The 'Kink Scale' value determines the size of the noise that produces the kinks. Smaller values will have a more dramatic tangled look, and larger values will look less tangled.

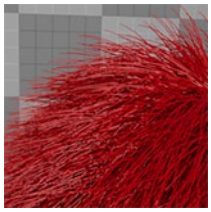
Kink Length Scale: The 'Kink Length Scale' determines the scale of the Kink along the length of the individual fiber, large values can be made to look more like waves.

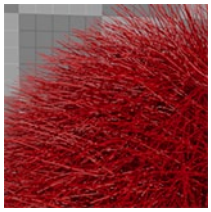
Kink Offset: The 'Kink Offset' value can be used to animate the noise, for example to simulate a grass field blowing in the wind.

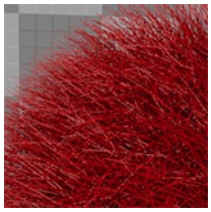
Root Kink--


Root Kink 0%

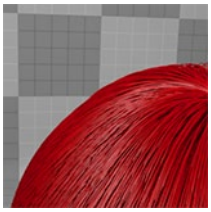

Root Kink 25%

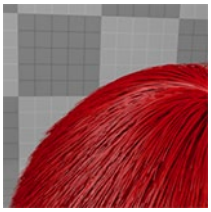

Root Kink 50%

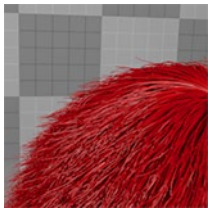

Root Kink 75%

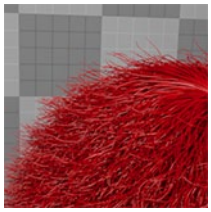

Root Kink 100%

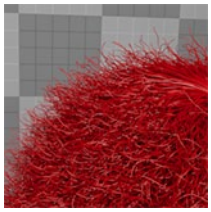
Tip Kink--


Tip Kink 0%


Tip Kink 25%


Tip Kink 50%

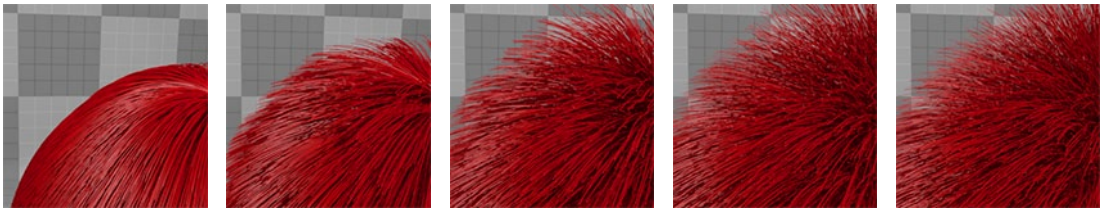

Tip Kink 75%


Tip Kink 100%

Frizz--

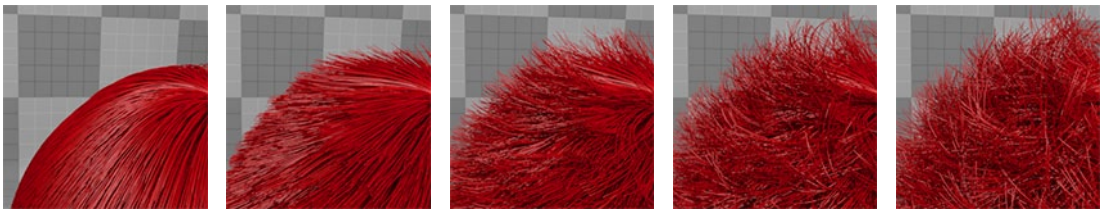
The Frizz effect will apply a somewhat random waviness to the individual fibers, making them look what would be described as 'Messy'. The detail introduced is dependant on the number of segments in the fiber, higher segments will produce more detailed results, but also require additional memory and time to render.

Root Frizz: The 'Root Frizz' value determines the amount of random waviness to the fiber fading from the Tip and increasing in strength toward the Root. Being at the base this will affect the fibers direction, much like jitter does.



Root Frizz 0% | Root Frizz 25% | Root Frizz 50% | Root Frizz 75% | Root Frizz 100%

Tip Frizz: The 'Tip Frizz' value determines the amount of random waviness to the fiber fading from the Root and increasing in strength toward the Tip.

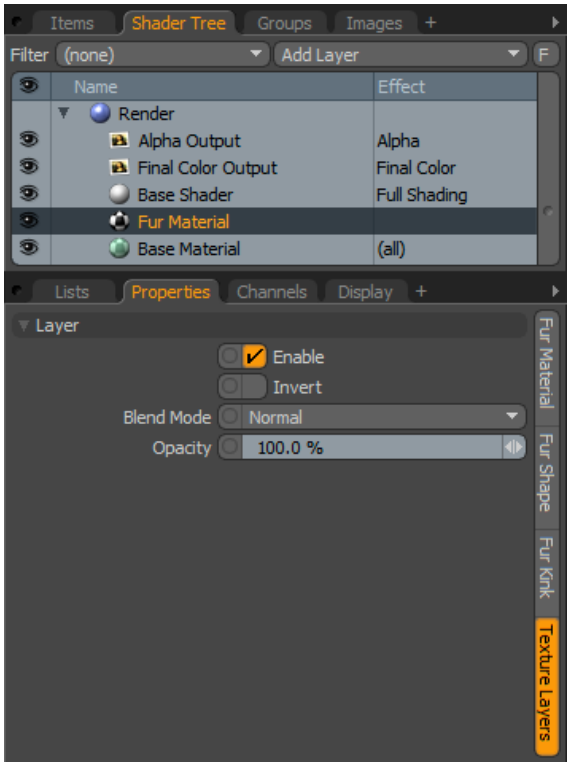


Tip Frizz 0% | Tip Frizz 25% | Tip Frizz 50% | Tip Frizz 75% | Tip Frizz 100%

Frizz Scale: The 'Frizz Scale' value determines the size of the waviness applied to the fibers. Smaller values will produce finer waves, while larger value will increase the size of the waves.

Frizz Offset: The 'Frizz Offset' value can be used to animate the noise, for example to simulate a grass field blowing in the wind.

Texture Layers



Layer--

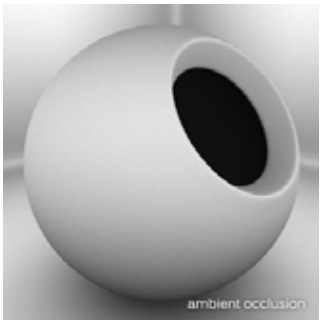
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Render Outputs



Samples demonstrating each render output, including detailed descriptions of each are available on the secondary Render Outputs page of the documentation listed under the Rendering section.

modo offers many different 'Render Outputs' useful in a variety of situations ranging from purely diagnostic, to having specific compositing utility in an external application. A default scene automatically includes a 'Final Color' and 'Alpha' output which should be fine for most situations. Additional layers can be added from within the Shader Tree viewport itself, by LMB-clicking the 'Add Layer' option of the full viewport window and selecting 'Special > Render Output' from the pop-up menu. This adds an additional layer to the top of the tree. Users may then RMB-click in the 'Effect' column of the shader tree for a pop-up context menu that allows one to select alternate output types. Render Outputs can be enabled either via the layer 'Enable' checkbox within the properties viewport, or by clicking the eyeball icon in the visibility column of the Shader Tree. All enabled outputs are rendered simultaneously each time the 'Render' or 'Render Animation' commands are invoked. Once rendered the individual outputs appear in the Render Display window and are accessible from the layer menu therein.

Selecting any of the 'Render Outputs' within the 'Shader Tree' makes their editable attributes available within the 'Properties' viewport. Selected layers are indicated by the dark bar highlighting the layer name. All Render Outputs share the same set of attributes, with 'Depth', 'Ambient Occlusion' and 'Final Color' having additional option covered below.

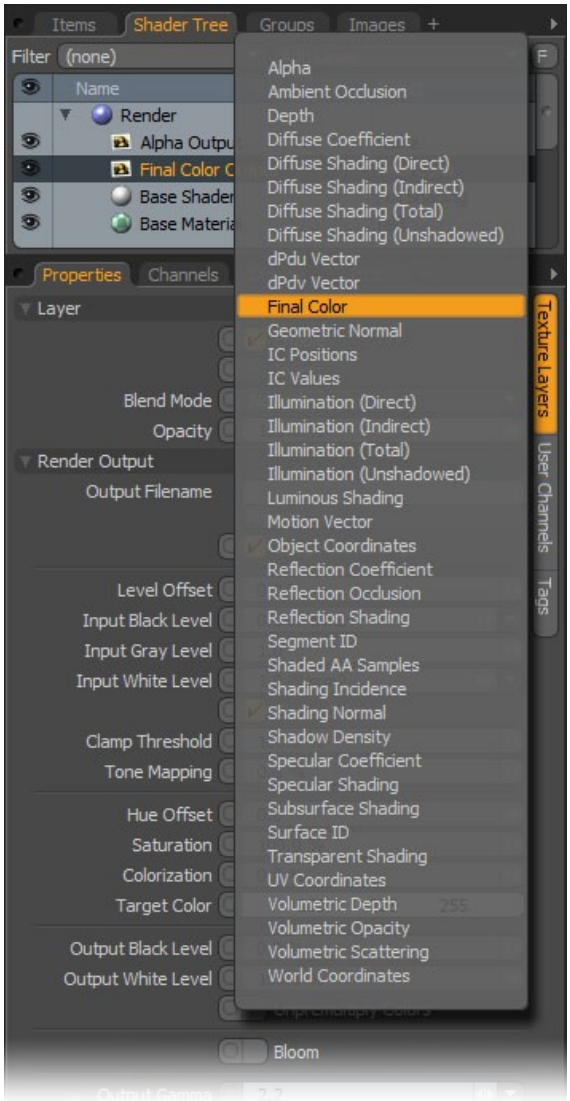
Layer--

Enable: The 'Layer Enable' checkbox allows the user to temporarily enable/disable a layer without losing stored values. Disabled layers are saved with the scene file.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing

user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.



Opacity: Disabled for Render Outputs.

Render Output--

Output Filename: This function allows the user to specify a filename and location for the saving of a rendered animation. Click the 'Browse' button to open the 'File Save' dialog box. Specify a name and location for the rendered output. Images will be saved the next time the user invokes the 'Render Animation' command from the menu bar. Additional automatic output naming options can be specified in the 'Frame' settings of the Render item as well. Note that Rendering scenes using the 'Render' command or 'F9' do not save the image automatically to disc, but merely displays the rendered result in the 'Render Display' window, for that use the 'Save As' option of the Render Dispaly window.

Image Processing: Users can toggle this option to control the post processing applied to the rendered image. When enabled post processing functions will apply to the rendered output, when disabled post processing settings will be disregarded. Post processing can be a powerful way to adjust the final rendered image, this options makes it easy for users to quickly see to what degree the image processing is contributing to the final render without disabling all the settings individually.

Level Offset: The 'Level Offset' option slides all color values up or down the value scale, shifting them identical amounts in the rendered image. For example, if an image

had a 0% black pixel and a 50% gray pixel, offsetting the values 50% would result in the black pixel changing to 50% gray and the gray pixel changing to 100% white with all other values changing corresponding amounts.

Input Black Level: The 'Input Black Level' control specifies the radiance or luminance level that corresponds to pure black in the final rendered image (a pixel color value of o.0). To use the input level controls effectively, 'Clamp Colors' must be disabled.

Input Gray Level: The 'Input Gray Level' is a non-linear luminance adjustment that applies a curve-like function to the pixels, modifying the mid-range the most while attenuating the adjustment amount for values moving toward the defined 'Black' and 'White' points. Values above 1.0 will lighten the midrange while values below 1.0 will darken the midrange.

Input White Level: The 'Input White Level' control specifies the radiance or luminance level that corresponds to pure white in the final rendered image (a pixel color value of 1.0). To use the input level controls effectively, 'Clamp Colors' must be disabled.

Clamp Colors: The "Clamp Colors" option will suppress all color values in a rendered image to truncated 24 bit values prior to anti-aliasing producing cleaner result in areas of extreme contrast. However, when this setting is enabled, you will lose any extended dynamic range present in the image (making proper tone mapping impossible). If you intend to save your image in an high dynamic range format, leave the setting disabled. Additionally, instance where 'Tone Mapping' may not be appropriate, any apparent blockiness can be minimized by adding a small amount of 'Bloom', available as an option of the 'Final Color' output. When enabled, a small amount of dithering is added to reduce or eliminate banding in gradations.

Clamp Threshold: Determines the threshold within the image where clamping takes place.

Tone Mapping: Dynamic Range is measured as the difference between the darkest shadow to the brightest highlight in an image. Most image formats dynamic range pale in comparison to that of the world around us. Much in the same way that a photographer struggles to capture all of the range our eyes can see using a camera, you may find your renders have shadow areas that are too dark, and highlight areas that are too bright. Luckily, modo renders in full floating point accuracy, providing dynamic range well beyond what your monitor is capable of displaying. This allows users to modify the overall tonal balance of the rendered image in several ways, including 'Tone Mapping', a technique that is used to compress the rendered dynamic range into something viewable; shadows will open revealing previously obscured details as will highlights. To use the 'Tone Mapping' function, 'Clamp Colors' must be disabled. You will often find it takes a combination of settings for White Level, Tone

Mapping and Gamma to achieve the best results, please reference the 'Tone Mapping' page of the documentation for further examples.

Hue Offset: The 'Hue Offset' option will adjust the color values of the rendered image independently of the luminosity or brightness values, shifting them across the entire spectrum in a sequential fashion, for example when shifting a red color, moving the values in a positive fashion will adjust reds toward an orange hue, then yellow and so on. The hue values should be thought of like a wheel, where a rotation of 180° inverts all the color values and a rotation of 360° brings them back to their initial state.

Saturation: The 'Saturation' options controls the concentration or amount of color in an image independent of the luminosity of brightness. At 100%, colors will be fully saturated as defined in the items material settings, reducing the value will reduce the overall color saturation down to a value of 0% producing an image only with gray shades. Since modo renders in full 32bit floating point accuracy, values above 100% will increase saturation without introducing the color banding and artifacts generally associated with oversaturating images.

Colorization: Users can use the 'Colorization' options to introduce an overall color tint into an image. This can be purely for artistic reasons but can also be helpful in reversing color casts introduced by Image Based Lighting among other things. Users need only to define a color with which to tint by setting a 'Target Color' and then adjusting the 'Colorization' amount determining the strength of the color into the image. A value of 0% produces no colorization attenuating toward 100% where the image is replaced fully by the target color.

Target Color: This value determines the color with which to tint the rendered image when the Colorization option is above 0%.

Output Black Level: This value determines the output value of what is considered fully black in the image. Adjusting the value upwards changes the Black point to a shade of gray. As values are adjust, intermediate values between black and white are adjusted as well.

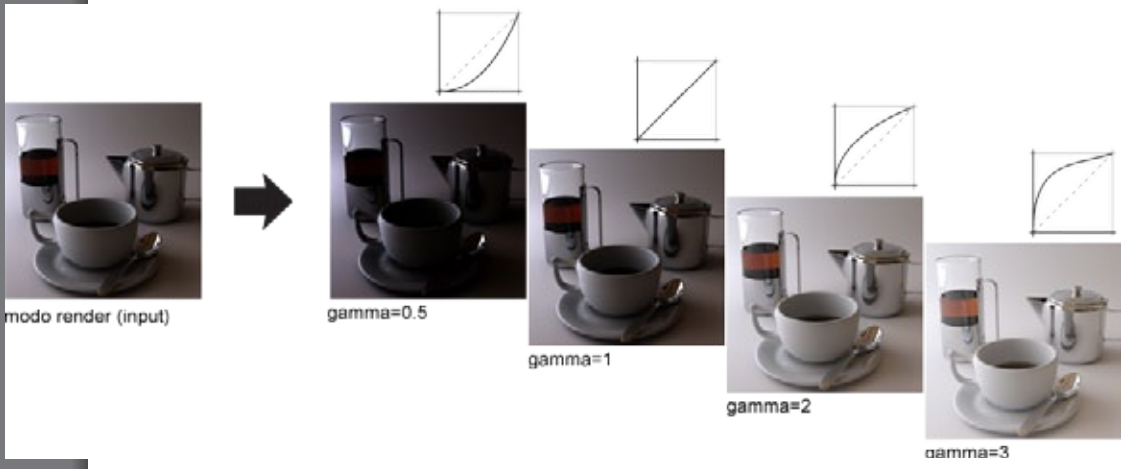
Output White Level: This value determines the output value of what is considered fully white in the image. Adjusting the value downwards changes the White point to a shade of gray. As values are adjust, intermediate values between white and black are adjusted as well.

Unpremultiply Colors: When generating pixel values in rendered images, modo 'multiplies' the rendered result with the background color to calculate the final result. This 'multiplied' final result is fine for most cases, however, when using alpha channels to composite multiple images together, partially transparent pixels will often exhibit

halos, an artifact of the previous background color. 'Unpremultiply' basically reverses this effect. Activating this option won't reveal any difference in modo's render display, you'll need to save the image and open it in an another application to see the result. This function requires that the object/scene is rendered over a solid black background with a corresponding alpha channel. This can be easily achieved by disabling the 'Visible to Camera' option in the 'Environment' Item (and the 'Visible to Refraction Rays' option in cases where there are transparent surface, note that this will affect how the render looks).

Output Gamma: Gamma is the measure of total contrast in an image and represents the relationship of an image input to an image output. In this case, the 'input' is modo's rendered image; what modo is holding in its memory and the 'output' is either the image being displayed in the 'Render Display' viewport (what you see) or the image that is saved. A gamma value of 1 is said to be 'linear', what modo rendered, is identical to what is displayed. Modifying the 'Gamma' value changes how the image is represented. Values higher than 1 lighten the image, while values lower than 1 darken it. Adjustments to the gamma value are not linear, i.e. they don't change all the pixels in an image the same amount, explaining why it is often referred to as a gamma curve. The default Gamma value is derived from the global preference found in the Preference Editor under 'Render'. Adjusting this Render Output 'Gamma' value overrides the global preference on a per item layer basis.

To separate the gamma of the 'Render Display' window from that of the saved file, open the Preferences window from the menu bar 'System > Preferences' and select the 'Rendering' options. By activating "Independent Display Gamma" you can set a gamma correction that will only apply to the image as it is viewed in the Render window. The saved image data will always use the Output Gamma value found on the Render Output in the shader tree.



Gamma is a complex matter in rendering, especially when maintaining consistent colors across a variety of applications as an image travels through a projects pipeline. Comprehensive coverage of the topic is beyond the scope of this document, but there have been numerous helpful threads at the Luxology Forum that cover the topic well. An internet search is certain to provide additional useful results.

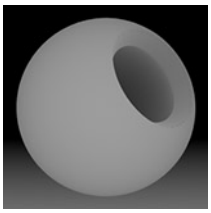
TIP: Since a render output is just another layer, it can reside anywhere in the shader tree. Typically they are at the top directly under 'Render' item, as this is where they see the result of all layers in the tree, but if a 'Render Output' is placed inside of a material group, the render output will only 'render' what it sees within that group, this makes it possible to render alpha channels for specific items based on a polygon tag, instead of making some item specific shader or hiding items in the item list. All active 'Render Outputs' will render each time the render command is invoked, and will appear within the layer menu of the Render Display viewport.

Depth

The 'Depth' output generates a Z-Buffer that represents the distance from the camera to the geometric surface in shades of gray. Rays begin white at the ray origin (camera position) and attenuate darker as they get longer becoming fully black at the specified 'Maximum Depth' value. Useful for calculating Depth of Field in an external application or for composing a fog effect into a rendered scene.

Maximum Depth: The Maximum Depth value sets a "clipping plane" for the black point so that modo knows at what distance to shade the Z-buffer pure black.

Depth 100mm


Depth 200mm

Depth 300mm

Depth 400mm

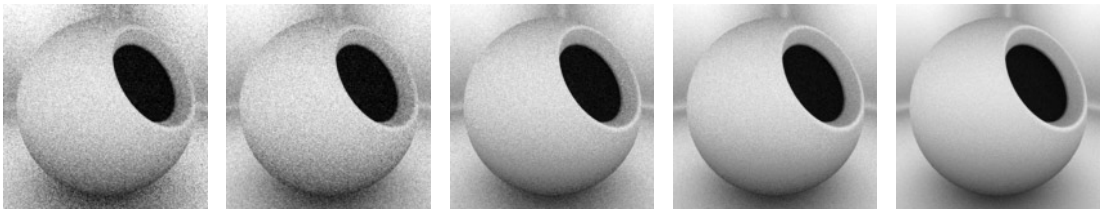
Depth 500mm

Ambient Occlusion

The 'Ambient Occlusion' output generates an occlusion map where occluded areas; crevices and other inaccessible areas of a scene shade darker, while open areas render lighter colors, approximating an overall global illumination type of shading. Grainy renders can be smoothed out by increasing the 'Occlusion Rays' value, controlling the density of the occluded areas is done through the 'Occlusion Range'.

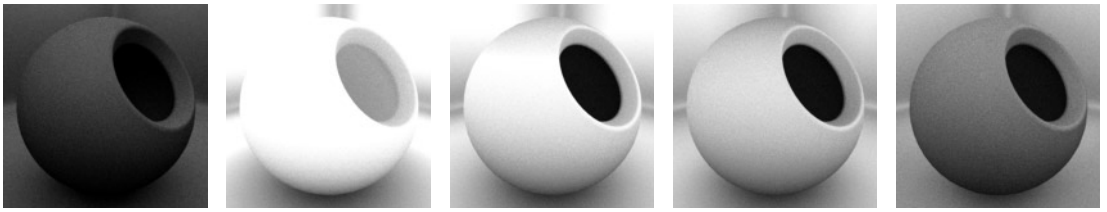
Occlusion Rays: The 'Occlusion Rays' setting specifies the number of random rays modo shoots out from a surface when shading a pixel for Ambient Occlusion. Generally,

the higher the number of rays, the more accurate (and smoother) the AO image will be, low numbers introduce noise as variations in neighboring pixels become more evident.



Occlusion Rays 32 | Occlusion Rays 64 | Occlusion Rays 128 | Occlusion Rays 256 | Occlusion Rays 512

Occlusion Range: The 'Occlusion Range' settings specifies the length of the rays that are fired from a surface. If you have a complex object with overlapping elements, limiting the ray range is a good way to control what recessed areas are shaded darker, it also helps to push the shading further into recessed areas. A value of 0m disables the range defaulting the value to the maximum length of the scenes overall bounding box size.



Range 0m (disabled) | Range 50mm | Range 100mm | Range 150mm | Range 200mm

Final Color

The 'Final Color' output represents the settings of the shader tree, leveraging all visible material and texture layers as well as shaders and the render item settings.

Bloom: In photography, when a very bright part of an image neighbors a very dark part, the bright part will appear to glow, this phenomenon is known as Bloom. modo's 'Bloom' simulates this effect by removing excess energy from a pixel and distributes it around to its neighbors. This checkbox enables the effect, once checked, two additional options become available-

Bloom Threshold: The 'Bloom Threshold' sets the lower threshold to which pixel are effected by 'Bloom'. When set to 100%, only the very hottest specular highlights will bloom, where at 0% every pixel in the scene will receive the bloom effect.

Bloom Threshold 0%

Bloom Threshold 25%

Bloom Threshold 50%

Bloom Threshold 75%

Bloom Threshold 100%

Bloom Radius: The 'Bloom Radius' sets the distance that energy is distributed, in effect controlling the size of the glow. 2% is the default value. Larger radius' impart and ethereal look to the rendered image.

Bloom Radius 1%

Bloom Radius 2%

Bloom Radius 4%

Bloom Radius 8%

Bloom Radius 16%

Contours: modo can render a contour outline around specified boundaries. The options available are-

None- The default value, renders no contouring outline.

Surface Boundaries- Contour line renders around all surface boundary definitions (modo Material tag boundaries).

Segment Boundaries- Contour line renders around all segment boundary definitions (modo Segment ID boundaries, which are not user definable).

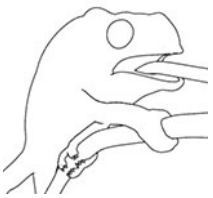
Contour Color: Defines the color of the contour outline using the standard modo color controls for specifying an R, G and B value.

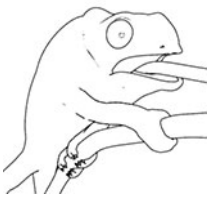
Contour Width: Defines the thickness of the contour line applied to the render output, defined as a pixels, so a value of 1 would render a line 1 pixel wide over all the boundaries.

Fade with Distance: Users can fade the application of the contour outline when enabling this option. Once enabled, an additional option 'Fade Start Distance' determines where the fading effect will start.

Crease Threshold: Determines the angle at which modo draws a contour line. Lower angles generate more lines.

Crease Threshold 180°


Crease Threshold 90°


Crease Threshold 45°

Crease Threshold 15°

Crease Threshold 5°

Shader

Each new scene, by default, has a single 'Base Shader' item. Shaders read the information from the Shader Tree and compute the effect of the illumination in a scene. They also control the rate that pixels are evaluated before being sent to the Render item. Additionally, they control visibility of surfaces in a scene and what rays are allowed to see the surfaces. In most cases, user won't ever touch the settings in the Base Shader, however, in cases where specific surfaces need additional control, such as limiting Monte Carlo shading to a Fur surface, or hiding lights from illuminating a surface, or even hiding an entire object from the Camera, additional Shader items will need to be added with appropriate settings applied. Due to the layered approach of the shader tree, the additional shaders will need to be placed above the Base Shader in the tree within their own material group mask (otherwise the settings of the base shader will simply override them).

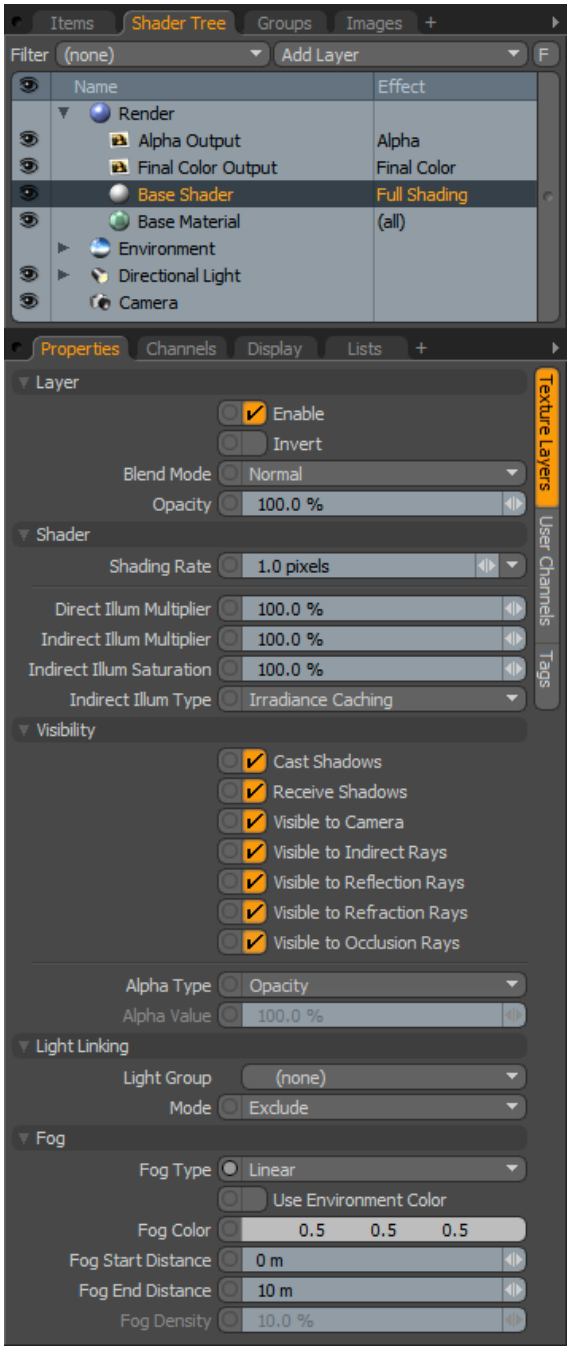
For information regarding adding and working with Shader Tree Items Layers, please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the illumination of the scene/layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.



Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Shader--

Shading Rate: The Shading Rate (also known as the "Coarse Shading Rate") determines how finely to shade each pixel in the materials feeding the selected Shader Item during initial shading. The default value of 1 indicates that the pixel will be evaluated once before being evaluated for further refinement (see Refinement Threshold). Decreasing the Shading Rate results in a more finely evaluated pixel which generally yields longer render times. For instance, if you change the value to .5 the pixel is effectively cut in half horizontally and vertically and each "sub pixel" is evaluated. In this case you have reduced the rate by half but increased the amount of evaluation by 4 times. Keep this in mind as small changes to the shading rate can yield large changes to performance.

Generally the Shading Rate can be left at one and further refinement can be achieved via the Antialiasing level or the Refinement Shading settings. However, in the cases of aliasing seen in reflections or refractions or on procedural textures it is possible to rectify them via the Shading Rate setting. Since this will effect all pixels evaluated it is a fairly 'brute force' solution. Therefore, you should manually restrict the pixels that are effected by the Shading rate by creating Shaders for specific problem areas via Masks

and only decrease the Shading Rates where absolutely necessary.

Direct Illumination Multiplier: Direct Illumination is light coming from traditional 3D light sources such as Directional, Spot, Area, Distant, Cylinder, Dome, and Point. At the Shader item level it is possible to globally affect the direct light intensity via the Direct Illumination Multiplier. By default the value is set to 100%. If we imagine a project with three lights of varying intensity such as 100%, 70% and 30%, setting the 'Direct Illumination Multiplier' to 50% would drive these values down to 50%. 35% and 15% respectively. This is often a handy method for tweaking overall light effects as well as balancing Direct and Indirect light contributions.

Indirect Illumination Multiplier: Indirect Light comes from global illumination sources such as luminous geometry, image based lighting via environments and simple bounced light from diffuse surfaces. The Indirect Illumination Multiplier provides a global method for controlling the Indirect light contribution to the final render. This is very useful when Indirect Light sources are too subtle/pronounced or for simply balancing Direct and Indirect Light contributions.

Indirect Illumination Saturation: One of the many benefits of indirect illumination is the color bleeding between surfaces. As rays take into account not only the color of the lights, and the environment, but the surfaces color as well. The 'Indirect Illumination Saturation' setting allows the user to control the strength of the colors generated by indirect illumination independent of the illumination itself. Increasing the value above 100% will make the colors stronger and more saturated, and values below 100 will decrease the effect.

Indirect illumination Type: Generally, indirect illumination is controlled globally through the Render Properties Global Illumination tab. In some instances, the user may want to control the type of global illumination per surface by using a shader with an associated mask. This gives the user the option of using Irradiance Caching for say an exterior scene, but on complex surfaces such as the trees and bushes, the monte carlo method would provide better results. The user could create a selection set for just the trees and bushes (Select > Assign Selection Set), and in the shader tree, add an additional material 'Group' item above the base shader. In the group properties panel set Tag Type to 'Selection Set' and highlight the named set under 'Polygon Tag'. In this group add its own 'Shader' and set the properties 'Indirect Illumination Type' to 'Monte Carlo'. Now most of the scene will use irradiance caching while only the trees and bushes would use Monte Carlo. The Indirect Illumination type will draw its values from the Render Properties Global illumination tab. The number of rays used for the Monte Carlo method is indicated by the 'Indirect Rays' value of the aforementioned properties tab of the Render item.

Visibility--

There are several toggle options to control visibility of various attributes of the elements feeding the shader. For instance, with these controls you could create a bright luminous polygon that is visible to Indirect Illumination but invisible to reflections, ray traced shadows, reflection and the camera. This would create a nice virtual light element that can be placed anywhere in the scene for illumination purposes without worrying about it effecting any other visual aspect of the project. The options include:

Cast Shadows: Disabling this toggle will cause all elements controlled by the shader to be ignored by shadow rays so they do not create any shadows in the scene. Light will simply pass through the elements. This will not disable shadow effects from indirect illumination as this only affects direct light sources.

Receive Shadows: Disabling this toggle causes the elements to stop receiving ray traced shadows from other geometry in the project.

Visible to Camera: Disabling this toggle will simply hide any item or element controlled by the shader from the camera. These elements may still appear in shadows, reflections and refractions even when hidden to camera.

Visible to Indirect Rays: Indirect Light is any light calculated during Global Illumination such as luminous geometry, Images used as light sources and other light bounced from geometry. Toggling this option off will remove the elements controlled by the shader from Indirect Light calculations.

Visible to Reflection Rays: Disabling this value will remove the elements controlled by the shader from reflecting in mirror surfaces.

Visible to Refraction Rays: Disabling this value will cause the elements controlled by the shader to be invisible when looking at them through refractive surfaces.

Visible to Occlusion Rays: Disabling this value will cause the elements controlled by the shader to be invisible to Occlusion Ray calculations.

Alpha Type: The 'Alpha Type' selector allows the user to choose how surfaces contribute to a rendered alpha channel. When set to 'Opacity' the transparency of each individual surface determines the alpha value. 'Constant' allows the user to specify a fixed alpha value for all associated surfaces. 'Shadow Catcher' is a slightly different function, as it makes all associated surface attributes invisible, exclusively rendering shadows cast by direct and indirect illumination. This is a great tool for composing rendered images with background photos right in modo. A ground plane can 'catch' the shadows cast by other object, without contributing any shadows of its own. When a

photo is specified as a background, the objects and shadows all render together in one pass.

Alpha Value: When 'Constant' is selected as the alpha type, the 'Alpha Value' overrides surface opacity settings and determines to what degree surfaces contribute to a rendered alpha channel.

Light Linking--

Light Group: Light linking allows users to control, on a shader level, what lights in a scene effect what surfaces. The 'Light Group' selector allows the user to specify which group of lights to control. Light groups can be easily made in the 'Groups' palette. Click 'new group' to create a base group, rename it if you wish, and add items by selecting the lights in the 3D viewport, hold down shift to select multiple items. Now right click on the group title and select 'Add Items' from the drop down menu. Users can also drag and drop items directly from the 3D viewport onto any group item to add them.

Mode: The 'Mode' selector allows the user to specify how the lights affect surfaces, If set to 'Include', the lights will affect the shaded surfaces and set to 'Exclude, the lights will not.

Fog--

The Fog options found on the Shader Item allow a simulated fog effect by applying the fog color to relevant materials with increased intensity as the geometry is further from the camera. This is also a great option to fade objects into a background increasing the apparent depth of an image.

Fog Type: The Fog Type popup allows you to choose the fog types options.

None- Disables Fog from rendering.

Linear- When selected, Fog will render in the scene based on the Start and End distances, where there is no fog up to the starting distance, then surfaces will attenuate toward the 'Fog Color' and maximum 'Fog Density' at the end distance.

Exponential- When selected, Fog will render starting at the cameras position, increasing in strength the further a surface is from the camera. Based on exponential values, the strength will increase non-linearly and is directly influenced by the 'Fog Density' setting

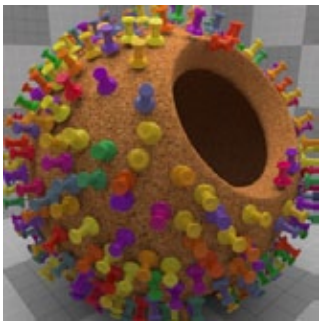
Use Environment Color: This option is only available when Backdrop Fog has been enabled. With backdrop fog modo adds the fog color to all materials with increasing intensity as the distance to the camera increases. With Use Environment Color active the Environment Material or textures will be used to color the fog. This option will give the illusion that the mesh is fading into the background.

Fog Color: This color component determines the color value that is applied to materials receiving fog.

Fog Start/End Distance: When the Fog type is set to linear, users can specify specific starting and ending distances for the fog effect.

Fog Density: This percentage value determines how thick the fog is just in front of the camera. The default value of 10% indicates that geometry just in front of the camera will have a 10% blend of the fog color. The density of the fog will increase as it recedes further from the camera.

Surface Generator



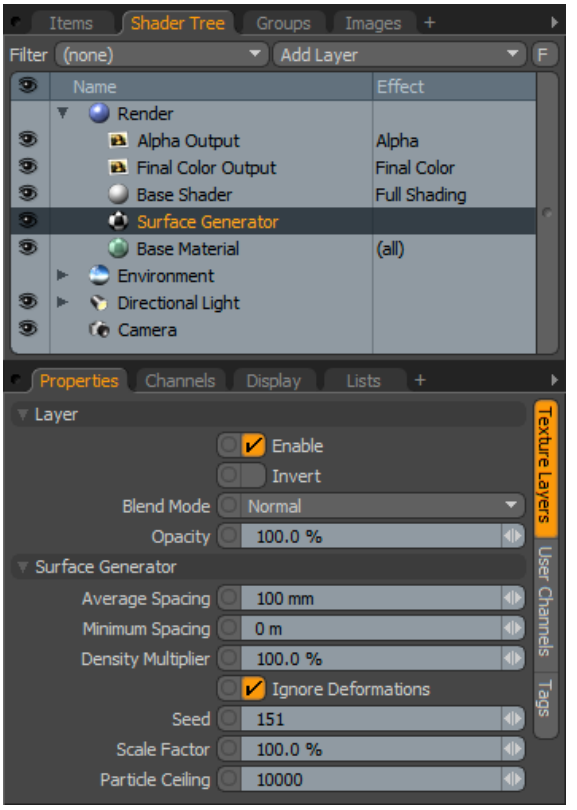
The Surface Generator works in tandem with a Replicator item offering a greater degree of control over the random placement and density of replicated objects. Once a Surface Generator is employed, additional texture layers can be used to control density, size, direction and even coloration, by setting the texture layer to the proper 'Effect'. Please reference the 'Shader Tree' page of the documentation for help with adding new layers and setting their 'Effect'. Replicators themselves use vertex data to propagate replicas, and even though there is an amount of randomization available, there is still a certain degree of regularity to the replicas. The 'Surface Generator' is a means of generating *random* points across the surface of mesh item, information that the surface generator then passes along to the replicator for *Prototype* propagation.

This is especially useful for situations like if you had modeled a tree and wanted to replicate leaves over its surface. You would want better control of the overall density than you could get by simply replicating them at only vertex positions, additionally, a weight map could be employed to limit the distribution the leaves on branches themselves and not the trunk with some additional color variation added by a gradient layer, producing far more realistic results.

Adding a Surface Generator--

When adding a surface generator, is is easiest to start with a working 'Replicator' in the item list, meaning the Prototype and Point Source items are defined as such in their respective properties. Once created, RMB+click in the Items List on the Replicator layer itself and select the 'Scatter Replicas' command from the context menu and that's it; an appropriate Surface Generator has been created in the shader tree updating the viewport with the new settings. Users may now select the 'Surface Generator' in the Shader Tree to modify its properties, making updates to the default settings.

If you prefer doing things manually, or you're just curious what happened in the background. Basically, modo created an item mask for the mesh that was specified as the 'Point Source'. An item mask is a shader tree group that calls a specific mesh item instead of the usual material tag. Under that shader tree group a 'Surface Generator' layer was added with some default values appropriate to the scale of the objects. Lastly, the 'Point Source' selector in the replicator now calls the 'Surface Generator' instead of the 'Point Source' item itself.



Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Surface Generator--

Average Spacing: Precisely as the name implies, the 'Average Spacing' value specifies the average distance between replicated items.

Minimum Spacing: Where the 'Average Spacing value only specifies an average, the 'Minimum Spacing' value allows one to specify the minimum amount of space between replicated items, especially useful in eliminating overlaps for neighboring replicas.

Density Multiplier: The 'Density Multiplier' offers simplified control over the overall density of replicated items. Lower values will reduce the overall number of replicas while higher values will increase the overall number of replicas up to the 'Particle Ceiling' value.

Ignore Deformations: When applying an animated deformer, such as a Morph or Weight deformer, changes to the underlying surface will trigger modo to re-distribute surface particles so they retain their 'Average' and 'Minimum' spacing values. When 'Ignore Deformations' is enabled, particle positions on a surface are maintained through the deformation as if they're were were stuck to their respective positions.

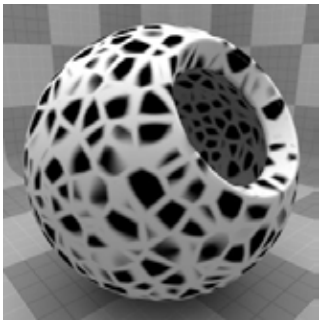
Seed: The 'Seed' value is an initial setting used when random numbers are generated. Different 'Seed' values will produce different random variations, in this case, initial clone positions. However you will want to use the same 'Seed' value when it is necessary that items retain the same variations.

Scale Factor: The 'Scale Factor' acts to control the scale of replicated objects, eliminating the need to manually scale the base 'Prototype' item. With a value of 100%, replicated items would be generated at the identical size of the prototype, smaller values would decrease each replicas size, while larger value would increase each replicas size.

Particle Ceiling: The 'Particle Ceiling' is a maximum for the total number of generated replicas.

TIP: If you're looking for a way to add some variety to replicated objects, one needs to look no further than Groups. By creating a group from multiple mesh items, and specifying that group as the Prototype to be cloned, the Replicator will pull random items from within the group adding further randomness and variation to the cloned objects. Once a group has been created, it will show up in the 'Prototype' drop down of the Replicator item. Additionally, surfacing for replicas can be further randomized with Gradients set to the 'Particle ID' Input Parameter.

Cellular Procedural



The Cellular texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The cellular pattern can be addressed by its two zones, the 'Cell' and the 'Filler'. The texture will modulate from one zone to the other based on the type and the users settings. Each zone can have a value, color and alpha value. Which of these are used is dependent on the 'Effect' to which the texture will be applied. For instance, if the texture is applied as a displacement, the 'Value' settings would be utilized whereas setting the texture Effect to 'Diffuse Color' would use the 'Color' and 'Alpha' settings for Cell and Filler. The Cellular texture is incredibly versatile and can create effects ranging from veins and cobblestone to crumply water and lizard skins.

For information regarding adding and working with Shader Tree Items Layers, please reference the the Shader Tree page of the documentation.

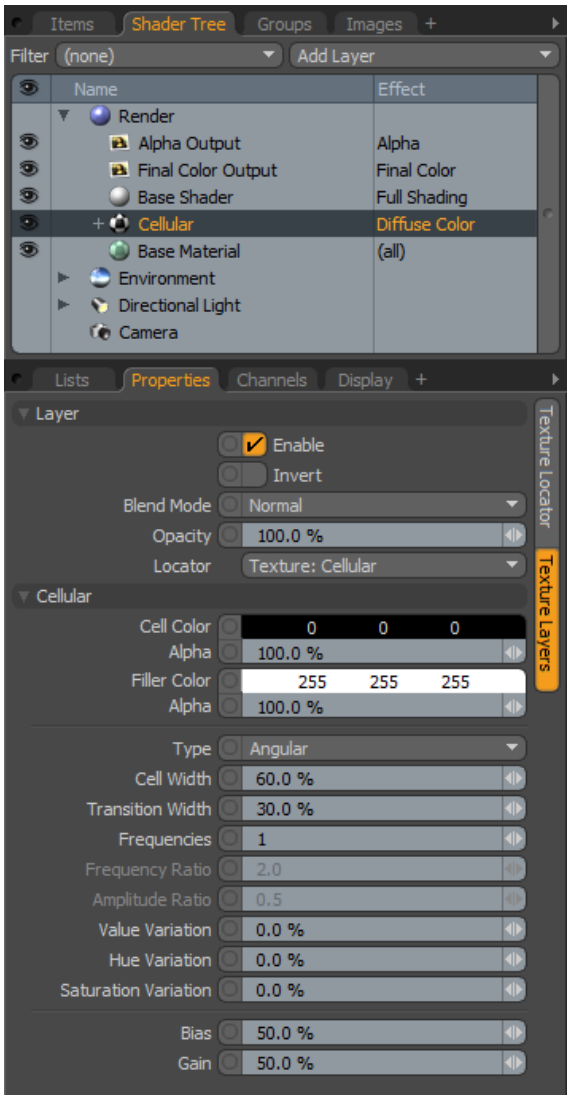
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (way the texture is applied) to the surface. The 'Locator' option sets that association. Users can choose alternate locators, however, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Cellular--

Cell Color: Determines the color of the texture where the Cell is most concentrated. This color will ramp to the Filler Color. The Color option is only used when the texture layer is set to an Effect that requires color. The dedicated Cell Color 'Alpha' will determine how transparent the Cell zone will be.

Cell Value: Determines the magnitude of the texture where the Cell is most concentrated. This value will ramp to the Filler value. This control is only active when the texture layers Effect is set to a non-color attribute.

Filler Color: Determines the color of the texture between the Cells. This color will ramp to the Filler Color. The Color option is only used when the texture layer is set to an Effect that requires color. The dedicated Fill Color 'Alpha' will determine how transparent the Fill zone will be.

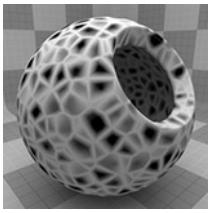
Filler Value: Determines the magnitude between the Cells. This value will ramp to the Filler value. This control is only active when the texture layers Effect is set to a non-color attribute.

Filler Alpha: Determines the alpha value of the texture between the Cells. This alpha value will ramp to the Filler Alpha value. The Alpha option is used in conjunction with the Color setting.

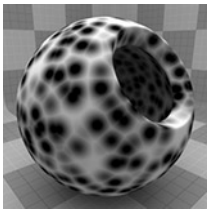
Type: The Type popup allows you to quickly change between the styles of Cellular texture. The options are 'Round' and 'Angular'. These options determine how the texture interpolates between its sample points.

Angular-- This option creates a somewhat linear interpolation between the sample points in the texture which results in an effect similar to cobblestone in the default setting. This mode can be used to generate effects ranging from lunar surface to craggy skin.

Round-- The Round type of cellular texture creates circular spots randomly distributed based on the texture. In the default settings the texture looks simply like overlapping soft edged dots. Adjusting the settings can yield effects ranging from cauliflower to chipped rock.

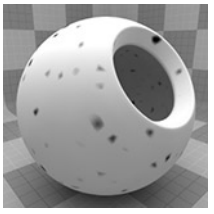


Type Angular

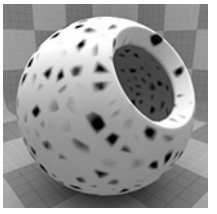


Type Round

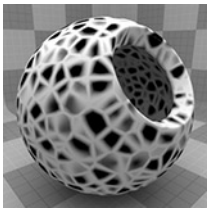
Cell Width: This setting determines the maximum scale of the texture details. At 100% the maximum scale of any single detail in the texture would be equal to the Texture size as set by the texture locator size values. If you want to change the scale of the details of the texture this is the value to use. To set the overall scale of the texture you will modify the texture scale on the Texture Locator tab.



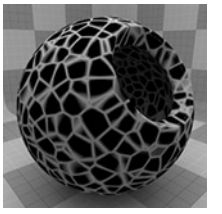
Cell Width 0%



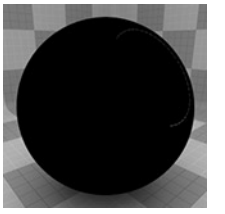
Cell Width 25%



Cell Width 50%

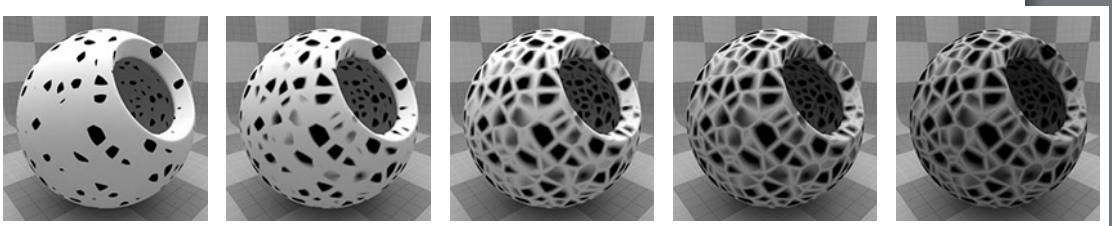


Cell Width 75%



Cell Width 100%

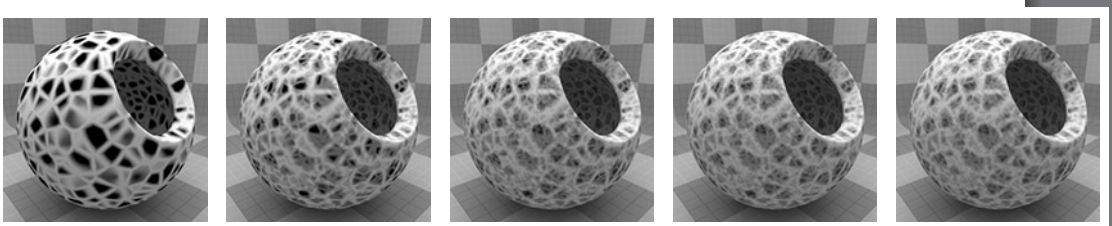
Transition Width: This value controls the amount of gradient falloff around each of the texture details. Driving this value over 100% can also yield interesting effects.



Transition Width 0% | Transition Width 25% | Transition Width 50% | Transition Width 75% | Transition Width 100%

All "noise" style procedurals can be "layered" to create a fractal effect. The traditional style of fractal includes layering a noise pattern multiple times with each new layer increasing in frequency (smaller details and more of them) and fading in opacity value. When you add a Cellular or Noise texture in modo the default settings are not "fractal" in that they use a default Frequency value of one, this means there is only one layer of the procedural effect. Adding multiple frequencies will create a more fractal look to the texture and will invoke the 'Frequency Ratio' and 'Amplitude Ratio' controls.

Frequencies: Determines the number of virtual layers used by the procedural. When set to less than 2, the 'Frequency' and 'Amplitude' Ratios are disabled. Adding more than one frequency increases the detail in the procedural texture which increases computation times.



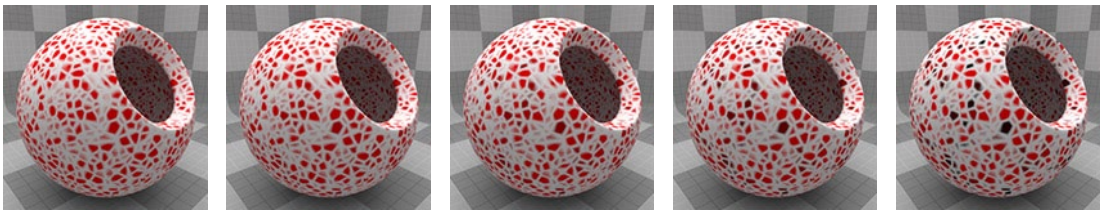
Frequency 1 | Frequency 2 | Frequency 3 | Frequency 4 | Frequency 5

Frequency Ratio: The Frequency Ratio determines how much detail to add into each additional application of the noise texture when the Frequencies value is set to a value greater than 1. With a value of 2, the result will be the second frequency of the texture being twice as high as the first. The visual effect of this is a pattern that seems to be half as large but with twice as many occurrences. With more than 2 frequencies each additional "layer" of the noise will have twice the frequency of the last.

Amplitude Ratio: This value determines the strength of each additional "layer" of noise

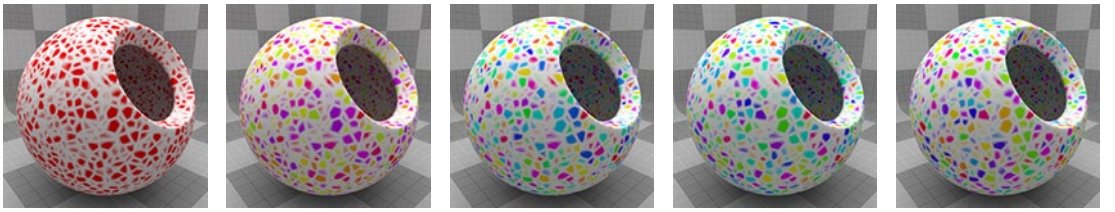
when frequencies are greater than 1. By default the setting is at .5 which yields an effect where each additional set of frequencies have half the value of the previous.

Value Variation: The 'Value Variation' percentage allows you to randomly vary the brightness or luminosity of each individual cell. Higher numbers will increase the differences between cells while lower ones will have decreased brightness variations.



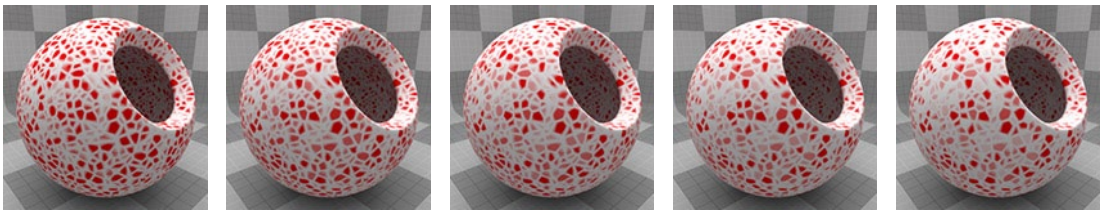
Value Variation 0% Value Variation 25% Value Variation 50% Value Variation 75% Value Variation 100%

Hue Variation: The 'Hue Variation' percentage allows you to randomly vary the hue or color of each individual cell. Higher numbers will increase the color differences between cells while lower ones will have decreased color variations.



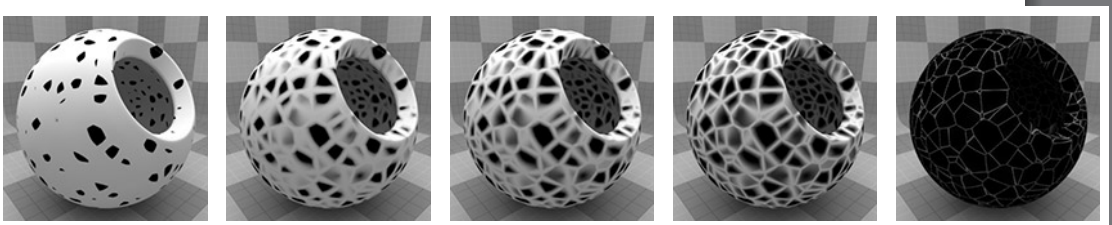
Hue Variation 0% Hue Variation 25% Hue Variation 50% Hue Variation 75% Hue Variation 100%

Saturation Variation: The 'Saturation Variation' percentage allows you to randomly vary the color saturation of each individual cell. Higher numbers will increase the differences between cells while lower ones will have decreased saturation variations.



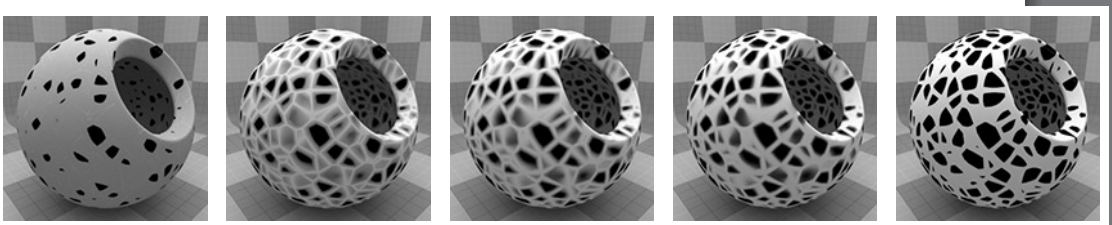
Sat. Variation 0% Sat. Variation 25% Sat. Variation 50% Sat. Variation 75% Sat. Variation 100%

Bias: Increasing this value will cause the texture to favor the Cell color or value over the Filler whereas decreasing the value causes the Filler color or value to be favored.



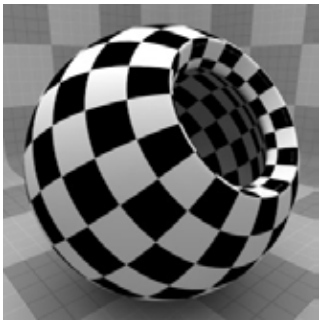
Bias 0% Bias 25% Bias 50% Bias 75% Bias 100%

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the cell and filler values. Setting the Gain to 100% will create a very sharp falloff effect whereas setting the value to 0 would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.



Gain 0% Gain 25% Gain 50% Gain 75% Gain 100%

Checkerboard Procedural



The Checker texture uses two colors or values and creates a pattern of alternating squares that resembles a Checker board. By changing the Transition Width, Bias and Gain, users may also create a variety of checkered and diamond like shapes. Additionally, as a procedurally created texture, mathematically generated at render-time, the Checker texture has no fixed resolution, and can be magnified nearly infinitely with no visual loss in detail.

For information regarding adding and working with Shader Tree Items Layers, please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

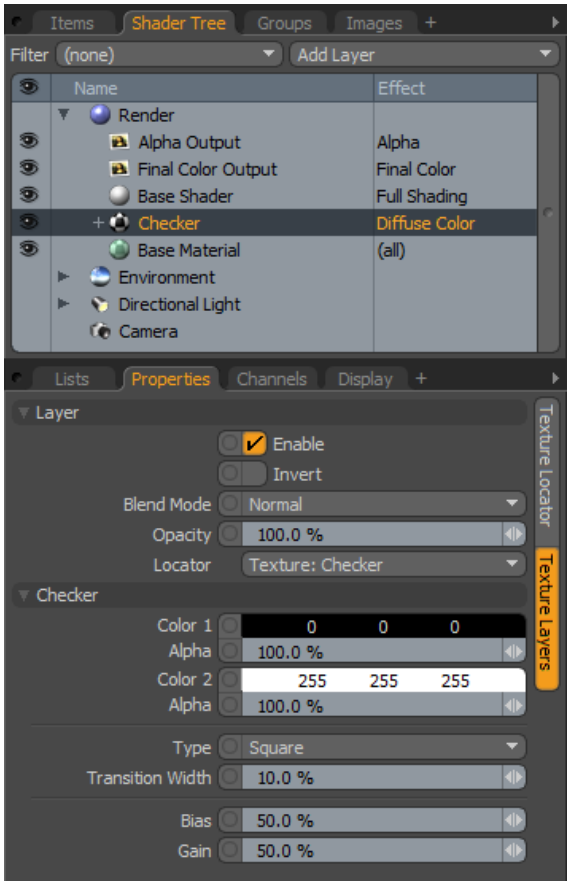
Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (way the texture is applied) to the surface. The 'Locator' option sets that association. Users can choose alternate locators, however, the need to do so will be very rare; still, there

are some possible instances where users may want multiple texture items to share a single locator.



Checker--

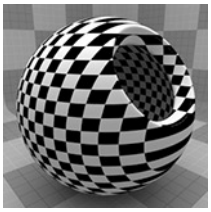
Color 1: Determines the color of the texture on the first set of alternating squares. This color will ramp to 'Color '2 based on the transition Width. The 'Alpha' value determines the level of transparency.

Color 2: Determines the color of the texture on the second set of alternating squares. The 'Alpha' value determines the level of transparency.

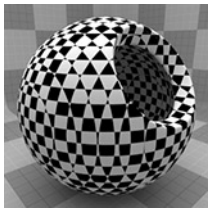
Type: The Type popup allows you to quickly change between the styles of Check texture. The options are 'Square' and 'Cubic' and determine the method of projection.

Square-- Projects the checkered grid from a single Axis (which can be defined in the 'Texture Locator').

Cubic-- Projects the checker grid from all three axes (which likely looks odd on curved surfaces as the edges interact with each other)

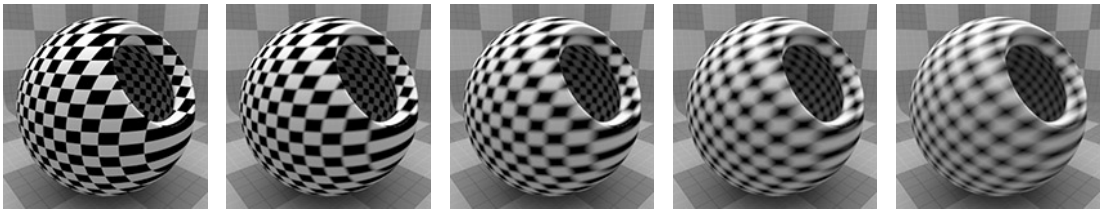


Type Square



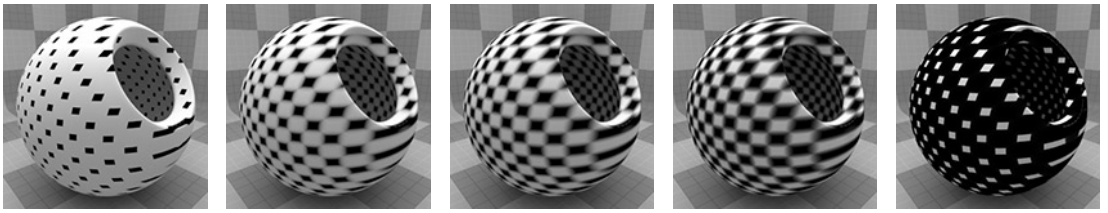
Type Cubic

Transition Width: This value controls the amount of gradient falloff across each of the texture edges. Driving this value over 100% can also yield interesting effects.



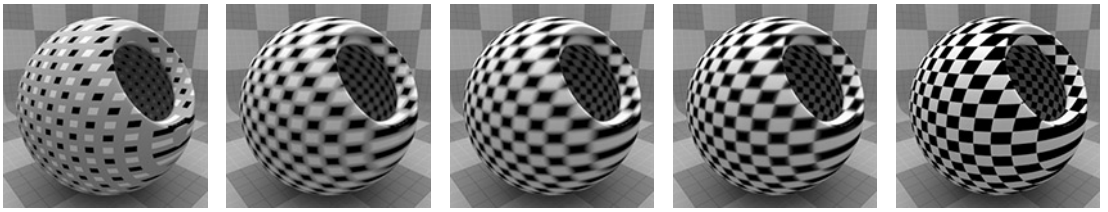
Transition Width 0% | Transition Width 25% | Transition Width 50% | Transition Width 75% | Transition Width 100%

Bias: Increasing this value will cause the texture to favor the first color or value over the other, whereas decreasing the value causes the secondary color or value to be favored. This value is dependant on the 'Transition Width' as it works across the gray tones of the image.



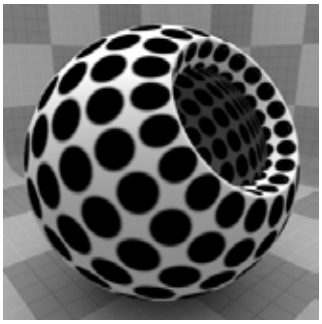
Bias 0% | Bias 25% | Bias 50% | Bias 75% | Bias 100%

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient. This value is dependant on the 'Transition Width' as it works across the gray tones of the image.



Gain 0% | Gain 25% | Gain 50% | Gain 75% | Gain 100%

Dots Procedural



The Dots texture creates a grid like pattern of dots across the surface of the mesh. These dots default to a fairly uniform position with hard edges, but can be modified for different placements and soft ramped dots. Procedurally created at render time, the dots texture has no fixed resolution, and can be magnified nearly infinitely with no visual loss in detail.

For information regarding adding and working with Shader Tree Items Layers, please reference the the Shader Tree page of the documentation.

Layer--

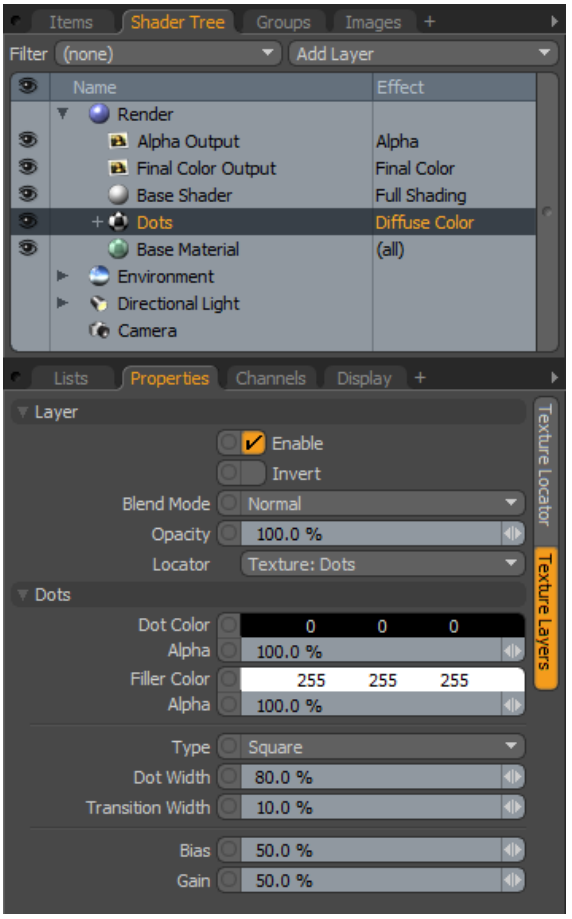
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (way the texture is applied) to the surface. The 'Locator' option sets that association. Users can choose alternate locators, however, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.



Dots--

Dot Color: Determines the color of the Dot itself. This color will ramp to the Filler Color based on the 'Transition Width. The 'Alpha;' value determines the opacity of the dot.

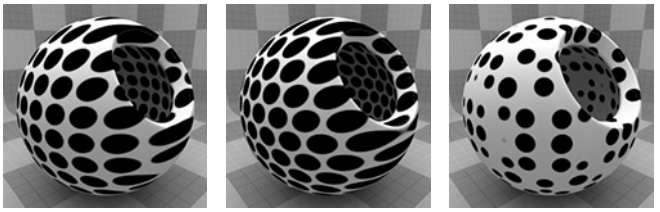
Filler Color: Determines the color of the texture between the dots. The 'Alpha;' value determines the opacity of the filler area.

Type: The Type parameter changes the method for applying the dots to the surface.

Square-- The default Square mode creates a series of perfectly aligned rows and columns of dots that are projected in 2D on a single axis against the mesh. You can see this effect in the image below.

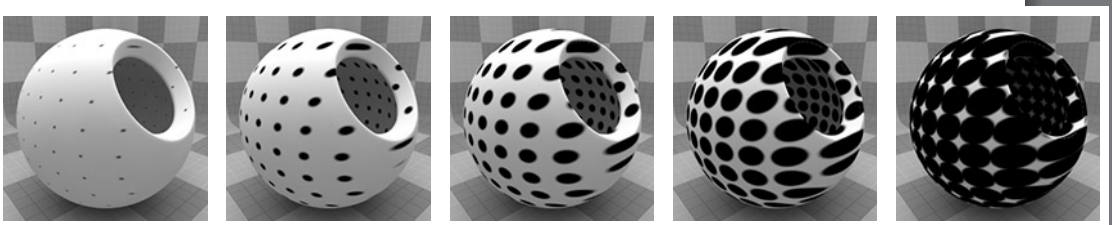
Hexagon-- This mode is similar to the Square method with the exception that the dots are offset in alternating rows creating a hexagonal pattern.

Cube-- This type setting creates a three dimensional array of spheres that intersects with the surface of the mesh. For any projection type other than UV, which causes the dots to lay perfectly on the surface, the mesh will intersect the spheres at different cross sections resulting in dots of varying size and thickness.



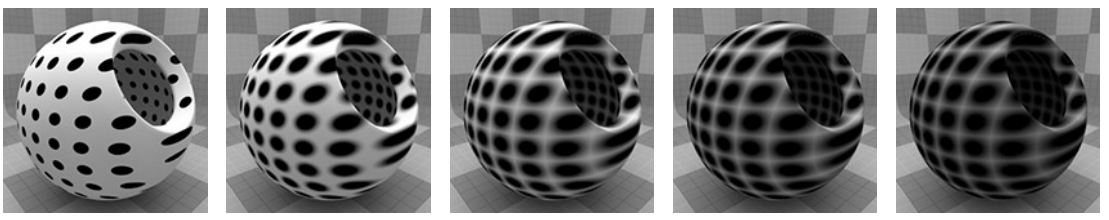
Type Square Type Hexagon Type Cube

Dot Width: This setting controls the diameter of the dots within the grid. At 100% the dots touch edge to edge.



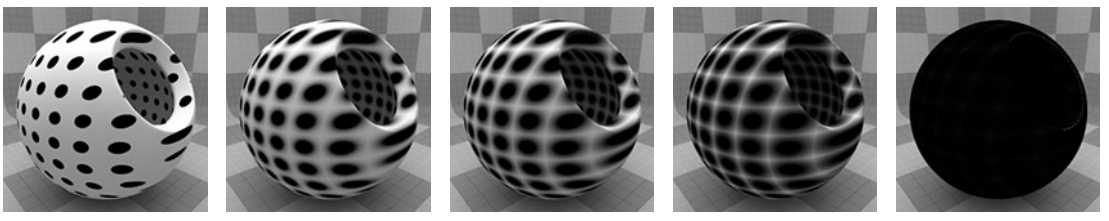
Dot Width 0% | Dot Width 25% | Dot Width 50% | Dot Width 75% | Dot Width 100%

Transition Width: The Transition Width creates a smooth ramp blend from the Dot color or value to the Filler color or value. This is set as a percentage of the distance between the dot and the filler. Extreme numbers can be used to create interesting effects. Don't be shy!



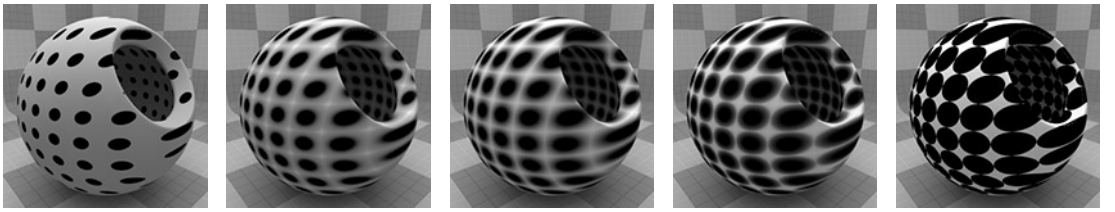
Transition Width 0% | Transition Width 25% | Transition Width 50% | Transition Width 75% | Transition Width 100%

Bias: Increasing this value will cause the texture to favor the Dot color or value over the Filler whereas decreasing the value causes the Filler color or value to be favored. This value is dependant on the 'Transition Width' as it works across the gray tones of the image.



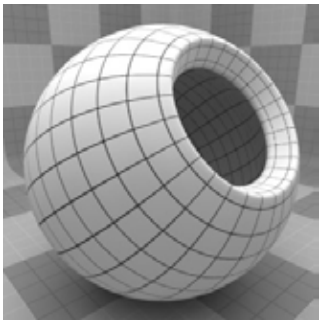
Bias 0% | Bias 25% | Bias 50% | Bias 75% | Bias 100%

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the dot and filler values. Setting the Gain to 100% will create a very sharp falloff effect whereas setting the value to 0 would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient. This value is dependant on the 'Transition Width' as it works across the gray tones of the image.



Gain 0% Gain 25% Gain 50% Gain 75% Gain 100%

Grid Procedural



The Grid texture creates a grid pattern between two value or color settings; the Line Value/Color and the Filler Value/Color. The pattern of the lines is determined by the Type setting and their thickness by the Line Width value. Between the lines and the filler exists a transition zone which is determined by the Transition Width. Procedurally created at render time, the grid therefore has no fixed resolution, and can be magnified nearly infinitely with no visual loss in detail.

For information regarding adding and working with Shader Tree Items Layers, please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

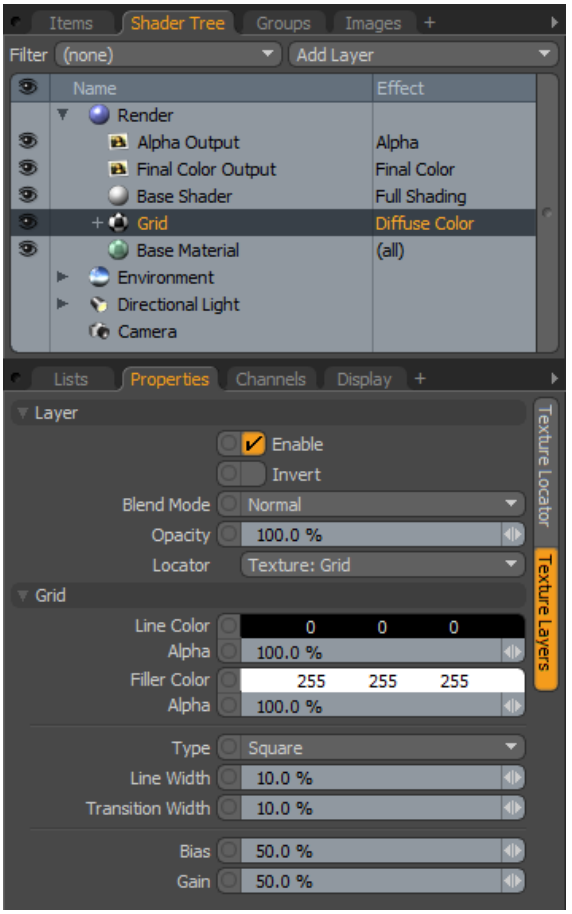
Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (way the texture is applied) to the surface. The 'Locator' option sets that association. Users can choose alternate locators, however, the need to do so will be very rare; still, there

are some possible instances where users may want multiple texture items to share a single locator.



Grid--

Line Color: Determines the color of the texture where lines appear ramping toward the 'Filler' color based on the 'Transition Width'. The 'Alpha' value determines the level of transparency.

Filler Color: Determines the color of the areas between the lines. The 'Alpha' value determines the level of transparency.

Type: The Grid can be generated in many different styles. These styles are determined by the Type setting.

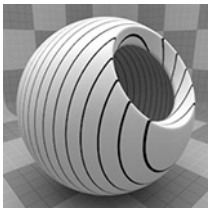
Line-- This pattern consists of parallel straight lines. The orientation of the lines are determined by the Texture Locator Axis setting.

Triangle-- This pattern interconnects the series of parallel lines with additional diagonal lines which intersect to create triangles.

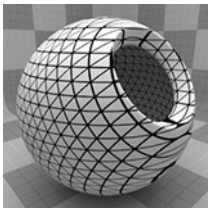
Square-- Two sets of lines running perpendicular to each other to create a square shaped grid. This is the default setting as it creates what we traditionally visualize when thinking of a "grid."

Hexagon-- One series of parallel lines is intersected by a series of interconnecting lines which create hexagons at each of the intersection points.

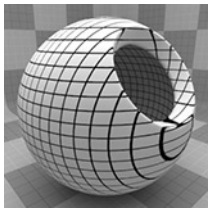
Cube-- This pattern is similar to Square generating a grid patern, with a third set of parallel lines projected from the Z axis that produces a grid with depth.



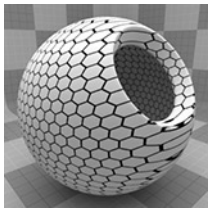
Type Line



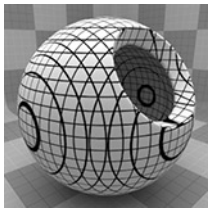
Type Triangle



Type Square

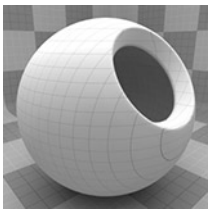


Type Hexagon

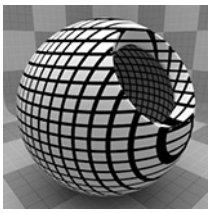


Type Cube

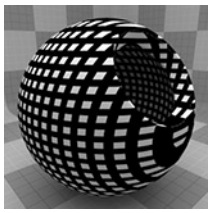
Line Width: This setting determines the width of the lines as a percentage of the texture area. Setting this value higher results in thicker lines until the lines completely cover the surface at 100%. Reducing the value results in thinner lines. (Tip: When using extremely small values to create very thin lines make sure to leave a small amount of Transition Width as this will provide an anti-aliasing effect on the fine texture.)



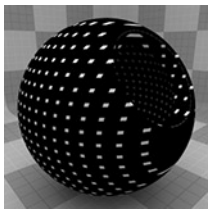
Line Width 0%



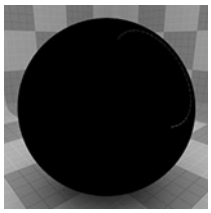
Line Width 25%



Line Width 50%

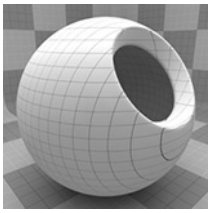


Line Width 75%

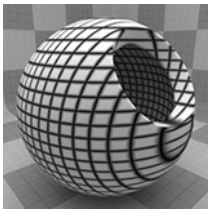


Line Width 100%

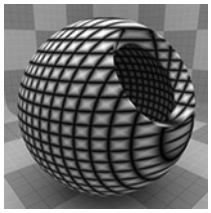
Transition Width: This setting determines the width of the gradient ramp between the line value or color and the filler value or color.



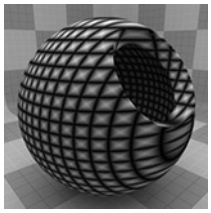
Transition Width 0%



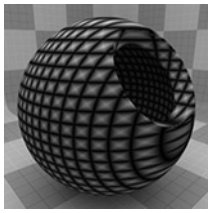
Transition Width 25%



Transition Width 50%

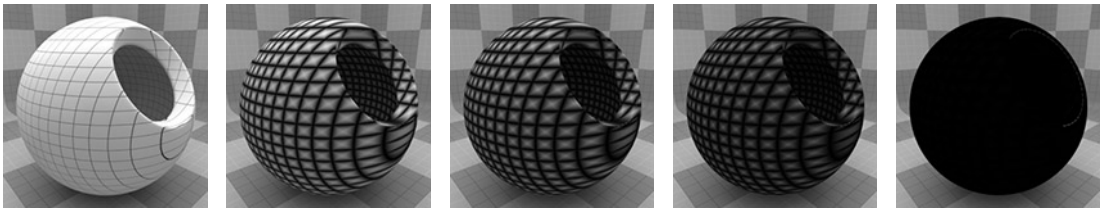


Transition Width 75%



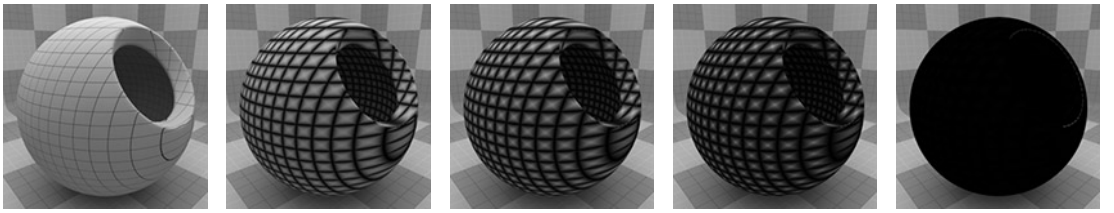
Transition Width 100%

Bias: Increasing this value will cause the texture to favor the first color or value over the other, whereas decreasing the value causes the secondary color or value to be favored. This value is dependant on the 'Transition Width' as it works across the gray tones of the image.



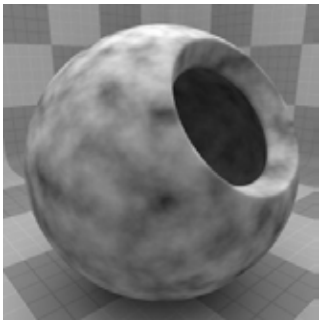
Bias 0% Bias 25% Bias 50% Bias 75% Bias 100%

Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% will create a very sharp falloff effect whereas setting the value to 0% would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient. This value is dependant on the 'Transition Width' as it works across the gray tones of the image.



Gain 0% Gain 25% Gain 50% Gain 75% Gain 100%

Noise Procedural



The Noise texture is one of the many procedurally generated textures provided with modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution; they can be magnified nearly infinitely with no visual loss in detail. The Noise texture has 2 zones, the Color/Value 1 and Color/Value 2. The texture will modulate between the two based on the particular algorithm and user settings. Essentially there are pseudo-random points created in 3D space and the texture blends from one value to the next based on the proximity of one sample point to the next. Each zone can have a value, color and alpha value. Which of these are used is dependent on the Effect channel to which the texture will be applied. For instance, if the texture is to be used as a displacement, the Value settings would be utilized whereas setting the texture Effect to Luminous Color would use the Color and Alpha settings for Value/Color 1 and Value/Color 2.

For information regarding adding and working with Shader Tree Items Layers, please reference the the Shader Tree page of the documentation.

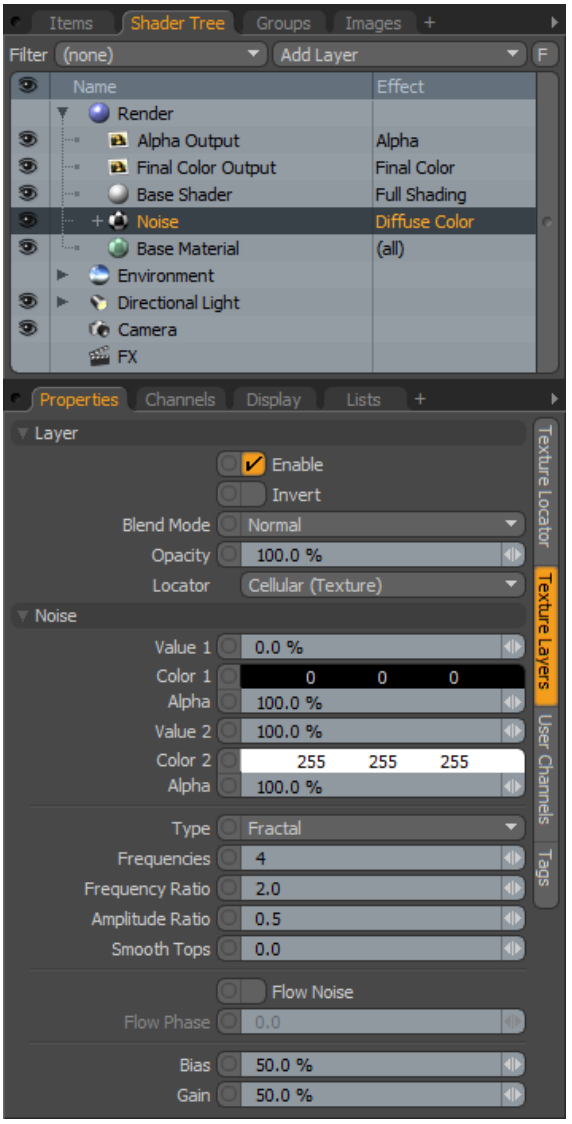
Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.



Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (way the texture is applied) to the surface. The 'Locator' option sets that association. Users can choose alternate locators, however, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

Noise--

Color 1: Sets the color that will be used at the center of each sample location. This color is blended into the Color 2 value based on proximity to the surrounding sample points. The Color option is only used when the texture layer is set to an Effect that requires color. The dedicated 'Alpha' will determine how transparent the Color 1 zone will be.

Value 1: Determines the magnitude of the texture at the center of each sample location. This value is blended into the Value 2 based on proximity to surrounding sample points. This control is only active when the texture layers Effect is set to a non-color attribute.

Color 2: Sets the color that will be used at the center of each secondary sample location. This color is blended into the Color 1 value based on proximity to the surrounding sample points. The Color option is only used when the texture layer is set to an Effect that requires color. The dedicated 'Alpha' will determine how transparent the Color 2 zone will be.

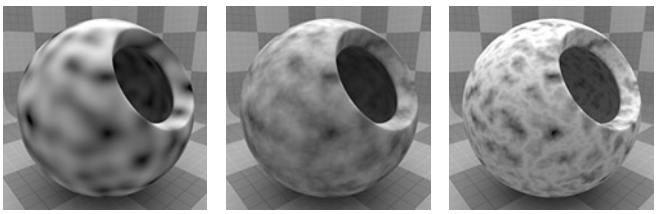
Value 2: Determines the magnitude of the texture at the center of each secondary sample location. This value is blended into the Value 1 based on proximity to surrounding sample points. This control is only active when the texture layers Effect is set to a non-color attribute.

Type: The Type popup allows you to quickly change between the styles of Noise. The options are 'Simple', 'Fractal' and 'Turbulence'.

Simple-- Merely modulates between Color/Value 1 and Color/Value 2 and does not allow for additional Frequencies to be added preventing it from becoming "fractal".

Fractal-- Unlike Simple, Fractal also allows for Frequency settings above 1. Once additional frequencies are added to a noise or cellular pattern then become "fractal". Each additional frequency of the texture is layered over the previous. The relative detail and amplitude of the additional frequencies is modulated by the frequency and amplitude ratios.

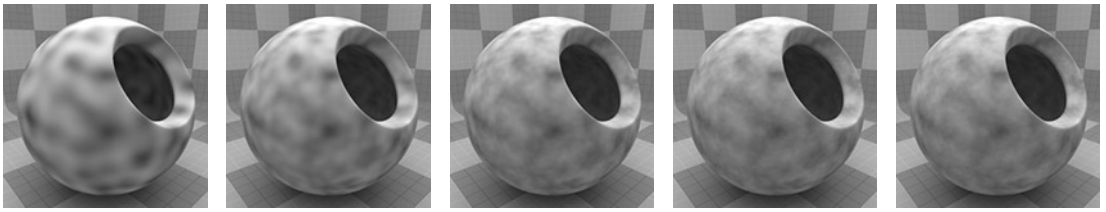
Turbulence-- The Turbulence setting is very similar to the Fractal setting but the gradient ramp between the between the two Color/Value modulates up and down rather than a straight ramp.



Simple Fractal Turbulence

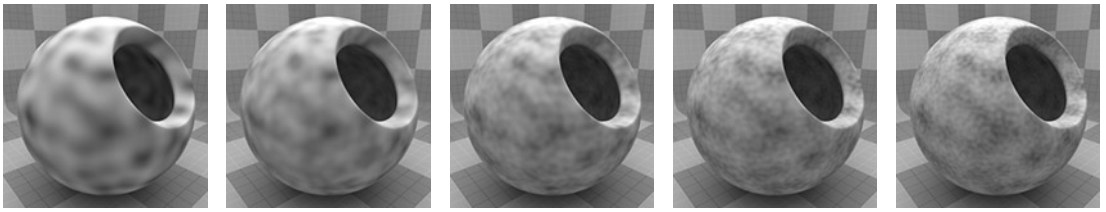
All noise style procedurals can be *layered* to create a fractal effect. The traditional style of fractal includes layering a noise pattern multiple times with each new layer increasing in frequency (smaller details and more of them) and fading in opacity. When you add a Cellular or Noise texture in modo the default settings are not "fractal" in that they only use a Frequency value of one, this means there is only single application of the procedural effect. Adding multiple frequencies will create a more fractal look to the texture and will invoke the Frequency Ratio and Amplitude Ratio controls.

Frequencies: Determines the number of "layers" used by the procedural. When set to less than 2 the Frequency and Amplitude Ratios are disabled. Adding more than one frequency increases the detail in the procedural texture which increases computation times.



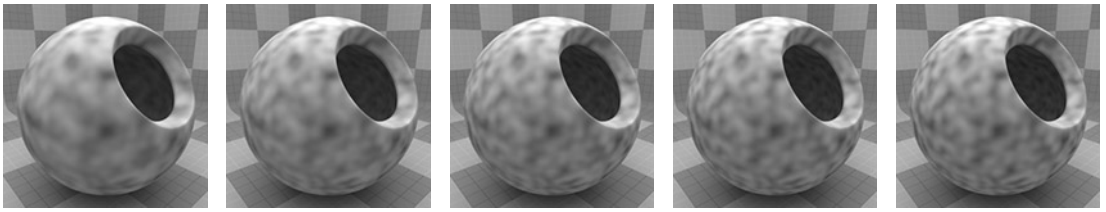
Frequency 1 Frequency 2 Frequency 3 Frequency 4 Frequency 5

Frequency Ratio: The Frequency Ratio determines how much detail to add into each additional application of the noise texture when the Frequencies value is set to a value greater than 1. The default value of 2 will result in the second frequency of the texture being twice as high as the first. The visual effect of this is a pattern that seems to be half as large but with twice as many occurrences. With more than 2 frequencies each additional "layer" of the noise will have twice the frequency of the last.



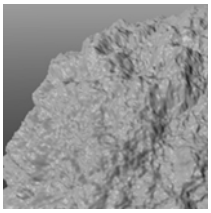
Frequency Ratio 1 Frequency Ratio 2 Frequency Ratio 3 Frequency Ratio 4 Frequency Ratio 5

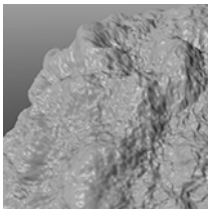
Amplitude Ratio: This value determines the strength of each additional "layer" of noise when frequencies are greater than 1. By default the setting is at .5 which yields an effect where each additional set of frequencies have half the value of the previous.

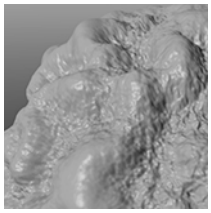


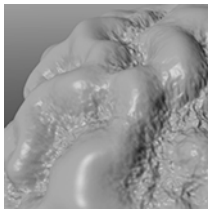
Amplitude Ratio 0.5 Amplitude Ratio 1.0 Amplitude Ratio 1.5 Amplitude Ratio 2.0 Amplitude Ratio 2.5

Smooth Tops: The 'Smooth Tops' option is used to vary the noise amplitude depending on the texture value. If the noise was applied to a surface as a displacement produces something that looks like mountains with a smooth top and ragged valleys as the noise amplitude increases as you go down. If the texture were inverted, the mountains would be ragged and the valleys will be smooth like is usually the case in nature.


Smooth Tops 0.0


Smooth Tops 1.0


Smooth Tops 2.0

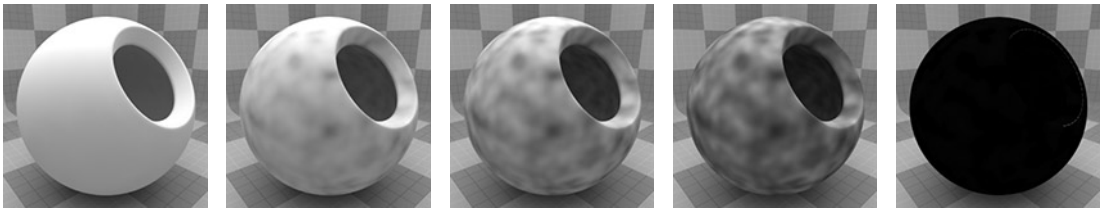

Smooth Tops 4.0

Smooth Tops 8.0

Flow Noise: Flow Noise provides a means to easily animate the noise field without having to physically translate (move) the noise across the surface it is shading, which often times looks artificial. Flow Noise is a Perlin Noise that simulates *advection* (which is simply the movement of something from one place to another, especially when they are different temperatures, such as creamer into coffee). The results could be described as *turbulence* or *boiling*, depending on how it is used. This setting can be especially handy when animating volumetrics, to produce the time-lapse-clouds sort of look that would be difficult to create otherwise. To use the 'Flow Noise' option, users need only to *enable* the setting and then animate the value for 'Flow Phase' option below.

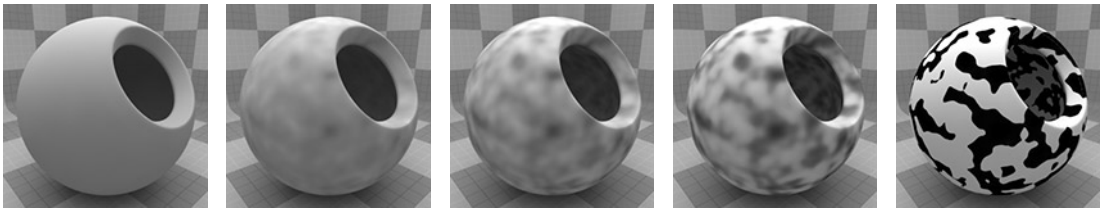
Flow Phase: The 'Flow Phase' option works in conjunction with the 'Flow Noise' setting when it is *enabled*. Animating this value produces a flowing type of animation to the noise that looks far more realistic than simply translating (moving) the noise across the surface or volume. To easily apply a continuously moving 'Phase', users can set a keyframe at frame '0' and then another value somewhere else (such as '1'). If the animation graph is set to 'Linear' and the 'Post Behavior' is set to 'Linear' as well, the Phase will continuously animate the length of the sequence. Users can then adjust the second keyframes value up and down to control the speed of the Flow Phase.

Bias: Increasing this value will cause the texture to favor the primary color or value over the secondary whereas decreasing the value causes the secondary color or value to be favored.



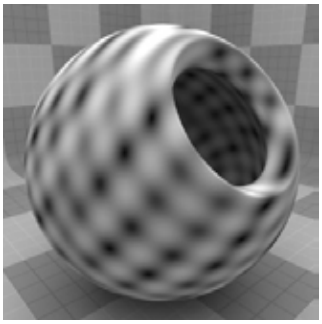
Bias 0% Bias 25% Bias 50% Bias 75% Bias 100%

Gain: The Gain setting is similar to a contrast control that effects the falloff of the gradient ramp between the Color/Value 1 and Color/Value 2. Setting the Gain to 100% will create a very sharp contrast whereas setting the value to 0% would reduce the contrast between the two to almost imperceptible amount.



Gain 0% Gain 25% Gain 50% Gain 75% Gain 100%

Ripple Procedural



Like raindrops on a puddle, the procedural texture Ripple creates a series of concentric circles that originate within a given area and continue outward toward infinity. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution; they can be magnified nearly infinitely with no visual loss in detail. The texture scale determines the area where the ripples will originate, while Sources, Wavelength and Phase determine the specific look of the ripples. The Ripple texture is a good choice when you need to simulate undulating waves.

For information regarding adding and working with Shader Tree Items Layers, please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

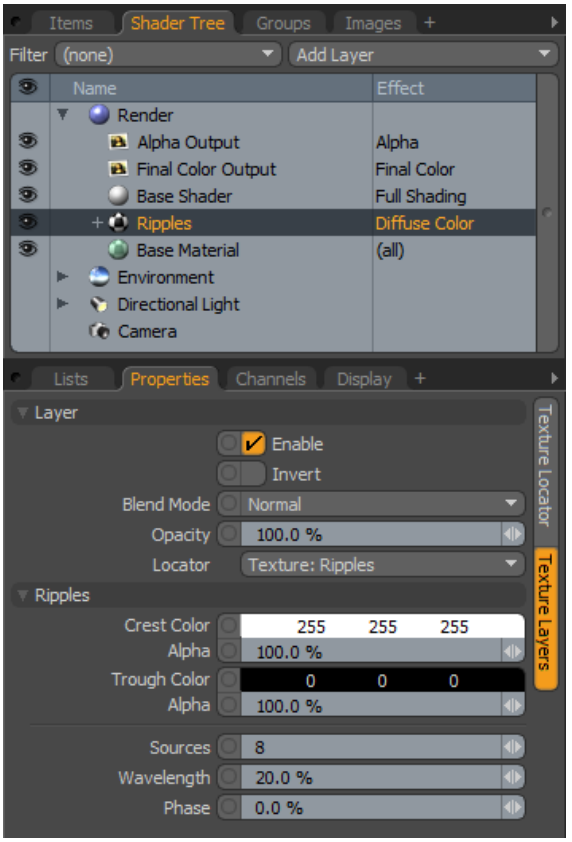
Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (way the texture is applied) to the surface. The 'Locator' option sets that association. Users

can choose alternate locators, however, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.



Ripple--

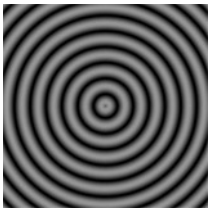
Crest Color: Only available when Ripple is applied to a color channel, such as diffuse color, 'Crest Color' will allow you to specify the RGB color value for what would be the top part of the wave along with its own Alpha value that ramps toward the trough determining how transparent the Crest will be.

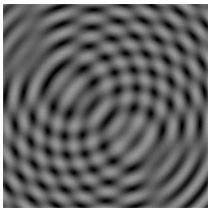
Crest Value: Only available when Ripple is applied to a non-colored channel, such as bump or displacement, the 'Crest Value' will specify the intensity or height of the top of the wave.

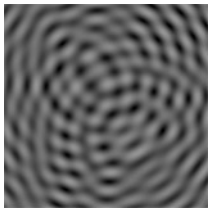
Trough Color: Only available when Ripple is applied to a color channel, such as diffuse color, 'Trough Color' will allow you to specify the RGB color value for what would be the lowest part of the wave along with its own Alpha value that ramps toward the crest determining how transparent the trough will be.

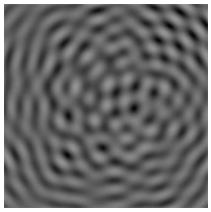
Trough Value: Only available when Ripple is applied to a non-colored channel, such as bump or displacement, the 'Trough Value' will specify the intensity or depth of the bottom of the wave.

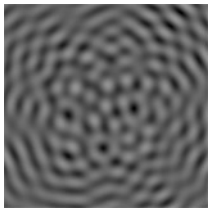
Sources: The 'Sources' value determines the number of ripple generators your surface will have, greater numbers give more complex rippled surfaces, while lower numbers will will more obviously show the concentric circles the ripples are made from.


Sources 1


Sources 2

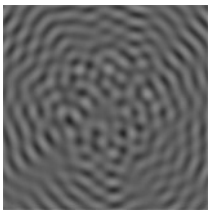

Sources 3

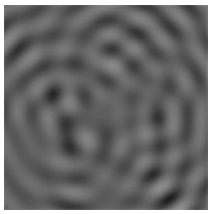

Sources 4

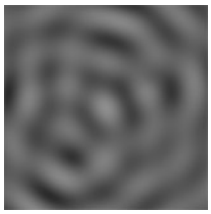

Sources 5

Wavelength: The 'Wavelength' value determines the distance between each ring. Higher values will move the rings further away from each other, while lower rings will increase their frequency, moving them closer together.

Wavelength 0.1


Wavelength 25


Wavelength 50

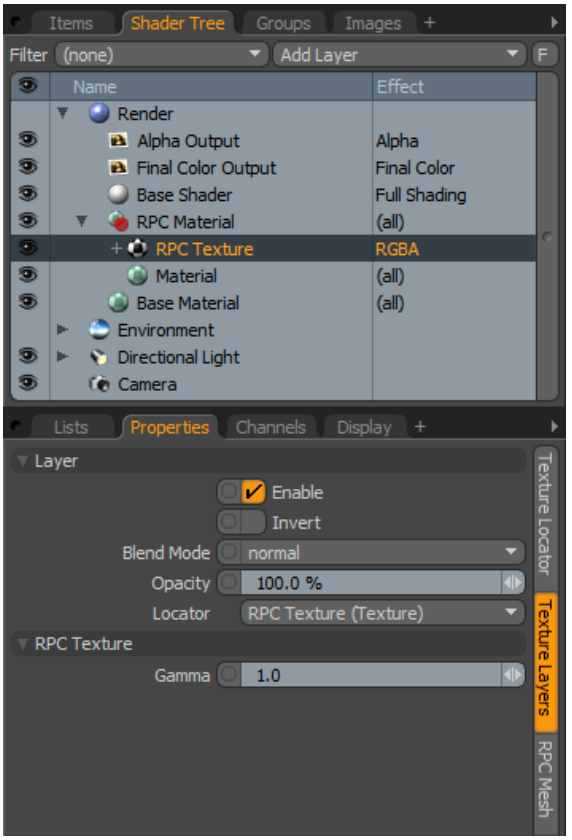

Wavelength 75

Wavelength 100

Phase: The 'Phase' value determines the position of the wave as it radiates from its center point. Animating this value in the negative direction for the waves to move outward, and in the positive direction to move the waves inward.

RPC Texture

RPC stands for "Rich Photoreal Content" and is useful in realistically adding photographic elements to a scene. The RPC Texture is associated to each RPC Mesh item added to a scene. THe main utility being to adjust the Gamma of the associated images. FOr more information on using RPC items in modo, please reference the RPC Mesh Item page of the documentation.



Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

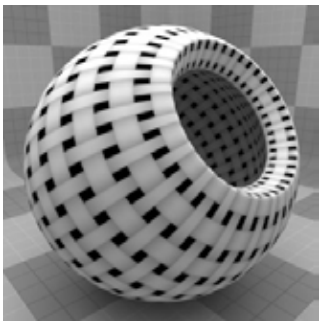
Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (way the texture is applied) to the surface. The 'Locator' option sets that association. Users can choose alternate locators, however, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.

RPC Texture--

Gamma: This value adjust the Gamma for images when rendered. If images appear washed-out or light, it is likely because of the gamma difference between the image and the Render Output. Please reference that page of the documentation for more on working with Gamma issues in modo.

Weave Procedural



The 'Weave' procedural texture simulates woven yarn, in a basic basket-weave pattern. A series of parallel rows, combined with columns that thread over and under alternating rows forming the basic pattern. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. Yarn Width and Roundness control the look of the actual woven strands.

For information regarding adding and working with Shader Tree Items Layers, please reference the the Shader Tree page of the documentation.

Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

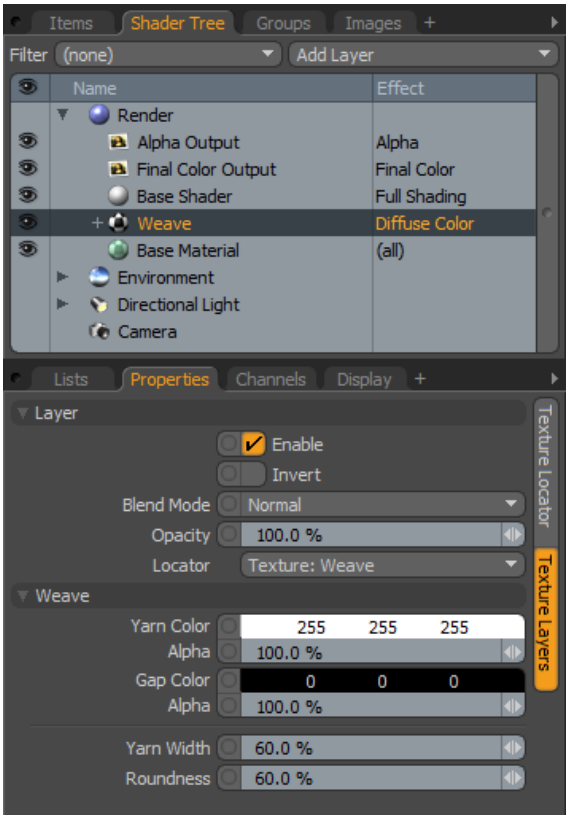
Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will

increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (way the texture is applied) to the surface. The 'Locator' option sets that association. Users can choose alternate locators, however, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.



Weave--

Yarn Color: Only available when Weave is applied to a color channel, such as diffuse color, 'Yarn Color' will allow you to specify the RGB color value for the columns and rows of virtual yarn along with its own Alpha value determining how transparent the Yarn will be.

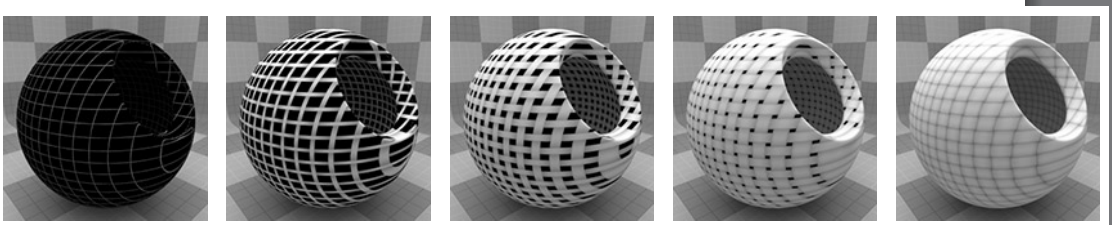
Yarn Value: Only available when Weave is applied to a non-colored channel, such as bump or displacement, 'Yarn Value' will determine the intensity or height of the columns and rows of the virtual yarn.

Gap Color: Only available when Weave is applied to a color channel, such as diffuse color, 'Gap Color' will allow you to specify the RGB color value for the spaces between the virtual yarn along with its own Alpha value determining how transparent the gaps will be.

Gap Value: Only available when Weave is applied to a non-colored channel, such as bump or displacement, 'Gap Value' will determine the intensity or depth of the gaps between the virtual yarn.

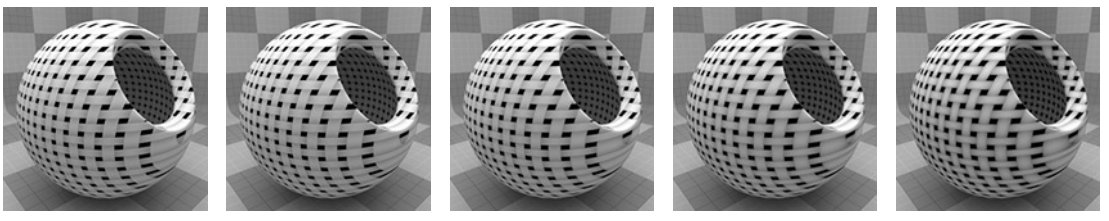
Yarn Width: The 'Yarn Width' setting determines the thickness of the virtual columns and rows of yarn. A value of 100% will completely eliminate the adjoining gaps looking

like a tightly woven basket, while smaller values will simulate loosely woven meshes such as a net or screen.



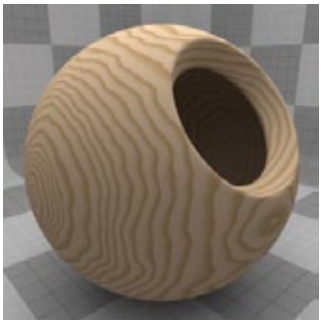
Yarn Width 0.1% Yarn Width 25% Yarn Width 50% Yarn Width 75% Yarn Width 100%

Roundness: The 'Roundness' setting determines how round the strands of virtual yarn look, A value of 0% would leave the strands completely flat while a value of 100% would render them as if tiny woven tubes.



Roundness 0% Roundness 25% Roundness 50% Roundness 75% Roundness 100%

Wood Procedural



The Wood procedural is a powerful texture that is designed to simulate wood grain. Of course, like any good procedural, the Wood texture can be used for effects far beyond its simple namesake. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The texture is composed of Rings which are distorted by *waves* and a layer of noise that simulates wood grain.

For information regarding adding and working with Shader Tree Items Layers, please reference the the Shader Tree page of the documentation.

Layer--

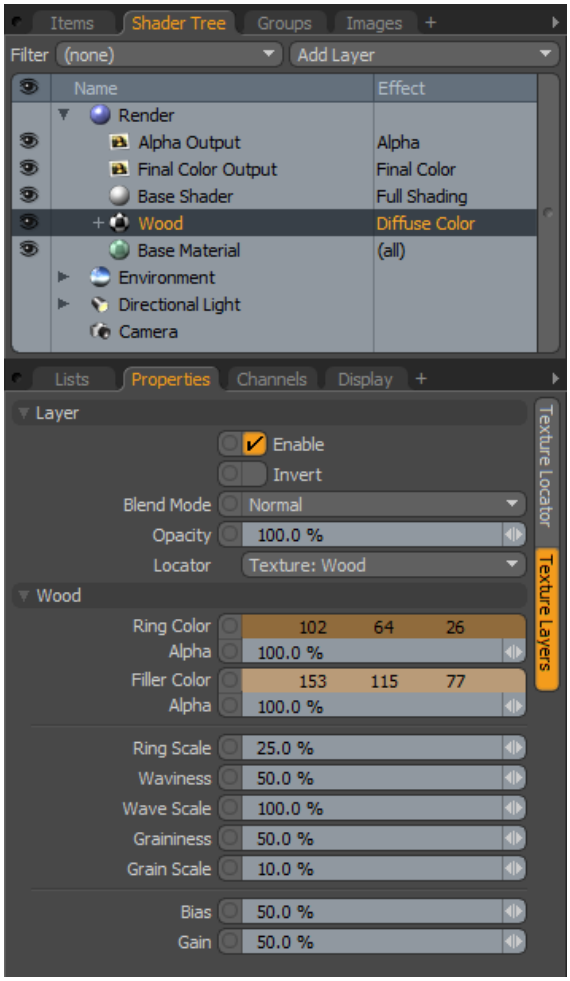
Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Locator: Most texture layers also have an associated 'Texture Locator' that is automatically created in the 'Item List'. This defines the mapping of the texture (way the texture is applied) to the surface. The 'Locator' option sets that association. Users can choose alternate locators, however, the need to do so will be very rare; still, there are some possible instances where users may want multiple texture items to share a single locator.



Wood--

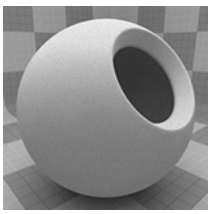
Ring Color: Determines the color of the texture where the Ring pattern occurs. This color will ramp to the Filler Color. The Color option is only used when the texture layer is set to an Effect that requires color. The dedicated Ring Color 'Alpha' will determine how transparent the Ring zone will be.

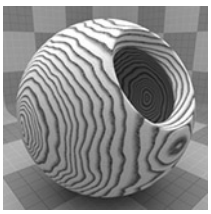
Ring Value: Determines the magnitude of the texture where the Ring pattern occurs. This value will ramp to the Filler value. This control is only active when the texture layers Effect is set to a non-color attribute.

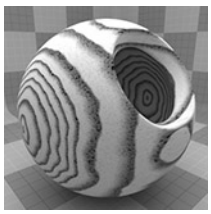
Filler Color: Determines the color of the texture where the Filler pattern occurs. The Color option is only used when the texture layer is set to an Effect that requires color. The dedicated Ring Color 'Alpha' will determine how transparent the Ring zone will be.

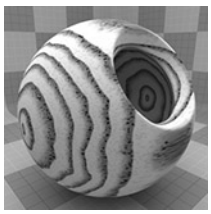
Filler Value: Determines the magnitude of the texture where the Filler pattern occurs. This control is only active when the texture layers Effect is set to a non-color attribute.

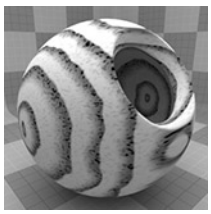
Ring Scale: This percentage value determines the size of the rings in the texture.


Ring Scale 0.1%

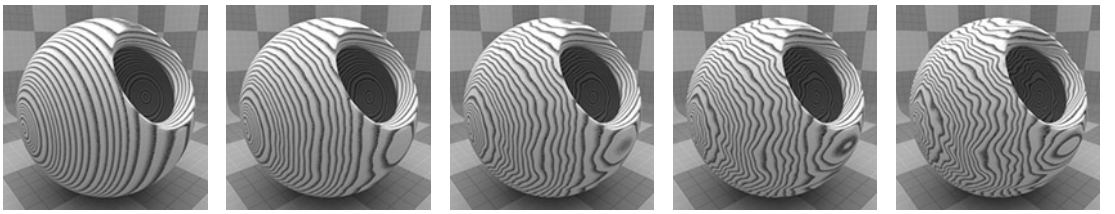

Ring Scale 25%


Ring Scale 50%


Ring Scale 75%

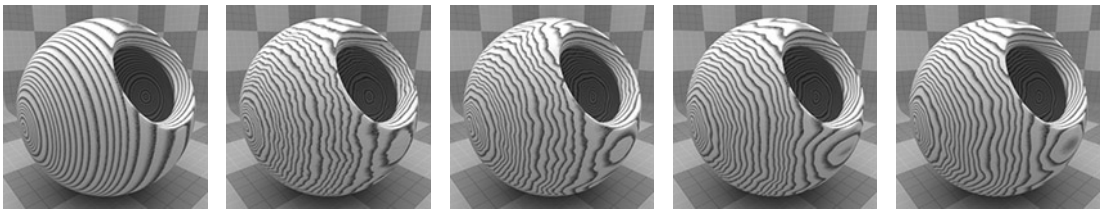

Ring Scale 100%

Waviness: This percentage value drives the amount of distortion that will occur to the rings in the texture.



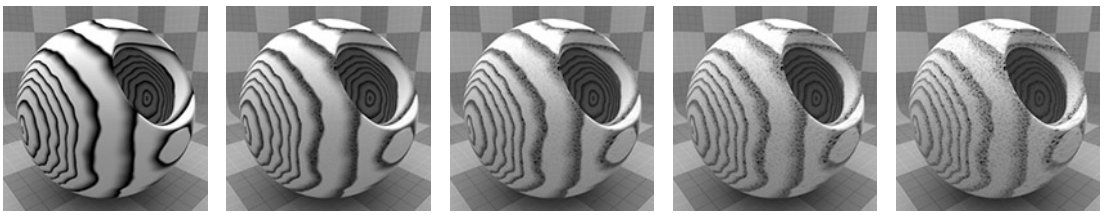
Waviness 0% | Waviness 25% | Waviness 50% | Waviness 75% | Waviness 100%

Wave Scale: This percentage value sets the size of the wave distortion that is used to deform the rings in the texture.



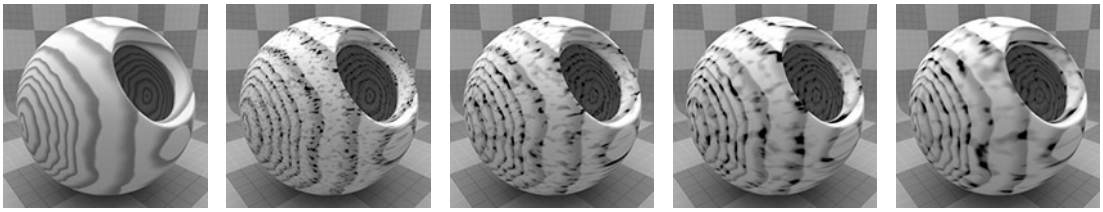
Wave Scale 0.1% | Wave Scale 25% | Wave Scale 50% | Wave Scale 75% | Wave Scale 100%

Graininess- This percentage value determines how visible the grain effect is when it is composited with the ring texture. Decreasing this value results in a more subtle grain appearance.



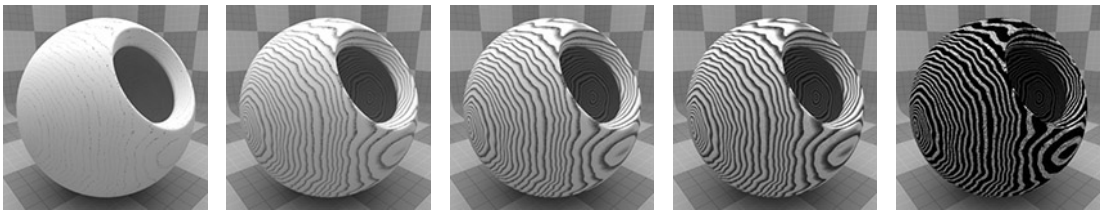
Graininess 0% | Graininess 25% | Graininess 50% | Graininess 75% | Graininess 100%

Grain Scale- This value determines how large the grain spots are.



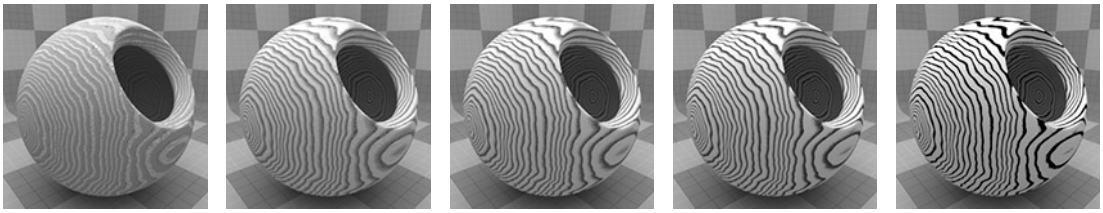
Grain Scale 0.1% Grain Scale 25% Grain Scale 50% Grain Scale 75% Grain Scale 100%

Bias: Increasing this value will cause the texture to favor the Ring color or value over the Filler whereas decreasing the value causes the Filler color or value to be favored.



Bias 0% Bias 25% Bias 50% Bias 75% Bias 100%

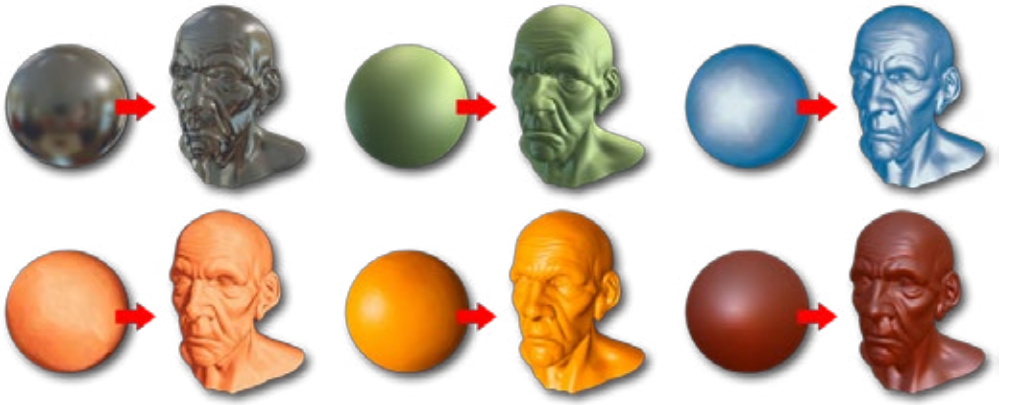
Gain: The Gain setting is similar to a gamma control that effects the falloff of the gradient ramp between the rings and filler values. Setting the Gain to 100% will create a very sharp falloff effect whereas setting the value to 0 would create a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.



Gain 0% Gain 25% Gain 50% Gain 75% Gain 100%

Matcap Shader

What if you could create a surface material and lighting environment simply by painting an object so that it looks like how you want your surface to appear? MatCaps, or "material captures" can do just that. Basically what happens is the shading of a spherical image is applied to the target surface by mapping corresponding normals from the captured surface onto that of the rendered surface. This opens up all sorts of interesting possibilities for non-photoreal image rendering demonstrated by these examples below; the original 'MatCap' is displayed on the left, with the resulting shaded surface shown on the right--

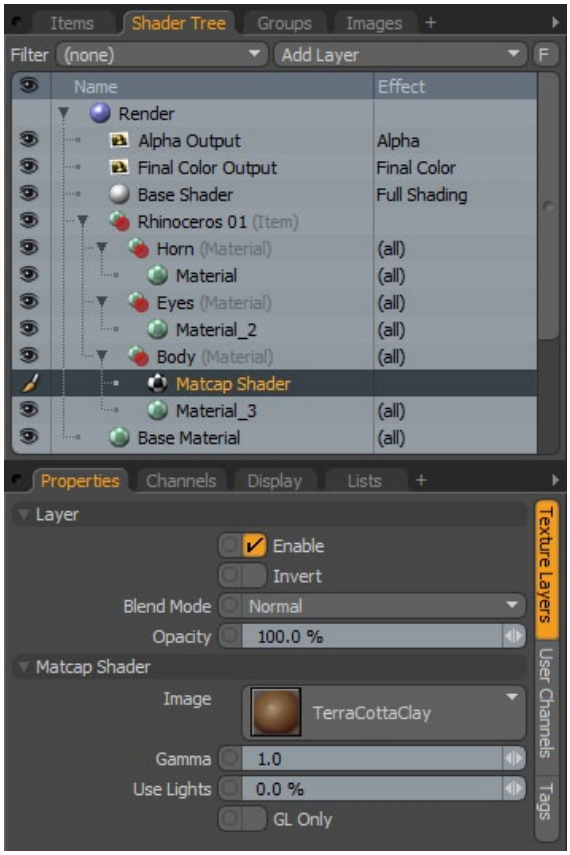


Creating Custom MatCaps

The 'Matcap' shader requires a spherical image as a source, users can create or photograph any spherical shape for application to the surface. However, make sure the final image is cropped in exactly to the outer edge of the sphere and that the image has the same width and height (square aspect ratio). When creating 'Matcaps' users will want to avoid surfaces with high-frequency details, these don't typically map well to complex surfaces, and they end up looking like the image is a reflection onto the surface instead; soft or subtle transitions work best. Resolution wise, make sure the image itself captures the necessary details for the target output resolution, but for soft, blurry Matcaps, sizes in the 512x512 to 1024x1024 pixel resolution should work just fine for most cases.

Applying Matcaps

Matcaps are applied with the Matcap shader, and can be added to the Shader Tree by simply selecting the Matcap item found under Add Layer button- "Add Layer > Other > Matcap". The Matcap layer works like any other Shader Tree layer; depending on its position in the tree, it can either affect the entire scene, or be limited to a specific surface through the use of a material mask. Also like other Shader Tree layers, it will overwrite the settings for layer that are below it in the tree (unless the 'GL Only' option is enabled). For more information on working with Shader Tree layers, please reference that page of the documentation. Once the Matcap shader is applied, users will need to specify an appropriate image to produce the effect. From there, users can adjust the additional settings depending on the intended use.



Layer--

Enable: Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across modo sessions.

Invert: Inverts the RGB values for the layer producing a negative effect.

Blend Mode: Affects blending between different layers of the same effect type, allowing user the ability to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current layer. Reducing this values will increasingly reveal lower layers in the shader tree if present, or dim the effect of the layer itself on the surface.

Matcap Shader--

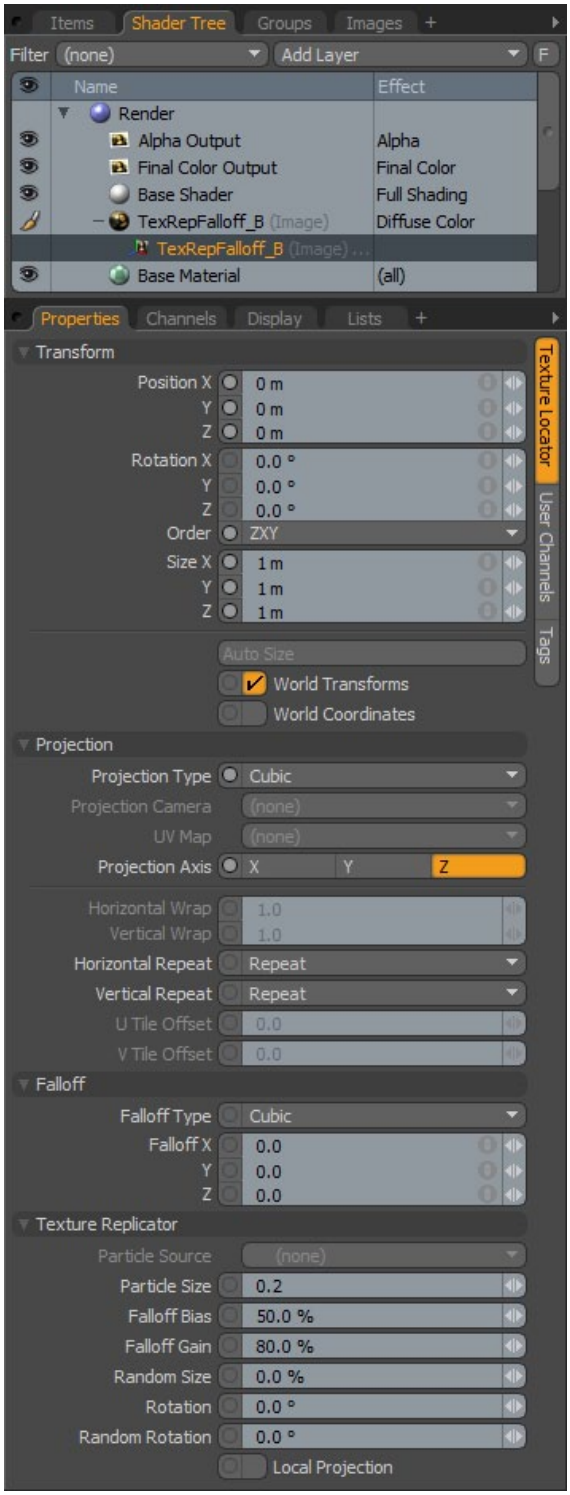
Image: This is where the users specify the spherical Matcap image that is to be applied to the target surface. If no images are loaded in the scene, a clip browser window will open allowing users to load an image for use, if images are already present in the scene, users can specify any existing images for use. Uses may need to load the image separately using the Clips Browser (image) viewport prior to applying a Matcap shader.

Gamma: When rendering surfaces with a Matcap applied, they may appear light or washed out. This is because modo always renders internally at gamma 1.0 (linearly) and then applies the user specified output gamma when displaying or saving the rendered result. Bitmapped images typically have automatic gamma applied to them in an image editor or digital camera (otherwise, they would look too dark to most users); because of this, rendering with modo's default gamma value (2.2) will make rendered images appear light or washed out as they have received double gamma. Therefore it is important to de-gamma the images. This can be done by inverting the gamma amount here, simply by dividing the render outputs Gamma amount (whatever it may be, 2.2 by default) by the image maps items 1.0 value. This can be done directly in the value input field using mathematical shorthand, the typed entry would look like "1.0/2.2" in the dialog, pressing enter will return the value "0.455" (modo does the math for you!). Now the rendered image will produce the correct result. When using the Matcap material strictly for OpenGL display, adjusting the gamma is unnecessary.

Use Lights: The shading of Matcaps surfaces are controlled directly by the Matcap image itself, and are not affected by scene lighting. The 'Use Lights' option provides users a means to incorporate the lights in the scene, blending them with the texture of the Matcap shader. A value of 0% disables all lights on the scene from having any affect on the surface moving up toward 100% which means the surface is completely controlled by the lights of the scene in effect changing the Matcap shader into a diffuse color channel for the surface.

GL Only: When this toggle is enabled, the Matcap shader will only affect the OpenGL viewport representation of the surface, and the rendered surface will display as normal (as if the Matcap shader were disabled). This is useful for when users wish to use Matcaps as a sculpting or modeling aid, giving users a clearer view of the geometry but still have the surface render normally otherwise.

Texture Locator



When any texture layer is created in the Shader Tree, image based or procedural, an associated 'Texture Locator' item is automatically generated and added to the items list and linked by the texture layers 'Locator' property. By default, Texture Locators are generally added to a folder (really a group locator item) called 'Texture Group'. This makes it easy to toggle the visibility of the locators in the 3D viewports, as some users find them distracting. Users may select the texture locator directly in the items list, or by selecting the locator icon in any 3D viewport. Most often though, users will select them via the shader tree; when any texture layer is selected in the tree, the locator is available in the named subtab on the right side of the items associated 'Properties' panel. Additionally, the locator may be toggled by the small '+' sign preceding the texture layers name in the Shader Tree, LMB+clicking the '+' icon reveals the layer.

Texture locators provide a very important function as they control the application of the texture layer over the rendered surface. A Texture Locator is like a regular 'Locator' Item, in that it has Position, Rotation and Scale information. Additionally there are settings specific to controlling the images projection method, meaning how it is mapped or applied to the surface. For certain

projection modes, such as 'Planar', users can interactively position textures in the 3D viewport by transforming the Texture Locator item itself. Other modes, UV in particular, completely ignore the positional and scale attributes as the mapping is defined by the UV map itself.

Transform--

Position: An Item transform that allows the user to numerically position the texture locator item in XYZ space, thus defining the position of the projected texture. This position defines the center of the textures overall volume as defined by the 'Size' setting. Useful in positioning of textures when using standard projections such as Planar, Cylindrical and Cubic, among others. Users can automatically define a textures Position and Size using the 'Auto Size' function.

Rotation: An Item transform that allows the user to numerically set the rotation of the texture volume. The Rotation transform originates from the locator items position value.

Order: Allows the user to set the order that rotations are applied to the locator item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock when animating.

Size: These XYZ values set the textures projection size, or boundary as the case may be defining the textures overall volume. Users can automatically define a textures Position and Size using the 'Auto Size' function.

Auto Size: This button automatically sets the 'Position' and 'Size' values; determined by the overall bounding box volume of the tagged surface(s).

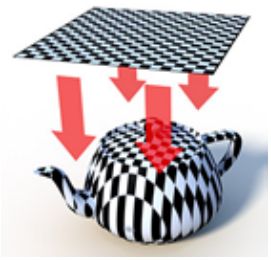
World Transforms: Typically, if you move a texture locator around, the texture will move over a surface that is sitting still, this allows for animated textures. If you parent the texture locator to the item and move the item around, both are moving in sync, so the texture stays still. With 'World Transforms' enabled, the parented texture will animate over the surface as if the surface was setting still and the locator was moving by itself. It is a subtle difference to World Coordinates.

World Coordinates: Typically, when you move an item, the texture moves with it, as if painted on. When 'World Coordinates' is enabled, textures will act as if frozen in place, with surfaces swimming through the texture. This option is extremely useful for texturing cloned or replicated items (that are NOT animated) with procedural textures as it provides the look of randomizing the texturing, without the need to modify any other surface attributes.

Projection--

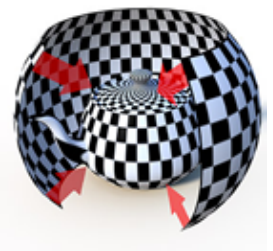
Projection Type: A projection defines how a texture, be it image based or procedural, is applied to a dimensional 3D surface. The various projection types can greatly affect the final rendered outcome. modo offers a variety of projection types (*illustrated below*).

Solid-- A 'Solid' projection will only be applicable for procedural texture layers. If you think of a procedural texture as a solid volume, this projection type will render a particular value at the intersection point of the surface within the virtual volume of the procedural. A block of marble is a good example, the surface will look as if it were carved from the procedural volume revealing the surface.

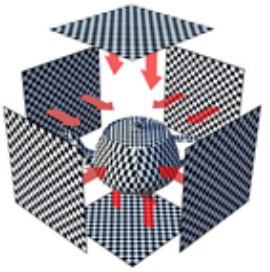


Planar-- A 'Planar' projection is similar in concept to a movie projector, but the associated image is projected onto the surface orthographically, meaning the projection rays travel perpendicular from the virtual projection plane onto the surface. This projection type is great for flat or nearly flat surfaces. Other instances will likely cause undesirable stretching of the texture. Users can specify the axis direction, either X, Y or Z, they wish the projection to face.

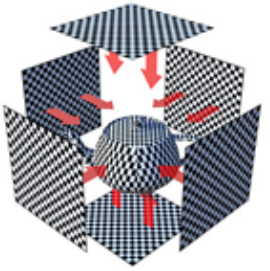
Cylindrical-- For the 'Cylindrical' projection, the texture image is warped into a cylindrical shape and projected onto the surface. Very useful in texturing cylindrical shapes, such as labeling on various cans and bottles. Surfaces perpendicular to the projection will produce undesirable stretching of the texture as exhibited by the lid of the teapot. Users can specify the axis direction for the cylinder along the X, Y or Z axes.



Spherical-- For the 'Spherical' projection, the texture image is warped into a spherical shape and projected onto a surface. Useful in texturing planets and round objects such as sports balls. Spherical projections are also useful in mapping 360° panoramic images for use as environment maps. Surfaces perpendicular to the projection will produce undesirable stretching of the texture as exhibited by the spout of the teapot. Users can specify the axis direction for the sphere along the X, Y or Z axes.

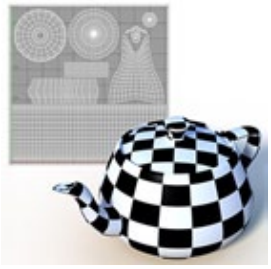


Cubic-- For the 'Cubic' projection, the texture is planar projected through a surface from all three axial directions, X, Y & Z. A polygon receives a certain projection based on its normal direction. This projection type is best on cube shaped objects, and occasionally on detailed surfaces where texture seams are not of great concern. The main drawback is for textures that have a proper orientation are flipped on reverse faces, this is resolved by using the 'Cube' projection method.



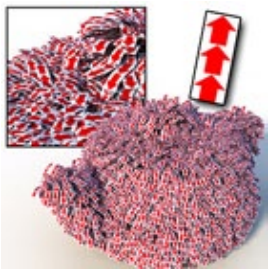
Box-- Similar to 'Cubic' projection, but the 'Box' projection type projects a texture in a planar fashion from all six directions (like an inward facing cube), eliminating reverse projections on rear facing polygons inherent to the Cubic method. This projection type is best on cube shaped objects, and occasionally on detailed surfaces where texture seams are not of great concern.

Front Projection-- A special projection type, useful in composing 3D elements into a photograph. A 'Front Projected' image (also called a plate) is projected out from a camera or lights position (as specified by the 'Projection Camera' setting below) often onto low resolution proxy geometry or as a background environment. If the cameras position and view settings are set appropriately, rendered elements can be made to match with elements of the projected background.



UV Map-- UV mapping is a technique where the user can precisely plot how a 2D flat image is applied onto a 3D surface by defining a 'U' horizontal and 'V' vertical position for each polygon vertex. Many primitive object are created with automatic UV maps. Otherwise, UV maps can be easily created using any of the various UV tools available in the 'Texture > UV Tools' menu. For surfaces with a UV map defined, this is the default projection method.

Light Probe-- Light probe images are images used to *light* scenes by way of image based lighting (IBL). They are typically high dynamic range imagery (HDRI) that when applied to a scene as an Environment (and Global Illumination is enabled), brightness information embedded in the image is used to light the scene producing very realistic and natural results. A popular method of of generating these HDRI images is based on taking multiple photographs of a reflective sphere. The 'Light Probe' projection will properly unwrap this chrome-sphere style of HDRI image, projecting it out as a proper spherical background.



Implicit UV-- 'Implicit UVs' are procedurally generated *implied* UVs used to apply image maps onto Fur fibers individually. Every fiber, when set as 'Strips' will fill the 0-1 UV space allowing appropriate images to be mapped onto the fibers, providing a means to make incredibly realistic grass and feathers, among other things.

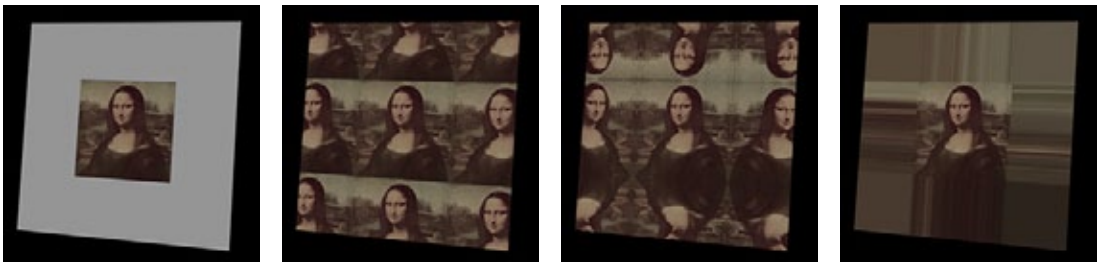
Projection Camera: When selecting the 'Front Projection' type one may use this setting to specify the camera to project from. Lights can be projected from as well.

UV Map: When selecting the 'UV Map' projection type, one may use this setting to specify an alternate UV map.

Projection Axis: Defines an axial direction for any of the primitive projection types -Planar, Cylindrical, Spherical, Cubic, Box and LightProbe.

Horizontal/Vertical Wrap: Controls wrapping amounts for UV image maps, effectively scaling the UV across the specified texture. A 'Wrap' value of 2.0 would repeat an image twice over the same UV space, while a wrap of 0.5 would stretch the image by half.

Horizontal/Vertical Repeat: Sets how bitmap images repeat beyond the defined image size/UV space boundary; several methods can be chosen: 'Reset' disables image tiling, 'Repeat' will tile repeatedly, 'Mirror' will tile the image flipping each subsequent image producing a backward and forwards type tiling, and finally 'Edge' will simply repeat the edge row of pixels. Users should note that these options are not visible in the GL Viewports.



Reset Repeat Mirror Edge

U/V Tile Offset: By default, modo uses only the 0-1 UV texture space, where 0,0 is the origin. Users can define an alternate UV area to use by offsetting the origin the amount determined by the 'Tile Offset' value.

Falloff--

Falloff Type: Falloff, when defined, will fade the texture layers opacity over the specified distance. There is cubic (3D), spherical (3D) and planar X, Y or Z.

Falloff: A setting of '0' disables falloff, when set to any value above '0', the falloff amount will act like a multiplier of the size. So a setting of 0.5 on a 1 meter texture scale will fade the texture to fully transparent over 2 meters from the locators position, a setting of 0.1 would scale it to fully transparent at 10 meters.

Texture Replicator--

The 'Texture Replicator' function works on basically the same principle as the geometry Replicator in modo, but provides users a means to procedurally clone an image map texture onto a surface while adding some randomization to the resulting clones. Consider the example below, on the left a seamless 'repeating' texture has been applied to the surface, but with the uniformity of the placement across the surface, the fact that it is a small repeating texture is painfully obvious. Utilizing the

exact same texture in the image on the right, a particle source has been defined and some falloff and Random Size and Rotation have been applied. While there is still an amount of regularity to the surface, the texture no longer exhibits the obvious tiling of the original.

Texture tiled across the surface looks repetitive.

Replicator randomizes tiles eliminating repetition.

To apply a texture Replicator to a surface, users must first assign an image map to the target surface, and in the texture layers 'Texture Locator' attributes, define a 'Particle Source', typically the target surfaces mesh item layer. Once specified, the texture is cloned to each vertex position across the surface. Users can then make adjustments to the Size, Falloff and Rotation options and then add some Random Scaling and Rotation as well.

Particle Source: Users can specify a mesh item layer as a point source for the cloned textures, when doing so, the Texture Replicator option is effectively enabled and its settings supercede all others of the Texture Locator. Once a mesh layer is defined, a single copy of the texture will be positioned and aligned to the vertices normal at each vertex point. Typically users would specify the mesh layer of the actual target surface with this option, however interesting affects can also be obtained when specifying alternate mesh layers. Keep in mind that the clones are aligned to the surface normal of the source mesh, so surface would need to be in somewhat close proximity and need to have similarly facing normals.

Particle Size: This option determines the size of the individual cloned images on the target surface. Larger numbers increase the clone sizes while smaller numbers decrease the resulting size.

Falloff Bias: Users can adjust this option to fade the texture out from its center toward the edge. Lower values keep the transparent edge toward the outer corner where higher values up to 100% push the transition toward the center.

Falloff Gain: This option determines the hardness of the falloff transition. Low values

produce a very soft fade over the Falloff while higher values increase the edge hardness up to 100% producing a crisp hard edge.

Random Size: Users can use this option to define a range of random scaling applied to the individual clones.

Rotation: Each clone is initially aligned to the vertex normal position (including its rotation) of the source, but users can use this option as an offset from the initial rotation of the clones.

Random Rotation: Users can use this option to define a range of random rotation applied to the individual clones.

Local Projection: The 'Local Projection' option, when enabled, warps the replicated images to the contours of the Particle Source geoemtry producing a unique effect that can be more apppropriate in some texturing situations.

TIP: Texture Replicators aren't limited to using a single image map. Once defined, users can specify a Group containing any number of images as the 'Image' source in the Texture Locator and the Texture Replicator will randomly grab a member of the group for each vertex location providing additional means to randomize the look of the surface.

Advancing the Shader Tree

Working with Shader Tree

The Shader Tree is both immensely powerful and tremendously flexible. It is this flexibility that can lead to it feeling unwieldy at times. Based on user feedback, it was a design requirement to allow materials and textures to be applied at any level in the scene; even globally if necessary, but as the saying goes, "With great power comes great responsibility". The result is user diligence is required to keep the Shader Tree organized and the effort necessary increases as the complexity of a project increases. Understanding how to work with layers in the Shader Tree and what each contributes is an important start in wielding the power that the Shader Tree provides. The basics are covered on the Shader Tree, Shader Tree Items and Layer Effect page of the documentation. Beyond the basics covered on these pages, it was important to add an additional page to go over some more advanced aspects of the tree.

Groups and Groups

The number one most important item in the Shader Tree is the Material Group item. In the 'Add Layer' function of the Shader Tree, it is simply referred to as a 'Group', but here I'm calling it a 'Material Group' to avoid any confusion with 'Groups'. The Material Group serves multiple purposes and understanding its few settings are extremely important in getting the most out of them. When users add a Material Group item to the tree, they are simply empty containers. These containers can hold any number of layers, including collections of other groups. In this case, they are strictly organizational devices, providing hierarchy to a scene, for instance associating like materials to an item in the scene, making it easier to navigate a complex tree. This can be illustrated by selecting a few layers in the tree and pressing the 'Ctrl+G' keyboard shortcut, grouping together like materials under an undefined Material Group item.

The Material Group item also provides the principal means for applying materials to surfaces by way of masks. By defining Items and/or Polygon Tags in the items 'Properties' panel, any contents (i.e. layers) within the Material Group can be limited to only affect geometry surfaces that carry that specific tag in the scene. This is done automatically every time a user applies a Material tag using the 'M' keyboard command. Users press 'M' opening the 'Polygon Set Material' dialog box. Defining a name and pressing 'OK' will generate a Material Group item in the shader tree automatically setting the 'tag type' and 'tag name' options to the particular name the user defined.

Masking Surfaces

While the Material Group is an easy way to surface geometry for rendering, its reliance on polygon tags limits what is possible with just Material Groups, therefore modo provides several other ways to mask textures that provide much finer control. The masking functions combined with the layering possibilities of the Shader Tree makes it simple to produce very complex surfaces with ease. Whenever a mask is applied underlying layers can be revealed.

Group Masks

The 'Group Mask' is a Layer Effect option, defined for a layer in the 'Effects' column of the Shader Tree. It allows the contents of an entire Material Group, regardless of complexity or number of layers, to be masked by using any single texture item. The texture can be a procedural texture such as a Noise layer, an image map or even a weight map using the Weight Map texture layer. The use of image maps provides very fine pixel-level control. Users can create these images externally using an image editor, or paint them directly in modo using the built in paint tools. The use of gradients opens up a number of interesting possibilities as well, where Material Groups can be masked by the numerous gradient input parameters, such as slope or incidence angle.

Layer Masks

The 'Layer Mask' result is similar to the masking functionality of the 'Group Mask', but it application is limited to a single layer within the Shader Tree providing even more granular control. There are two ways to apply a layer mask, either choosing the 'Layer Mask' effect and placing the layer directly below the layer the user wishes to mask, or by simply dragging and dropping the layer onto the target layer (therefore automatically defining it as a layer mask effect). Users can modify the settings of the texture layer, toggling the visibility of this kind of mask using the '+' icon preceding the layers name.

Alpha Channels

Modo opens and respects embedded alpha channels or layer transparency on imported image map layers. Their actual application can be a point of confusion because some users are surprised when they can't see-through their surface in the transparent areas of the image map. This is because modo respects the alpha channel for the image map as a texture layer (revealing any lower surface shading), but it doesn't serve double-duty as 'Transparency' (or 'Stencil' or 'Dissolve') for the surface when set as the default 'Diffuse Color' effect. Users would need to also apply the image as a 'Layer or Group Mask, or assign the Alpha channel as 'Transparency' (utilizing the 'Alpha Only' option of the images properties). Additionally, users can simply use the 'RGBA' layer

effect to designate both diffuse color and transparency in a single layer.

By virtue of how modo handles alpha transparency for a image maps, this makes it possible to layer any number of transparent images and combine the result into a single complex surface. Even when painting, using transparent image layers, Photoshop-like multi layered images are possible by stacking several image maps layers with Alpha channels. User can simply select the layer in the Shader Tree to designate which image receives the brush strokes while painting, leaving other layers unaffected.

Item Masks

Within the Material Group item, users can specify item masks in addition to the other setting. Item Masks can be extremely useful limiting the application of images or shading to a specific mesh item, without resorting to a bunch of different polygon tags. This makes it possible to apply different shading or image maps to an Instance simply by specifying the Instance items name in the Item mask property of the Material Group item.

Layering Surfaces

Not only can users layer images and textures in a single material, users can also layer tags for interesting results. The polygons tagged do not have to be identical, or even similar. Its therefore possible to create a 'Selection Set' that spans polygons from two separate material surfaces, define it as a mask and apply an image that straddles the two surfaces. Use a straight material tag for the two regular surfaces and then assign the smaller areas the Selection Set with a custom UVs to apply labels and such.

Cleaning Up the Shader Tree

On occasion, users may find the Shader Tree encumbered with unnecessary layers that need to be removed. Choosing which to delete can be difficult. The 'Purge Unused Materials' command will remove all Shader Tree material and/or texture layers that do not contribute toward the final rendered image, including those that reference unassigned Polygon Tags. These could occur when geometry is deleted, or new tags are assigned, as polygons can only hold a single material or part tag. It also possible when moving items layers into a scene from another that extraneous layers are added. To use, simply select the command from the menu bar 'Texture > Purge Unused Materials'.

Some Additional Notes

Users often forget that texture items such as image maps and procedurals can be added to lights and environments. By adjusting their 'Projection' types and 'Effect' settings, a whole range of interesting effects can be obtained.

Texture layers can be used to mask any other types of layer, including 'Surface Generators' and 'Fur'. The only layer that cannot be used as a mask for Surface Generators or Fur is the 'Occlusion' layer; since it requires rays to be fired to calculate, this only happens at render time, and it is therefore too late in the process to have produce the intended result. In these cases users should bake the occlusion layer out to a image map texture and then reapply the image map in place of the occlusion.

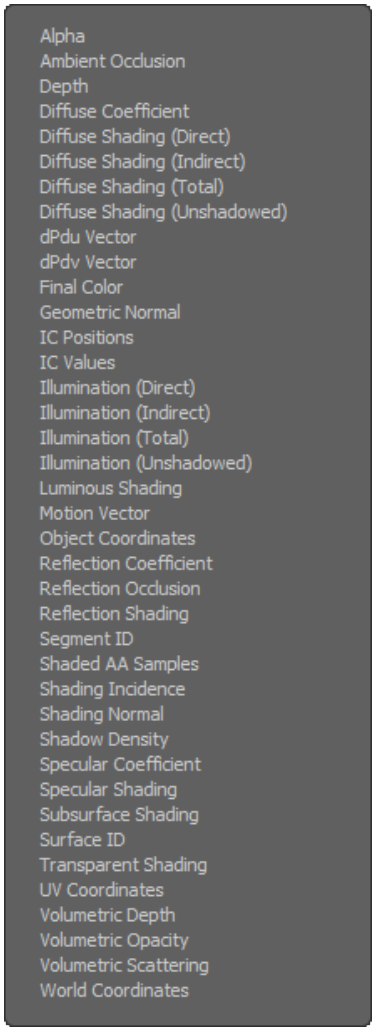
Layers

Layer Effects

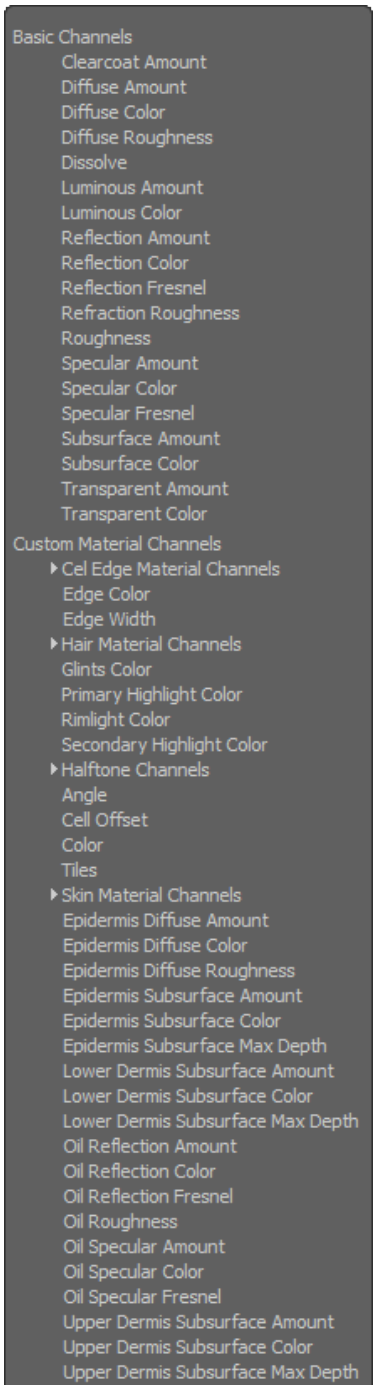
By default, any new texture layer added to the shader tree will modulate the 'diffuse color' channel, meaning the layer will affect the apparent visual color of the surface. Should you wish any layer to effect the surface differently, such as applying a bump or displacement map, this is specified via the 'effect' column of the shader tree. A layer can be set to modulate a specific effect, by way of a RMB click over the name in the effect column and selecting a new type from the pop up contextual menu. Layers can also be added and effect other items, such as changing the function of a render output, or using procedural textures to modulate a volumetric lights intensity.

Depending on the type of layer selected, different contextual menus will open, revealing different options. Each is fully covered on their respective document pages, click the titles below to view the associated documentation pages.

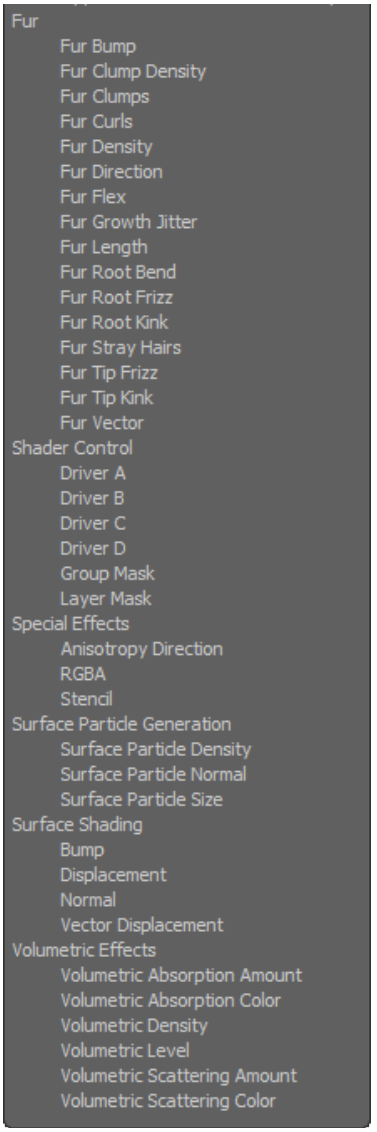
Effect - Render Output



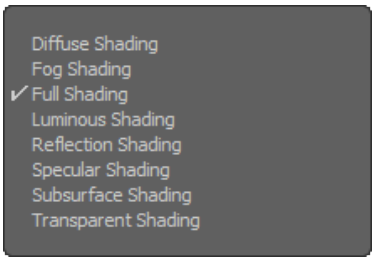
Effect - Texture Items



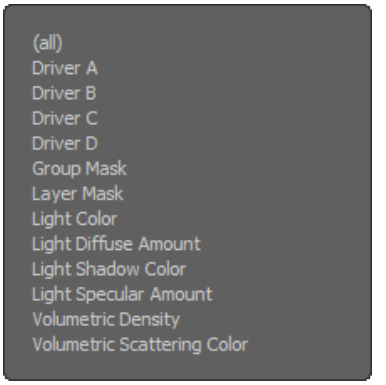
Effect - Texture Items Continued



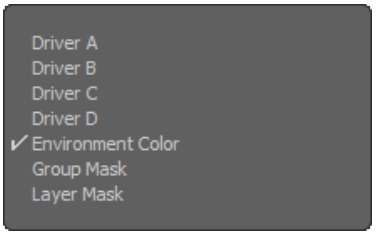
Effect - Shader Item



Effect - Light Item

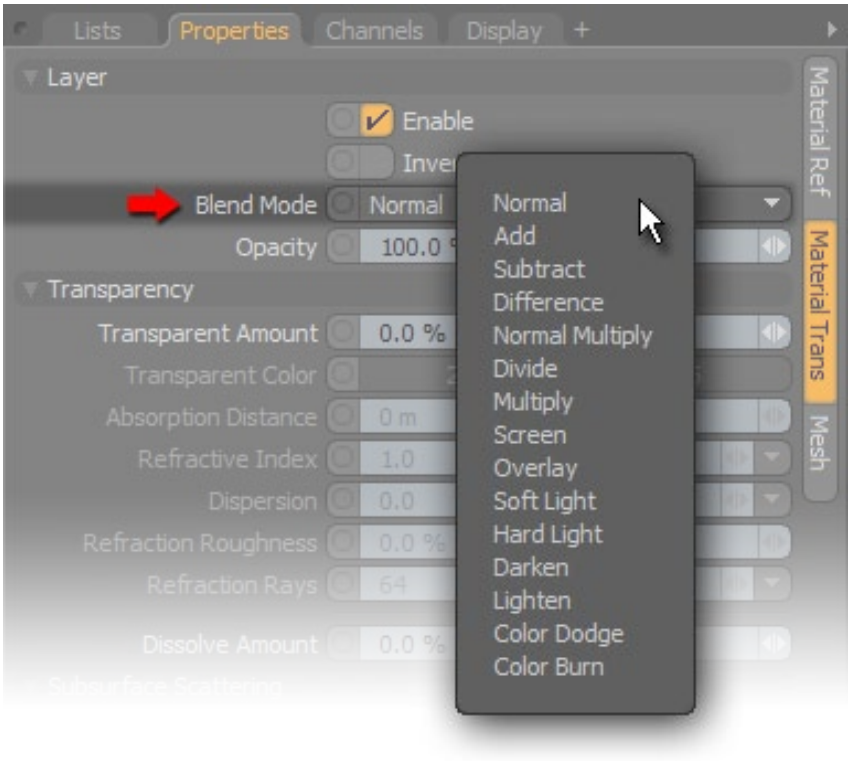


Effect - Environment Item



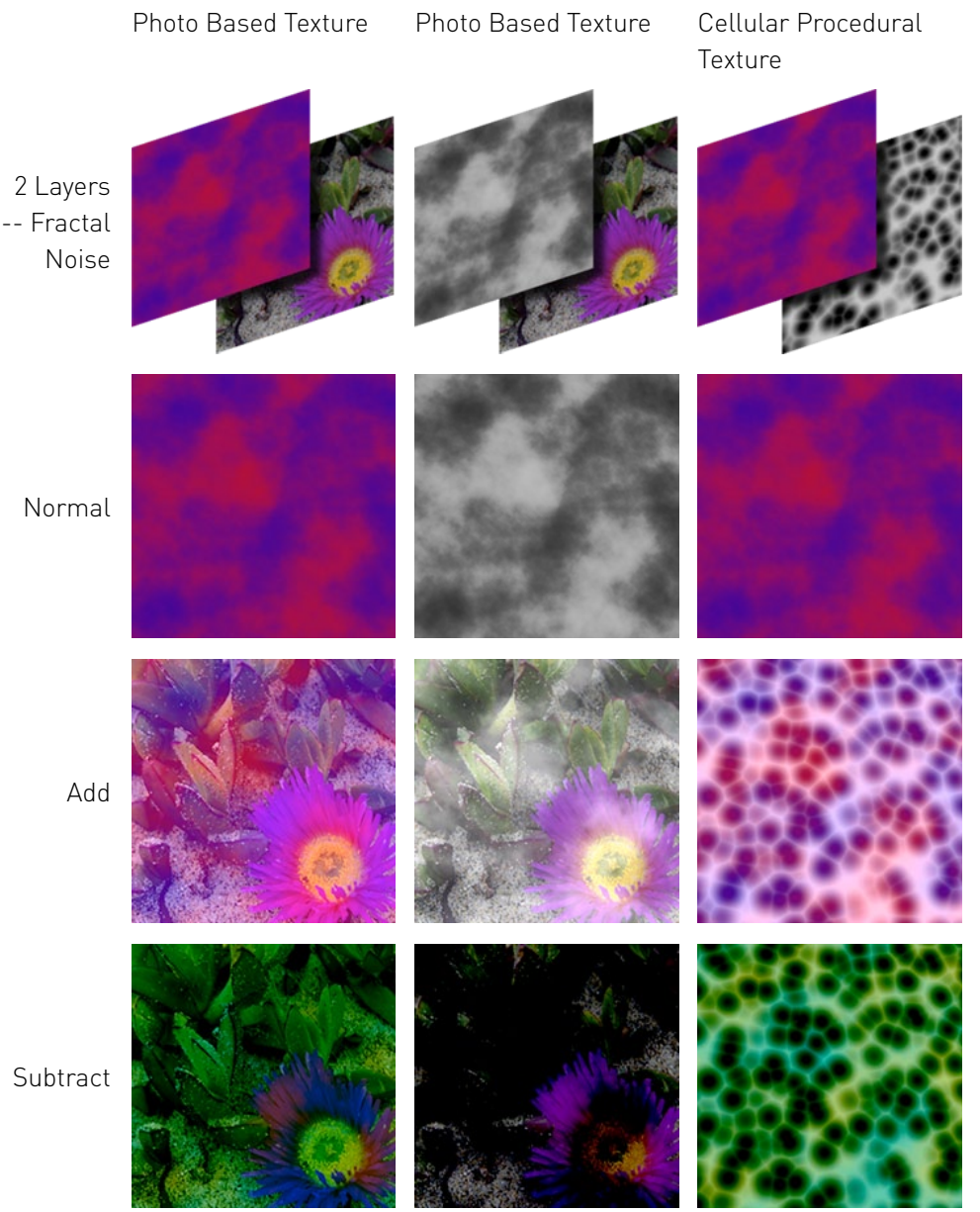
Layer Blend Modes

The flexibility and power of the Shader Tree is extended with the use of 'Blend Modes', a means of determining how multiple layers interact with each other. The key point being that there must be more than one layer in order for blending to work. The shader trees layered approach to shading surfaces works in a fashion similar to layers in an image editor such as Photoshop; any layer above another will obscure layers below unless one (or more) of three options is applied, 1. A mask is applied, either as a layer or a surface tag, this limits the affect of the layer exclusively to areas inside the mask, revealing lower layers outside of the mask. 2. The transparency of the layer is changed to reveal the lower layers. 3- A Blend Mode other than 'Normal' is specified, changing the way the layer interacts with the lower layers.



Nearly every item layer in the shader tree offers a Blend Modo option. The mode can be easily specified by LMB-clicking on the 'Blend Mode' option itself, this opens a contextual popup menu, users can then highlight the blend mode from the available options and LMB-click to select. Please note that Blend Mode results will only be visible in the final render or in a Preview viewport. Blend modes will affect the RGB color values (even if you just have black and white specified) and have a couple rules

that need to be adhered to -- 1. There must be more than a single layer of the same 'Effect' type, for example, a 'Diffuse Color' layer will only 'Multiply' with another layer specified as 'Diffuse Color' and 2. A layer with a blend mode will only affect other layers below it in the shader tree. The various modes available are as follows (with samples to illustrate the effect).



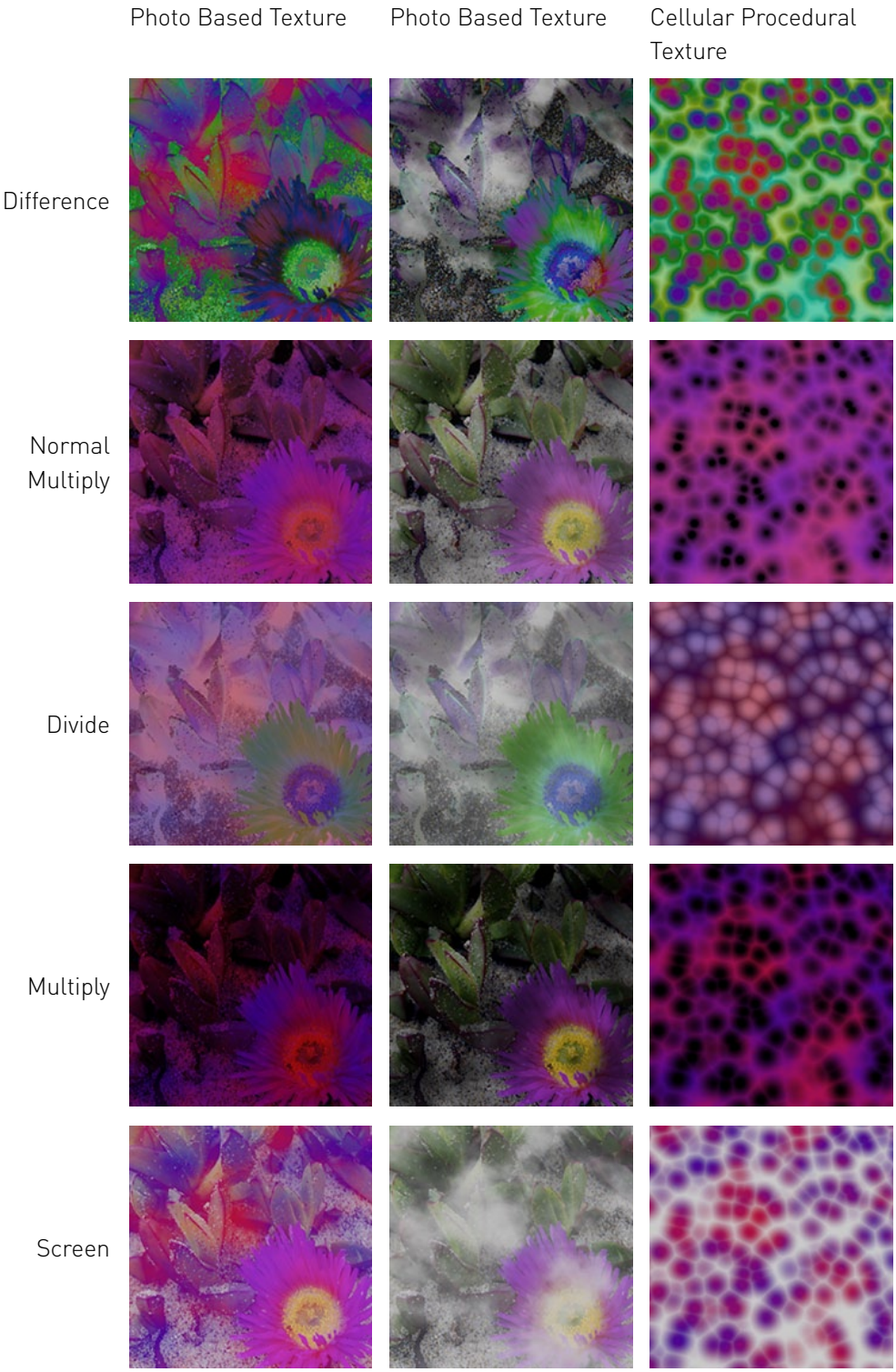


Photo Based Texture
Photo Based Texture
Cellular Procedural Texture
Difference
Normal Multiply
Divide
Multiply
Screen

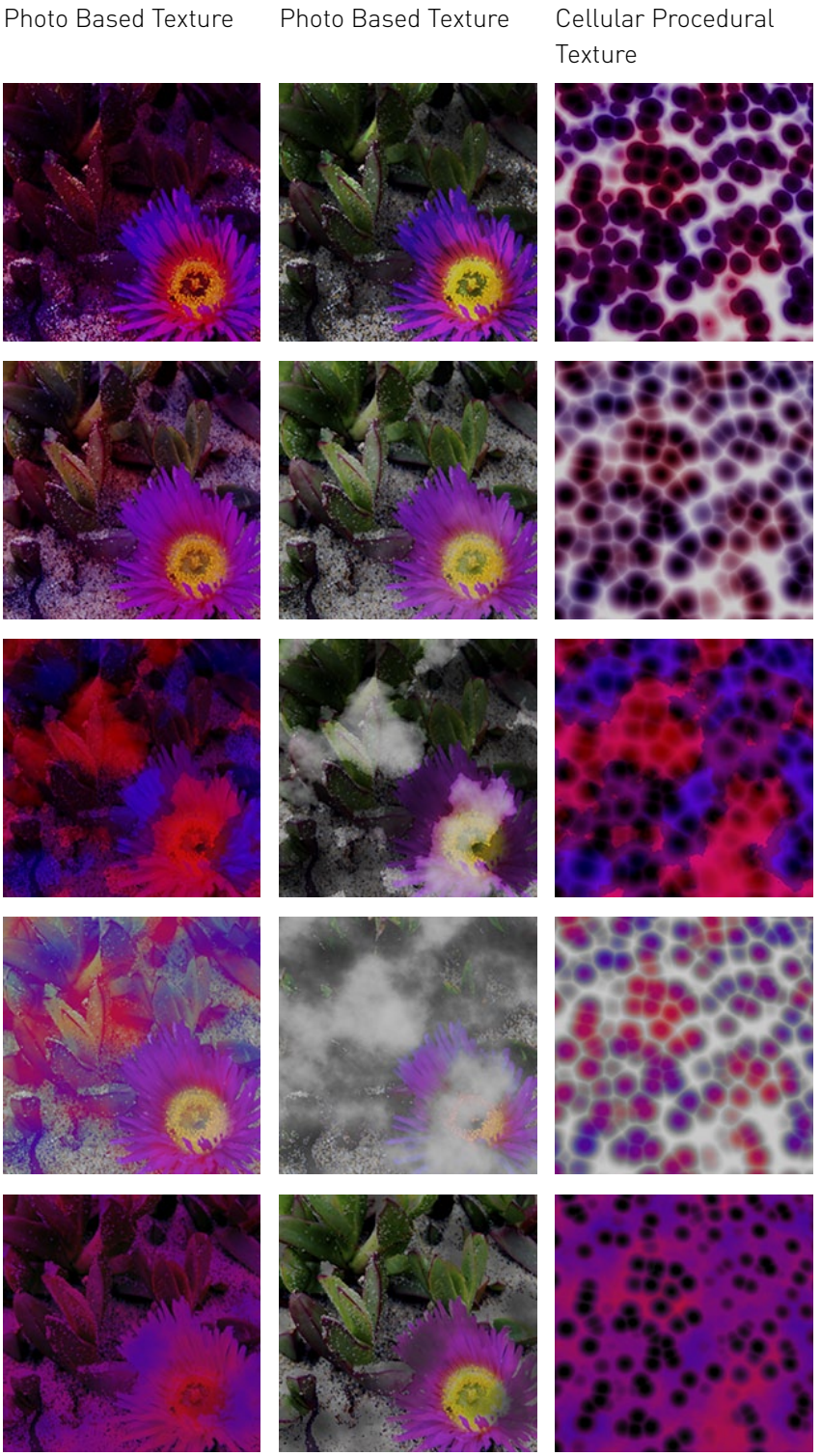
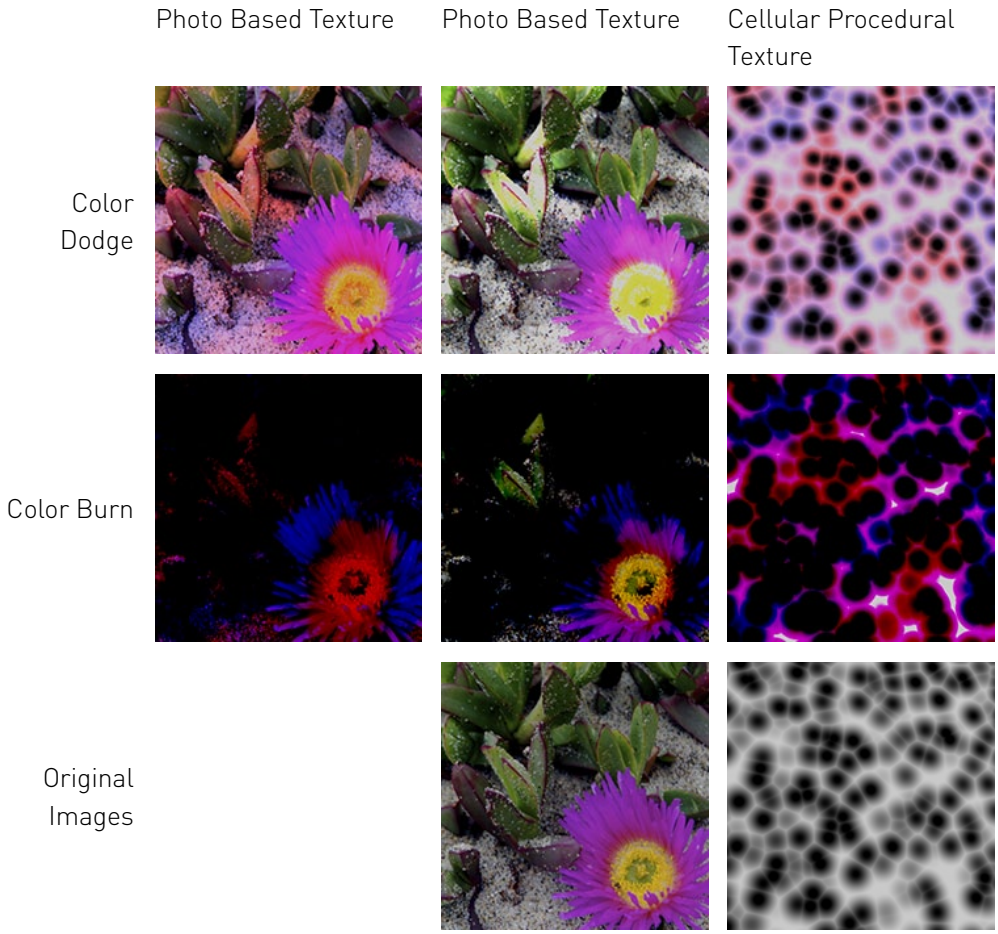


Photo Based Texture
Photo Based Texture
Cellular Procedural Texture
Overlay
Soft Light
Hard Light
Lighten
Darken



Doing the Math

Blend modes work by calculating the math (as specified) on the individual RGB channels. Take for instance 'Add', this function takes the values from the two layers and adds them together for the result. Depending on the texture and the blend mode, it may produce values beyond 100%. To reduce or eliminate clipping for certain effects that can't be driven higher than 100% (such as Displacement), users may also need to adjust a layers transparency. Let's explain how this works with an example-

1.) Cellular displacement layer

2.) Noise layer overwrites the Cellular layer

3.) Noise layer set to 'Add' produces clipped values

4.) Transparency of both layers set to 50%

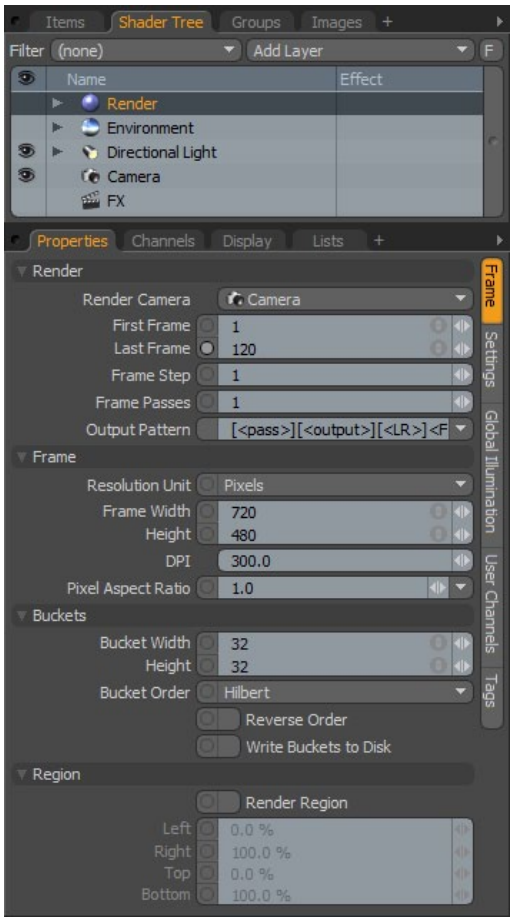
Example one shows a Cellular procedural layer is applied as a displacement map. In the shader tree, a Noise layer was added above it, also set as displacement for example two; this supersedes the lower layer in the tree in example one. To produce a combination of both, the noise layer was set to 'Add' as a blend mode, but this produces 'clipped' values in example three, this is because math that produces values beyond 100%, i.e. 75% + 75% = 150%, but there is no value higher than 100% (meaning all the way displaced), so that value is truncated to 100. By changing the Opacity value of both layers to 50%, each layer now contributes equally to the displacement. This can be biased either direction by adjusting the opacity values either direction, as long as both equal 100% you wont receive any clipped values (although, that may be the look you are going for). In some instances, certain effects can be driven higher than 100%, in those cases, it won't be necessary to adjust the the transparency.

TIP: A single 'Material' item produces values for multiple effects, such as diffuse color and amount, reflection, transparency, etc. Single texture layers in the shader tree can effectively 'Blend' with the values produced by the material item without the need for an additional matching layer.

Render Item

The 'Render' Item always occupies the top spot in the shader tree, this is no mistake considering its importance. The Render item can be thought of as a general container for global scene render settings. When the Render item is selected, its various attributes become available in the Properties viewport. It is broken into three sub tabs of related settings. The 'Frame' tab allows users to adjust render settings specific to the rendered frame itself such as its dimensions. The 'Settings' tab allows users to adjust render settings specific to rendering quality such as anti-aliasing. The 'Global illumination' tab allows users to adjust render settings specific to Global Illumination, a rendering technique that utilizes bounced light to produce incredibly realistic rendered images.

Frame



Render--

Render Camera: The 'Render Camera' popup option allows the user to select a specific camera from which the rendered image will be created. While a basic modo project begins with a single default camera, additional cameras may be added within the Item List by clicking (new item) and selecting 'Camera'. These cameras, like any item in the mesh list, can be renamed by RMB-clicking and choosing Rename. By default each new camera is assigned a numeric differentiator such as Camera(2), Camera(3) etc. Once additional cameras have been added to a project the Render Camera popup can be used to select them.

Frame Range: In animated scenes, the frame range specifies the sequence of frames that will render when you use the 'Render Animation' command. The top number specifies the first frame rendered, while the lower number specifies the last frame that will render.

Frame Step: When rendering out animated sequences, the frame step value will skip the specified number of frames. Essential for testing rendered sequences, rendering with a frame step of 2 (every other frame) would cut the overall render time in half, a frame step of 4 (every fourth frame) would cut the overall render time by three quarters and so on, while still giving a good idea of the final animated scene albeit at a lower frame rate. This function is relative to the Frame range. If you specify frame 7 as your first frame with a step of 2, frames 7,9,11,13 and so on would be rendered.

Frame Passes: The 'Frame Passes' option controls how many times a single frame is rendered (Not to be confused with 'Render Passes' which are entirely different). The results of each pass are evenly blended to make up the final frame. The random numbers used to generate fur are varied for each pass, so the resulting blended fur

will look softer. If motion blur is enabled, then each frame pass will cover just part of the exposure. For example, if there are four frame passes then the first pass will cover the first quarter of the interval from shutter open to shutter close. Increasing this value above '1' will increase render times significantly, so unless users are in need to softer fur or higher quality motion blur, there is no added benefit to increasing this value.

Output Pattern: These pattern options provide users a means to automatically name files when saving rendered animations, as either flat images or layered images. Instructions appear in the "< >" brackets, and those in the "[ ]" square brackets are ignored in instances that don't apply.

<LR> adds an 'L' for Left or an 'R' for Right when rendering stereo images.

<output> names file for render output item

<pass> names the file for the render pass

<F> Frame number (add extra 'F' for extra leading zeros).

So in practice, you could put Robot_[<LR>]_<pass>_<FFFF> and the resulting file name for the left render of the 'BG' named render pass would be "Robot_L_BG_0001.exr".

Frame--

Resolution Unit: This popup allows you to select how the rendered frame's image size is specified, either as pixels, or as inches.

Frame Width/Height: With Frame 'Width' and 'Height', the user can specify the size of the frame rendered when you invoke the 'Render' command. Dependant upon the resolution unit, if 'Pixels' is selected, then you can specify the size in pixel units, or if 'Inches' is specified, the user can specify the printed resolution of the rendered image in inches. When set to 'Inches', the DPI setting is used to determine the number of pixels rendered. The typical video size of 640 pixels by 480 pixels is the default size.

DPI: The DPI input field allows the user to directly determine the number of "Dots Per Inch". This is particularly useful when defining image height and width using the 'Inches' resolution unit, as it will be used to determine the final pixel resolution of the rendered images. It will also be added to files on export determining the eventual printed size of the image.

Pixel Aspect Ratio: Most computer screens, by default, use a 'Pixel Aspect Ratio' of 1 which yields "square pixels", what most people are used to. However, some target media use non-square pixels to display images such as NTSC television (0.90). Use this setting to properly render the image for its intended destination. Note: modo will still display the image as if they were square pixels, so rendered images will look squashed/

stretched depending on this setting.

Buckets--

Bucket Width/Height: The bucket size defines the number of pixel values calculated concurrently and is controlled via the Height and Width settings. The default value of 32 by 32 means that each bucket will be 32 pixels wide and 32 pixels high. Using larger buckets will increase the amount of memory required but can result in speed improvement. However, if multiple threads are being used it is important to remember that using smaller buckets can result in a better load balanced total frame. It is likely that when using multiple buckets they will not finish at exactly the same instant therefore there will usually be one bucket/thread computing when the other bucket(s)/thread(s) are finished. If the buckets are very large the amount of time the render engine is using only one thread will increase which of course decreases the impact of the additional processors on final render time. One might come to the conclusion that they should set the bucket size as small as possible, this is incorrect as well, system performance may decrease with the overhead to manage the buckets, while reducing anti-aliasing quality (as bucket borders cannot be anti-aliased to their neighboring pixels outside the bucket; decreasing the size of the bucket increases the number of 'border buckets'). Luxology Chief Scientist Allen Hastings, concludes that the 32 by 32 pixel bucket is typically the ideal size bucket when rendering approximately video resolution sized frames.

Bucket Order: The bucket order determines the pattern in which the buckets are rendered. In some cases there can be a small performance benefit to certain patterns. In particular the Hilbert pattern is designed to maximize bucket edge concurrency to reduce the amount of data that must be purged and loaded whereas Random would be most likely the least efficient in that regard. Aside from some slight variance in performance, using different bucket orders is simply entertaining and can be used to impress your friends who use rendering engines with boring fixed render patterns.

Rows-- Renders the buckets in rows from left to right starting and the top and working down.

Columns-- Renders the buckets in columns from top to bottom starting on the right side and working to the left.

Spiral-- Renders the buckets from the center spot outward in a clockwise spiraling pattern.

Hilbert-- Uses a specially formulated pattern that snakes across the screen making certain that there is as much concurrency from bucket to bucket across the entire frame. This pattern can result in slightly improved render performance on

certain hardware architectures that are not as efficient with bus bandwidth as it will reduce the amount of data being loaded and purged from bucket to bucket.

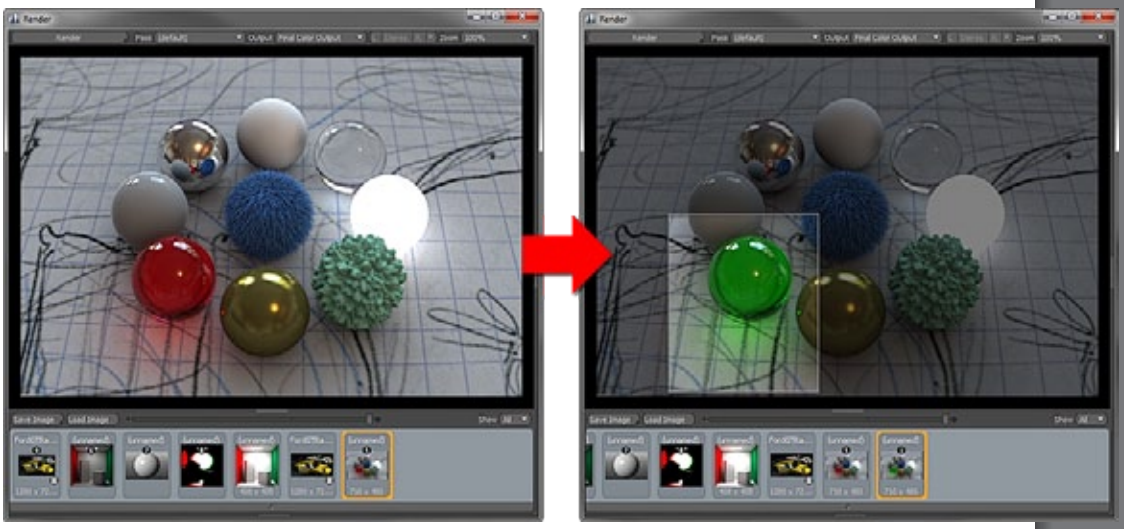
Random-- Renders the buckets in a random order across the screen.

Reverse Order: When this option is checked the order of buckets will be reversed so that, for example, the Columns pattern would start at the bottom right hand side of the image and work its way bottom to top in columns towards the left of the screen.

Write Buckets to Disk: Designed to facilitate renders of enormous resolution, this option will cause each completed bucket to store its frame buffer data on disc rather than in system memory. The frame buffer of an image contains a tremendous amount of data and can become quite large for even video resolution images. For instance a 640 by 480 pixel image will require upwards of 5 megabytes to hold the frame buffer. Doubling the image resolution will quadruple the number of pixels and memory requirement for the frame buffer. When rendering images for large format media such as printed billboards or IMAX film, the frame buffer size can become unwieldy. Activating Write Buckets to Disk will drastically reduce the memory overhead.

Region--

The 3D process is one of constant iteration and refinement. Rendering entire images repeatedly is a significant waste of time and a disruption to the creative workflow. To alleviate some of this pain modo provides the 'Render Region' rendering functionality. Users may use the Left/Right/Top/Bottom functions to specify the region manually, or use the 'Render Region Tool' to specify an area by dragging in any 3D viewport. Additionally, the function can activated through a button of the 'Render Display' window, under the options tab. When *enabled*, the next time a render command is invoked (such as by pressing 'F9'), only the area within the region specified will be rendered. The function can be disabled by toggling the 'Render Region' setting.



Render Region: This button toggles the 'Render Region' function on and off.

Left/Right/Top/Bottom: Left/Right and Top/Bottom percentage settings work in conjunction with each other to specify the total rendered region width and height. One could easily render the left half of the image, regardless of the output resolution or aspect ratio, by setting the 'Right' value to 50%. One can also easily set a specific area by click dragging over the rendered image in the Render Preview Window. When you do that, the render region toggle is automatically activated. Outside of disabling the function with the 'Render Region' toggle, a single click anywhere on the 'Render Display' window will restore the render to the full frame.

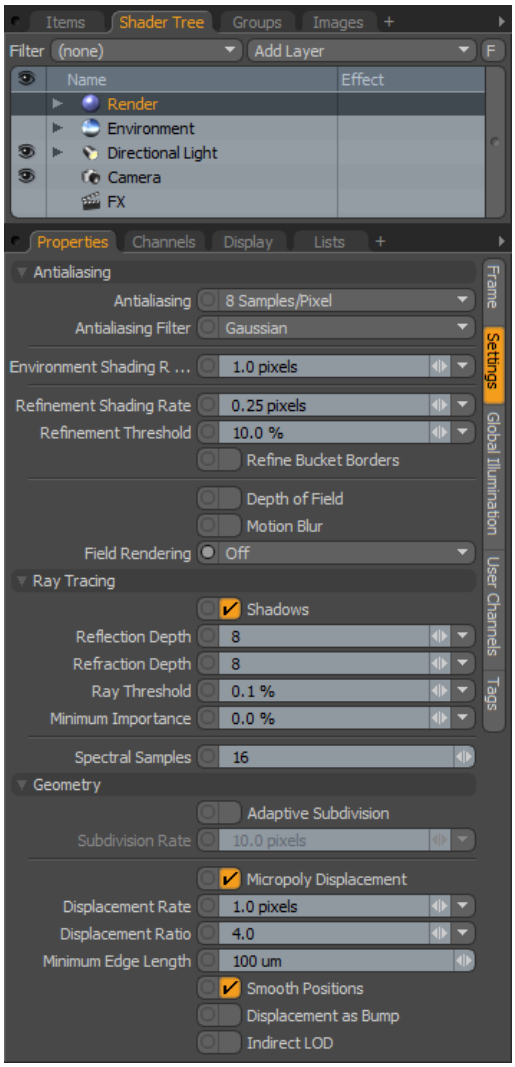
Render Region Tool

The render region tool provides users a means to easily select a region within a 3D viewport limiting the RayGL rendering to a specific area of the frame. As the RayGL option can be applied to any 3D viewport, the Render Region tool further streamlines the process of iteration and refinement when creating a scene. Users can find and activate the tool in the menu bar under "Render > Render Region Tool". Users need only to select the command, and then LMB+click and drag within a 3D viewport to define the region. One a region has been defined, enabling the RayGL function will limit the subsequent updates to the area inside frame. Users can adjust the quality settings of the RayGL option using the controls found in the Preferences panel. Dropping the tool does not disable the limited region so users may continue to work on a scene while limiting the region of the RayGL rendering. To disable the 'Render Region' option, users can toggle the 'Enable Option' found in the tools properties panel or disable the 'Render Region' option in the Render Items 'Frame' subtab.

TIP: The 'Render Region' function is also very useful when a user wants to render a rather large image for print but is having difficulty doing so. Use 'Render Region' to render smaller sections of the total image and assemble the pieces in an external image editing application.

Settings

The 'Render' Item always occupies the top spot in the shader tree, this is no mistake considering its importance. The Render item can be thought of as a general container for global scene render settings. When the Render item is selected, its various attributes become available in the Properties viewport. It is broken into three subtabs of related settings. The 'Frame' tab allows users to adjust render settings specific to the rendered frame itself such as its dimensions. The 'Settings' tab allows users to adjust render settings specific to rendering quality such as anti-aliasing. The 'Global illumination' tab allows users to adjust render settings specific to Global Illumination, a rendering technique that utilizes bounced light to produce incredibly realistic rendered images.



Antialiasing--

There are three main settings that control the "quality" of rendered pixels. These include Antialiasing, Refinement Shading Rate and Refinement Threshold. To create the highest quality imagery modo does all of its computations at the sub-pixel level; taking multiple tiny samples within the space of a single pixel and then averaging their result. However, rather than computing the entire image this way over and over; the "brute force" method, modo uses elegant logic to determine how to best create the image. To balance quality with performance modo provides methods to determine which areas of the image should receive the greatest number of samples.

Antialiasing: At the top most level the Antialiasing control dictates the numbers of "samples" taken per pixel. modo allows you to specify from 1 to 1024 samples per pixel in powers of 2 (1,2,4,8,16 and so on up to 1024). We can think of these samples as sub-pixel fragments; imagine a pixel as a square that can be divided into smaller squares.

When Antialiasing is set to 8, the pixel is broken down into 8 pieces, each of these pieces is evaluated and then averaged together to give us the final high-quality pixel. Since it would be expensive time wise, and overkill, to evaluate every single pixels in an image repeatedly we have some additional controls to limit the areas that receive the maximum number of evaluations.

Antialiasing Filter: This popup option allows you to choose different antialiasing filters. The antialiasing filter determines the pattern to use when evaluating a pixel. The default is 'Gaussian' which performs very well in most cases and offers a good balance of performance and quality. Box, Triangle, Catmull-Rom and Mitchell-Netravali are additional options. Catmull-Rom will generally yield slightly 'sharper' results than Gaussian while Mitchell-Netravali offers good results when dealing with problematic moire created by fine texture patterns.

Environment Shading Rate: This option specifies the approximate spacing between shading samples measured in pixels, used for the environment. It also acts as a cap on the rates specified in the various shader items, allowing shading detail to be increased across the entire frame. The main reason one would want to lower this value would be to improve the motion blur of volumetrics. (*In previous versions of modo this option was called the* 'Coarse Shading Rate').

Refinement Shading Rate: The 'Refinement Shading Rate' control tells the engine how finely to evaluate any pixel that falls within the 'Refinement Threshold'. When a pixel is being evaluated, first, it is rendered through the standard 'Antialiasing' application, neighboring pixels are then tested for 'Refinement Threshold'. If a pixel is within the threshold, then the pixel is finished and the engine moves on to the next, but if it is outside of the threshold, then the 'Refinement Shading Rate' kicks in, and the pixel is further evaluated. The value itself basically acts a as a multiplier of the pixels size, so a setting of .5 would split the pixel in half both ways resulting in 4 additional shading samples calculated, a setting of .1 would split the pixel 10 times both directions resulting in 100 additional shading samples calculated, overkill in most cases.

Refinement Threshold: The Refinement Threshold is essentially a contrast tolerance which is to say this control dictates how much contrast between adjoining samples is acceptable. Set to 100% the Refinement Threshold would do nothing at all as it would accept brightness steps as high as 255 (the upper bound of a 24 bit image). Set to 0% no pixels would be inside the threshold and every pixel would be refined. The percent amount spans the 0 to 255 brightness steps so a threshold of 50% would accept pixel contrasts of 128 steps whereas a setting of 10% would only tolerate 25 brightness steps between samples. Just like the refinement shading rate, we see that an increase in quality when set to lower values, but it also increase computation requiring longer render times.

Refine Bucket Borders: Due to the 'bucket' nature of modo's render engine, adjacent pixels in neighboring render buckets can't be used to decide whether a border pixel needs further refinement because they are either not rendered yet, or fully rendered and no longer active. By enabling 'Refine Bucket Border' modo will tell modo to assume all border pixels in a given bucket require refinement.

TIP: The Antialiasing setting's effect is most apparent on geometric edges, a setting of 8 is generally acceptable and is a good trade off between quality and speed, though in reproducing effects such as 'Depth of Field' or 'Motion Blur' you will need to increase this value possibly all the way up to 1024 to produce a pleasing result. If you find your shaded edges, such as cast shadows, refractions, reflections, procedural and image based textures are aliased (not smooth), then reducing the shading rate value or lowering the refinement threshold will help to increase shaded edge quality. If you are finding specific areas of an image to be noisy, that cannot be solved by Antialiasing and Refinement, such as soft reflections, blurry refractions, soft shadow edges, subsurface scattering, it may simply be a case where raising the number of samples specific to the effect will eliminate the noise. These setting can be found in the associated properties panels. Always keep in mind though, as more calculations are taken into account, the longer an image will take to render.

Depth of Field: This setting is the global toggle to enable/disable the 'Depth of Field' function when rendering. Just as with real world cameras the depth of field effect is highly dependent on the Cameras F-Stop which can be adjusted on the Camera Items properties along with the 'Focus Distance settings which defines the focal plane, the area of acceptable focus in an image. Lower F-Stop values will increase the softening of edges as they recede from the focal plane while higher values will decrease this effect. As edges become increasingly softened, it may be necessary to also increase antialiasing levels to smooth out any artifacts that appear. While modo performs better than most rendering engines when rendering depth of field, it is still possible to create depth of field effects that are so strong it is impossible to reduce the noise.

If you run into this case, some options to consider are rendering the image larger than the actual target resolution and then scale the image down in an image editor. This effect is called "super-scaling" and will improve Depth of Field as well as other antialiasing effects. But an easier and less time consuming method would be to apply Photoshop's 'Lens Blur' filter (or equivalent) that will work in conjunction with a Z-Depth buffer to smooth out problematic renders. modo can create a Z-buffer and put it into the alpha channel of the image to be used for just this sort of effect. With the Render Item selected, you can use the 'Add Layer' drop-down to add a render output, set to 'Depth'. Within the 'Depth' properties you will need to set the maximum depth of your scene to maximize the values available.

Motion Blur: This setting is the global toggle to enable/disable the 'Motion Blur' function when rendering, an effect that simulates the blurring of fast moving objects in a photographed image. When enabled, modo will take 'time' samples forward and

backward from the rendered frame blending them together to create the blur effect. Depending on the strength of the effect or the speed of motion in your frame, you may find it necessary to increase the number of antialiasing samples. Adjustments to Blur Length (like adjusting the exposure time for more or less blur) and Blur Offset can be found in the Camera Items properties.

Field Rendering: For scanline televisions (older tube televisions) there were originally two half frames that made up each whole frame, interlaced on top of each other to make a single image. Enabling the 'Field Rendering' option will properly render two discreet moments in time and interlace the resulting half-frames into a single image (in essence doubling the frame rate). This option is useful to compose modo generated elements into video that was originally captured this way. When using Field Rendering, it is recommended that users switch to 'Box' antialiasing to eliminate field cross-talking. Also, post effects, such as 'Bloom' will be generated on a whole image basis, so should only be applied in a compositing package. Modern film and video capture is largely progressive, where each frame represents only a single moment of time, so in most cases Field Rendering will not be necessary.

Ray Tracing--

Shadows: Enabled by default, the 'Shadows' checkbox directs modo to globally render all shadows cast by direct lights. Direct lights can be any of the light items, such as Distant Lights, Area Lights, Point Lights, Spot Lights, Dome Lights Cylinder Lights, and Photometric Lights. Shadows cast by luminous polygons when Global illumination is enabled are not affected by this setting (their shading is part of the indirect illumination). Shadows can be disable on a per light basis in each lights properties panel, by setting shadow type to 'none'.

Reflection Depth: modo renders images by ray tracing, sort of like shooting tiny arrows out from the cameras lens and seeing what they hit. If a ray hits an object that is reflective, that ray bounces off the object until it hits another, should that surface also be reflective modo will again bounce that ray until it finds a surface that is not reflective. If the ray doesn't find a surface that isn't reflective, the 'Reflection Depth' threshold stops the ray from continuing on its path keeping modo from infinitely tracing rays between reflective surfaces. The default of 8 bounces in fine for most scenes, and is a good compromise between render accuracy (quality) and render time. In some instances you may find a need to increase this setting to accurately portray all the reflective surfaces in a scene, or find you need to reduce it to shorten render times.

Refraction Depth: Refraction Depth, like Reflection Depth is calculated by ray tracing. When a ray enters a refractive surface with some transparency, the ray bends the appropriate amount and travels until it hits its next surface, if that surface is also

refractive, again the ray is bent and sent along its path until it hits a non-refractive surface or is stopped by the Refraction Depth threshold from traveling further. The default setting of 8 is often good enough for most scenes and is a good compromise between render accuracy (quality) and render time. In scenes with many transparent surfaces, you may find a need to increase the setting to accurately portray all your transparent surfaces, or find that you need to reduce it to shorten render times.

Ray Threshold: For some scenes, firing all those refraction and reflection rays can get expensive time wise. What if there were a way to eliminate those rays that have little to no impact on the final rendered image but are just as expensive to calculate as the rays that are important. Ray threshold does just that. As rays are fired, they are given an importance value that can decrease or increase at each bounce depending on the surface value. Should the rays importance fall below the 'Ray Threshold' modo randomly decides to kill the ray or trace it further (up until the total depth value specified) This random 'killing' of rays reduces the overall number and depth of rays traced but has little visual impact on the final image. Increasing the Ray Threshold value will further eliminate rays, though at the expense of image quality.

Minimum Importance: The 'Minimum Importance' option specifies the how important rays are when shading a set of anti-aliasing samples. The default value is 0%, in which case Importance is the number of samples in the set being shaded, multiplied by the total number of samples in the pixel. Increasing the minimum importance can be used to avoid artifacts that can occur with finer shading rates when 'Clamp Colors' is enabled, but render time will increase, possible significantly. (At a values of '100%', each antialiasing sample is geven the same importance as a full pixel!)

Spectral Samples: When rendering any transparent refractive surfaces with 'Dispersion', the 'Spectral Samples' value determines the number of rays that each dispersion ray is broken into. The higher the number of rays, the smoother the dispersion calculations will be, at the expense of longer render times.

No Dispersion
- 1 Minute

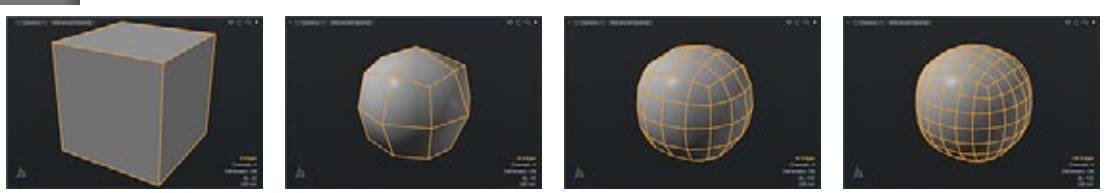
16 Spectral Samples
- 3 Minutes

256 Spectral Samples
- 41 Minutes

Geometry--

The first two settings under the Geometry section are global controllers that relate to Subdivision Surface (SDS) models in a scene. 'Subdivision Surfaces' is a means of producing a smooth continuous mesh at render time based on a low resolution proxy model (sometimes called the 'Cage; or 'Limit Surface'). Generally, when straight polygon modeling, surfaces are made from 'faces', each its own tiny flat plane that when shaded together simulates a smooth surface, but silhouette edges can reveal the faceted nature of the model. The traditional fix for this limitation is to simply add more polygons, but controlling all those real polygons can quickly become unwieldy. For a SDS model, the polygons in the low resolution model are automatically divided and refined recursively in such a way that the resulting model is a smooth organically curving surface.

In modo, users can toggle any object as a SDS model by simply pressing the 'Tab' key. Users can control the level (number of times the models is subdivided) using the 'Subdivision Level' setting found in the mesh item properties viewport. Higher values will produce smoother models, but generate more geometry that requires more processing time and memory to render. When the 'Adaptive Subdivision' setting is enabled, this function will override the mesh item setting at render time allowing users to adaptively control all the Subdivision Surfaces in a scene globally.



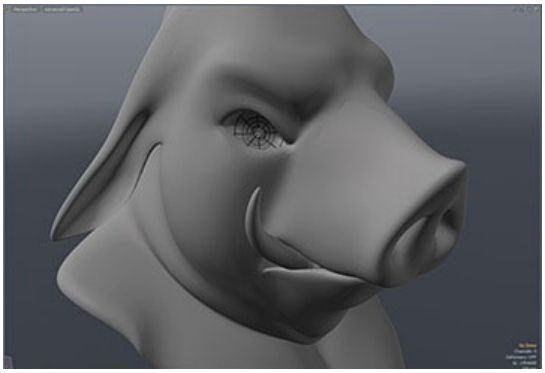
Original 'Cage' Geometry | SubD level of 1 | SubD level of 2 | SubD level of 3

Adaptive Subdivision: This setting toggles the 'Adaptive Subdivision' function in modo. When enabled, modo will adaptively tessellate all SDS meshes at render time. Which is to say, depending on the users settings, the mesh is refined repeatedly until all polygonal edges fall within the users subdivision rate.

Subdivision Rate: When 'Adaptive Subdivision' is enabled, the subdivision rate sets the Level threshold for all SDS geometry in a scene. Defined as pixels, the default rate is 10, which means that modo will pick a subdivision level for that mesh such that the length of edges for any group poly's in the largest subdivision patch appear no more than 10 pixels long (most will be shorter than that). This is very useful since it means that subdivision levels automatically adapt to each mesh's distance from the camera, the zoom factor, etc. If you really want to minimize your polygon count, you can set the

subdivision rate to a huge number such as 1000. On the other hand, if you never want to see a faceted edge in a render you can reduce this number to 5 or less. Be warned that reducing the subdivision rate will be at the expense of additional system memory use and performance loss.

The next five settings control modo's Micropolygon Displacement function, a means of adding fine geometric detail to surfaces by way of a texture based controller. Similar in nature to Subdivision Surfaces, polygonal surfaces are automatically divided and refined recursively producing finer and finer micropolygons that are then displaced (pushed in or out in the polygons normal direction) based on a grayscale value (and users settings in the Texture Locator item). This can result in an incredible amount of detail for models that would be difficult or impossible to create otherwise. Similar to a bump map, but where a bump map is a rendering trick that simulates detail on a surface and often looks off, displacement is actual geometry. 'Displacement' maps can be created in modo using any of the painting or sculpting tools, generated in an external application and applied to the object, or simply applied as a procedural texture layers in the shader tree. For best results, users should enable SDS for any surfaces with displacement applied. Please reference the Texture Effects page of the documentation for more on working with displacement maps.



SDS Limit Surface

Displacement Map applied

Micropoly Displacement: When an objects surface has displacement applied in the shader tree, this checkbox will toggle the rendering of micropolygon displacement on a global level.

Displacement Rate: When Micropoly Displacement is enabled, the 'Displacement Rate', sets the threshold to which the geometry is subdivided into micropolygons. Similar to the Subdivision rate, Micropolygon displacement is adaptive, evaluating every polygon edge separately. The distance from the camera to the center of the edge (the focal length), and the render resolution are used to convert the Displacement Rate (in

pixels) into a distance in world space. If the edge is longer than this and also longer than the Minimum Edge Length, it is split in half. This process continues recursively until all edges satisfy those requirements.

Displacement Ratio: The 'Displacement Ratio' is the ying to 'Displacement Rates' yang. When set to 1.0, objects are tessellated a consistent amount across their entire surface, but when displacing large areas, such as a ground plane in an environment, where a good deal of the geometry lays outside of the view of the camera, most of those polygons are going to waste filling up your systems memory. The Displacement ratio reduces the number of displaced polygons outside the cameras view; the higher the number, the greater the reduction in polygons. Keep in mind though that reflective and refractive objects will still be able to 'see' these areas outside the cameras view, so high settings may begin to introduce artifact to your image.

Minimum Edge Length: Each edge in the mesh is tessellated until polygons satisfy the Displacement Rate setting. Ungoverned, this can go out of control resulting in ridiculous numbers of polygons. The Minimum Edge Length does exactly what the name implies. It sets a minimum length for any polygonal edge. Once an edge has reached that minimum length it can no longer be split. A simple and effective throttle for controlling the amount of polygons in your scene.

Smooth Positions: When a surface is diced into micropolygons, the initial positions of the micropolygon vertices lie on a curved surface based on the original smooth vertex normals. So if you start with a regular non-SDS sphere, for example, it'll be diced into a smooth sphere, and then the displacement texture will be applied to that. You could even use this as an alternative to subdivision surfaces by just applying a displacement texture that is zero everywhere (although the resulting surfaces are different from SDS in that they will actually pass through the original vertices). The ability to turn off position smoothing, which means that all the initial positions of the micropolygon vertices will lie exactly on the flat planes of the original polygons. Doing this in the non-subdiv sphere example would cause the diced version to look faceted. The reason this feature was added was that it's necessary to turn off position smoothing when using a displacement map created by object-to-object baking, since the distances in the map are measured based on the original polygons of the low-poly object and not the smoothed polygons.

Displacement as Bump: When this option is enabled, modo will apply any displacement maps also as a bump map, providing a finer detail evaluation that requires lower number of subdivisions to achieve. This can be a great way to reduce the amount of memory necessary to render complex scenes when combined with lower subdivision rates.

Indirect LOD: When this option is enabled, modo will hold an additional lower resolution version of displaced geometry used strictly in calculating Indirect lighting reducing overhead in GI calculation, however it may end up using slightly more memory as two distinct copies of the same geometry may be held in memory simultaneously.

NOTE: Highly triangulated surfaces, such as imported CAD models are not good candidates for micropolygon displacement.

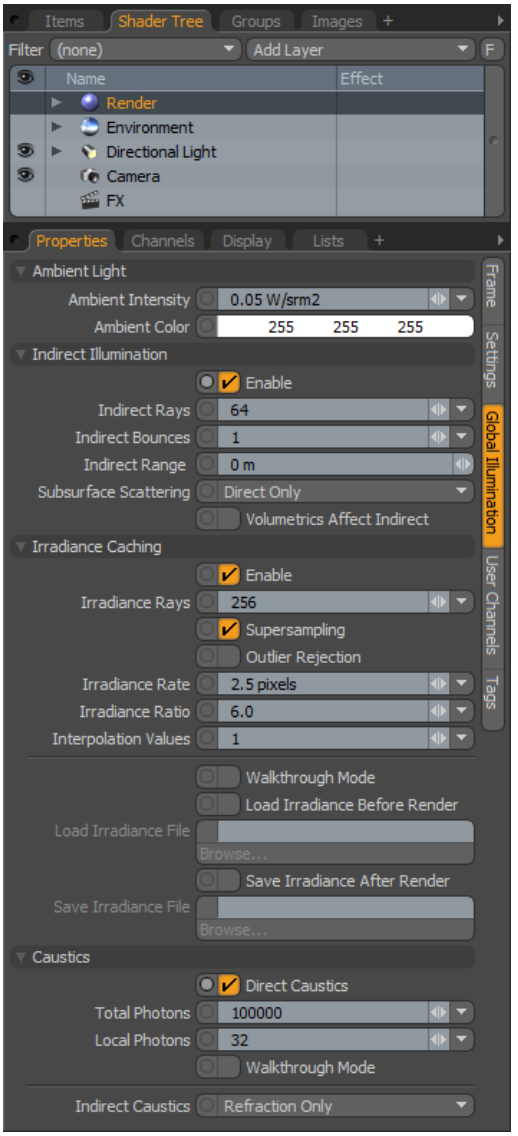
Global Illumination

The 'Render' Item always occupies the top spot in the shader tree, this is no mistake considering its importance. The Render item can be thought of as a general container for global scene render settings. When the Render item is selected, its various attributes become available in the Properties viewport. It is broken into three subtabs of related settings. The 'Frame' tab allows users to adjust render settings specific to the rendered frame itself such as its dimensions. The 'Settings' tab allows users to adjust render settings specific to rendering quality such as anti-aliasing. The 'Global illumination' tab allows users to adjust render settings specific to Global Illumination, a rendering technique that utilizes bounced light to produce incredibly realistic rendered images.

Ambient Light--

Ambient Intensity: The Ambient Intensity control add an even illumination globally to all diffuse surfaces in a scene. This is most useful for simulating additional bounces of light. Some 3D artists recommend driving this value very low or off altogether and favor adding more complex lighting scenarios or using actual Global Illumination bounces to achieve adequate light fill. However, using a little Ambient light can be a very useful optimization to any scene; if setup properly can look very natural and at very little cost in render time. Especially when compared to additional lights and/or indirect illumination bounces.

The ambient intensity control is a real world radiance value for lighting simulation, and simply acts as a multiplier for the ambient color. You can change it to show luminance units (candela per square meter) in Preferences. Start with the default value of .05 W/srm and then work the value up or down until the desired lighting result is achieved.



Ambient Color: The 'Ambient Color' function allows you to specify a color value for the global light added by the 'Ambient Intensity' setting. Choosing a good Ambient Color and Intensity is very important when using 'Ambient Intensity' feature. If you consider that this setting is trying to simulate the effect of the accumulation of all the scattered light rays in a project it is easier to imagine how to choose the color and intensity. To assist the color selection you can decrease the Ambient Intensity to zero and render your image. Look at all the pixels and their brightness and think about what the "average" overall tone or color is. That color is what you should choose for the Ambient Color control.

Indirect Illumination--

Traditionally, 3D rendering applications have used direct light sources to illuminate a scene, various virtual lights would be defined, such as spot light or a point light, and rays could be traced from their origin. Based on their position it was simple to calculate if a surface was illuminated or in shadow. This rendering method was fast and accurate but didn't produce the most realistic results.

Light in the real world isn't static, but actually bounces around in an environment, absorbing and reflecting the colors each ray encounters. A green ball placed next to a white wall will pick up some subtle color bleeding between the two surfaces when illuminated; the brighter the light, the more pronounced the bleeding effect will be. Global illumination is a means with which to simulate this behavior of light producing far more realistic results more easily than could ever be obtained with direct light sources only.

Modo provides users with the ability to use both direct and indirect lighting sources in any scene. These two models are computed separately, and their respective products are added together to give the final shaded result. Each model has specific advantages

(and disadvantages). When used together wisely, users can optimize both the speed and quality of rendered images with an incredible amount of control.

To fully understand the indirect illumination settings, it is critical to first understand the technical process of indirect illumination. While it is intuitive to imagine light coming from a 3D light or a luminous surface and traveling to reach the surfaces then bouncing around a room, the actual process of rendering with indirect illumination is quite the opposite. As the surface is evaluated rays are cast outward from the surface randomly and evaluated when they strike other surfaces in the scene. The sum of those evaluations is what contributes to the color and brightness of the original surface.

To get a more precise idea of how indirect illumination is estimated at a point on a surface, imagine the top half of a transparent globe resting on the surface, so that the point's surface normal is poking through it's north pole. Rays are fired from the surface point through random positions within each "cell" formed by the latitude and longitude grid lines, with one ray per cell. These rays go out and hit either other surfaces or the distant environment, and the average color that they see is the indirect irradiance estimate ("irradiance" being the technical term for incoming light).

Now imagine we need to shade a surface on a moonlit night; the environment is practically all black except for one concentrated bright region(the moon). Each shading point on the surface will send rays as described above. For some points, maybe two of their rays will be fortunate enough to hit the bright region, while for other points only one ray hits it and the rest of the rays see black. With some points getting twice the irradiance of others, you can predict that the surface will look quite splotchy if irradiance caching is on (or grainy if it's off). However if we subdivide the transparent hemispheres more finely (i.e. use more rays), the number of hits and misses will be much more consistent between neighboring surface points, smoothing out the shading.

While indirect illumination in modo is based on this hemispheric sampling technique, there are two very different approaches to the use of these samples. The default method utilizes a technique called 'Irradiance Caching'. The concept behind this technique is that by leveraging a smaller number of more accurate samples and blending between them, you can achieve an image of perceived quality in a shorter amount of time than sampling every pixel with lesser quality, which often results in 'grainy' or 'noisy' images. When Irradiance Caching is disabled modo will fall back to generating a hemispherical shading sample for every pixel in the image (Monte Carlo method). As a result you must be careful about the number of rays you use as this number will be multiplied by the millions of pixels in your image. That's a lot of rays. Really, it is. With irradiance caching active, modo intelligently samples the scene at strategic locations and then interpolates between them for a smoother overall final frame.

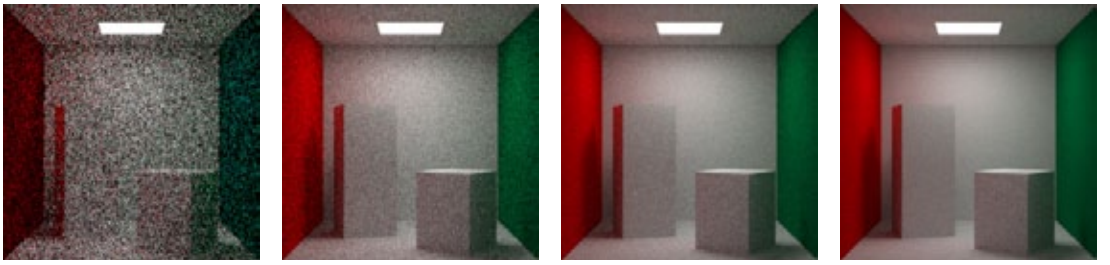
The simplest way to think about Monte Carlo versus Irradiance Caching (IC) is that the Monte Carlo method uses a lower quality (fewer rays) sample at every single pixel location, whereas IC uses fewer, much higher quality (more rays) samples and blends them together. As a result when the samples are not accurate enough in Monte Carlo, there will be significant variance between neighboring pixels which appears visually as grain. When using IC the variance is spread across from sample to sample which yields splotches visually. With Monte Carlo there is one remedy which is to simply increase the number of rays per pixel. This can cause render times to increase dramatically. IC provides several approaches to reducing artifacts which include increasing the number of rays, adding 'Super sampling', and increasing the number of samples required to create a blend called 'Interpolation Values'.

Now that you have an overview of the basic concept, the following definitions will have more context and enable you to more effectively balance your scene in performance/quality when working with indirect illumination.

Indirect Illumination Enable Toggle: Activating this toggle obviously enables modo ability to calculate global illumination.

Indirect Rays: This value represents the number of samples taken for each pixel in the image when utilizing the Monte Carlo indirect illumination model *(Irradiance Caching disabled*). Increasing this value will result in higher quality at the expense of longer render times.

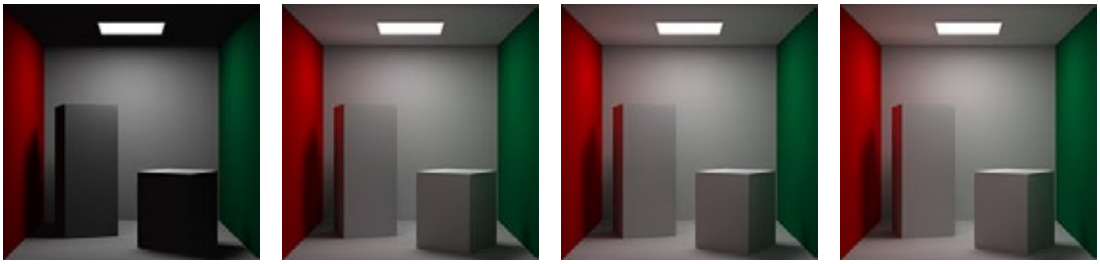
Global Illumination - Monte Carlo (IC disabled), 4 bounces, Ray Threshold 0%



(default) 64 Indirect Rays -12s | 1024 Indirect Rays -3m 13s | 4096 Indirect Rays -13m 22s | 16384 Indirect Rays -51m 40s

Indirect Bounces: The default value of 1 means that a single bounce is used to calculate how a surface's environment affects it. This is less accurate than a multi-bounce solution since in the real world photons bounce all over when illuminating an environment. However, the sacrifice in quality using fewer bounces is rewarded by increased performance. By increasing the number of indirect bounces, the indirect rays are fired from the initial surface and bounce off the first surface they hit, then continue traveling and hitting surfaces until the maximum number of bounces is met. Keep in mind that as you add bounces the number of calculations will increase as will render times and while the result is technically more accurate there are naturally occurring diminishing returns with each additional bounce. It is recommended that you adjust your scene with a single bounce and then add additional bounces to see how great an impact they have on the end result. If a scene is mostly illuminated by a single small area of light, adding bounces can dramatically improve the overall brightness and look of the render.

Indirect Bounces (IC enabled), Ray Threshold 0%

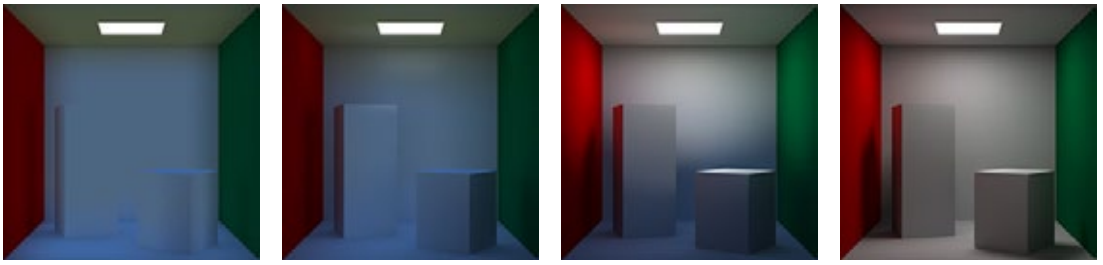


(default) 1 Bounce -30s | 2 Bounces -57s | 3 Bounces -1m 5s | 4 Bounces -1m 14s

TIP: To reduce the number of bounces required, users can increase the Ambient Intensity value. Since the effect of multiple bounced illumination rays is to simply boost the overall lighting by some averaged color, you can reduce the number of bounces and set the Ambient Color and Intensity to mimic those final bounces. This saves time and yields similarly realistic results.

Indirect Range: This value determines how far an indirect ray can travel before it is terminated. In the case of a ray being terminated due to the indirect range value, it is assumed that the ray would eventually hit the environment background, so this value is returned to the shading engine. This is a very useful way to optimize render times. Reducing the indirect range will generally improve render speed. However, you should keep in mind that setting this value too low will create unnatural lighting effects as many rays that otherwise would have ultimately hit a geometric surface resulting a shadow may get cut off early illuminating the surface with the backdrop color rather than shading it with the surrounding geometry. A setting of zero disables indirect range, utilizing the full scale of your scene.

Indirect Range (IC enabled inside a 1m cube w/ default Environment), Ray Threshold 0%



200mm Range -29s | 400mm Range -36s | 800mm Range -48s | 1.6m Range -57s

Subsurface Scattering: When rendering surfaces that employ 'Subsurface Scattering' (SSS), you can choose how the indirect lighting in the scene will affect those surfaces.

Direct Only-- When selecting 'Direct Only', modo will not use any indirect (bounced) lighting, but only calculate SSS based off the direct lighting within the scene, i.e. Distant Lights, Spotlights, Point Lights etc. This mode is useful in many cases as global illuminations effect upon SSS is usually fairly subtle. However, in instances where there are no direct lights in your scene, such as those lit entirely by image based lighting (HDRI), one will need to choose either 'Indirect Affects SSS' or 'Both' to see the effects of SSS for any given surface.

Indirect Affects SSS-- When selecting 'Indirect Affects SSS', modo will use any global illumination in your scene, as well as the direct lighting to calculate the subsurface scattering effects on your objects surface.

SSS Affects Indirect-- When selecting 'SSS Affects Indirect', modo will calculate bounced indirect light, taking in to consideration the subsurface scattering effects of any given surface.

Both-- The 'Both' setting will allow global illumination to affect subsurface

scattering, as well as for the subsurface scattering to affect the indirect illumination calculations. This setting is the most render intensive.

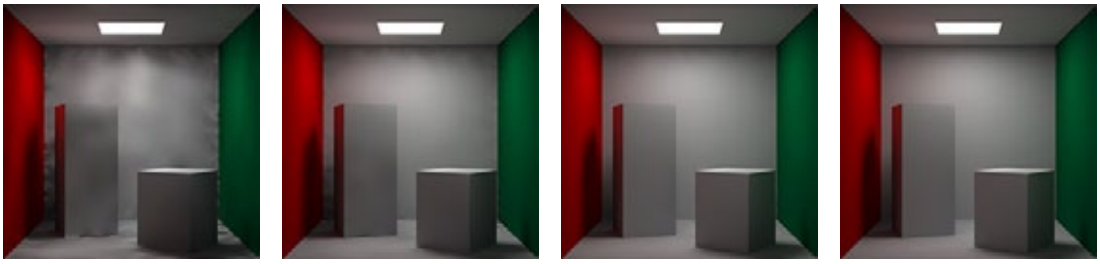
Volumetrics Affect Toggle: Enabling this checkbox will direct modo to consider volumetric lighting effects when calculating global illumination.

Irradiance Caching--

Irradiance Caching Enable Toggle: When rendering indirect illumination in modo, Irradiance caching is the default method utilized. Disabling this toggle directs modo to use the more intensive Monte Carlo method to calculate global illumination.

Irradiance Rays: Irradiance rays are technically the same as Indirect Rays in that they are the rays fired out from the surface in order to sample indirect illumination. As a matter of convenience modo has both Irradiance and Indirect Rays settings since the two forms of indirect illumination typically require vastly different numbers of samples. Irradiance Caching relies on much higher quality samples distributed sparsely across the project whereas Indirect Illumination without Irradiance Caching uses lower quality samples at every pixel. Having two values allows you to easily switch between Irradiance Caching and traditional Indirect Illumination without constantly adjusting the number of Rays for each sample.

Irradiance Rays, Ray Threshold 0%



64 Rays -3.1s | 256 Rays -4.1s | 1024 Rays -6.7s | 4096 Rays -12.4s

Supersampling Toggle: After all rays have been fired for a particular irradiance evaluation, this feature looks at the resulting ray color of each hemisphere cell and then sends additional rays through those cells that differ a lot from their neighbors, getting a more detailed look at high contrast areas of the environment and therefore a more accurate estimate. Currently about 25% more rays are fired, so for a setting of 100 rays you really get 125, but the results will be better than if you had just used a setting of 125 without supersampling, because the rays will be going in the more important directions. (Chief Scientist Allen Hastings recommends leaving this option on all the time. It will generally improve your render quality with minimal performance impact.)

Outlier Rejection: The 'Oulier Rejection' option specifies whether to disregard rays that are much brighter or darker than the averaged values in a particular direction when Irradiance Cache values are being computed and 'Supersampling' is enabled. It can be useful for smoothing of indirect illumination in some cases.

Irradiance Rate: As Irradiance Caching smoothly blends between neighboring IC values, a little breathing room is necessary between samples to provide space for that blend to occur. The Irradiance Rate provides that space as it sets a minimum distant between IC samples, calculated by pixels. So a setting of 2.5 means the minimum distance between two samples would be 2 and a half pixels away. Increasing this value will cause IC values further away from each other to blend together, providing a smoother result, however increasing the value too high may result in reduced accuracy in areas of high detail.

Irradiance Ratio: Where the Irradiance Rate sets the minimum distance between IC values, the Irradiance Ratio sets the maximum distance. The ratio is a multiplier of the rate, so a setting of 6 multiplied by a rate 2.5 would return a value of 15, meaning IC values would be no closer than 2.5 pixels and no further apart than 15 pixels. The ratio ultimately reduces the number of samples necessary across smooth surfaces where shading changes are minimal.

Interpolation Values: The 'Interpolation Values' setting specifies a minimum for how many nearby values to interpolate. Let's say it's set to three for example. If we're shading a point and can only find two nearby previously computed values in the cache, then it will force the computation of a new value at the current point, and the final irradiance at that point will be a blend of all three values. So it tends to smooth out the shading. Increasing this value will generally improve render quality at the expense of render time.

Walkthrough Mode: Enabling 'Walkthrough' mode tells modo to calculate global illumination only on the first rendered frame and utilizing that same calculation for every subsequent frame. While a great time saver, this setting assumes no objects move in the scene except the camera. Exceptionally useful for architectural renderings where a viewer is given a 'rendered' tour of a modeled space, hence the walkthrough name. While a huge time saver, it also eliminates the jittery shading sometimes evident on pristine architectural surfaces.

Load Irradiance Before Render: Using the 'Load Irradiance' function allows one to load a previously calculated irradiance cache solution created by the 'Save Irradiance After Render' function. To use, enable the 'Load Irradiance' checkbox and use the browse button to locate the previously created solution in .lxi format. When enabled, subsequent renders will load the specified solution eliminating the need to calculate the irradiance cache each time the scene is rendered. Especially useful for rendering

series of images where only the camera moves.

Save Irradiance After Render: Using the 'Save Irradiance' function allows one to save an irradiance cache solution for a particular scene and load it at a later time for reuse. To use, enable the 'Save Irradiance' checkbox and click the browse button to choose a filename and location. Click 'Save' to continue. Now the next time you render (F9) your scene, the irradiance cache will save to disc upon completion. Note that only the render directly following the activation will generate the IC file, subsequent renders will not overwrite the initial save, unless you disable and re-enable the function.

Caustics--

When light refracts through or reflects off of a surface, these bent rays are focused together creating a bright pattern referred to as a caustic. Think of the light dancing on the swimming pools floor, or the bright flash you see from a cars windshield on a sunny day and you'll know what it is. Simulating this effect in modo is quite easy with global illumination as caustics are a natural rendering byproduct of image based lighting. Caustics from direct light sources, such as area lights, distant lights, spot lights and such, require the use of photons map tracing to calculate the effect, enabled via these settings.

Direct Caustics Toggle: This checkbox tells modo to calculate caustics for all reflective and transparent surfaces.

Total Photons: The total photons control sets a ceiling on the total number of photons fired in the scene. The photon count is divided amongst all active 3D lights in a scene with a bias based on the power of the light, ensuring energy conservation.

Local Photons: Local photons indicate the number of photons required for each pixel sampled. When the pixel is rendered a search along the surface locates the nearest photons up to the local photon count. The default setting of 32 indicates that 32 photons will be used for each pixel rendered. Increasing this value will create a smoother overall caustic effect at the expense of detail, while decreasing the value results in a sharper caustic effect with increasing graininess.

(default)
32 Local Photons

64 Local Photons

128 Local Photons

256 Local Photons

Walkthrough Mode: When rendering animations containing caustics, where the camera is the only item moving in the scene, it will be helpful to enable the 'Walthrough Mode' option. This tells modo to cache the caustic calculations, frame to frame, reusing the initial calculations, thus ensuring each subsequent frame matches precisely producing a smoother rendered result.

Indirect Caustics: Indirect caustics are a natural rendering byproduct of image based lighting, in instances where you wish to control their creation, you have several choices to do so-

None-- None disables all indirect caustics within a scene.

Reflection Only-- Caustics are only calculated off of reflective surfaces.

Refraction only-- Indirect caustics are only calculated though refractive surfaces.

Both-- Indirect caustics are calculated for both reflective and refractive surfaces.

When modeling overlapping transparent surfaces such as glasses or bottles containing liquids, it is recommended that the 'liquid' layer overlap the 'glass' layer by a small amount. In no cases should there be a gap between the two surfaces. This image illustrates a cutaway view of a shot glass, the black outline, filled with a liquid, the red shading. Note how the red object extends beyond the interior surface of the glass, this is necessary to obtain the desired results.

Environments

Environment Item

The 'Environment Item' allows the user to set options related to the environments visibility and intensity in a modo scene. When selected, its associated attributes can be edited within the 'Properties' viewport panel. Environments offer a quick, flexible way to fill a modo background with colors, gradients and image maps. With the addition of an HDR image and global illumination, environments can also 'light' a scene providing an easy path to create stunningly realistic results. For more information on setting up IBL (Image Based Lighting) please reference the 'Image Based Lighting' page of the documentation.

The rendered rhino object; the background area is solid black as no geometry is visible nor has an environment been inserted.

Environments can give the impression of a much larger scene without adding additional geometry.

Also, keep in mind that since the Environment is just a shader tree item, they can have additional shader tree item layers added to them; lightprobe images can be added producing full 360° environments, or camera projected images can provide 'plates' allowing users to incorporate 3D elements into a filmed scene by simply adjusting the settings of the 'Texture Locator'. Procedural layers can be added and layered producing interesting effects. By RMB-clicking on the items effect column, different options can be selected producing a variety of different and interesting blending effects as materials are layered. More on Environment Effects can be found in the 'Environment Effect' page of the documentation.

Modo also allows for multiple Environments; additional Environment Items can be added to the shader tree by selecting in the menu bar 'Item > Create Environment' or by RMB-clicking the item itself and choosing 'Duplicate' from the popup menu. This allows the user the ability to specify an environment for reflections, refractions and the background itself, with a completely different version of the environment fir 'lighting' the scene.

Environment

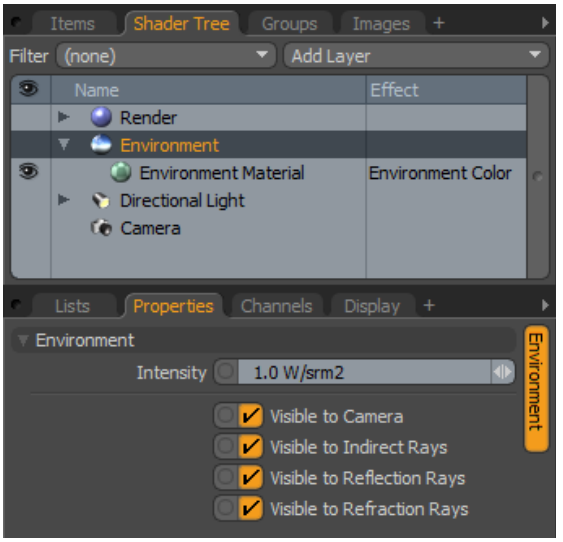
Intensity: 'Intensity' acts as a multiplier, allowing the user a convenient way to control and modify the overall brightness of the selected Environment. This setting will have the greatest effect when global illumination is enabled, controlling to what degree the overall environment contributes to the lighting of the scene.

Visible to Camera: Toggles whether or not the selected Environment is visible to the camera when rendered. This setting is independent to the others, even when not visible to the camera, it can still contribute to the lighting of the scene providing users the ability to specify alternate images or textures for the background in an additional environment item.

Visible to Indirect Rays: Toggle whether or not the environment contributes lighting to global illumination. This setting will only affect scene that have 'Global Illumination' enabled.

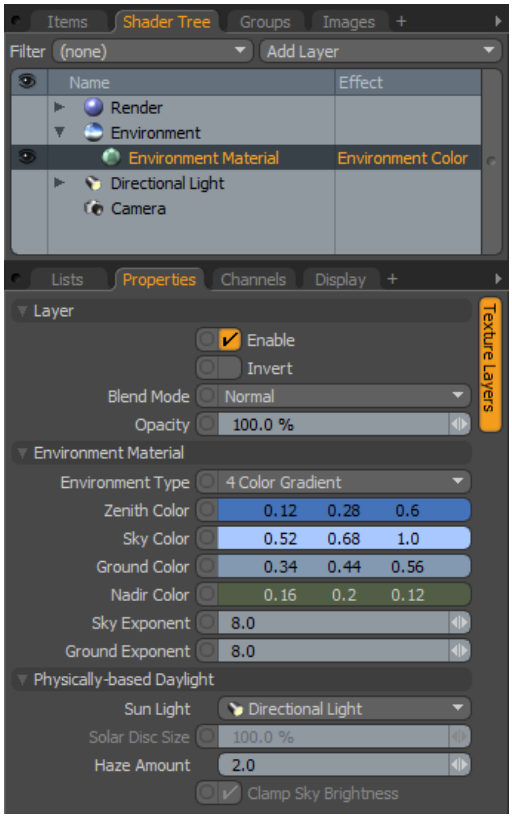
Visible to Reflection Rays: Toggles whether or not environments have an effect on any reflective surfaces within a scene.

Visible to Refraction Rays: Toggles whether or not environments have an effect on any refractive surfaces within a scene.



Environment Material

The 'Environment Material' provides a simple method for adding a basic color or a color gradient to the background of a project. Each scene by default includes a single Environment Material. Additional material layers can be added from the menu bar with the command "Texture > Add Shader Tree Layer > Environment Material" or from within the shader tree viewport itself, by LMB-clicking the 'Add Layer' option of the full viewport window and selecting the 'Environment Material' item from the pop-up context menu. Environments, when combined with modo's 'Global Illumination' functionality, can produce incredibly realistic lighting simulations as well. Please reference the 'Global Illumination' page of the documentation for information on enabling this function.



Layer--

Enable: The 'Layer Enable' checkbox allows the user to temporarily enable/disable a layer without losing stored values. Disabled layers are saved with the scene file. This is the same as toggling the eye icon "" in the visibility column of the shader tree viewport.

Invert: This setting inverts any RGB color values specified in the Environment Material.

Blend Mode: Affects blending between material layers allowing the user to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: At 100% a layer is completely opaque, values below 100% change the transparency of the current material layer allowing other lower layers to become visible, ramping toward completely transparent at 0% essentially disabling the layer. If only one environment material is present, modifying the opacity will fade the brightness of the background and affect how much it contributes to global illumination.

Environment Material--

Environment Type: Within the 'Environment Type' drop down are several options for specifying colors and gradients that appear in the background of rendered images. Once selected, the appropriate options for Zenith, Sky Ground and Nadir colors are available as well as the Sky and Ground exponent values that adjust the bias of the gradients between the Zenith/Sky and Ground/Nadir respectively.

4 Color Gradient-- This easy to use gradient allows you to set 4 color values which are used to simulate the sky and ground. The Zenith color is the color that is directly overhead and this ramps into the Sky color which starts at the horizon and ramps upward. The Nadir color is directly below the camera and ramps upwards into the Ground color which terminates at the horizon when it meets the Sky color. There is a soft blend between the Zenith and Sky colors and a soft blend between Ground and Nadir. The boundary between Ground and Sky is hard edged to give the illusion of a distant horizon.

When using a 4 Color Gradient, the Sky and Ground Exponents are used to compress or expand the gradation between Zenith and Sky and the Nadir and Ground. Higher value push the gradient transition closer to the horizon, while lower values spread the gradient further across the available spectrum.

2 Color Gradient-- This option reduces the colors to Zenith and Nadir and creates a soft ramp between the two.

Constant-- Uses only the Zenith color for the entire background.

CIE Overcast Sky-- This option provides two user defined color values; Zenith and Nadir. The Nadir color will be used in without any ramp effect for everything below the horizon. The Zenith color starts directly above the camera and then has a slight ramp to a somewhat darker color. By definition the CIE overcast sky is three times brighter at the Zenith than it is at the horizon so the Zenith color will be reduced in brightness by that amount by the time it reaches the horizon. When this setting is applied with global illumination, you can recreate the traditional overcast render look popular in image that want to show off an untextured models surface detailing.

Physically-based Daylight-- The Physically Daylight option works in tandem with global illumination to create renders lit by an incredibly realistic daylight simulation. Especially useful for architectural exterior and interior renders or in any scene that needs nice outdoor lighting.

Physically-based Daylight--

When 'Physically-based Daylight' is selected as the Environment type, a few additional option are made available allowing the user to fine tune the look of the lighting simulation. Global Illumination needs to be enabled in order for the Physically-based Daylight function to work.

Sun Light: The' Sun Light' setting allows the user to set a distant light (other light types wont show up here) that will be considered the sun in the rendered images. Additional properties specific to the light itself are available in the Directional Lights 'Physical Sun' settings.

Solar Disc Size: A small disk is added to the rendered image, representative of the actual sun, above the horizon. It also appears in reflections, refractions and so forth. The default value of 100% is correct to the scale of what you would see in the real world, however, you can change the size depending on your particular scene needs using this setting.

Haze Amount: In the real world, as the sun nears the horizon, light has to pass through a greater amount of atmosphere to reach our eyes, additionally, moisture, dust and pollutants in the air (collectively described as Haze) further affect the light. The further the light has to pass through these particles, the more shorter wavelengths of light are absorbed, tending the light toward warmer hues. Increasing the 'Haze' setting in modo will push the horizon colors toward oranges and yellows for midday lighting scenarios and deeper reads (sunsets) in evening times. Time of day can be controlled in the Distant Lights "Physical Sun' settings.

Clamp Sky Brightness: The Physically-based daylight function is a highly accurate lighting simulation generating light intensities outside of normal render display settings. The 'Clamp Sky Brightness' toggle will limit the overall brightness of the sky producing pleasing results more easily. When disabled, you might think your scene is completely 'blown out' and overexposed, but actually, since modo renders in full floating point accuracy, you can bring back all that detail using a combination of white level, gamma and tone mapping. Please reference the 'Tone Mapping' page of the documentation for more information.

Image Based Lighting

Any environment in modo can be used to light a scene with the addition of 'Global Illumination', wether it is an image map, a gradient or even a solid color or procedural texture layer. Individually, layers dont have specific brightness or luminous settings, but their contribution to a scene can be adjusted by modifying the 'Opacity' value of

the layer or globally with the 'Intensity' setting of the 'Environment Item' producing brighter, more evenly illuminated scenes or darker scenes, depending on how the settings are adjusted.

When global illumination is combined with High Dynamic Range (HDR) images, incredibly realistic simulations of real world light interaction can be produced, a technique known as image based lighting (IBL). HDR imagery is generated in such a way as to not only capture the colors of a scene but faithfully capture the brightness values as well, creating a dynamic range that far exceeds what most monitors are cabable of displaying. A full 360° panoramic HDR would capture any visible lights, exterior lighting (including the sun) and the skys contribution, and most importantly bounced light. modo in turn is able to utilize this extended information and illuminate the scene. Because of HDRs abilty to capture such subtleties in a scene, scenes utilizing image based lighting are often indistiguishable from photographs producing a look that has become extremly popular, especially amongst visual effects artists, art directors wishing to absolutly control every aspect of an advertisers image and the CAD visulization crowd seeking to produce ever more realistic simulations for planning and marketing purposes.

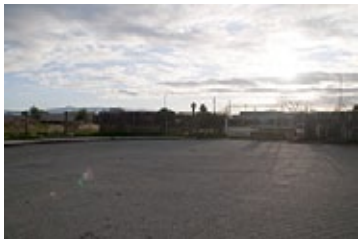
A series of IBL renders utilizing the same settings, only the environment image has changed.

Robot model provided by John Hayes

Where Do HDRI's Come From?

HDR images used for IBL are often comprised of a series of photographs taken of the same scene. Each having a slightly different exposure (length of time the shutter is open). This sequence of images, ranging from extremely dark to extremely bright, are then combined in such a way that for every pixel, a color value and a brightness value can be calculated. (Photoshop has an automation command "Merge to HDR" that is capable of doing this; several free tools are availalbe that are capable of doing this as well. See HDR Labs for more info.)

Sky exposure (-2 EV)



Default Exposure (0 EV)

Forground Exposure (+2 EV)

The resulting HDR image can then be saved in a format that retains this information, such a EXR or HDR (16bit and lower formats will work but don't retain the brightness values necessary to properly illuminate the scene). When loaded into modo, the brightness information is then extracted and used to illuminate the scene producing an incredibly realistic simulation of the lighting from the original scene.

TIP: Don't happen to have any HDR images available? A great alternative that produces equally impressive results is the Physically Based Sky feature. This function can be enabled in the Environment Material item; use the 'Environment Type' dropdown to select. Please reference that page of the documentation for more information on setting it up.

Lighting

Light Types

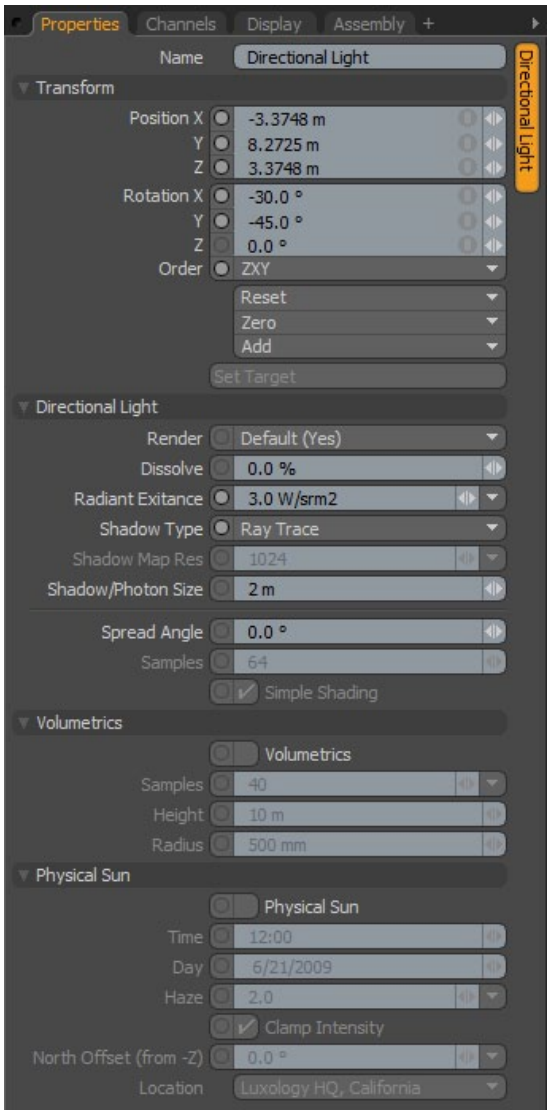
modo offers a variety of different light types that simulate real world sources of illumination. A light can be added to the scene by way of the menu bar, by selecting "Item > Create Light" and then selecting the type of light you wish to add. Lights may also be added from the 'Item List' viewport by LMB-clicking the 'Add Item' function and selecting one of the available light types from the contextual menu. Settings related to the lights illumination are set within the 'Properties' viewport when light items are selected, material specific properties are available via the light material item, selectable within the 'Shader Tree' viewport. Specific documentation for each of the lights are on the following pages.

Direct Light Types			
Directional Light	Point Light	Dome Light	IES Light
Spot Light	Area Light	Cylinder Light	Portal

Directional Light

The Directional Light is most similar to the Sun. This light type emits a wall of parallel rays in the direction that the light is facing. The position of a Directional light is mostly irrelevant as the rays travel infinitely along their path through the scene, as such, only the Rotation is used to set the direction of the light's rays. Also, as a result of the infinite nature of the rays there is no falloff for a Directional Light.

As with all Items, the Directional Light controls are located on the Properties tab when the light is selected. Directional lights have Position and Rotation options as well as controls for the intensity of the light, shadow type, and other settings covered fully below. To control the lights color, shadow color and other material related properties, refer to the Light Material page of the documentation.



Directional Light

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the light item in XYZ space. By default, Position transforms originates from the items center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the light item. By default, Rotation transforms originates from items center position.

Order: Allows the user to set the order that rotations are applied to the light item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Reset: Resets the selected transform values to (0,0,0) returning the items back to the world space center position.

Zero: Returns the light items Center position to the world space center (0,0,0) without changing the position of the light item itself.

Add: The 'Add' function will add the selected set of transforms to the channel

list. Useful when animating complex motions, providing a layered approach to the transforms.

Set Target: By selecting the Light item, and a single additional item in the item list, and then pressing 'Set Target', this function allows users the ability to target specific items in a scene, automating the rotation of an item, so that it continuously points toward the targeted item. Once activated, additional option appear--

Remove Target: Removes the target link between the two items.

Enable: Toggling this option off temporarily disables the targeting function while retaining the link between the items.

Set Focus Distance: This option sets the focal distance for DOF when targeting camera items to other item.

Roll: Provides the ability to offset the items rotation angle away from the target.

Time Offset: Provides users the ability to time offset, by a number of frames, how the light follows the target item. It can either be delayed behind it with a negative value, or run ahead of it with a positive value.

Directional Light--

Render: This drop down menu allows the user to select from 3 choices, when set to 'Default', the user can enable/disable lights using the visibility function "" of the item list. When the light is visible, it contributes to the final rendered scene and when invisible, it will not. On some instances the user may prefer to fix this state, setting the light as 'On' (enabled) or 'Off' (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving the user from manually enabling lights for test renders.

Dissolve: When the 'Dissolve' function is set to any value above 0%, the lights overall affect on the scene will attenuate as the value increases. When set to 100%, the lights effect on the scene is completely disabled. This function providing a convenient way to dim a lights effect within a scene.

Radiant Exitance: This value controls the intensity of the light and uses the standard physically based unit of Watts per volumetric meter. As you would expect, increasing this value will increase the amount of light coming from the Directional Light and decreasing the value reduces the light intensity. With directional lights, the area of light generated in the scene is very large and therefore small values are all that are necessary to illuminate an entire scene.

Shadow Type: This popup offers options between 'Ray Traced' , 'None' and 'Deep Shadow Maps'. In situations where users want a light to cast a shadow, Ray Traced will

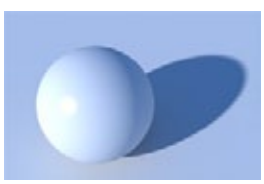
give the most accurate results. The traditional hard edge of ray traced shadows can easily be softened using 'Spread Angle' value in any light item. Deep Shadow maps are useful for volumetric lights and fur rendering, where a great deal of calculations are required to produce shadows; producing similar results to ray traced shadows while reducing the number of calculations.

Shadow Map Res: The 'Shadow Map Res' Sets the resolution for the calculated deep shadow map in pixels. This option is only available when Deep Shadow Map is selected.

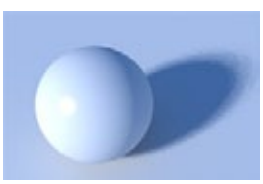
Shadow/Photon Size: Since directional lights are infinite, 'Shadow Map Size' allows the user to specify the area that the shadow map will affect. Think of it like a rectangular panel aligned with the light items position in the 3D viewport (an outline appears around the light signifying the map size), shadow map 'rays' will fire from this panel forward and objects within this space will have shadows. As the shadow map size increase, you may find you need to increase the resolution to reduce blocky artifacts. When generating caustics with Directional Lights, adjustments to the 'Shadow/Photon' size also effect the photon emit area.

Spread Angle: The default value of 0° results in crisp hard-edged shadows. If you want to produce a more realistic soft-edged shadow users may increase this value. Since the Directional Light is most often used to simulate the Sun, a very small amount of Spread Angle is recommended. Values between 2 and 5 degrees will give a subtle soft edge to the shadow similar to sunlight. Keep in mind that increasing this value to high will result in noisy shadows unless the user also increases the number of samples.

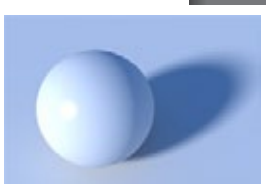
A spread angle of 0° produces a crisp shadow.



A spread angle of 2° produces a softer edged shadow.



A spread angle of 10° produces a soft but noisy shadow.



Increasing the 'Samples' value produces a smoother result.

Samples: As the Spread Angle increases, additional samples are required to smooth out the resulting soft edge shadow. If the shadow appears grainy, increasing this sample setting is the most likely solution. The default value of 16 is a good starting point for balancing speed and quality. As the spread increases or the resolution of the image increases you can drive this number higher to account for any grain that appears. Keep in mind that increasing the number of samples will also increase render times.

Simple Shading: When generating soft shadows with a spread angle greater than 0°,

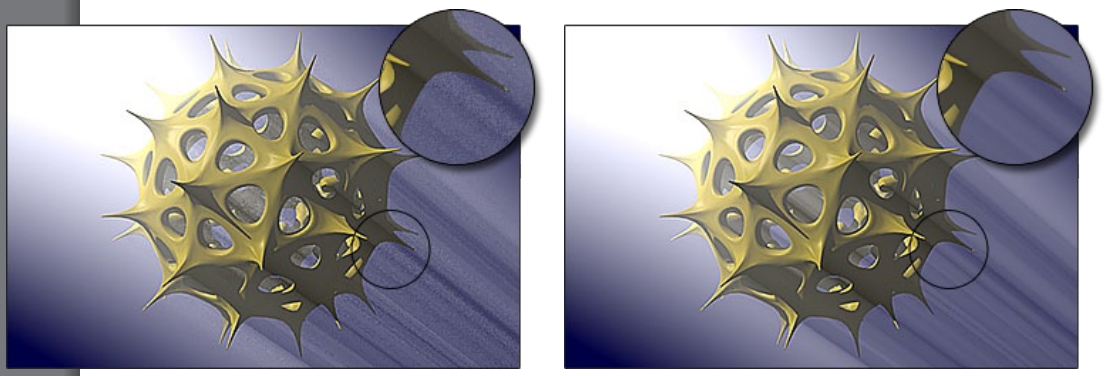
modo calculates the shadow as if sampled from an array of lights, these samples are blended together to produce the smooth result. With the 'Simple Shading' checkbox enabled, modo will calculate the specular and diffuse values for a surface as if the light were a single point, regardless of the spread angle. However, with 'Simple Shading' disabled, modo uses the same light array as the shadow to calculate the specular and diffuse values. This results in diffuse and specular shading that is 'soft', like the shadow but can also become increasingly grainy with a greater 'Spread Angle'. Increasing the number of 'Samples' will reduce grain in the diffuse and specular shading as well as the shadows at the expense of longer render times.

Volumetrics--

When light shines on particular matter suspended in the air, such as water vapor, smoke or even pollution, the light rays become visible. In computer graphics this effect is known as volumetric lights, their affect can be subtle or pronounced, either way it can add a good amount of believability and atmosphere to renders. Additional Settings related to volumetrics can be found in the Light Material item.

Volumetrics Checkbox: The 'Volumetrics' checkbox toggles the effect on or off. When enabled, the samples, height, and radius options become available.

Samples: Any objects that intersect the light beam volume will cast shadows through it. You may have witnessed this when sunlight shining through the clouds creates streaks of light sometimes referred to as 'god rays'. The 'Sample' setting defines the degree of accuracy that modo uses to calculate these shadows through the volume, higher numbers of samples will give smoother and more accurate results, while lower values will render more quickly but become increasingly grainy. The sample below demonstrates the difference between 32 samples on the left and 256 samples on the right.



Height: This is the depth of the volume as measured extending out from the front of the light. A setting of 20m will extend the rays 20m from the light including falloff.

Radius: The 'Radius' sets the circular region around the light where volume effect will originate from.

Physical Sun--

The Physical Sun function works in tandem with the 'Physically-based Daylight' setting under the Environment Material properties. It is in that dialog box you enable the Daylight lighting function and select which light will act as the sun. Here in the directional light properties, you can activate that light as the sun and use the time and day values to set the suns position.

Physical Sun Checkbox: This setting toggles the directional light as the sun in the scene. You'll want to enable this for Physically-based Daylight renders.

Time: This sets the time of day for your lighting, and effect the position of the sun as well as the colors in the atmosphere -- afternoons offer cooler lighting and shorter shadows and evening times tend toward warmer oranges and reds and longer shadows.

Day: The sun rises and sets at different times during the year, based on the date supplied, this setting will position the sun precisely in your sky based on the location specified below.

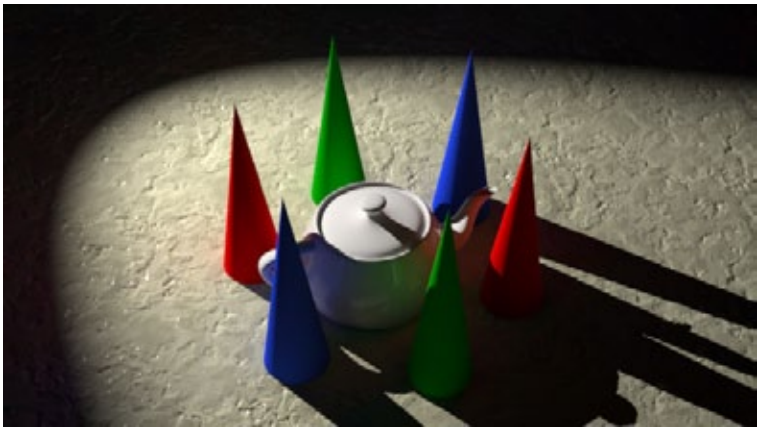
Haze: In the real world, as the sun nears the horizon, light has to pass through a greater amount of atmosphere to reach our eyes, additionally, moisture, dust and pollutants in the air (collectively described as Haze) further affect the light. The further the light has to pass through these particles, the more shorter wavelengths of light are scattered. The longer wavelengths pass through tending the light toward warmer hues. Increasing the 'Haze' setting in modo will push the horizon colors toward oranges and yellows for midday lighting scenarios and deeper reads (sunsets) in evening times. This is the same setting as in the Physically-based Daylight properties.

Clamp Intensity: The Physically-based daylight function is a highly accurate lighting simulation generating light intensities outside of normal render display settings. The 'Clamp Intensity' toggle will limit the overall brightness of the sun producing pleasing results more easily. When disabled, you might think your scene is completely 'blown out' and overexposed, but actually, since modo renders in full floating point accuracy, you can bring back all that detail using a combination of white level, gamma and tone mapping. Please reference the Tone Mapping page of the documentation for more information.

North Offset: You've probably modeled your scene relative to the XYZ planes in modo and didn't take into consideration whether it was facing north or south. That's ok, you can use the 'North Offset' to make it face the proper direction without having to resort to item rotations. Your model can stay in its comfortable XYZ position and the world will revolve around it.

Location: The sun will appear in different positions in the sky from different locations on earth. The 'Location' setting allows the user to set the location for their scene so the Sun will appear in the proper position for your simulated sky. A number of well known locations is provided, but you can also specify the latitude and longitude positions for your custom location as well as save it in the config file for future use.

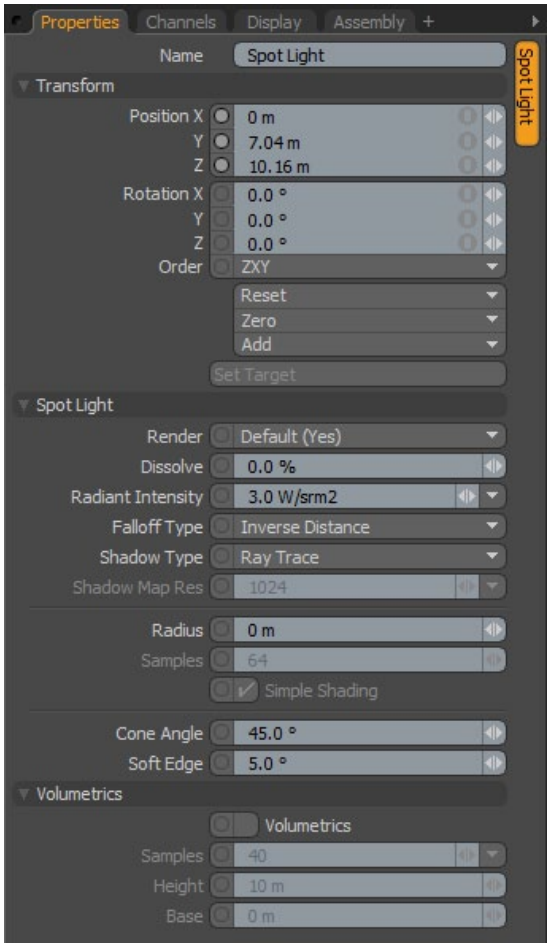
Spot Light



The Spot Light looks and acts like a real world Spot Light and obeys the physical law of inverse square falloff of intensity. Light radiates from a single point outward in a cone shape widening the further the subject is from the light itself. As with all Items, the Spot Light controls are located on the Properties tab when the light is selected. Spot lights have Position and Rotation options as well as controls for the intensity of the light, shadow type, and other settings covered fully below. Using the Transform tool with the Spot Light selected provides interactive handles for editing position, rotation, cone and soft edge angles. To control the lights color, shadow color and other material related properties, refer to the Light Material page of the documentation.

Spot Light

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.



Transform--

Position: An Item transform that allows the user to numerically position the light item in XYZ space. By default, Position transforms originates from the items center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the light item. By default, Rotation transforms originates from items center position.

Order: Allows the user to set the order that rotations are applied to the light item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Reset: Resets the selected transform values to (0,0,0) returning the items back to the world space center position.

Zero: Returns the light items Center position to the world space center (0,0,0) without changing the position of the light item itself.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Set Target: By selecting the Light item, and a single additional item in the item list, and then pressing 'Set Target', this function allows users the ability to target specific items in a scene, automating the rotation of an item, so that it continuously points toward the targeted item. Once activated, additional option appear--

Remove Target: Removes the target link between the two items.

Enable: Toggling this option off temporarily disables the targeting function while retaining the link between the items.

Set Focus Distance: This option sets the focal distance for DOF when targeting camera items to other item.

Roll: Provides the ability to offset the items rotation angle away from the target.

Time Offset: Provides users the ability to time offset, by a number of frames, how the light follows the target item. It can either be delayed behind it with a negative value, or run ahead of it with a positive value.

Spot Light--

Render: This drop down menu allows the user to select from 3 choices, when set to 'Default', the user can enable/disable lights using the visibility function " " of the item list. When the light is visible, it contributes to the final rendered scene and when invisible, it will not. On some instances the user may prefer to fix this state, setting the light as 'On' (enabled) or 'Off' (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving the user from manually enabling lights for test renders.

Dissolve: When the 'Dissolve' function is set to any value above 0%, the lights overall affect on the scene will attenuate as the value increases. When set to 100%, the lights effect on the scene is completely disabled. This function providing a convenient way to dim a lights effect within a scene.

Radiant Intensity: This value controls the intensity of the light and uses the standard physically based unit of Watts per volumetric meter. As you would expect, increasing this value will increase the amount of light coming from the Spot Light and decreasing the value reduces the light intensity. With Spot lights, the area of light generated in the scene is very small and therefore large values are likely necessary to illuminate the scene how the user would expect.

TIP: Keep in mind that a lights intensities and falloffs adhere to real world scale values, scenes that do not adhere to a real world scale factor will likely require intensities to be adjusted accordingly to achieve appropriate illumination.

Falloff Type: Light in the real world isn't a uniform brightness, its intensity diminishes with distance. Photographers will be familiar with the concept that a light will be a quarter as bright at twice the distance away, know as the Inverse Square Law. modo lights default to this setting providing a realistic way to light a scene, however there are times when users may not wish to have this behavior, so modo provides three falloff type options-

None- No falloff, light is consistently bright across its distance.

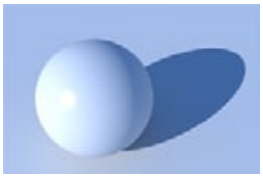
Inverse Distance- Light is half as bright at twice the distance.

Inverse Distance Squared- Default behavior, Light will be a quarter as bright at twice the distance.

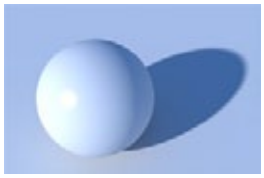
Shadow Type: This popup offers options between 'Ray Traced' , 'None' and 'Deep Shadow Maps'. In situations where users want a light to cast a shadow, Ray Traced will give the most accurate results. The traditional hard edge of ray traced shadows can easily be softened using 'Spread Angle' value in any light item. Deep Shadow maps are useful for volumetric lights and fur rendering, where a great deal of calculations are required to produce shadows; producing similar results to ray traced shadows while reducing the number of calculations.

Shadow Map Res: The 'Shadow Map Res' Sets the resolution for the calculated deep shadow map in pixels. This option is only available when Deep Shadow Map is selected.

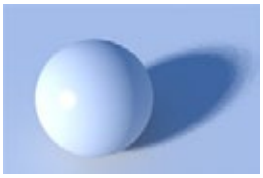
Radius: The default value of 0m results in crisp hard-edged shadows. Increasing the size prompts modo to sample lighting from an array of positions within the radius, the various samples are blended together producing the smooth result. Keep in mind that increasing this value too high will result in noisy shadows unless the user also increases the number of samples.



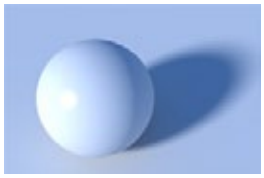
A radius of 0m produces a crisp shadow.



A radius of 2mm produces a softer edged shadow.



A radius of 10mm produces a softer but noisy shadow.



Increasing the 'Samples' value produces a smoother result.

Samples: As the Spread Angle increases, additional samples are required to smooth out the resulting soft edge shadow. If the shadow appears grainy, increasing this sample setting is the most likely solution. The default value of 16 is a good starting point for balancing speed and quality. As the spread increases or the resolution of the image increases you can drive this number higher to account for any grain that appears. Keep in mind that increasing the number of samples will also increase render times.

Simple Shading: When generating soft shadows with a radius greater than 0m, modo calculates the shadow as if sampled from an array of lights, these samples are blended together to produce the smooth result. With the 'Simple Shading' checkbox enabled, modo will calculate the specular and diffuse values for a surface as if the light were

a single point, regardless of the radius size. However, with 'Simple Shading' disabled, modo uses the same light array as the shadow to calculate the specular and diffuse values. This results in diffuse and specular shading that is 'soft', like the shadow but can also become increasingly grainy with a greater 'Radius' value. Increasing the number of 'Samples' will reduce grain in the diffuse and specular shading as well as the shadows at the expense of longer render times.

Cone Angle: Spot lights emit their light in a cone shape that originates from the lights position toward the default -Z direction. The 'Cone Angle' defines the shape of the cone that is generated; larger values produce wider conical shapes and therefore larger areas of illumination, while smaller values produce tighter cones with smaller areas of illumination. A setting of 360° produces light similar to a point light radiating in all directions equally. This value can also be set interactively with the Transform tool in the 3D viewport.

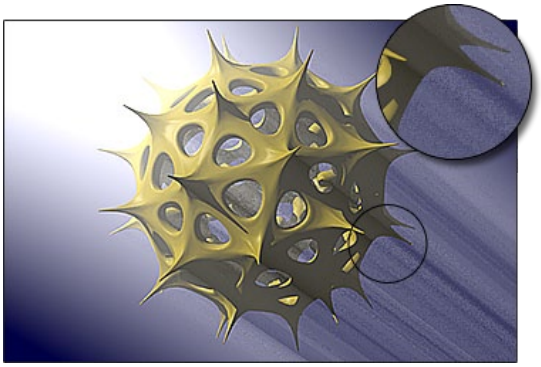
Soft Edge: The soft edge defines an illumination falloff inside the edge of the spot lights cone producing a more natural look. The soft edge is contained within the 'Cone Angle', and attenuates illumination at the edges; a 40° Cone Angle with a 10° falloff would really be a 20° cone angle of full illumination that would falloff 10° across the outer edges of the cone. This value can also be set interactively with the Transform tool in the 3D viewport.

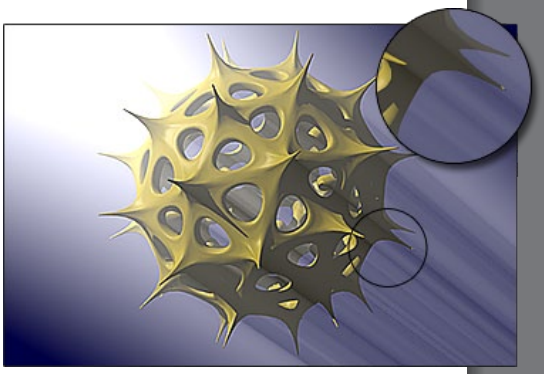
Volumetrics--

When light shines on particular matter suspended in the air, such as water vapor, smoke or even pollution, the light rays become visible. In computer graphics this effect is known as volumetric lights, their affect can be subtle or pronounced, either way it can add a good amount of believability and atmosphere to renders. Additional Settings related to volumetrics can be found in the Light Material item.

Volumetrics Checkbox: The 'Volumetrics' checkbox toggles the effect on or off. When enabled, the sample, height, and radius options become available.

Samples: Any objects that intersect the light beam volume will cast shadows through it. You may have witnessed this when sunlight shining through the clouds creates streaks of light sometimes referred to as 'god rays'. The 'Sample' setting defines the degree of accuracy that modo uses to calculate these shadows through the volume, higher numbers of samples will give smoother and more accurate results, while lower values will render more quickly but become increasingly grainy. The sample below demonstrates the difference between 32 samples on the left and 256 samples on the right.

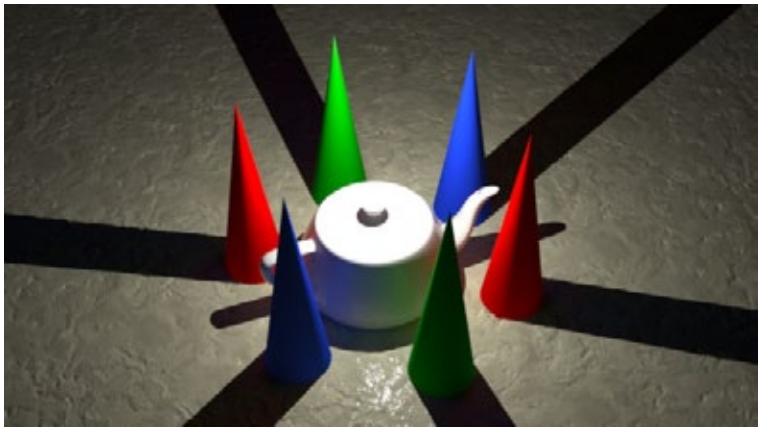




Height: This is the depth of the volume as measured extending out from the front of the light. A setting of 20m will extend the rays 20m from the light including falloff.

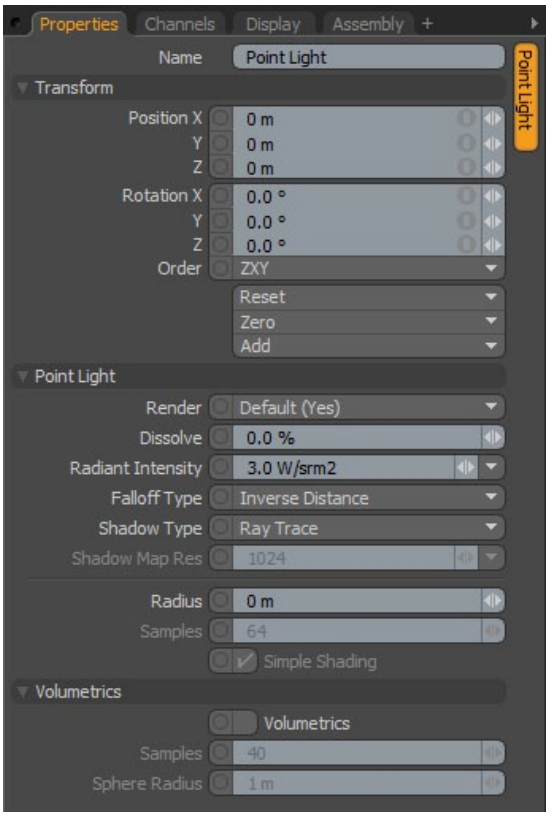
Base: The 'Base' value sets the circular region around the light where volume effect will originate from.

Point Light



The Point Light emits light equally in all directions, similar to a light bulb or a candle flame as all rays originate from a single point. Point lights obey the physical law of inverse square falloff for intensity. As with all Items, the Point Light controls are located on the Properties tab when the light is selected. Point lights have Position and Rotation options as well as controls for the intensity of the light, shadow type, and other settings covered fully below. To control the lights color, shadow color and other material related properties, refer to the Light Material page of the documentation.

Point Light



Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the light item in XYZ space. By default, Position transforms originates from the items center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the light item. By default, Rotation transforms originates from items center position. Since Point Lights radiate omni-directionally, rotation has no effect this type of light.

Order: Allows the user to set the order that rotations are applied to the light item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Reset: Resets the selected transform values to (0,0,0) returning the items back to the world space center position.

Zero: Returns the light items Center position to the world space center (0,0,0) without changing the position of the light item itself.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Point Light--

Render: This drop down menu allows the user to select from 3 choices, when set to 'Default', the user can enable/disable lights using the visibility function "👁" of the item list. When the light is visible, it contributes to the final rendered scene and when

invisible, it will not. On some instances the user may prefer to fix this state, setting the light as 'On' (enabled) or 'Off' (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving the user from manually enabling lights for test renders.

Dissolve: When the 'Dissolve' function is set to any value above 0%, the lights overall affect on the scene will attenuate as the value increases. When set to 100%, the lights effect on the scene is completely disabled. This function providing a convenient way to dim a lights effect within a scene.

Radiant Intensity: This value controls the intensity of the light and uses the standard physically based unit of Watts per volumetric meter. As you would expect, increasing this value will increase the amount of light coming from the Point Light and decreasing the value reduces the light intensity. With Point lights, the area of light generated in the scene is very small and therefore large values are likely necessary to illuminate the scene how the user would expect.

TIP: Keep in mind that a lights intensities and falloffs adhere to real world scale values, scenes that do not adhere to a real world scale factor will likely require intensities to be adjusted accordingly to achieve appropriate illumination.

Falloff Type: Light in the real world isn't a uniform brightness, its intensity diminishes with distance. Photographers will be familiar with the concept that a light will be a quarter as bright at twice the distance away, know as the Inverse Square Law. modo lights default to this setting providing a realistic way to light a scene, however there are times when users may not wish to have this behavior, so modo provides three falloff type options-

None- No falloff, light is consistently bright across its distance.

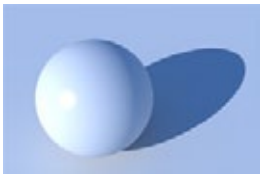
Inverse Distance- Light is half as bright at twice the distance.

Inverse Distance Squared- Default behavior, Light will be a quarter as bright at twice the distance.

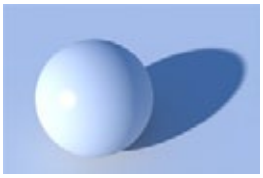
Shadow Type: This popup offers options between 'Ray Traced' , 'None' and 'Deep Shadow Maps'. In situations where users want a light to cast a shadow, Ray Traced will give the most accurate results. The traditional hard edge of ray traced shadows can easily be softened using 'Spread Angle' value in any light item. Deep Shadow maps are useful for volumetric lights and fur rendering, where a great deal of calculations are required to produce shadows; producing similar results to ray traced shadows while reducing the number of calculations.

Shadow Map Res: The 'Shadow Map Res' Sets the resolution for the calculated deep shadow map in pixels. This option is only available when Deep Shadow Map is selected.

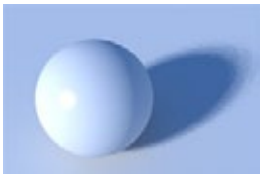
Radius: The default value of 0m results in crisp hard-edged shadows. Increasing the size prompts modo to sample lighting from an array of positions within the radius, the various samples are blended together producing the smooth result. Keep in mind that increasing this value too high may result in noisy shadows unless the user also increases the number of samples.



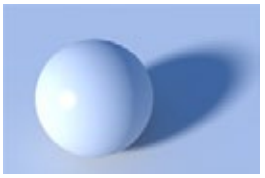
A radius of 0m produces a crisp shadow.



A radius of 2mm produces a softer edged shadow.



A radius of 10mm produces a softer but noisy shadow.



Increasing the 'Samples' value produces a smoother result.

Samples: As the Spread Angle increases, additional samples are required to smooth out the resulting soft edge shadow. If the shadow appears grainy, increasing this sample setting is the most likely solution. The default value of 16 is a good starting point for balancing speed and quality. As the spread increases or the resolution of the image increases you can drive this number higher to account for any grain that appears. Keep in mind that increasing the number of samples will also increase render times.

Simple Shading: When generating soft shadows with a radius greater than 0m, modo calculates the shadow as if sampled from an array of lights, these samples are blended together to produce the smooth result. With the 'Simple Shading' checkbox enabled, modo will calculate the specular and diffuse values for a surface as if the light were a single point, regardless of the radius size. However, with 'Simple Shading' disabled, modo uses the same light array as the shadow to calculate the specular and diffuse values. This results in diffuse and specular shading that is 'soft', like the shadow but can also become increasingly grainy with a greater 'Radius' value. Increasing the number of 'Samples' will reduce grain in the diffuse and specular shading as well as the shadows at the expense of longer render times.

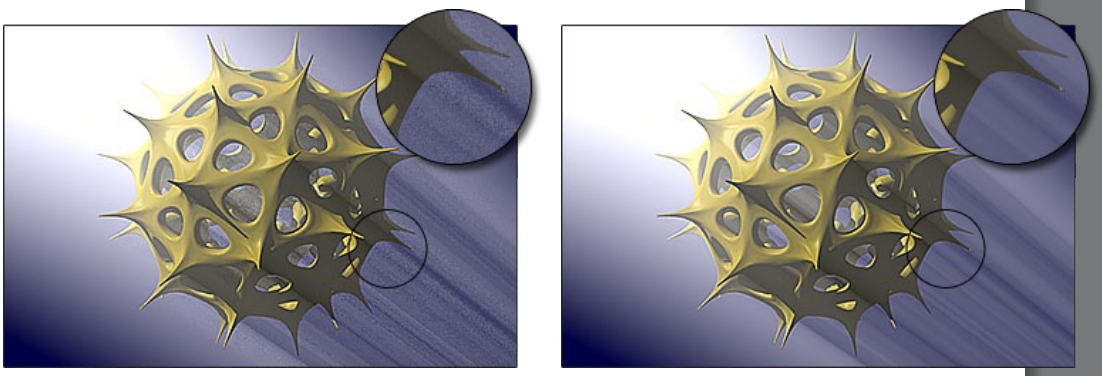
Volumetrics--

When light shines on particular matter suspended in the air, such as water vapor, smoke or even pollution, the light rays become visible. In computer graphics this effect is known as volumetric lights, their affect can be subtle or pronounced, either way it

can add a good amount of believability and atmosphere to renders. Additional Settings related to volumetrics can be found in the Light Material.

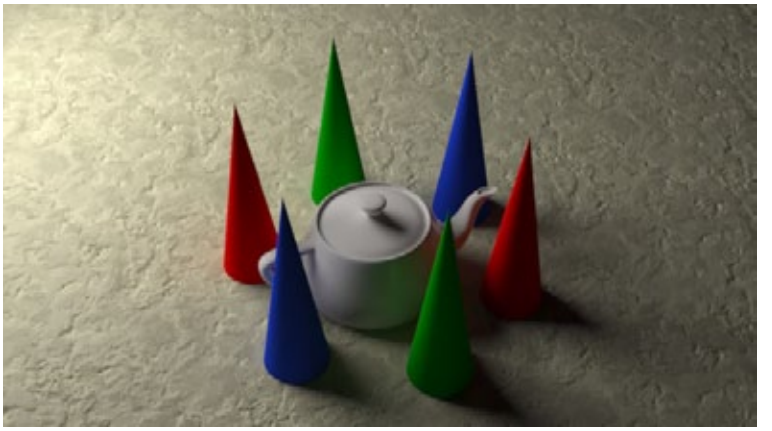
Volumetrics Checkbox: The 'Volumetrics' checkbox toggles the effect on or off. When enabled, the sample, height, and radius options become available.

Samples: Any objects that intersect the light beam volume will cast shadows through it. You may have witnessed this when sunlight shining through the clouds creates streaks of light sometimes referred to as 'god rays'. The 'Sample' setting defines the degree of accuracy that modo uses to calculate these shadows through the volume, higher numbers of samples will give smoother and more accurate results, while lower values will render more quickly but become increasingly grainy. The sample below demonstrates the difference between 32 samples on the left and 256 samples on the right.



Sphere Radius: This is the size of the volume overall as measured from the lights center positions extending out in all directions around the source. A setting of 10m will produce a sphere with an overall radius of 20m including falloff.

Area Light



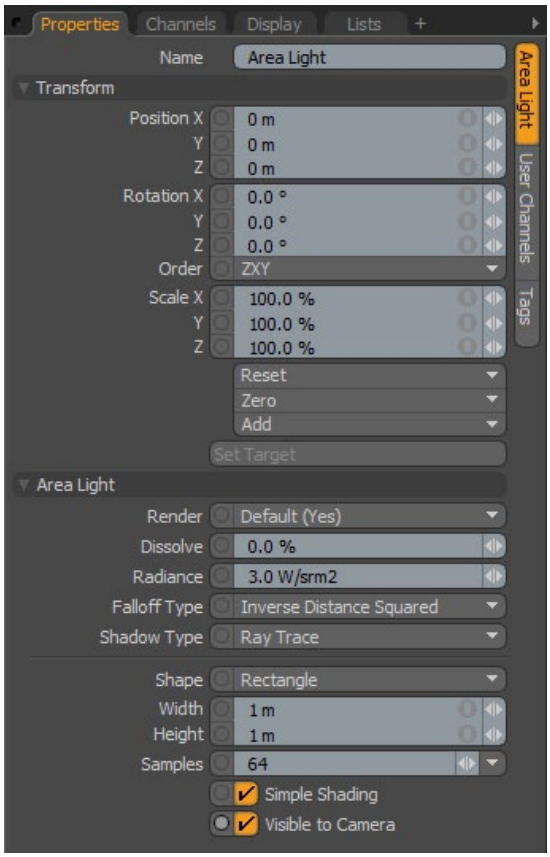
The Area Light is similar to a studio soft box style light, often used in professional photography because of the soft shadows they produce. Light comes from a square or round plane depending on the light settings. Area lights are excellent for creating photo realistic lighting effects as they create a very natural light emission and smooth shading. They are typically more expensive, render time wise, than other light types because they require more samples due to their size but their natural tendency to create soft edged shadows is worth the tradeoff in many situations. Area Lights use a physically accurate light model so the intensity of the light falls off with the inverse square of the distance to the light, for example a surface will be half as bright when the light is twice as far away. As a physically based light, the intensity is also increased as the volume (size) of the light increases. Increasing the size of the light relative to the models it affects will also increase the amount of shadow spread. Options for the Area Light include Position, Rotation, Scale as well as Radiance, Shadow Type, Samples, Sample and Area Width/Height. When adjusting the Area Light in the 3D view be sure to use the Item Tranform tool. This will provide additional handles so you can adjust the length and width of the light directly in the 3D viewport. To control the lights color, shadow color and other material related properties, refer to the Light Material page of the documentation.

Area Light

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the light item in XYZ space. By default, Position transforms originates from the items center position.



Rotation: An Item transform that allows the user to numerically set the rotation of the light item. By default, Rotation transforms originates from items center position.

Order: Allows the user to set the order that rotations are applied to the light item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. This scale transform is a multiplier of the Height and Width options.

Reset: Resets the selected transform values to (0,0,0) returning the items back to the world space center position.

Zero: Returns the light items Center position to the world space center (0,0,0) without changing the position of the light item itself.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Set Target: By selecting the Light item, and a single additional item in the item list, and then pressing 'Set Target', this function allows users the ability to target specific items in a scene, automating the rotation of an item, so that it continuously points toward the targeted item. Once activated, additional option appear--

Remove Target: Removes the target link between the two items.

Enable: Toggling this option off temporarily disables the targeting function while retaining the link between the items.

Set Focus Distance: This option sets the focal distance for DOF when targeting camera items to other item.

Roll: Provides the ability to offset the items rotation angle away from the target.

Time Offset: Provides users the ability to time offset, by a number of frames, how the light follows the target item. It can either be delayed behind it with a negative value, or run ahead of it with a positive value.

Area Light--

Render: This drop down menu allows the user to select from 3 choices, when set to 'Default', the user can enable/disable lights using the visibility function " " of the item list. When the light is visible, it contributes to the final rendered scene and when invisible, it will not. On some instances the user may prefer to fix this state, setting the light as 'On' (enabled) or 'Off' (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving the user from manually enabling lights for test renders.

Dissolve: When the 'Dissolve' function is set to any value above 0%, the lights overall affect on the scene will attenuate as the value increases. When set to 100%, the lights effect on the scene is completely disabled. This function providing a convenient way to dim a lights effect on your scene.

Radiance: This value controls the intensity of the light and uses the Watts per steridian per square meter. As you would expect, increasing this value will increase the apparent brightness of the Area Light and decreasing the value reduces the lights intensity. Area light shading is calculated from the size and shape of the light, so larger lights will produce more illumination in the scene than smaller ones will and adjustment will be necessary when scaling the size of the light.

Falloff Type: Light in the real world isn't a uniform brightness, its intensity diminishes with distance. Photographers will be familiar with the concept that a light will be a quarter as bright at twice the distance away, know as the Inverse Square Law. modo lights default to this setting providing a realistic way to light a scene, however there are times when users may not wish to have this behavior, so modo provides three falloff type options-

None- No falloff, light is consistently bright across its distance.

Inverse Distance- Light is half as bright at twice the distance.

Inverse Distance Squared- Default behavior, Light will be a quarter as bright at twice the distance.

Shadow Type: This popup offers options between 'Ray Traced' , 'None' and 'Deep Shadow Maps'. In situations where users want a light to cast a shadow, Ray Traced will give the most accurate results. Deep Shadow maps are useful for volumetric

lights and fur rendering, where a great deal of calculations are required to produce shadows; producing similar results to ray traced shadows while reducing the number of calculations.

Shape: Area lights can be specified as 'Rectangular' or 'Elliptical'. The shape of the light will slightly impact total distribution of light.

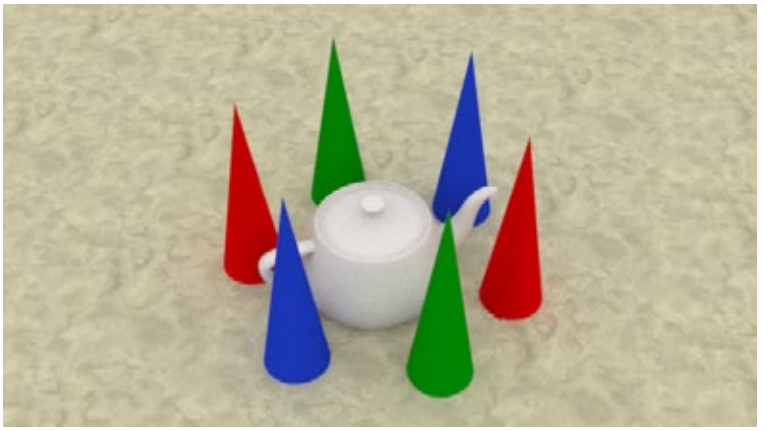
Width/Height: These values set the size for the light as an absolute value for width and height and is based on the current units. By default the unit system is Metric so the width and height would be set directly as meters. When the light item is selected in the 3D viewport, additional handles are available allowing users to interactively scale the area light item. Keep in mind that the size of the light affects the apparent intensity as mentioned above.

Samples: As the size of the area light increases, so does the soft edge spread of the shadow. If the resulting shadows appear grainy, additional samples may be required to smooth out the results. The default value of 16 is a good starting point for balancing speed and quality. One may also need to increase the number of samples as image resolution increases, keeping in mind that increasing the number of samples will also increase the render time.

Simple Shading: With the 'Simple Shading' checkbox enabled, modo will calculate the specular and diffuse values for a surface as if the area light were a single point, regardless of the width and height. However, with 'Simple Shading' disabled, modo uses the same light array as the shadow to calculate the specular and diffuse values. This results in diffuse and specular shading that is 'soft', like the shadow but can also become increasingly grainy with a lower number of samples. Increasing the number of 'Samples' will reduce grain in the diffuse and specular shading as well as the shadows at the expense of longer render times.

Visible to Camera: Users can enable this option to make the Area Light itself visible to cameras and reflective surfaces. The size and shape of the visible plane is determined by the 'Shape', 'Width' and 'Height' attributes of the light.

Dome Light



The Dome light creates a virtual dome of lights around the scene producing a flat, even illumination over surfaces similar to photos taken on an overcast day. Dome lights have often been used to simulate Global Illumination; however, with the modo rendering engine, Global Illumination is just as fast, if not faster than dome lights and generally yields more pleasing results. This light has Position and Rotation controls as well as Radiance, Shadow Type and Samples. Note that there is no size or scale value as the dome will always scale itself to encompass the entire scene, regardless of size. To control the lights color, shadow color and other material related properties, refer to the Light Material page of the documentation.

Dome Light

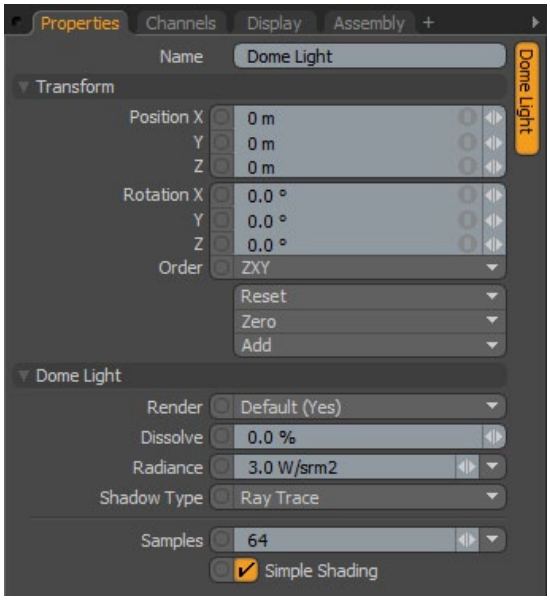
Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the light item in XYZ space. By default, Position transforms originates from the items center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the light item. By default, Rotation transforms originates from items center position.

Order: Allows the user to set the order that rotations are applied to the light item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.



Reset: Resets the selected transform values to (0,0,0) returning the items back to the world space center position.

Zero: Returns the light items Center position to the world space center (0,0,0) without changing the position of the light item itself.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Dome Light--

Render: This drop down menu allows the user to select from 3 choices, when set to 'Default', the user can enable/disable lights using the visibility function "👁" of the item list. When the light is visible, it contributes to the final rendered scene and when invisible, it will not. On some instances the user may prefer to fix this state, setting the light as 'On' (enabled) or 'Off' (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving the user from manually enabling lights for test renders.

Dissolve: When the 'Dissolve' function is set to any value above 0%, the lights overall affect on the scene will attenuate as the value increases. When set to 100%, the lights effect on the scene is completely disabled. This function providing a convenient way to dim a lights effect within a scene.

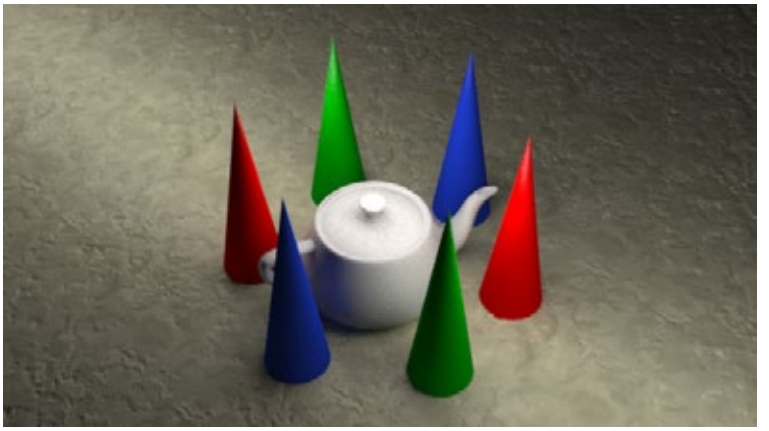
Radiance: This value controls the intensity of the light and uses the standard physically based unit of Watts per volumetric meter. As you would expect, increasing this value will increase the apparent brightness of the Dome Light and decreasing the value reduces the lights intensity. Since the area that produces the light is so large, small values are all that are necessary to illuminate a scene properly.

Shadow Type: This popup offers options between 'Ray Traced' , 'None' and 'Deep Shadow Maps'. In situations where users want a light to cast a shadow, Ray Traced will give the most accurate results. Deep Shadow maps are useful for fur rendering, where a great deal of calculations are required to produce shadows; producing similar results to ray traced shadows while reducing the number of calculations.

Samples: A Dome light illuminates a scene in a way similar to sampling lights from multiple positions across the dome and blending them for the final result. The 'Samples' setting defines how many samples are taken. Low sample values often produce shading that appears too grainy, increasing this value will reduce or eliminate the grain at the expense of longer render times.

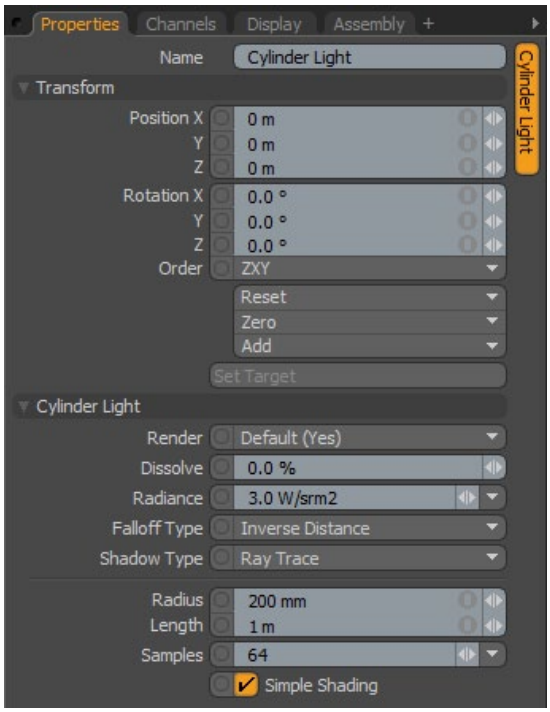
Simple Shading: With the 'Simple Shading' checkbox enabled, modo will calculate the specular and diffuse values for a surface as if the dome light were a single point. However, with 'Simple Shading' disabled, modo uses the same light array as the shadows to calculate the specular and diffuse values. This results in diffuse and specular shading that is 'soft', matching the shadows, but can also become increasingly grainy with a lower number of samples. Increasing the number of 'Samples' will reduce grain in the diffuse and specular shading as well as the shadows at the expense of longer render times.

Cylinder Light



The Cylinder light is similar to a common fluorescent tube in that it emits light in a radial fashion outward from a cylindrical shape. Cylinder Lights use a physically accurate light model so the intensity of the light falls off with the inverse square of the distance to the light, for example a surface will be half as bright when the light is twice as far away. As a physically based light, the intensity is also increased as the volume (size) of the light increases. Increasing the size of the light relative to the models it affects will also increase the amount of shadow spread. The light has Position and Rotation controls as well as Radiance, Shadow Type, Samples, Cylinder Radius and Length. When adjusting the Cylinder light in the 3D view be sure to use the Item Transform tool. This will provide additional handles so you can adjust the length and diameter of the Cylinder light directly in the 3D viewport. To control the lights color,

shadow color and other material related properties, refer to the Light Material page of the documentation.



Cylinder Light

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the light item in XYZ space. By default, Position transforms originates from the items center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the light item. By default, Rotation transforms originates from items center position.

Order: Allows the user to set the order that rotations are applied to the light item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Reset: Resets the selected transform values to (0,0,0) returning the items back to the world space center position.

Zero: Returns the light items Center position to the world space center (0,0,0) without changing the position of the light item itself.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Set Target: By selecting the Light item, and a single additional item in the item list, and then pressing 'Set Target', this function allows users the ability to target specific items in a scene, automating the rotation of an item, so that it continuously points toward the targeted item. Once activated, additional option appear--

Remove Target: Removes the target link between the two items.

Enable: Toggling this option off temporarily disables the targeting function while retaining the link between the items.

Set Focus Distance: This option sets the focal distance for DOF when targeting camera items to other item.

Roll: Provides the ability to offset the items rotation angle away from the target.

Time Offset: Provides users the ability to time offset, by a number of frames, how the light follows the target item. It can either be delayed behind it with a negative value, or run ahead of it with a positive value.

Cylinder Light--

Render: This drop down menu allows the user to select from 3 choices, when set to 'Default', the user can enable/disable lights using the visibility function " " of the item list. When the light is visible, it contributes to the final rendered scene and when invisible, it will not. On some instances the user may prefer to fix this state, setting the light as 'On' (enabled) or 'Off' (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving the user from manually enabling lights for test renders.

Dissolve: When the 'Dissolve' function is set to any value above 0%, the lights overall affect on the scene will attenuate as the value increases. When set to 100%, the lights effect on the scene is completely disabled. This function providing a convenient way to dim a lights effect on your scene.

Radiance: This value controls the intensity of the light and uses the standard physically based unit of Watts per volumetric meter. As you would expect, increasing this value will increase the apparent brightness of the Cylinder Light and decreasing the value reduces the lights intensity. Keep in mind larger lights will appear brighter in the scene because of the increased surface area. When scaling lights, 'Radiance' will need to be adjusted appropriately.

Falloff Type: Light in the real world isn't a uniform brightness, its intensity diminishes with distance. Photographers will be familiar with the concept that a light will be a quarter as bright at twice the distance away, know as the Inverse Square Law. modo lights default to this setting providing a realistic way to light a scene, however there are times when users may not wish to have this behavior, so modo provides three falloff type options-

None- No falloff, light is consistently bright across its distance.

Inverse Distance- Light is half as bright at twice the distance.

Inverse Distance Squared- Default behavior, Light will be a quarter as bright at twice the distance.

Shadow Type: This popup offers options between 'Ray Traced' , 'None' and 'Deep Shadow Maps'. In situations where users want a light to cast a shadow, Ray Traced will give the most accurate results. Deep Shadow maps are useful for fur rendering, where a great deal of calculations are required to produce shadows; producing similar results to ray traced shadows while reducing the number of calculations.

Radius/Length: These values set the size for the light as an absolute value for 'Radius' and 'Length' and is based on the current units. By default the unit system is Metric so the radius and length would be set directly as meters. The size of the light affects the apparent intensity as mentioned above.

Samples: As the size for the cylinder light increases, the shading produced from the light may appear grainy and additional will likely be required to produce a smooth result. Keep in mind that increasing the number of samples will also increase render times.

Simple Shading: With the 'Simple Shading' checkbox enabled, modo will calculate the specular and diffuse values for a surface as if the area light were a single point, regardless of the width and height. However, with 'Simple Shading' disabled, modo uses the same light array as the shadow to calculate the specular and diffuse values. This results in diffuse and specular shading that is 'soft', like the shadow but can also become increasingly grainy with a lower number of samples. Increasing the number of 'Samples' will reduce grain in the diffuse and specular shading as well as the shadows at the expense of longer render times.

Photometric Light

Photometric lights use data provided by lighting manufacturers in the form of IES or EULUMDAT format photometry files. These files contain 'photometric' measurements of a light's intensity in different directions, as well as information about the size of the actual luminous surface. Modo uses this directional intensity information to shape the distribution of light and match real-world "luminaires" (light sources or fixtures). These photometric files also describe a luminaire's real-world intensity, and this value is extracted from the lighting distribution and displayed in the light's Intensity channel, where you can edit it to suit your scene. Keep in mind, editing the intensity means that it is no longer an accurate simulation of real lighting. The photometry files remain external to the modo scene files, and like images, must be available when loading the scene.

Photometric Light

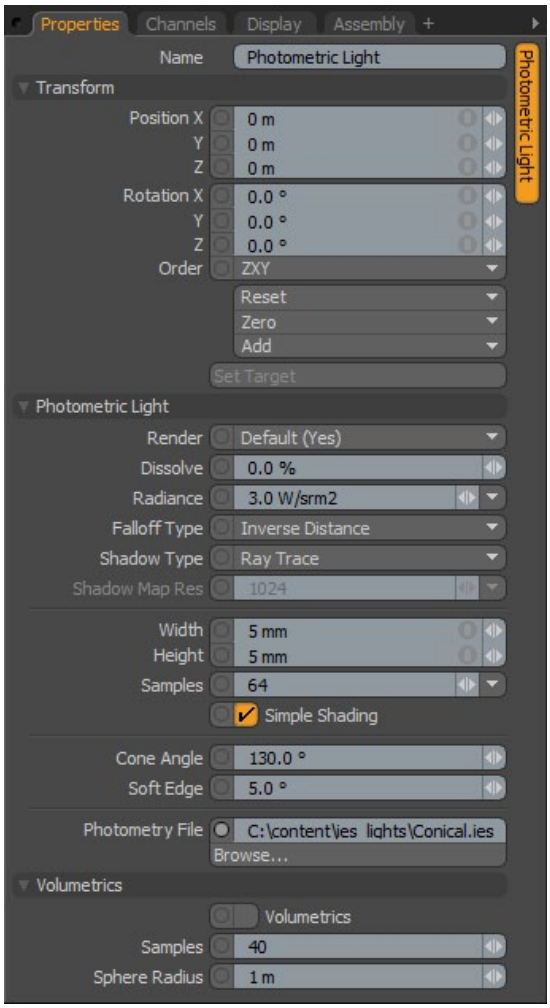
Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the light item in XYZ space. By default, Position transforms originates from the items center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the light item. By default, Rotation transforms originates from items center position.

Order: Allows the user to set the order that rotations are applied to the light item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.



Reset: Resets the selected transform values to (0,0,0) returning the items back to the world space center position.

Zero: Returns the light items Center position to the world space center (0,0,0) without changing the position of the light item itself.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Set Target: By selecting the Light item, and a single additional item in the item list, and then pressing 'Set Target', this function allows users the ability to target specific items in a scene, automating the rotation of an item, so that it continuously points toward the targeted item. Once activated, additional option appear--

Remove Target: Removes the target link between the two items.

Enable: Toggling this option off temporarily disables the targeting function while retaining the link between the items.

Set Focus Distance: This option sets the focal distance for DOF when targeting camera items to other item.

Roll: Provides the ability to offset the items rotation angle away from the target.

Time Offset: Provides users the ability to time offset, by a number of frames, how the light follows the target item. It can either be delayed behind it with a negative value, or run ahead of it with a positive value.

Photometric Light--

Render: This drop down menu allows the user to select from 3 choices, when set to

'Default', the user can enable/disable lights using the visibility function " " of the item list. When the light is visible, it contributes to the final rendered scene and when invisible, it will not. On some instances the user may prefer to fix this state, setting the light as 'On' (enabled) or 'Off' (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving the user from manually enabling lights for test renders.

Dissolve: When the 'Dissolve' function is set to any value above 0%, the lights overall affect on the scene will attenuate as the value increases. When set to 100%, the lights effect on the scene is completely disabled. This function providing a convenient way to dim a lights effect within a scene.

Radiance: This value controls the intensity of the light and uses the standard physically based unit of Watts per volumetric meter. The initial value is extracted from the Photometry file, so modifying this 'Radiance' means the light is no longer an accurate simulation of the light. As you would expect, increasing this value will increase the amount of light coming from the light and decreasing the value reduces the lights overall intensity.

TIP: Keep in mind that a Photometric Lights intensities and falloffs adhere to real world scale values, scenes that do not adhere to a real world scale will not provide accurate lighting simulations as specified by the photometric file.

Falloff Type: Light in the real world isn't a uniform brightness, its intensity diminishes with distance. Photographers will be familiar with the concept that a light will be a quarter as bright at twice the distance away, know as the Inverse Square Law. modo lights default to this setting providing a realistic way to light a scene, however there are times when users may not wish to have this behavior, so modo provides three falloff type options-

None- No falloff, light is consistently bright across its distance.

Inverse Distance- Light is half as bright at twice the distance.

Inverse Distance Squared- Default behavior, Light will be a quarter as bright at twice the distance.

Shadow Type: This popup offers options between 'Ray Traced', 'None' and 'Deep Shadow Maps'. In situations where users want a light to cast a shadow, Ray Traced will give the most accurate results. Deep Shadow maps are useful for volumetric lights and fur rendering, where a great deal of calculations are required to produce

shadows; producing similar results to ray traced shadows while reducing the number of calculations.

Shadow Map Res: The 'Shadow Map Res' Sets the resolution for the calculated deep shadow map in pixels. This option is only available when Deep Shadow Map is selected. Increasing this value will increase the quality and accuracy of the generated map, while increasing renter times as well.

Width/Height: The 'Width/Height' setting defines the size of the illumination area for the Photometric light, much like it does for an area light. Larger values produce more light while smaller values produce less. The size value are extracted from the photometry file, modifications to this value would no longer produce an accurate simulation of the light.

Samples: As the 'Width/Height' increases, additional samples are required to smooth out the resulting soft edge shadow. If the shadow appears grainy, increasing this sample setting is the most likely solution. The default value of 16 is a good starting point for balancing speed and quality. As the size increases or the resolution of the image increases you can drive this number higher to account for any grain that appears. Keep in mind that increasing the number of samples will also increase render times.

Simple Shading: When generating soft shadows with a width/height greater than 0m, modo calculates the shadow as if sampled from an array of lights, these samples are blended together to produce the smooth result. With the 'Simple Shading' checkbox enabled, modo will calculate the specular and diffuse values for a surface as if the light were a single point, regardless of the lights width and height. However, with 'Simple Shading' disabled, modo uses the same light array as the shadow to calculate the specular and diffuse values. This results in diffuse and specular shading that is 'soft', like the shadow but can also become increasingly grainy with a greater total area. Increasing the number of 'Samples' will reduce grain in the diffuse and specular shading as well as the shadows at the expense of longer render times.

Cone Angle: In a way, Photometric Lights are like a combination of area lights and spot lights where light is emitted in a cone shape within a defined area (width/height). The 'Cone Angle' defines the shape of the cone that is generated; larger values produce wider conical shapes and therefore larger areas of illumination, while smaller values produce tighter cones with smaller areas of illumination. A setting of 360° produces light similar to a point light radiating in all directions equally. The cone angle value is extracted from the photometry file, modifications to this value would no longer produce an accurate simulation of the light.

Soft Edge: The soft edge defines an illumination falloff inside the edge of the spot

lights cone producing a more natural look. The soft edge is contained within the 'Cone Angle', and attenuates illumination at the edges; a 40° Cone Angle with a 10° falloff would really be a 20° cone angle of full illumination that would falloff 10° across the outer edges of the cone. The soft edge value is extracted from the photometry file, modifications to this value would no longer produce an accurate simulation of the light.

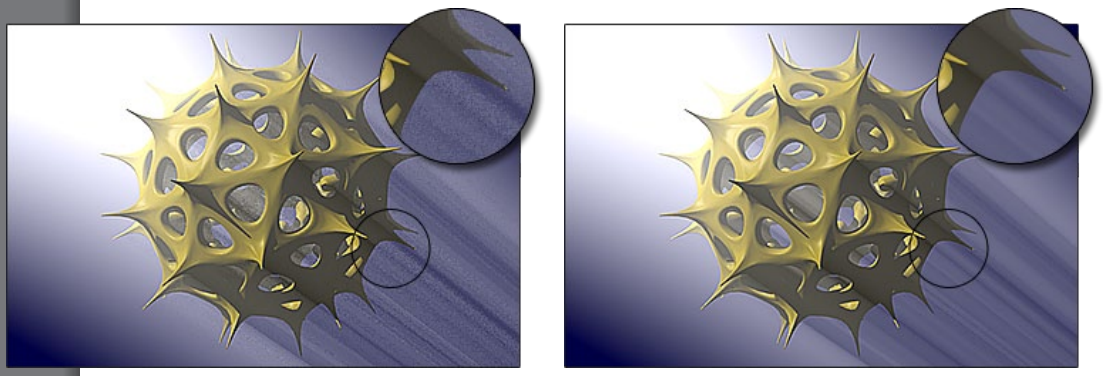
Photometry File: The browse function opens a file dialog window that allows you to locate and load a Photometry File in '.LDT' or '.IES' formats. Like image maps, Photometry files are external need to be available to modo each time a scene is loaded. When moving files, make sure to include the Photometric files as well.

Volumetrics--

When light shines on particular matter suspended in the air, such as water vapor, smoke or even pollution, the light rays become visible. In computer graphics this effect is known as volumetric lights, their affect can be subtle or pronounced, either way it can add a good amount of believability and atmosphere to renders. Additional Settings related to volumetrics can be found in the Light Material item.

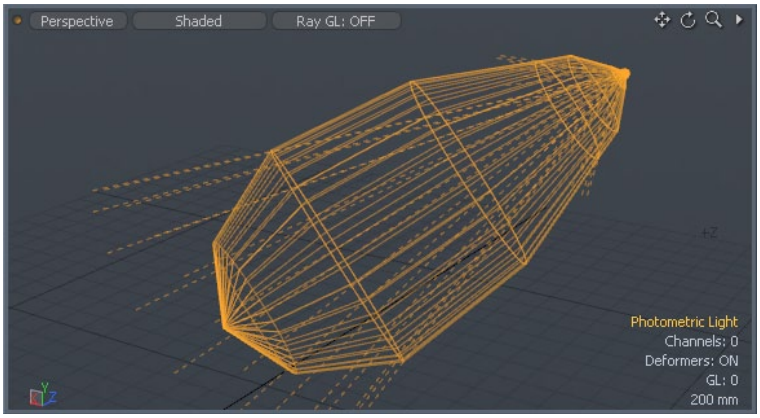
Volumetrics Checkbox: The 'Volumetrics' checkbox toggles the effect on or off. When enabled, the sample, height, and radius options become available.

Samples: Any objects that intersect the light beam volume will cast shadows through it. You may have witnessed this when sunlight shining through the clouds creates streaks of light sometimes referred to as 'god rays'. The 'Sample' setting defines the degree of accuracy that modo uses to calculate these shadows through the volume, higher numbers of samples will give smoother and more accurate results, while lower values will render more quickly but become increasingly grainy. The sample below demonstrates the difference between 32 samples on the left and 256 samples on the right.



Sphere Radius: This is the size of the volume overall as measured from the lights center positions extending out in all directions around the source. A setting of 10m will produce a sphere with an overall radius of 20m including falloff. The actual falloff within the spheres volume will be dictated by the supplied photometric information.

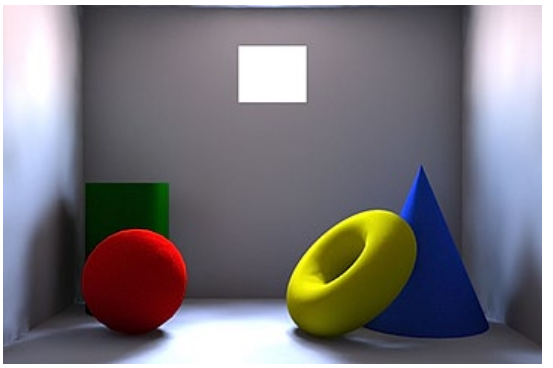
Photometric Light Shape Preview



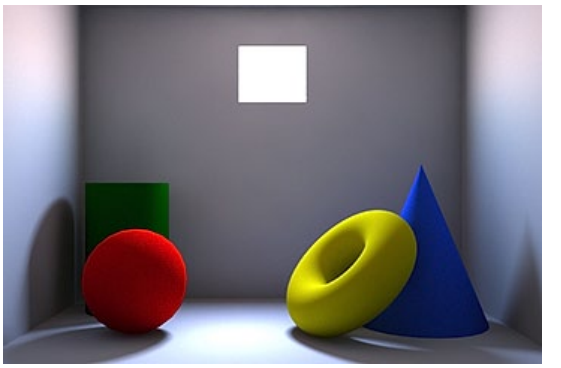
Photometric lights display a displaced wireframe sphere at the lights position with the displacement radius based on the intensity of the photometric data file (IES or EULUMDAT). You can adjust the size of the preview by adjusting the light's display size.

Portal

Portals are not lights in the traditional sense, as they produce no illumination by themselves. They are meant more to control lighting, especially that of interior renders that rely mainly on Global Illumination. Think of them as gathering rays and focusing them into a specific area. In this following example, the interior is illuminated solely by the environment, no lights are being used whatsoever. At the default settings, the image on the left exhibits some chattering and splotches, especially in the area of shadow transitions. In the second image on the right, a Portal has been added to the scene and set with 512 light samples. This produces a much smoother result with no other setting or value changes.



No Portal, 512 IC Samples

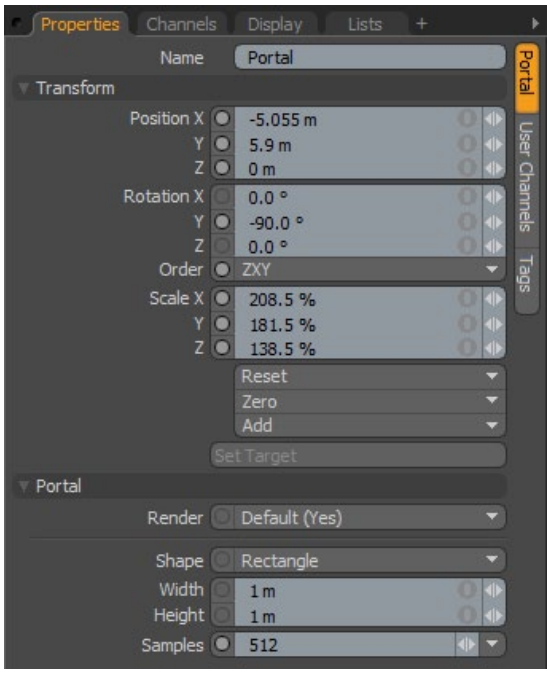


Portal Added with 512 Samples

The main use for 'Portals' is not surprisingly to be placed at openings; anywhere light would enter into an interior, such as a window, door or skylight. For best results users should position the portal directly outside the portal itself (or inside depending on the shape of the opening) and scale it to be slightly larger than opening. Users can also adjust the shape of the portal to better represent the shape of the window, either as round or rectangular. Users should also note it is better to use fewer Portals than more, as this can add significantly to render calculation times, for example when representing a multi-paned window, using a single Portal scaled to the over-all window size would be better than individual Portals for each pane. Users should also note that Portals take their color from the environment behind them and therefor have no Light Material setting, like those of other direct light sources.

Portal

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.



Transform--

Position: An Item transform that allows the user to numerically position the Portal item in XYZ space. By default, Position transforms originates from the items center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the Portal item. By default, Rotation transforms originates from items center position.

Order: Allows the user to set the order that rotations are applied to the Portal item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock when animating.

Scale: An Item transform that allows the user to numerically set the scale of the item. By default, Scale transforms originate from the Center position.

Reset: Resets the selected transform values to (0,0,0) returning the items back to the world space center position.

Zero: Returns the Portal items Center position to the world space center (0,0,0) without changing the position of the Portal item itself. This is done by the addition of a negative transform added to the Transform channels.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Portal--

Render: This drop down menu allows the user to select from 3 choices, when set to 'Default', the user can enable/disable Portals using the visibility function " " of the item list. When the Portal is visible, it contributes to the final rendered scene and when invisible, it will not. On some instances the user may prefer to fix this state, setting the Portal as 'On' (enabled) or 'Off' (disabled) regardless of visibility. Also useful for

workflows that auto toggle visibility, saving the user from manually enabling Portals for test renders.

Shape: This option determines the final shape of the Portal, set as either Rectangular or Elliptical. User can then adjust the 'Width' and 'Height' options to further define square, rectangular, circular or oval type shapes.

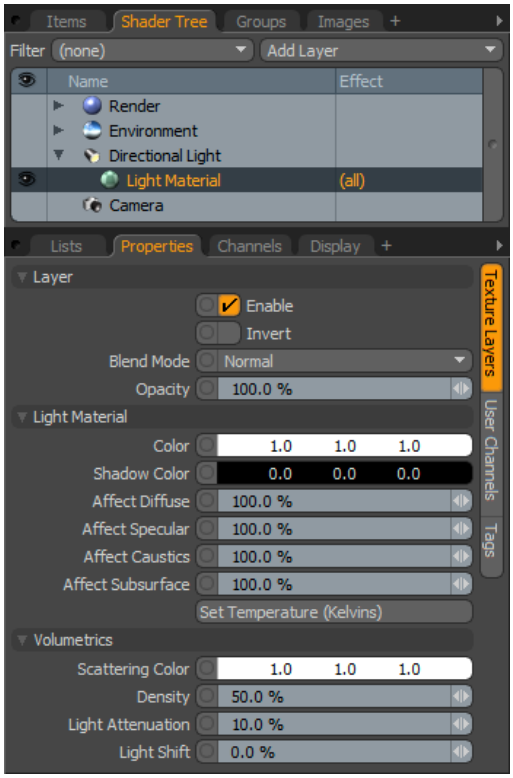
Width/Height: This value determines the actual size of the Portal. Users should size the Portal to approximate the overall size of the window it is representing.

Samples: This value determines the Portals light gathering ability, having greater numbers of Samples will increase the overall accuracy of the final rendered image, but also keep in mind that additional rays will also adversely affect render times, so users should adjust the rays to find a balance of quality versus render time.

Light Material

Lights derive settings for their illumination color, shadow color and how much they contribute to Diffuse and Specular shading from the Light Material item, as well as their volumetric settings as appropriate to the light. While Light items are located within the Item List, the Light Material item is located inside the Shader Tree; outside of the camera which has no material settings, Lights are the only items the have a dual identity in modo existing in both places. The Light Material item can be found by LMB-clicking the small arrow widget next to the Light Items name to reveal/hide the material item. Selecting the item will display its properties within the Properties viewport. All lights have a set of default values that can be easily modified, providing an intuitive and flexible method for customizing a lights effect. Additional settings related to the various light items can be found within the Light Items properties.

Also, keep in mind that since lights are shader tree items, they can have additional shader tree item layers added to them; images can be added to produce 'slide projector' types of effects (by adjusting the settings of the Texture Locator), and by RMB-clicking on the items effect column, different options can be selected producing a variety of different and interesting effects. More on Light Effects can be found in the 'Lights Effect' page of the documentation.



Layer

Enable: Toggles the layer on and off. If there is only one material attributed to the light, this will disable/enable the light itself.

Invert: Inverts the RGB values for the lights material layer.

Blend Mode: Affects blending between material layers applied to the light allowing the user to stack several layers for different effects. For more on blending, please reference the 'Blend Modes' page of the documentation.

Opacity: Changes the transparency of the current material layer. If only one material is associated with your light, this will also dim the light itself.

Light Material

Color: The color control uses the standard modo color component of three RGB values and allows direct control over the color of light emitted. Our brains are used to seeing pure white lights, but often times, there are many reasons to change the color for artistic affect. Also, when simulating outdoor lighting, setting a lights color to cooler values such as blue-ish tones tends to simulate mid to late day-time lighting where as warmer colors such as yellow or orange-tones will be more like evening-time.

Shadow Color: This setting will tint the shadows produced by the associated light. In most cases this color is left as black, producing the 'absense-of-light' shadows we are all used to seeing, but other times, there can be creative reasons to modify the shadow color to either simulate a specific phenomenon, or simply for artistic affect. Some consider it good practice to tint the shadow with a complementary color to that used for the Light 'Color' setting.

Affect Diffuse value of 100%

Affect Diffuse value of 66%

Affect Diffuse value of 33%

Affect Diffuse value of 0%

Affect Diffuse: The diffuse lighting is the light illuminating a surface that is only affected by position and angle of the light source, but is unaffected by the viewers position or angle. Light that behaves this way is described as being isotropic. In simpler terms, this is the normal shading users see over a surface. This percentage value provides the ability to control how much the lights illumination contributes to the overall Diffuse shading within a scene on a per light basis. You can set this value to 0% if you want to disable the Diffuse shading, allowing the creation of a "Specular only" light source.

Affect Specular value of 100%

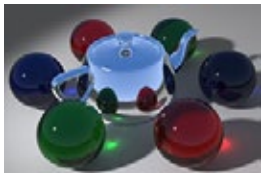
Affect Specular value of 66%

Affect Specular value of 33%

Affect Specular value of 0%

Affect Specular: The specular lighting is light that is dependant on the viewers position and angle. Specular is meant to simulate the reflection of a bright light source upon a surface and is calculated as an angle vector between the camera and the light source itself. This percentage value provides the ability to control how much the light contributes to any Specular shading within a scene on a per light basis. You can set this value to 0% if you want to eliminate Specular shading produced by the light, allowing the creation of a "Diffuse only" light source.

Affect Caustic value of 100%



Affect Caustic value of 66%

Affect Caustic value of 33%

Affect Caustic value of 0%

Affect Caustic: Caustics are the refraction or reflection of light, either through a transparent object or off of a reflective object and projected onto a surface. This percentage value provides the ability to control how much the lights illumination contributes to the overall caustic shading within a scene on a per light basis. You can set this value to 0% if you want to disable the light from contributing to generated Caustics.

Affect Subsurface value of 100%

AffectSubsurface value of 66%

Affect Subsurface value of 33%

Affect Subsurface value of 0%

Affect Subsurface: Subsurface Scattering is the appearance of light entering a translucent surface, diffusing (bouncing around) and then exiting. Typically when this happens, the exiting light will be tinted by the interior of the surface producing a unique effect. The 'Affect Subsurface' percentage value provides the ability to control how much the lights illumination contributes to the overall subsurface scattering within a scene on a per light basis. You can set this value to 0% if you want to disable the light from contributing to subsurface scattering. Values beyond 100% will boost the lights affect on the subsurface amount.

Some common values for K	
1500 K	Candlelight
2680 K	40 W incandescent lamp
3000 K	200 W incandescent lamp
3200 K	Sunrise/sunset
3400 K	Tungsten lamp
3400 K	1 hour from dusk/dawn
5000-4500 K	Xenon lamp/light arc
5500 K	Sunny daylight around noon
5500-5600 K	Electronic photo flash
6500-7500 K	Overcast sky
9000-12000 K	Blue sky

Set Temperature: This setting allows you to set a lights 'Color' based on the familiar kelvin scale for color temperature.

Volumetrics

When light shines on microscopic particles suspended in the air, such as water vapor, smoke or even pollution, the light rays become visible. In computer graphics this phenomenon is known as a volumetric light, their affect can be subtle or pronounced, either way it adds a good amount of believability and atmosphere to rendered images. For Volumetrics to work, they must first be 'Enabled' in the Light items properties where additional settings specific to the volume itself are located.

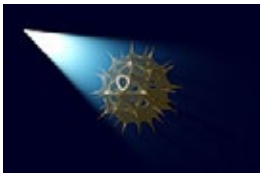
Scattering Color: The 'Scattering Color allows the user to set an RGB color value for the light volume itself. You would usually want this to be the same color as the light, however, there may be some creative reasons one might want to change it to something else entirely.

Density value of 25% | Density value of 50% | Density value of 75% | Density value of 100%

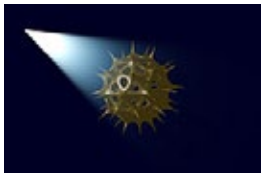
Density: This value sets the 'Density' or thickness for the volumetric effect, higher values would make the volume thicker and more opaque and lower values would make it appear more sparse and transparent.

Attenuation value of 25% | Attenuation value of 50% | Attenuation value of 75% | Attenuation value of 100%

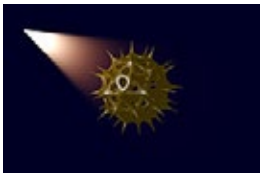
Light Attenuation: Attenuation is the gradual loss of intensity when light is passing through any medium. With an 'Attenuation' setting of 0%, modo produces a smooth gradual falloff along its length, as the 'Light Attenuation' value increase, the volumetric effect gradually fades in overall intensity with a bias toward the light source. Combine with 'Light Shift' to mimic large volumes of light scattering.



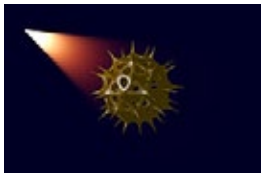
Light Shift value of -100%



Light Shift value of -50%



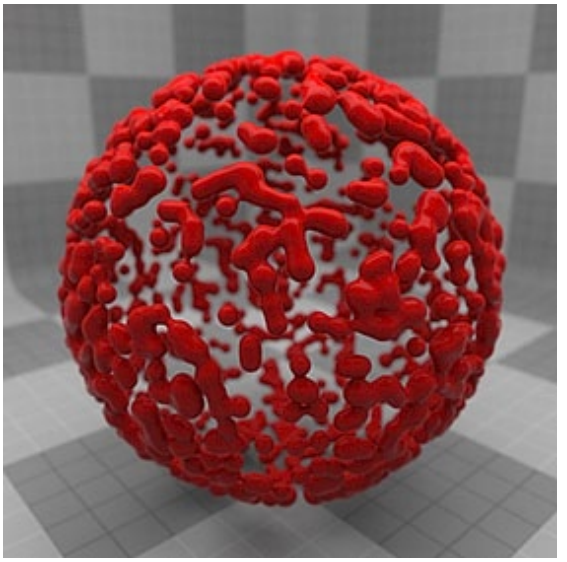
Light Shift value of 50%



Light Shift value of 100%

Light Shift: As the light is increasingly attenuated, certain wavelengths are absorbed or scattered, the 'Light Shift' function can simulate this phenomenon. Negative values push the color of the volume increasingly toward cooler light blues, while positive values push the color toward warmer orange and red hues. The 'Light Attenuation' value must be a number above 0% to see any effect

Volumes



Blobs

The 'Blob' item provides a means for users to procedurally create blobby meta-ball like shapes; combinations of spherical forms with an interpolated smoothly curved surface covering them. The positions of individual blob shapes are determined by the vertices of a 'Particle Source' item, where each vertex in the Particle Source produces a single blob (The Particle Source can be a number of items described under the 'Particle Source' option below). The form itself is dependant on the radius of the Blob particles and the distance between adjoining vertices, as the surfaces get closer together, they will try to join their surfaces (think like water droplets when they get close to each other). The resulting surface can be further controlled with the addition of textures added via the 'Shader Tree' producing complex surfaces very easily.

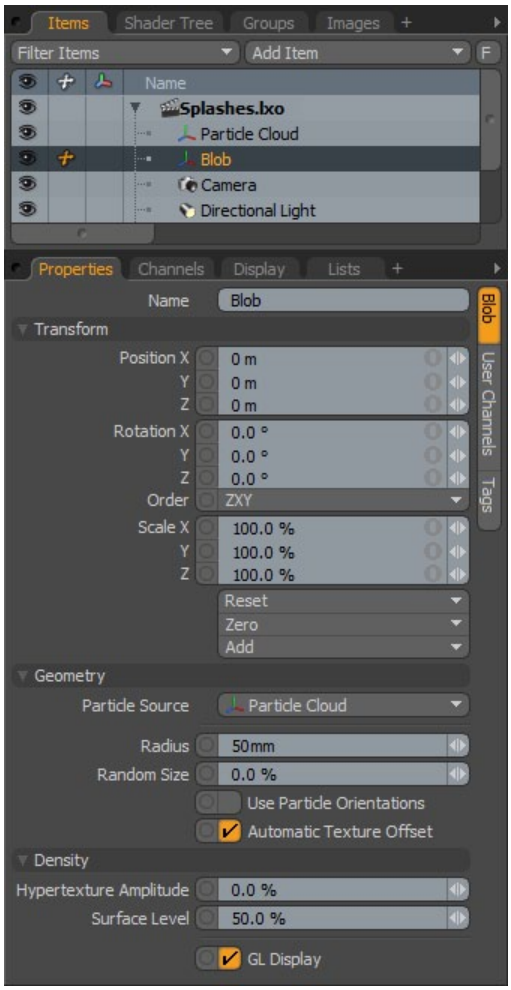
Adding Blobs

A Blob item can be added to a scene by utilizing the 'Add Item' function found in the 'Items List'. Users can LMB+click the button itself, opening the popup menu and select the "Volumes > Blob" option. Once added, the item will appear at the bottom of the Items List viewport. When selected, a number of attributes controlling the blobby shape appear in the 'Properties' viewport.

Applying Materials

Normally, surfaces are applied through the use of polygon tags that define masks in the Shader Tree controlling material placement, however, 'Blobs' are a procedural

surface created at render time and have no user selectable polygons to apply a tag to, so surfacing is applied to the 'Blob' item through the use of an Item Mask. Users can quickly create an 'Item Mask' by RMB+clicking on the Blob item in the Items list and choosing the 'Create Item Mask' option from the contextual menu. Users can then add 'Materials' and other Shader Tree items, setting the appropriate 'Effect' for each layer building up surfacing attributes just as one would for any other surface. Users may also wish to control additional attributes of the Particle Source that are not surface specific (such as 'Particle Size' and 'Particle Density' for the Source), this can be done by creating an additional Item Mask and applying texture layers set to the appropriate 'Effect' under the 'FX Item' of the Shader Tree. For more on using FX, see that page of the documentation.



originate from the Center position.

Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. By default, Position transforms originates from the Center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. By default, Rotation transforms originate from the Center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. By default, Scale transforms

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Mesh position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Geometry--

Particle Source: When a 'Sprite' item is added to a scene, it is represented by a single locator type point at the world origin producing a single sprite image. To produce additional sprites, users will want to define a 'Particle Source', this can be any Mesh Item layer that contains vertices, where each vertex position is used to determine the position of the Sprite particle (any type of geometry can be used- single point polygons, vertices, two-point polygons and regular polygons, however, only the vertex positions of any component will be used). Source layers can be normal geometric forms, point clouds that are imported or created using the 'Particle Cloud' item or any of the 'Particle Tools'. They can be imported particle simulations controlled by way of a MDD deformer, RealFlow Point Cache or an Alembic format data file.

Radius: This option determines the base size of the Sprite image. From here, the size can be altered via the 'Random Size' option, the addition of a 'Particle Size' vertex map that allows users to paint the scale for individual particles interactively, or with the application of a texture layer, under the FX Item of the shader tree, set to control 'Particle Size'.

Random Size: This option controls the amount of randomness added to the scale of each individual particle blob. The amount is determined by a scale percentage- a setting of 25% would adjust the scale of the base radius size by an amount of 0%-25% randomly across all particles, a setting of 50% would adjust the size by 0%-50% randomly across all particles and so on.

Use Particle Orientations: Aligns the particle to the direction vector from the particle source. The direction vector is usually the surface normal for surface particles or the particle velocity vector for simulation particles (like from RealFlow). Note that in the case of blobs, this option is disabled by default as it adds a bit to render time (around 30%), so don't use it if you don't need it.

Automatic Texture Offset: Adds an offset to the hypertexture different for each particle so that each particle looks different.

Density--

Hypertexture Amplitude: Determines the amplitude (intensity) of a 'Volume Density' texture layer applied to the Blob volume. The higher the value the more you see the effect of the density texture. It's similar in concept to displacement amplitude but applies to volumes.

Amplitude 0%

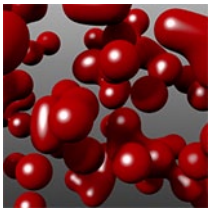
Amplitude 25%

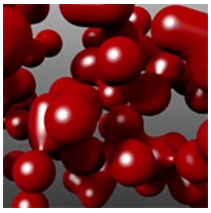
Amplitude 50%

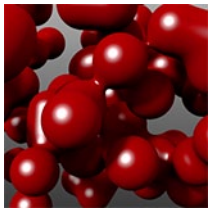
Amplitude 75%

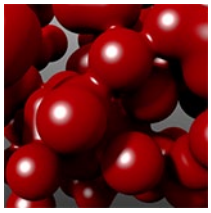
Amplitude 100%

Surface Level: The surface level option determines the height of the limit surface, in-turn controls how smooth or abrupt the transitions are between neighboring Blobs. Low values produce smoother blobs that are slightly smaller, and higher value will produce more spherical shapes that are slightly larger. At 100%, individual blobs will no longer blend and simply look like a collection of spheres.


Surface Level 0%


Surface Level 25%


Surface Level 50%


Surface Level75%

Surface Level 100%

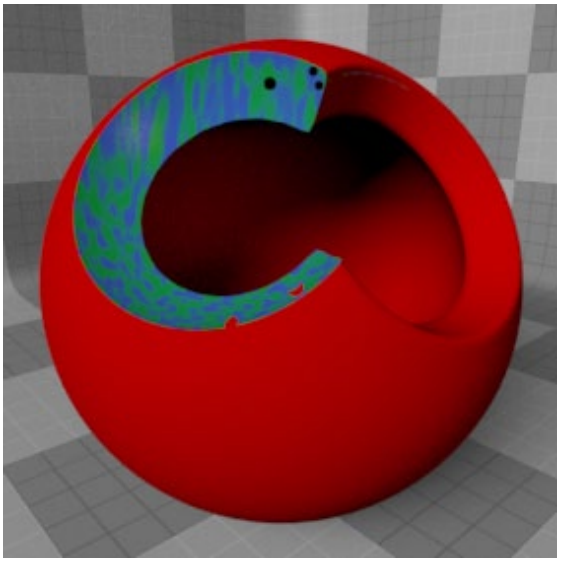
GL Display: This option toggles the display of the 'Blob' proxy in the 3D OpenGL viewport, generally user will want this enabled to get an approximation of how the blob particles will render (the default state), but in heavy scenes it may be useful to disable this option to increase scene interactivity or reduce overall visual clutter.

Render Sprite

Please refer to the on-line documentation.

Volume

Please refer to the on-line documentation.



Render Boolean

Booleans, as they relate to 3D computer graphics, are a technique of using geometry from one mesh item to modify the geometry of another; where they intersect produces a subtractive trimmed surface that would be difficult to create otherwise. The 'Render Boolean' item provides users a means to trim away areas of an object or surface at render time without needing to physically modify the mesh itself. The benefits are that the mesh remains fully intact, the results are fully animateable and it produces a texturable surface where the Render Boolean intersects with the trimmed mesh. Also, since the boolean is generated at render time on the final subdivided geometry, there is a less likely chance of producing the errors generally associated with boolean trimming of Subdivision Surfaces.

Adding Render Booleans

A Render Boolean item can be added to a scene by utilizing the 'Add Item' function found in the 'Items List'. Users can LMB+click the button itself, opening the popup menu and select the "Volumes > Render Boolean" option. Once added, the item will appear at the bottom of the Items list viewport. When selected, a number of attributes controlling the Booleans functions appear in the 'Properties' viewport.

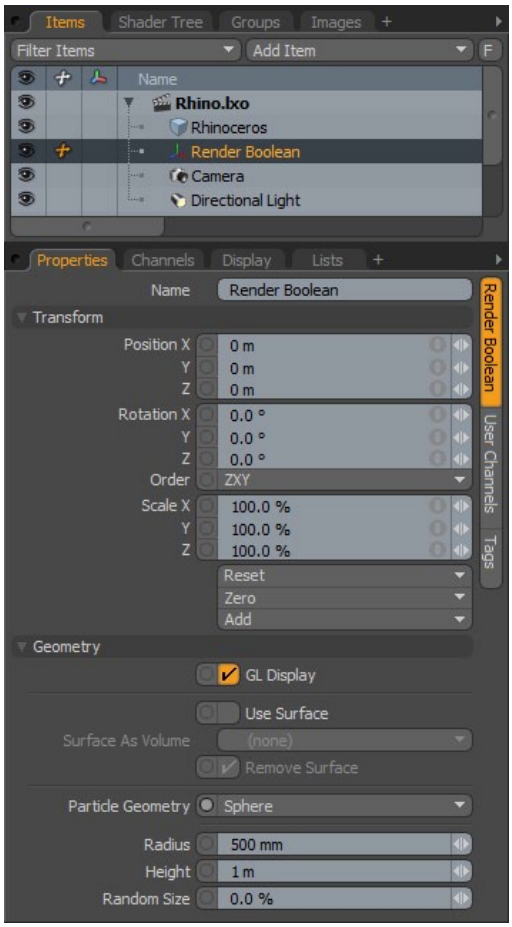
Applying Materials

When utilizing the procedural geometry options of the Render Boolean, users can texture the resulting boolean surface with the use of an item mask. By RMB+clicking on the 'Render Boolean' item in the 'Items List' and selection the 'Create Item Mask'

option, a new Material Group is added to the Shader Tree. Users can use the 'Add Layer' option to add 'Materials' and 'Texture' layers that will then be confined only to the areas where the booleans intersect with the target mesh. For more on working with layers in the 'Shader Tree', please reference that page of the documentation. The 'Render Boolean' item will trim all items/surfaces in the scene unless the user explicitly disables the 'Enable Surface Clipping' option of the Material Item on a per surface basis (This option is enabled by default for all surfaces).

Animated Booleans

Users can animate the Render Boolean operation by creating keyframes on the Render Boolean item when using the procedural shape option, or when a 'Surface as Volume' mesh layer is specified, animating the target mesh layer will produce the animated boolean operation. For more on setting keyframes and animating in modo, please reference the 'Rigging and Animation' section of the documentation.



Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the item in XYZ space. By default, Position transforms originates from the Center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the item. By default, Rotation transforms originate from the Center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.

Scale: An Item transform that allows the user to numerically set the size of the item. By default, Scale transforms originate from the Center position.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Resets the chosen transform property values to '0', leaving the 'Center' position and Mesh position intact.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Geometry--

GL Display: This option toggles the display of the Particle Geometry proxy in the 3D OpenGL viewport, generally user will want this enabled to get an approximation of how the boolean item will render (the default state), but in certain instances it may be useful to disable this option.

Use Surface: When this option is enabled, users can select any arbitrary mesh layer from the scene and use it as a boolean volume. While Render Booleans work for nearly any geometry type, for best results both should be fully closed watertight models, meaning they could virtually hold water without leaking.

Use Surface As Volume: This option determines the mesh layer that will be used for the boolean operation, only when 'Use Surface' is enabled.

Remove Surface: When this option is enabled (the default behavior), the specified mesh layer will be hidden at render time, and will only show the resulting intersection of the Boolean volume and the trimmed geometry.

Particle Geometry: The Render Boolean item also generates procedural geometry, users can choose between a 'Cube', a 'Sphere' and a 'Cylinder'.

Radius/Height: Defines the outside dimensions of the Particle Geometry object. 'Radius' refers to the overall dimensions of the 'Sphere' and 'Cube' shapes, and 'Radius' and 'Height' work together to define the dimensions of a 'Cylinder'.

Random Size: Adds random size variations to the particle geometry.

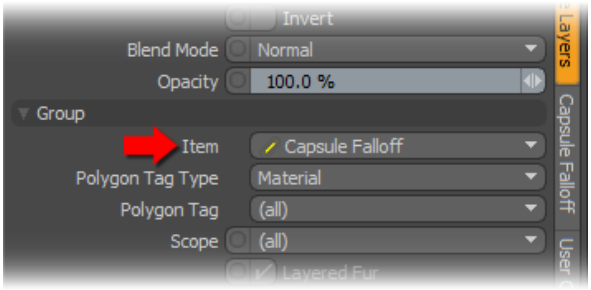
TIP: Render Booleans aren't strictly for trimming geometry, they can also affect volume items such as 'Blobs', 'Sprites' and 'Volumes'.

FX Item

The FX Item provides users a way to control scene related functions that aren't directly surface or shading related using Shader Tree textures. This could be modulating the strength of an animation Falloff using a 'Noise' procedural, or controlling the density of particles in a 'Particle Generator' item with a Gradient. The same procedures of the rest of the Shader Tree apply to the 'FX" item, however since the target items one would apply these textures to are procedural in nature (generated numerically), they can't be tagged in the traditional sense. For the 'FX' item, textures are applied by way of 'Item' masks.

Creating Item Masks

To apply a texture to an item, users will first need to create an 'Item Mask'. This can be done easily by RMB+clicking directly over the target item in the Items list or 3D viewport (making certain to be in 'Items' mode first) opening the context menu. By selecting the 'Create Item Mask' option toward the bottom of the menu, a 'Material Group' item is generated. Now any textures added to the group will only apply to the target item. In some cases the items will be created properly in the 'FX" item (such as with a 'Falloff), but in other cases modo may place the Item mask under the 'Render' Item and users will need to drag and drop the layer to the proper location. For information on adding and working with Shader Tree layers, please reference that page of the documentation.

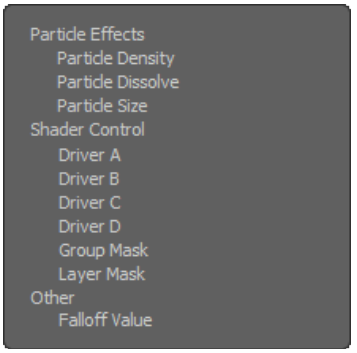


An Item Mask is nothing more than a Material Group with the target layer specified under the 'Item' option.

Adding Textures

Once the mask is generated, users can use the 'Add Layers' function of the Shader Tree to insert the actual texture(s) for use. If the Group Item mask is selected when

the menu is opened, users should note that the list of available layers is reduced, as only applicable layers are present. Once the specified layer is added, users can select the layer by LMB+clicking on the layer name and its associated attributes can be edited in the 'Properties' panel. Please reference the Shader Tree 'Item Layers' menu for information on specific texture layers. In order to direct how the texture layer modulates the target item, users will need to specify the layer 'Effect' type. This can be done by RMB+clicking on the Effect Title of the target layer in the 'Effect' column opening the contextual menu where users may choose an alternate effect.



Effect Types

The effect types available to the FX item layers are as follows-

Particle Density--

The Particle Density option allows users to modulate the frequency of particles generated by a 'Particle Generator' or a 'Particle Cloud' item using procedural textures with a 'Solid' projection.

Particle Dissolve--

The Particle Dissolve option allows users to modulate the opacity of particles in a 'Volume' or a 'Sprite' item using procedural textures with a 'Solid' projection.

Particle Size--

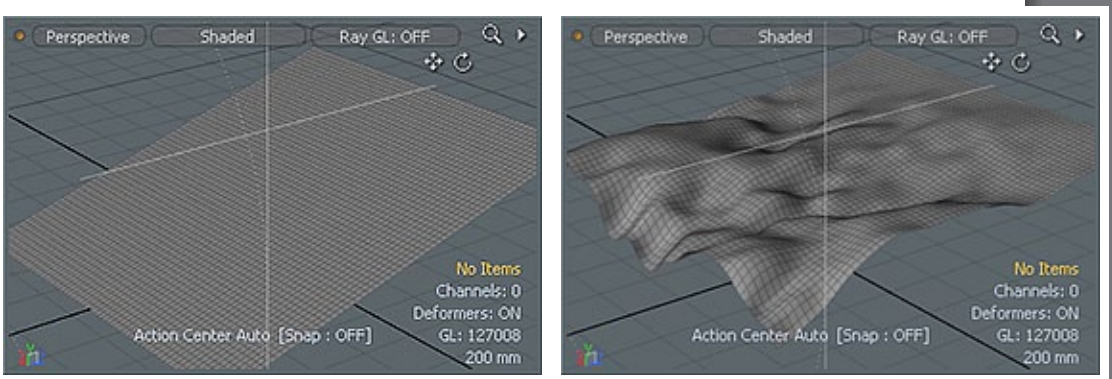
The Particle Size option allows users to modulate the size of particles in a 'Blob', 'Volume' or 'Sprite' item using procedural textures with a 'Solid' projection.

Driver A,B,C & D-- The Driver options allow texture layers to modulate a Gradient layer with a matching Input Parameter. The mapping of the images brightness (luminosity) is passed onto the Gradient whose keyed values replace those of the driving layers and are output as specified by the Gradient layers 'Effect' setting.

Group Mask-- When any texture layer resides within a Group item and is specified as a 'Group Mask', the brightness values of the the layer will mask all layers contained within the group revealing any lower texture layers in the Shader Tree. Black areas will be completely transparent attenuating toward completely opaque represented by white.

Layer Mask-- The layer mask works identically to the Group Mask, but where a Group Mask affects an entire group, a layer mask only affects a single layer. When any texture layer is set as such, it will modulate the the texture layer directly below it in the shader tree. Black areas will be completely transparent attenuating toward completely opaque represented by white.

Falloff Value--



The 'Falloff Value' option allows users to modulate the strength of a 'Falloff' deformer using a modo texture layer. In the example above, a Linear Falloff Deformer is applied to a plane on the left, shearing it. On the right a 'Noise' procedural is modulating the Falloffs strength.

Camera

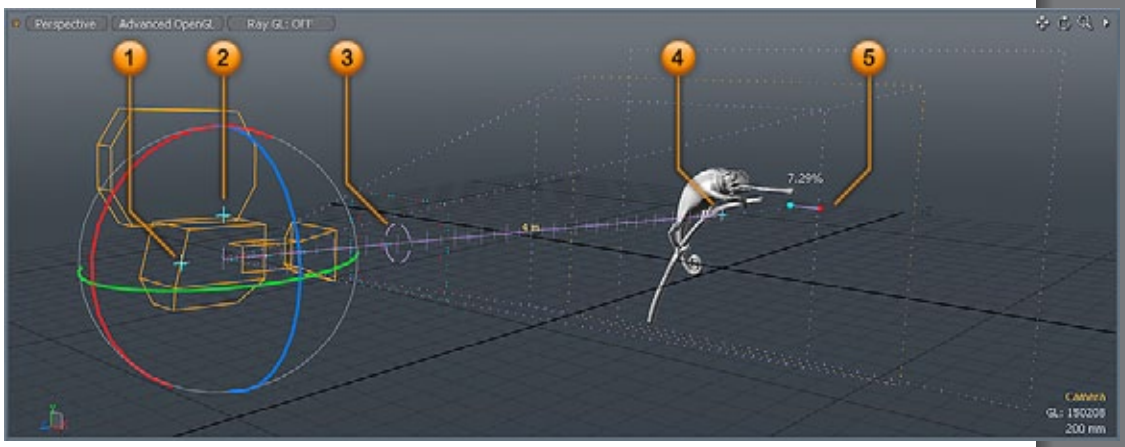
Camera Item

The Camera Item is a virtual lens into the 3D world inside modo. Everything rendered is seen from this point of view, as such, these settings determine how your camera sees your 3D world. Much like *how* a real camera operates, the camera *captures* your image, and works in conjunction with the 'Render Item' making your virtual snapshot viewable in the 'Render Display'. What is displayed depends on your 'Render Output(s)', which acts as virtual *developer* determining what actually gets rendered when 'F9' is pressed (or any render command is invoked). While the Camera Item contains the properties related to the projection, frame aspect ratio, DOF, etc., the actual frame resolution is set within the 'Render Item' under the 'Frame' properties.

Each new modo scene by default has a single camera item, additional cameras may be added to the scene with the menu bar command 'Item > Create Camera' or within the 'Item List' viewports 'Add Item' function; LMB-click the add item button and select 'Camera' from the popup menu. Additional cameras are helpful in providing alternate views of a scene, easing navigation, or for providing close-up views of a particular area, especially in situations where the *hero* camera is animated and locked, avoiding accidental changes. Only a single camera may be 'rendered' from at a time, the particular camera used is specified in the 'Render Item's 'Frame' subtab under the 'Render Camera' option. Additionally, alternate camera may be selected within the 'Render Preview' viewport using the select camera option available, without affecting the 'Render Camera' used for *final rendering*.

Working with the Camera

The cameras attributes, such as Position and Rotation, can be modified in the properties panel whenever the camera item is selected in either the 'Item List' or the 'Shader Tree'. By default, the camera icon is only visible in the 'Animate' and 'Render' 3D viewports, but by changing the visibility in the viewport options, they can be made viewable in any viewport; hover the mouse pointer over the 3D viewport you wish to modify, and press 'o' on the keyboard to open the visibility options panel.



The camera item may also be interactively modified in the 3D viewport while in 'Items' mode (camera items have no editable components). Use the standard transform tools to move (W) and rotate (E) the camera. When making fine adjustments, users may find it easier to see how the camera is viewing the scene. This can be done by changing the viewport view button in the upper left to 'Camera'. When set as such, black bars may be drawn over the viewport if the *aspect ratio* of the 'Frame Size' is different from the viewports aspect ratio; the area inside the bars represents the area of the scene that will render when a render command is invoked. To position the camera while in this view mode, various keyboard/mouse options are available--

Shift+Alt+LMB Drag	=	Move camera item
Shift+Alt+RMB Drag	=	Move Up/Down direction only
Alt+LMB Drag	=	Rotate around Target Distance
Alt+RMB Drag	=	Rotate around Camera center
Ctrl+MMB Drag	=	Rotate around 'Z' axis
Ctrl+Alt+LMB Drag	=	Move Forward/Backward only
Ctrl+Alt+RMB Drag	=	Adjust Focal Length
Roll Mouse Whee	=	Move Forward/Backward only

When tool handles are set to 'Advanced' (within the 'Preferences' or 'Tool Pipe'), additional handles become visible allowing users to interactively adjust camera setting in the viewport (*mouse over the above image for handle reference*). The handles are--

1.) Adjust 'Focal Length'/'Field of View'.

2.) Adjust F-Stop for Depth of Field.

3.) Adjust 'Convergence Distance'.

4.) Adjust 'Focus Distance'.

5.) Parallax Offset Difference value (distance between items for left & right eye in final rendered frame) expressed as percentage of frame width. Users can drag the handle around the scene to get an idea how far apart elements will appear on the screen when rendered as a stereo image. Generally speaking, for comfortable viewing users will want to keep the sum of the values for the closest elements in frame to the farthest elements 5% and under. Adjustments can be made to 'Interoccular Distance' and the 'Convergence Distance' options to help in obtaining this result.

TIP: User will likely find it more intuitive to adjust the camera when the 'Action Center' is set to 'Local', keeping the transform handles aligned to the camera item itself.

On occasion, users may wish to modify the size of the camera icon within the 3D viewport, this can be accomplished with the 'Size' setting in the 'Display' viewport, which offers additional settings for modifying the cameras iconic representation including the ability to label the item, enable the 'Safe Area' display and toggle the display of the 'Stereo Volume'. Users may also adjust the 'Scale' option of the cameras in the 'Properties' panel. Note that changing the size of the camera icon in no way affects the rendering outcome or the overall scale of the scene.

Camera View

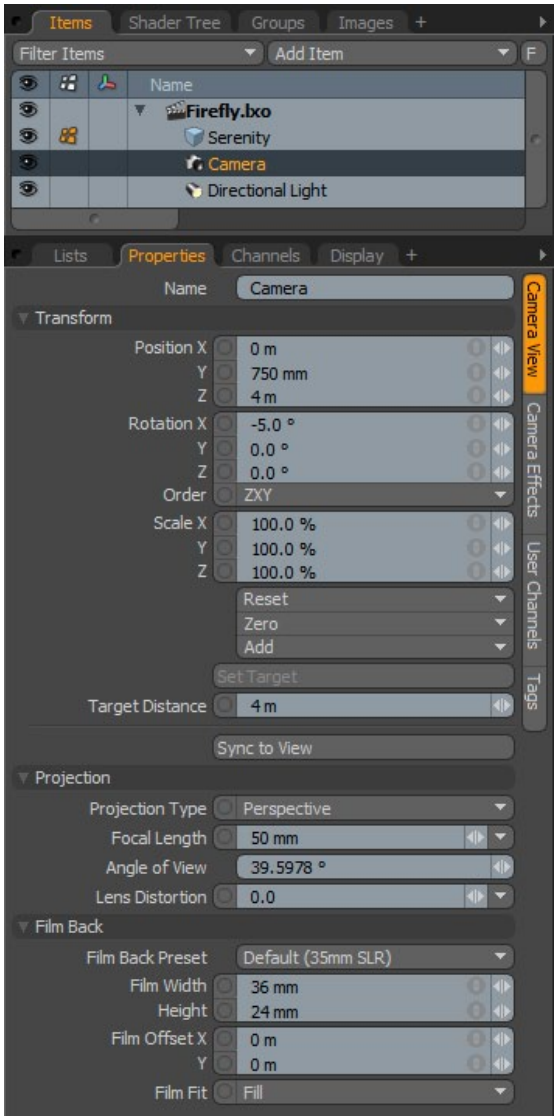
Name: This data field displays the current item name. Users may easily change it by LMB-clicking within the field and typing the new name.

Transform--

Position: An Item transform that allows the user to numerically position the camera item in XYZ space. By default, Position transforms originates from the items center position.

Rotation: An Item transform that allows the user to numerically set the rotation of the camera item. By default, Rotation transforms originates from items center position.

Order: Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.



Scale: An item transform that determines the size of the icon in the 3D viewport. Scale otherwise has no effect on the final rendered image.

Reset: Resets the selected transform values to (0,0,0) returning the items back to their default state.

Zero: Returns the camera items Center position to the world space center (0,0,0) without changing the position of the camera item itself.

Add: The 'Add' function will add the selected set of transforms to the channel list. Useful when animating complex motions, providing a layered approach to the transforms.

Set Target: The 'Set Target' option allows users the ability to target specific items in a scene, automating the rotation of an item, so that it continuously points toward the targeted item. To use, make sure you are in 'Items' mode, and select the Camera item (and only the camera item), then, holding down the 'Ctrl' key, select a single additional item in the item list that you wish to target, (such as a locator) in that order; as the selection order is important, and then pressing 'Set Target' option in the Camera View options to enable the function. Remember though, that the camera will look directly at the item's center position regardless of the location of geometry within the target layer. Once activated, additional options appear--

Remove Target: Removes the target link between the two items.

Enable: Toggling this option off temporarily disables the targeting function while retaining the link between the items.

Set Focus Distance: This option sets the focal distance for DOF when targeting camera items to other item.

Roll: Provides the ability to offset the items rotation angle away from the target.

Target Distance: This value defines the point around which the camera will orbit. Note that when you start a fresh project the camera is back 4M on Z and the Target distance is 4M. That is why it orbits around 0. This value is reset automatically when the user performs a zoom. modo fires a ray out from the camera and automatically sets the target distance based on the first surface the ray intersects.

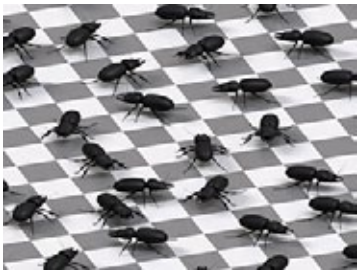
Sync to View: This command will change the position and orientation of the currently selected Camera item to match the currently selected 3D viewport as closely as possible. There are some caveats in that the Cameras view scale is dependent upon the focal length and image resolution (for aspect) and therefore cannon be matched precisely.

Projection--

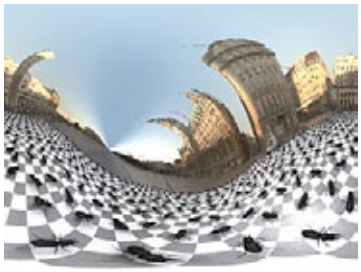
Projection Type: modo can render traditional perspective cameras as well as orthographic cameras and spherical cameras. This popup allows you to choose the desired camera type. Perspective cameras will show perspective distortion where the scale of an element appears to diminish in the distance. This traditional perspective view is what most artists would expect when rendering; similar to what a real world camera might capture. Orthographic cameras do not exhibit any perspective distortion as all rays are fired from the camera in parallel lines. Spherical cameras capture the entire 360 degree view of a scene.

Perspective cameras are found most often in visual FX, product and architectural visualization and/or anywhere artists are trying to mimic reality. Orthographic rendering is very useful in news/info graphics and architectural plan rendering. Spherical cameras are useful in capturing images that can be later applied as image based lighting and reflection maps, among other things.

Perspective Camera



Orthographic Camera



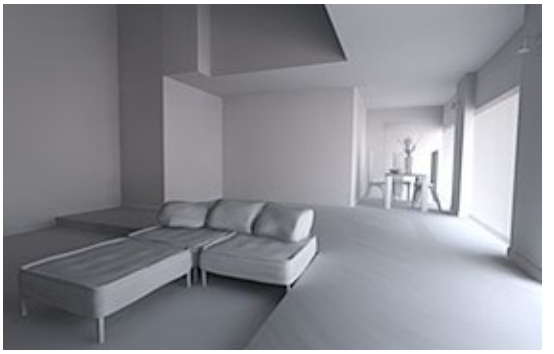
Spherical Camera

Focal Length: The Focal Length setting controls the amount of zoom just as it would with a real world camera. The actual definition of Focal Length is the distance between the lens and the point where the rays converge after having been refracted by the lens. Wide angle lenses (those with short focal lengths) capture more of a given scene than their telephoto counterpart. Lower focal lengths will exhibit more perspective distortion, conversely, longer focal lengths will minimize the effect of perspective. To zoom out you should decrease the focal length and to zoom in you will increase the value. This control is set in real world metric units to match physical cameras.

Focal Length can be adjusted on the Camera property form or interactively in the 3D viewport. If you are looking through the Camera (and 3D Viewport set to 'Camera') you can use Ctrl+Alt+RMB to adjust the Focal Length interactively (Ctrl+Alt+LMB moves the camera itself forwards and backwards).

Angle of View: In some instances, it may be easier to specify an angle of view instead of a Focal Length. This setting allows one to specify the 'Angle of View' as a number of degrees horizontally, once set, the Focal Length will be updated to reflect the new value.

Lens Distortion: In a real world camera, light rays are bent through a spherical lens element resulting in a slightly distorted image. 3D rendered images don't pass through a lens and likewise don't exhibit the distortions we're accustomed to in photography. The Lens Distortion setting can be used to introduce these familiar distortions in a realistic way. Positive values create an increasing 'Barrel' distortion, and negative value create increasing 'Pincushion' distortion effects.



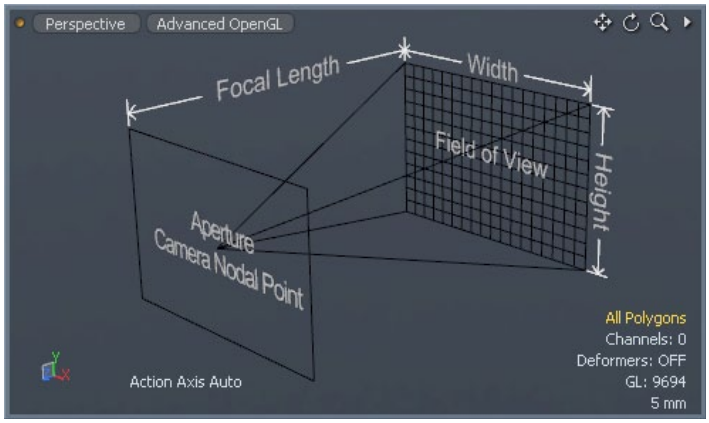
Standard CG Wide Angle View

Same view w/ Lens Distortion applied

Film Back--

In the real world, every camera has certain attributes that are a fixed physical size, in the analog days, this was based on the film size (i.e. 35mm, 70mm etc.), with digital cameras, it is the sensor size (i.e. 1/3, 1/2, APS etc.). The size of this fixed capture area works in conjunction with the focal length to control the field of view that is captured. modo offers the ability to modify the virtual capture area to any conceivable size. Should one wish to compose 3D rendered elements together with images from a camera, for the the most convincing results, matching the real-life sizes and aspect ratios of the original camera is highly recommended. Many popular preset sizes are provided for convenience, but users may also define custom values with the Width and Height settings. The Film Fit setting allow users to control modo's behavior when the frame size specified in the render item, is a different aspect ratio than the film back settings.

Film Back Preset: The 'Film Back Preset' allows you to choose from a number of available preset real world cameras. Choosing a preset will automatically set the Film Width and Height, and the corresponding Frame Size (in the Render Items Frame Tab). You can also save your own preset by selecting "(new preset)" from the bottom of the list. When selecting this option, a dialog will popup asking you to name your preset. You're new preset is saved in modo's config file and is available immediately for selection.



Film Width/Height: 'Film Width' and 'Film Height' correspond to the exposed region of the film negative in a real camera set in real world measurements. In some applications this is known as the Film Gate. One way to visualize this is to imagine the camera frustum as a pyramid whose base dimensions are the Film Width and Height and terminating at the apex. The distance from the base to the apex is the Focal Length (The apex being the position of modo's virtual aperture and the position where the camera rotates which incidentally is also the nodal point). The base of the pyramid can be shifted around within its plane (shearing the pyramid) by using the Film Offset controls.

Film Offset X/Y: The 'Film Offset' function allows users the ability to offset the position of the Film Back without changing the position of the camera item itself providing unique perspective views not otherwise possible. The function is especially useful when rendering architectural type images that prefer two point perspective as opposed to three. By rotating the camera on the X axis to 0° (eliminating the vertical perspective), the offset may be adjusted to modify the cameras view (framing the subject properly) while retaining the perspective of the cameras original position.

The three images below illustrate the differences between film offset and simply moving the camera. A.) the camera standard perspective view with the subject centered in the frame. B.) rotates the camera X-axis to 0° and uses a 'Film Offset' to frame the subject eliminating the vertical perspective. C.) is the default perspective camera again with no offset, but the camera was rotated to 0° on the x-axis again and the cameras position was modified upward to frame the subject producing a dramatically different perspective than that offered by 'B'.

A.) Normal Perspective Camera

B.) 0° X-axis rotation w/ Film Offset

C.) 0° X-axis rotation w/ camera offset

Film Fit: 'Film Fit' only matters when the camera item's film gate (defined by the Film Width and Height above) has a different aspect ratio than the Render Item's resolution gate (defined by the actual Frames Width, Height and Pixel Aspect Ratio). In that case, the two possibilities are that the horizontal borders coincide, or the vertical borders coincide. The Fill and Overscan options automatically select whether to match the horizontal or vertical borders in order to make the resolution gate fit within the film gate or vice versa.

You can think of the film gate (Film Width and Height) as the size of the film in the camera in real world units, with the film being analog (not divided into pixels). Then you take the exposed film frame and lay it on a digital scanner (the resolution gate), which is divided into pixels, and which might have a different shape (wider or taller). It's the resolution gate that gets saved to the output image file.

Camera Effects

Distant Chair in Focus

Near Chair in Focus

Depth of Field--

Depth of Field (DOF), as it relates to photography, is the area of acceptable focus in an image; any area outside gets increasingly blurry as it recedes from the focal plane. The softening of edges as they recede from the plane is highly dependant on the 'F-Stop' setting. Smaller values represent larger apertures producing softer blurs, conversely, higher values represent smaller apertures producing less blurriness. Users must first enable DOF in the Render Item properties under the settings subtab to see its effect.

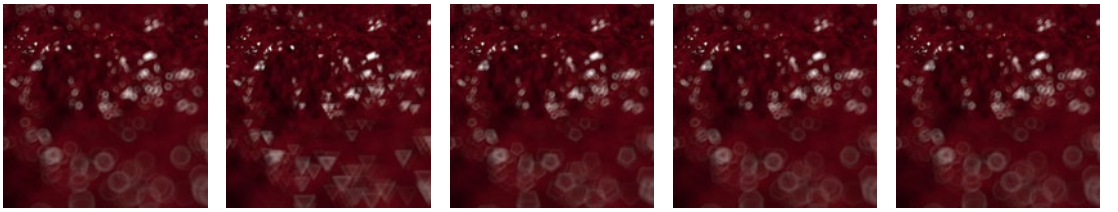
Focus Distance: When rendering with 'Depth of Field', this value sets the distance out from the camera center at which the image will be perfectly in focus. To quickly set the focus distance for the camera you can use the 'Autofocus' command or by pressing Ctrl-A with the mouse over the Camera viewport. You can also set this value interactively with the Item Transform tool ("Y") when the Camera item is selected.

Autofocus: This command sets the Focal Distance to whatever geometry is at the direct center of the current camera view. Setting the Focal Distance is only useful when rendering with Depth of Field active. You can also use the keyboard shortcut 'Ctrl-A' when the mouse is over a 3D Viewport set as 'Camera' or any Render Preview window.

F-Stop: The F-Stop control on the modo camera is only effective when Depth of Field is enabled. In this case lowering the F-Stop will decrease the depth of field (apparent area of focus) just as it would in real-world cameras. Note that F-Stop does not effect exposure of light since the modo camera captures real world radiance values as they are striking the virtual CCD. This provides linear light values to the final image which provides an opportunity for post processing the exposure for maximum flexibility.

Iris Blades: The 'Iris Blades' setting controls the way the blurriness looks in images

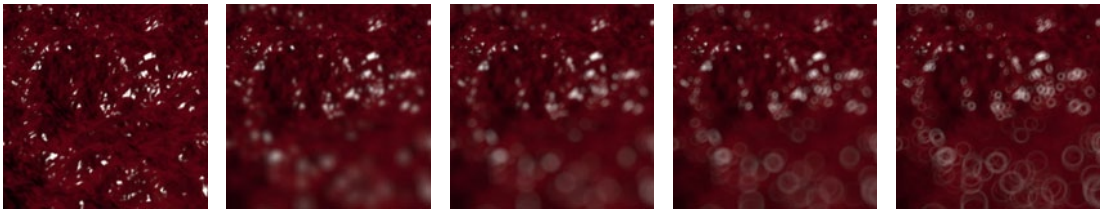
with shallow depth of field, generally referred to as "Bokeh". Bright areas, when out of focus tend to take on the shape of the cameras Iris, smaller numbers of blades make more obvious geometric patterns to the bokeh shapes, while larger values look more circular. A setting of '0' makes a perfectly circular shape.



Iris Blades 0 (default) | Iris Blades 3 | Iris Blades 5 | Iris Blades 7 | Iris Blades 9

Iris Rotation: This option changes the rotation angle of the virtual Iris when rendering shallow depth of field in images, which in turn rotates the bokeh shapes in the final rendered output.

Edge Weighting: When modo calculates Depth of Field, rays are fired in a pattern that simulates the iris blades producing the bokeh effect shapes. The edge weighting option concentrates those rays toward the center or edges of the iris, depending on the value. A value of '0%' concentrates all the values at the center of the iris, while a setting of '100%' concentrates the rays to ward the outer edge. A value of '50%' evenly distributes the rays. A light tends to be stronger in the outer edge of the iris in the real world, setting around 60% to 70% tend to produce the most realistic results.



Edge Weight 0% | Edge Weight 25% | Edge Weight 50% | Edge Weight 75% | Edge Weight 100%

Motion Blur--

Real cameras don't freeze time, but capture brief moments, causing moving objects to blur in the frame. This natural motion blur that our eyes are accustomed to can be simulated in modo. Users must first enable 'Motion Blur' in the Render Item properties under the settings subtab to see its effect. Once set, modo will now take 'time' samples forward and backward from the rendered frame blending them together to create

the blur effect. Depending on the strength of the effect or the speed of motion in your frame, you may find it necessary to increase the number of anti-aliasing samples. Adjustments to Blur Length (like adjusting the exposure time for more or less blur) and Blur Offset can be found in the Camera Items properties.

Animated Ball, Motion Blur Disabled

Animated Ball, Motion Blur Enabled

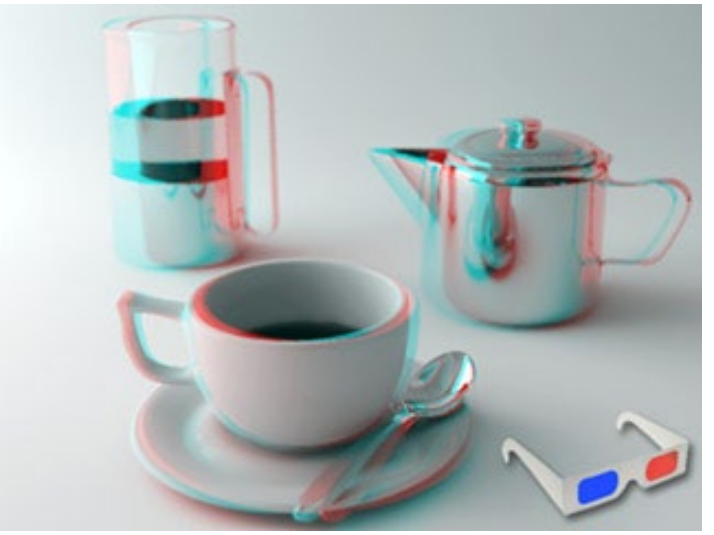
Blur Length: The Blur Length control will affect the length of the motion blur effect when rendering animations. The default value of 50% would be similar to a real-world camera set with a 180 degree shutter. In modo the 50% value will result in the shutter being open for half of the exposure time of that frame. The motion blur effect is calculated by taking time samples both forwards and backwards from the current frame time and blending them together for the final blurred result. As such, it may be necessary to increase the level of anti-aliasing to produce a smooth result.

Blur Offset: The blur offset value lets you control the center position in time where modo samples its motion blur from. A setting of zero centers the motion blur on the current rendered frame, while a setting of 100% moves the position backwards one entire frame and a setting of -100% moves the position forward one entire frame.

Stereoscopic--

We see with two eyes, each slightly offset form the other, providing our brains with enough information to create a sense of depth to the world around us. modo scenes are likewise 3D with depth, but the monitors we view scenes with are two-dimensional, as are the renders modo creates. Activating the stereoscopic function in modo will bring back that missing depth information by rendering 2 offset images that when viewed in tandem reveals the depth present in your original scene. The stereoscopic function, when enabled, will render two separate frames, one appended 'L' for left and 'R' for right noting which eye the frame was rendered for. Note that when activated, the Stereoscopic function is rendering 2 completely separate frames, so increased render

times are to be expected. If applicable, only a single Irradiance Cache is calculated for stereoscopic renders.



Stereoscopic Enable Toggle: This setting is the global toggle to enable/disable the 'Stereoscopic' function when rendering. When enabled, modo will render two frames slightly offset from each other (like our eyes). The resulting images can them be combined a variety of way that when properly viewed reveal the depth present in your original scene.

Stereo Eyes: When 'Stereoscopic' rendering in *enabled*, users can use this option to choose which eye is rendered, choosing to render either only the 'Left' eye, only the 'Right' eye or rendering 'Both' simultaneously. When rendering Both, users can additionally choose a 'Stereo Composite' option to determining how modo will combine the resulting stereo images for final output.

Stereo Composite: modo offers a variety of options for composing stereo images into a single image for viewing. Which option you choose is largely based on how the resulting image will be viewed.

Anaglyph Red-Cyan- This method replaces the red channel or the right image with the red channel from the left image.

Anaglyph Grayscale- This method makes both frames into grayscale (luminance) images, then puts the left one in the Red output channel and the right one in the Green and Blue channels.

Anaglyph Red-Cyan Half Color- This method converts the left image to grayscale

and puts it into the Red channel, and uses the right frame's Green and Blue channels for the output Green and Blue channels. This technique reduces 'retinal rivalry' which is a disturbing phenomenon caused by differing brightness of what should be the same objects seen by either eye.

Anaglyph Red-Cyan Optimized- This method further combats retinal rivalry by discarding the red channel from both images, keeping the right frame's cyan component, and mixing the left frame's Green and Blue channels at 70%/30% ratio to use in the output red channel.

Anaglyph Red-Blue Least Squares- This algorithm was invented by Eric Dubois. He used the optical properties of standard red and blue filters used for anaglyph glasses and derived magic numbers for blending stereo pairs into color anaglyphs by minimizing the error using a "least squares" method.

Side-by-Side- Half-width frame is rendered and composed side by side into a full width regular frame, with Left on Left and Right on Right.

Side-by-Side Full Width- Full sized frames are rendered separately and composed side by side in a double width frame, with Left on Left and Right on Right.

Side-by-Side Cross Eyed- Half-width frame is rendered and composed side by side into a full width regular frame, with Left on Right and Right on Left.

Side-by-Side Cross Eyed Full Width- Full sized frames are rendered separately and composed side by side in a double width frame, with Left on Right and Right on Left.

Interocular Distance: The 'Interocular Distance' technically means the distance between the center point of a persons two pupils, but here it is the measured offset between the two rendered images in modo. The default setting of 65mm is the average distance between the eyes of an adult. For scenes modeled to real world values, this is an appropriate setting that produces good results. Scenes of varying scale will likely require this distance to change as will renders of exceptionally close or exceptionally far objects.

Convergence Distance: The convergence distance is the point within 3D space where the 2 offset images converge. In front of this point, objects appear to come forward, and behind this point, object appear to recede. It is easily understood by imagining it as the focal point of your scene, much like with DOF where the focal distance is the position in space where all objects are in focus, objects in front or behind this position gradually blur, the Convergence Distance is the point in the scene you wish to direct your viewers eyes, as it will be the most comfortable to view. This is exactly how your eyes work as well. If you were sitting in a restaurant and your eyes were focused on your glass of water on the table, this would effectively be the convergence point of your two eyes,

your dinner plate would be in front of the glass coming forward, and your guest would be further away on the other side of the table.

Clipping Plane--

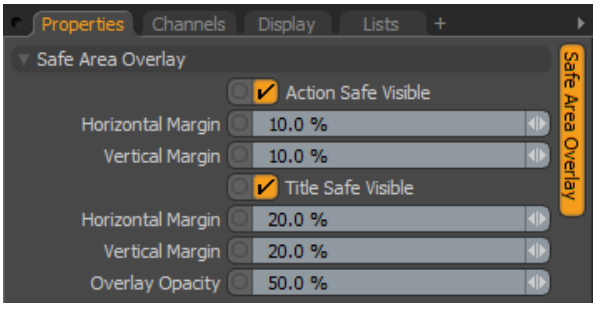
Clipping refers to the removal of part of a scene from a rendered image while leaving the actual scene geometry intact. Users can define a distance where all (or selected) surfaces will be clipped, by enabling the 'Clipping Plane' option and then defining a 'Clipping Distance'. Users can disable clipping on a per-material basis by disabling the 'Enable Surface Clipping' option in the Material item. Users should note that a similar effect can be obtained through the use of 'Render Booleans' that is not camera view dependant.

Enable: When *enabled*, any scene items within the bounds of the camera position and the clipping plane will be clipped from the scene, removing them from the rendered image while retaining the actual scene geometry.

Clipping Distance: Defines the distance away from the camera where items will be clipped.

Camera Safe Overlay

Safe Area Overlay--



Overscan was a display technique used in older tube-type televisions sets, because of this *overscanning* of the screen, the final size of the displayed image was unreliable, so a 'Safe' area was determined that would allow content creators to design within a reliable boundary to ensure visibility of elements on the screen. Users can enable the 'Safe Area Overlay' option in the 'Display' viewport displaying guide elements on the GL viewport. Once the option has been enabled, additional camera properties become visible when the 'Camera' item is selected providing users a means to adjust the default values.

Action Safe Visible: The action safe area is the space determined to be where important action of a scene would be clearly visible, but the bounds could possibly be against the edges of the screen. The 'Action Safe Visible' option toggles the visibility of the outer guide of the Safe Area overlay.

Horizontal/Vertical Margin: The Horizontal and Vertical margins determine the distance away from the edge where the guide is actually drawn in the viewport.

Title Safe Visible: The title safe area is the space where all on screen text and graphics would be guaranteed to be visible, ensuring clear legibility. The 'Title Safe Visible' option toggles the visibility of the inner guide of the Safe Area overlay.

Horizontal/Vertical Margin: The Horizontal and Vertical margins determine the distance away from the edge where the guide is actually drawn in the viewport.

TIP: The Camera Safe Overlay guidelines will only appear when a viewport is set to the 'Camera' view type. Also the Camera must be set to 'Visible' within the viewport (check the 'Display > Options' to ensure 'Show Camera' is *enabled*.

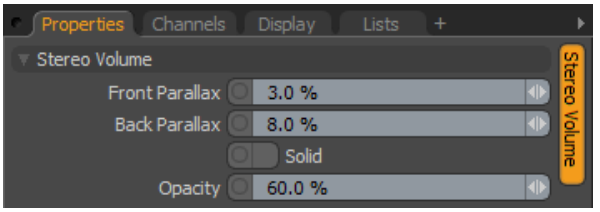
Overlay Opacity: The 'Overlay Opacity' option determines the strength of the guides when drawn in the viewport, a higher opacity value would mean the guide lines were more apparent, and a lower opacity would be more subtle.

Stereo Camera Volume

The separation of *matching* pixels in left and right stereo images is referred to as *parallax*. The Camera's convergence plane, which corresponds to the TV/Movie screen as far as the audience is concerned, there is no parallax. Objects at that distance appear to be in the same plane as the screen. Objects further than the convergence plane appear to be pushed back into the screen. The farther back they are, the greater the separation (parallax) between those pixels. Things in front of the convergence plane appear to be in front of the screen, and also have increasing parallax, though the images are shifted in the opposite direction.

If a scene exceeds a certain parallax amount, this will cause discomfort for the viewer, or ruin the effect entirely. Conversely, if the scene doesn't have enough *volume*, then the stereo effect is barely noticeable. Thus the amount of parallax into and out of the screen are critical to monitor when setting up a shot. Directors are even starting to plot *depth scores* so they can keep the parallax in a safe range, using it as a storytelling element as well as to catch sharp jumps which can be disturbing to viewers.

The Stereo Volume 'package' on the camera lets users define your stereo volume by setting the front and back parallax amounts. You can then either use it to make sure your objects stay in range, or you might use it to measure shots by setting keyframes so it always matched the nearest and farthest elements in your scene. Then you get your 'Depth Score' just by screen capturing the 'Graph Editor'.



Stereo Volume--

This refers to the amount of depth of your scene between the frontmost element and backmost. It is an important quantity for designing a stereo image. It is referred to as a *volume* because it includes the entire space- width, height and depth of the frame. It is also related to the *stereo budget* which is yet another way of saying the amount of depth used by the scene. Users can activate the display of this important visual guide in the 3D viewport by *enabling* the 'Show Stereo Volume' option in the 'Display' viewport. Once enabled, the following options appear in the 'Properties' panel when the 'Camera' Item is selected.

Front/Back Parallax: This parameter sets the desired front/back parallax amount(s). It is given as the separation distance's percentage of the screen width. When you set this, modo computes the distance from the convergence plane where objects will have that parallax amount, and draws the front or rear plane there. There are also two circles drawn with centers that are separated by the parallax amount.

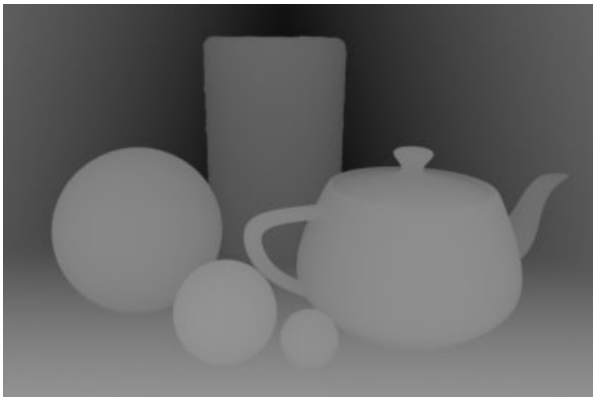
Solid: When enabled, draws volume as filled quads, instead of outlines making its apparent area more obvious.

Opacity: Determines the opacity of the drawn volume guides.

Render Output Items

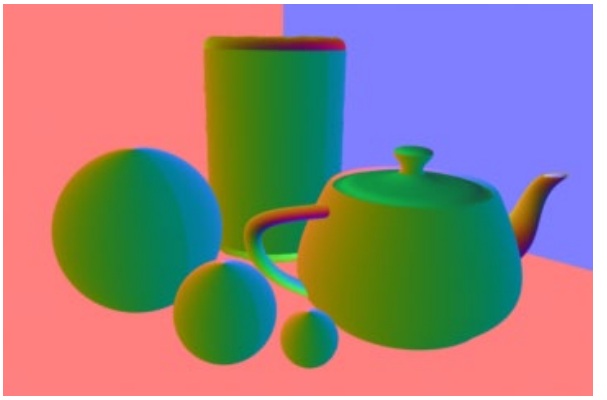
modo offers many different 'Render Output' types useful in a variety of situations ranging from purely diagnostic, to specific compositing utility in an external application. A default scene automatically includes a 'Final Color' and 'Alpha' output which should be fine for most situations. Additional layers can be added by using the menu bar command "Texture > Add Shader Tree Layer > Render Output", or from within the shader tree viewport itself, by LMB-clicking the 'Add Layer' option of the full viewport window and selecting 'Render Output' from the pop-up context menu. This adds an additional layer to the top of the tree. Users may then RMB-click in the 'Effect' column of the shader tree for a pop-up context menu that allows one to select alternate output types. All enabled outputs are rendered simultaneously each time the 'Render' or 'Render Animation' commands are invoked. Once rendered, the individual outputs appear in the Render Display viewport accessible from the layer menu. Please reference the primary Render Outputs page of the documentation listed under the Shader Tree Layers section for for information regarding the editable attributes of the individual outputs. The various output types available are as follows--

Depth



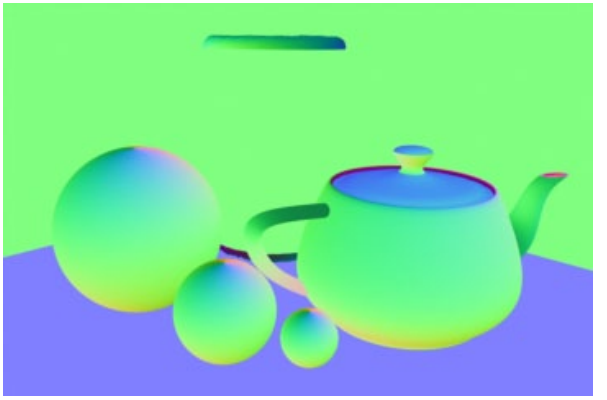
The 'Depth' output generates a Z-Buffer that represents the distance from the camera to the geometric surface in shades of gray. Rays begin white at the ray origin (camera position) and attenuate darker as they get longer becoming fully black at the specified 'Maximum Depth' value. Useful for calculating Depth of Field in an external application or for composing a fog effect into a rendered scene.

dPdu Vector



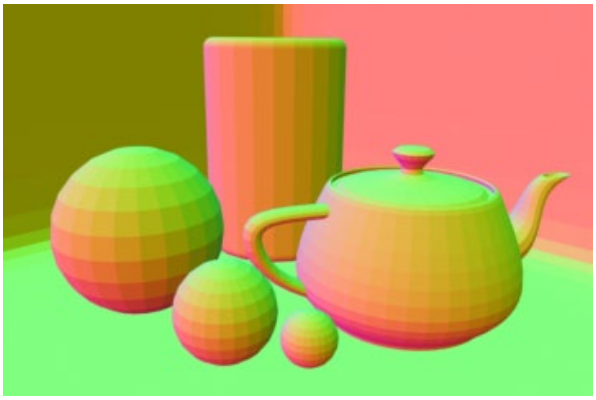
The 'dPdu' and 'dPdv' are world space vectors that point in the direction of increasing U and increasing V at each point on a surface that has a UV map. dPdu and dPdv have a big effect on the anti-aliasing of UV image maps and also define the reference directions for anisotropic highlights and reflections, so the outputs are sometimes useful for the purposes of debugging.

Dpdv Vector



The 'dPdu' and 'dPdv' are world space vectors that point in the direction of increasing U and increasing V at each point on a surface that has a UV map. dPdu and dPdv have a big effect on the antialiasing of UV image maps and also define the reference directions for anisotropic highlights and reflections, so the outputs are sometimes useful for the purposes of debugging.

Geometric Normal

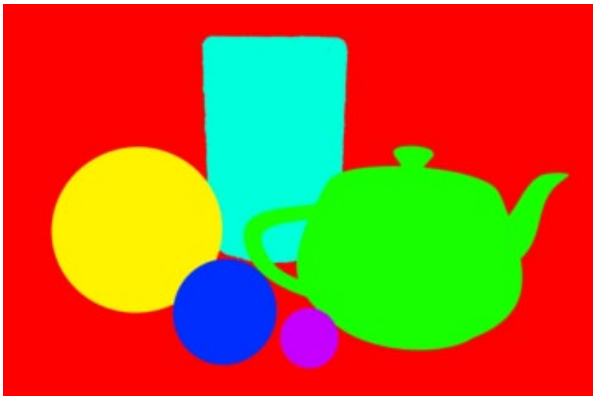


The 'Geometric Normal' output simply renders all meshes with a color ramp based on the angle of their polygonal normal. It is useful for evaluating the continuity of a mesh's surface. In this example smoothing has been disabled.

Segment ID

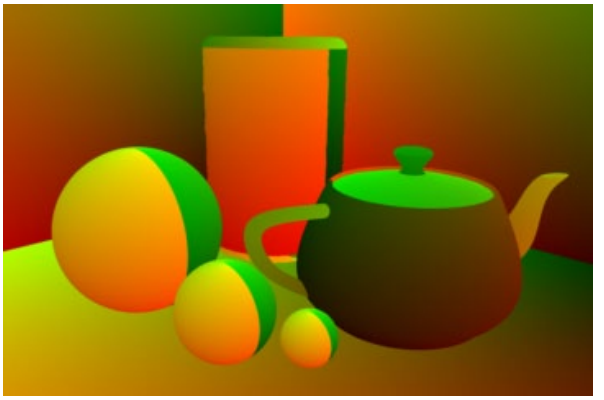
When modo renders a project, all meshes are broken down into "segments" for reasons of optimization and general scene management. Using the Segment ID shading option will render each generated segment with a random color. This shading method is purely a diagnostic option.

Surface ID



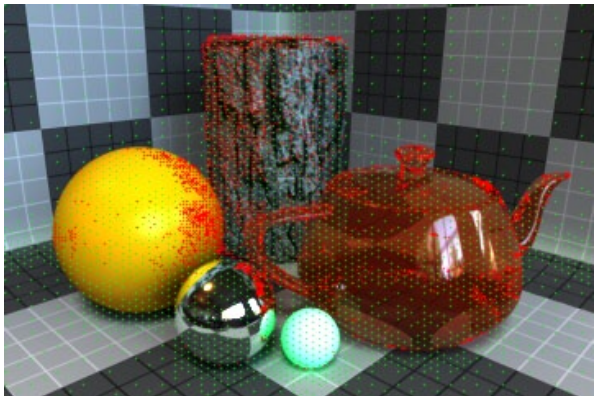
The Surface ID output render each material with a random color. This is purely a diagnostic shading option, as it may be helpful in spotting surface relationships in complex scenes.

UV Coordinates



The 'UV Coordinates' output generates a channel that converts UV positions to a color value. Some software is then capable of utilizing this image to reapply a texture map to the object after it has been rendered.

IC Positions

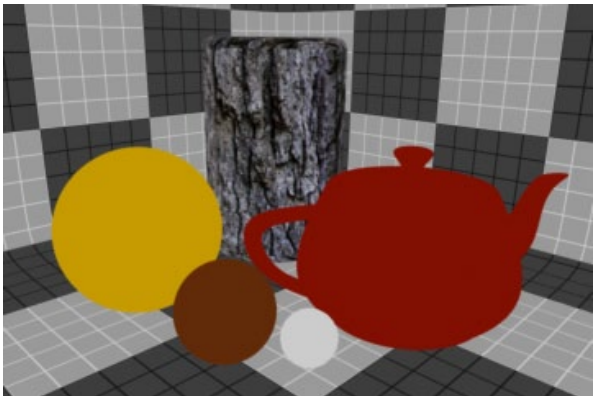


The settings within the 'Render Item > Properties > Global Illumination' tab allow you to specify values that the Irradiance Cache uses to shade a rendered scene. The 'IC Positions' output will display the pixel positions used in generating the shading based on the Irradiance Cache values. Green points are positions sampled during the initial pass, Red points signify additional sample generated by the Interpolation Values setting. Useful in analyzing various IC settings.

IC Values

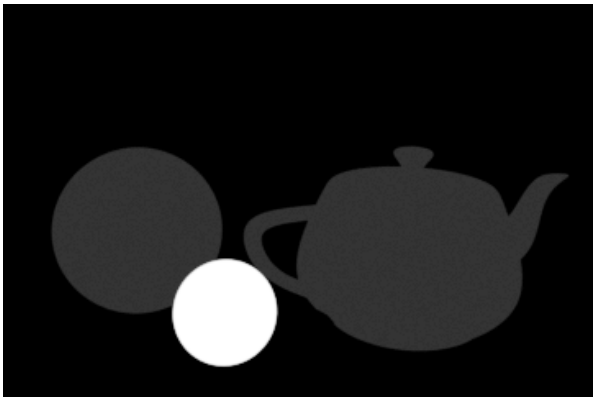
The settings within the Render Item > Properties > Global Illumination Tab allow you to specify values that the Irradiance Cache uses to shade a rendered scene. The 'IC Values' output will display the diffuse values calculated during the initial irradiance cache render pass. The image will appear splotchy as each irradiance sample is not blended with nearby samples; giving one a much better understanding of the area affected by individual IC positions. Useful in analyzing various IC settings.

Diffuse Coefficient



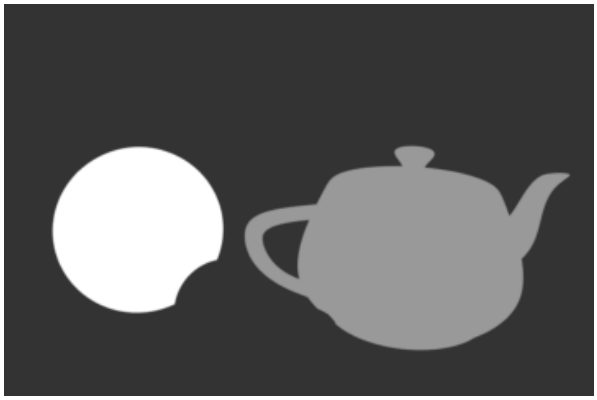
The 'Diffuse Coefficient' output renders the diffuse colors of a scene independent of illumination or shading.

Reflection Coefficient



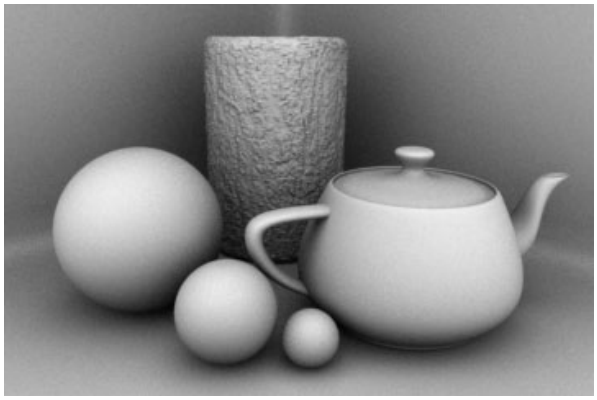
The 'Reflection Coefficient' output renders a scene as grayscale image that represents the reflection amount per surface. Black for 0% reflectivity ramping toward white at 100%.

Specular Coefficient



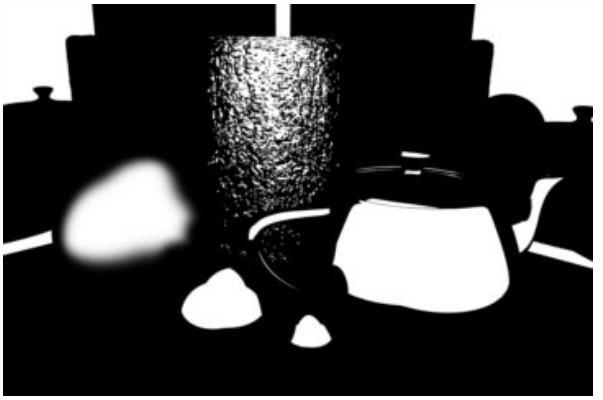
The 'Specular Coefficient' output renders a grayscale image that represents the amount of specular on surfaces within a scene. Black for 0% specularity ramping toward white at 100%.

Ambient Occlusion



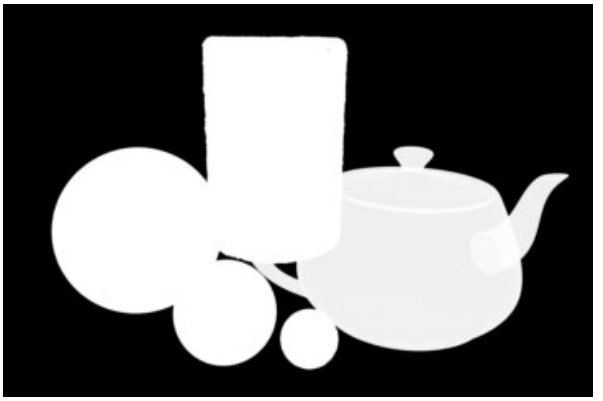
The 'Ambient Occlusion' output generates an occlusion map where occluded areas; crevices and other inaccessible areas of a scene shade darker, while open areas render lighter colors, approximating an overall global illumination type of shading. Grainy renders can be smoothed out by increasing the 'Occlusion Rays' value, controlling the density of the occluded areas is done through the 'Occlusion Range'. Reference the 'Ambient Occlusion' section on the alternate Render Output page for more information.

Reflection Occlusion



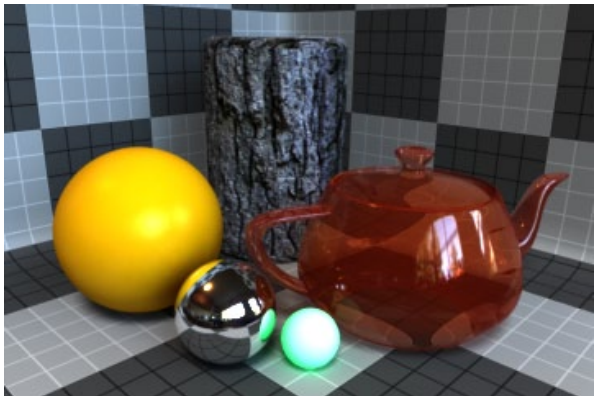
The 'Reflection Occlusion' output renders a channel that is strictly black and white, determining if a reflection ray off of an object has hit any other geometry in the scene. White if it has and black if it has not. The areas that appear as white here would be shaded with the background environment in a Final Color render.

Alpha



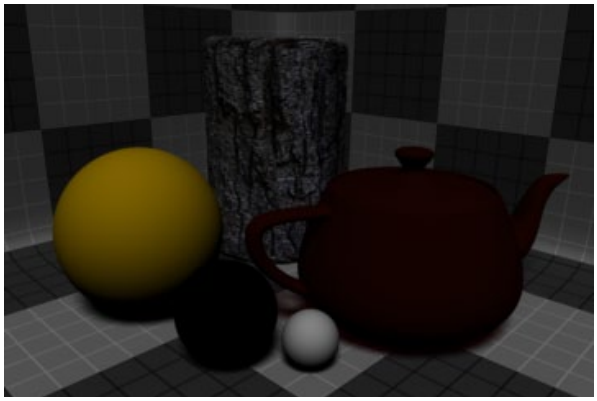
The 'Alpha' output will calculate the opacity of the objects based on their material settings to determine an alpha (transparency) value. With this setting, environment colors and textures will not contribute to the alpha channel, so that when you view the final image with an application that supports alpha channels, only the rendered geometry will be visible. If an object or surface has a level of transparency this will also be considered in the alpha channel. In this way it is possible to render out images with semi-transparent elements that can easily be composited over other backdrops in an application such as Shake, Motion or Photoshop. For best results when compositing multiple images, you should enable 'Unpremultiply Colors' in the corresponding 'Final Color' output.

Final Color



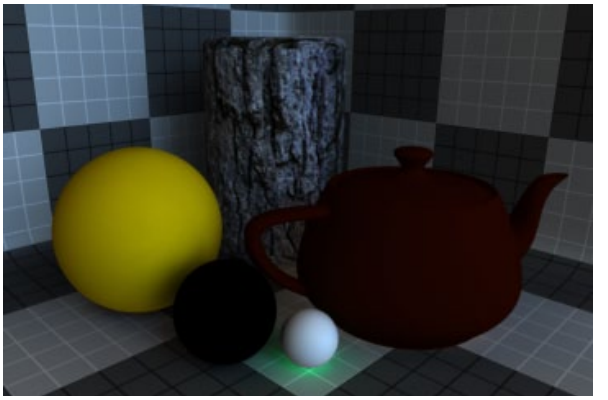
The 'Final Color' output represents the settings of the shader tree, leveraging all visible material and texture layers as well as shaders and the render item settings.

Diffuse Shading (Direct)



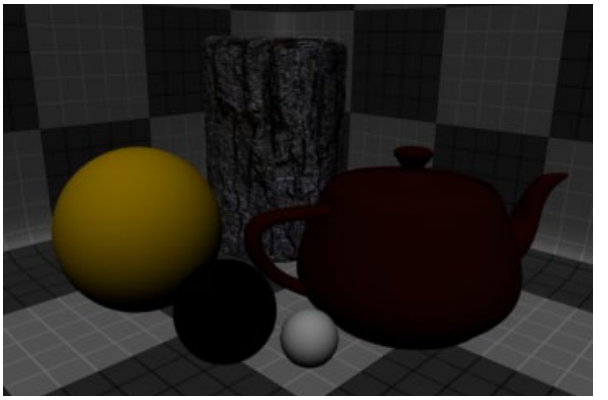
The 'Diffuse Shading (Direct)' output renders a channel of the diffuse color and shading of all objects in scene by all direct light sources independent of other surfacing attribute. Basically a combination of diffuse coefficient and direct illumination.

Diffuse Shading (Indirect)



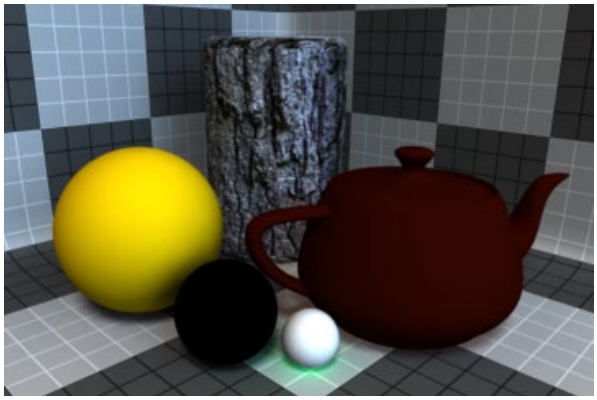
The 'Diffuse Shading (Indirect)' output renders a channel of the diffuse shading of all objects in scene by all indirect light sources independent of other surfacing attribute. Basically a combination of diffuse coefficient and indirect illumination.

Diffuse Shading (Unshadowed)



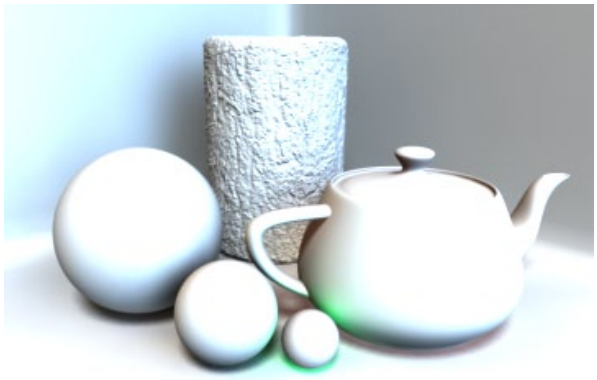
The 'Diffuse Shading (Unshadowed)' output renders a channel of the diffuse color and shading of all objects in a scene by all light sources, independent of other surfacing attribute while excluding shadows cast by direct and indirect lights sources.

Diffuse Shading (Total)



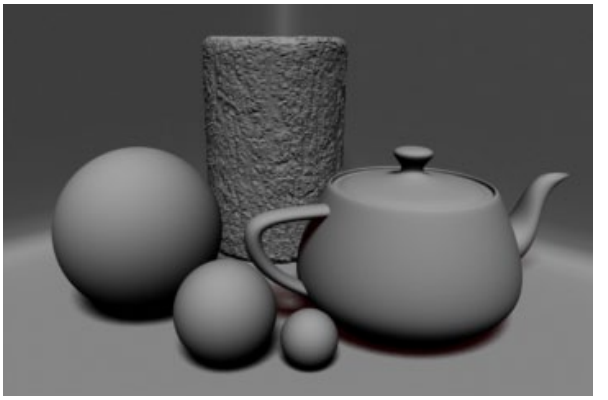
The 'Diffuse Shading (Total)' output renders a channel of the diffuse color and shading of all objects in a scene by all light sources, independent of other surfacing attribute.

Illumination (Total)



The 'Illumination (Total)' output generates a channel of the full illumination in a scene including shadows cast by direct sources, independent of all other surfacing attributes.

Illumination (Direct)

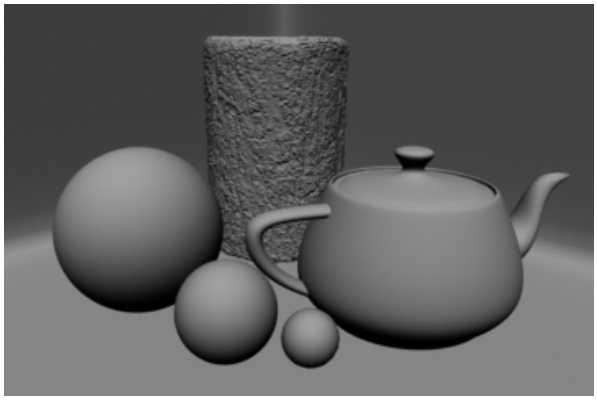


The 'Illumination (Direct)' output generates a channel exclusive to the illumination in a scene from all direct light items such as distant lights, area lights, point lights and spot lights, independent of all other surfacing attributes.

Illumination (Indirect)

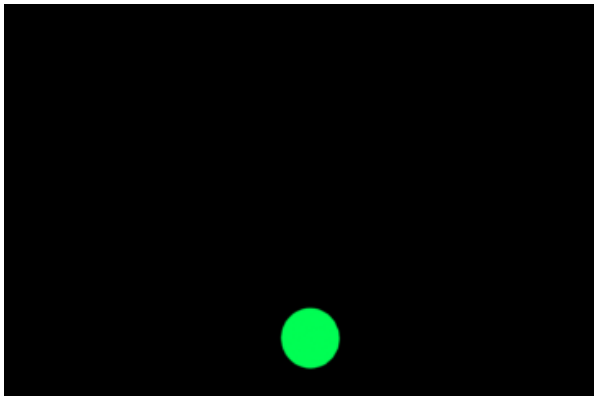
The 'Illumination (Indirect)' output generates a channel exclusive to the illumination in a scene from all indirect sources such as image based lighting and luminous polygons, independent of all other surfacing attributes.

Illumination (Unshadowed)



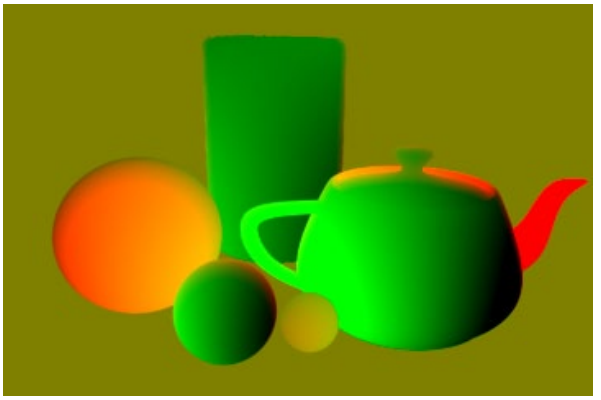
The 'Illumination (Unshadowed)' output generates a channel of the full illumination in a scene excluding shadows cast by direct sources, independent of all other surfacing attributes.

Luminous Shading



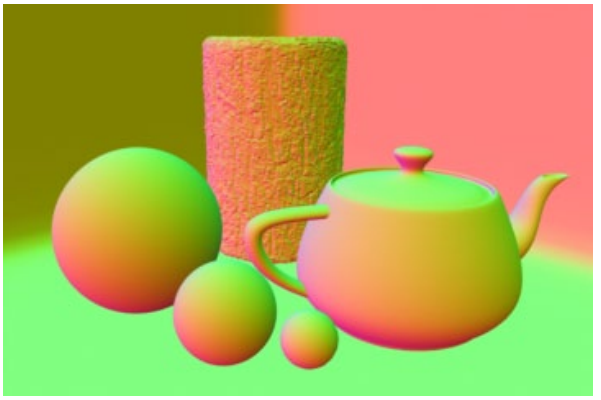
The 'Luminous Shading' output generates a channel of all the luminous shading within a scene independent of all other attributes.

Motion Vector



The 'Motion Vector' output encodes the direction and speed of each pixel into an RGB value, which can then be further processed in an external application to apply motion blur to a rendered image as a post production effect. This can greatly enhance the smoothness of motion blur and save rendering time when processed this way, however, since each pixel can only have a single color value (and therefore a single motion vector), incorrect result will be produced when motion blurring transparent or translucent objects moving in a scene.

Shading Normal

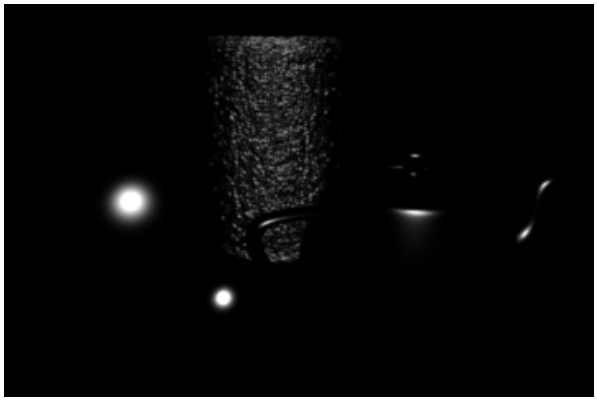


The 'Shading Normal' output renders all meshes with a color ramp based on the angle of their surfaces shaded normal. Where the geometric normal is calculated exclusively by the polygonal geometry, the shading normal also takes in to account surfacing attributes affect on normals such as bump maps.

Reflection Shading

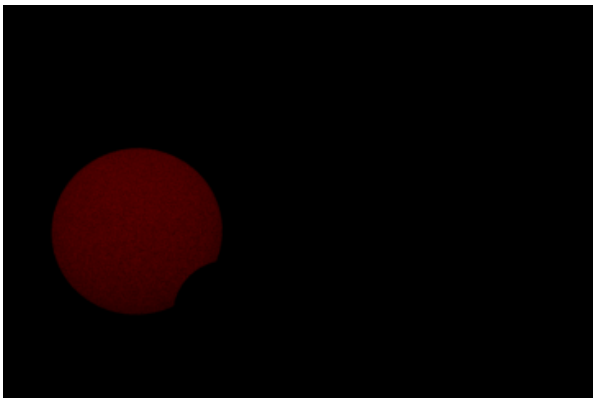
The 'Reflection Shading' output generates a channel of all the reflection calculated within a scene independent of all other attributes.

Specular Shading



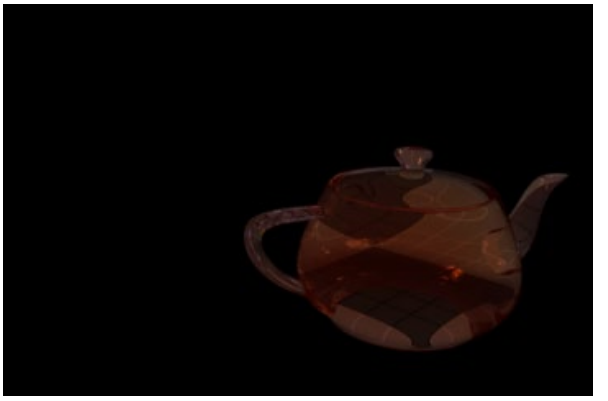
The 'Specular Shading' output generates a channel of all the specular shading within a scene independent of all other attributes.

Subsurface Shading



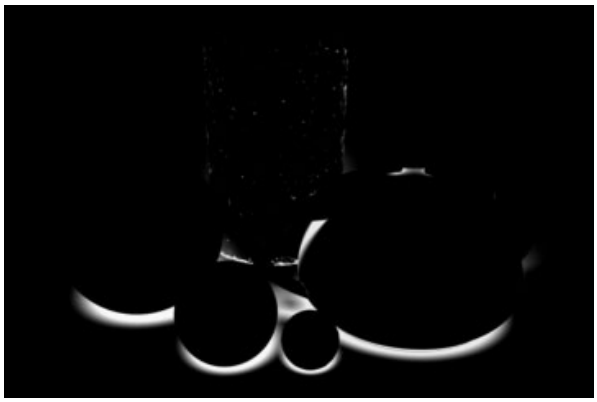
The ‘Subsurface Shading’ output generates a channel of all the subsurface scattering shading within a scene independent of all other attributes.

Transparent Shading



The ‘Transparent Shading’ output generates a channel of all the Transparent shading within a scene independent of all other attributes.

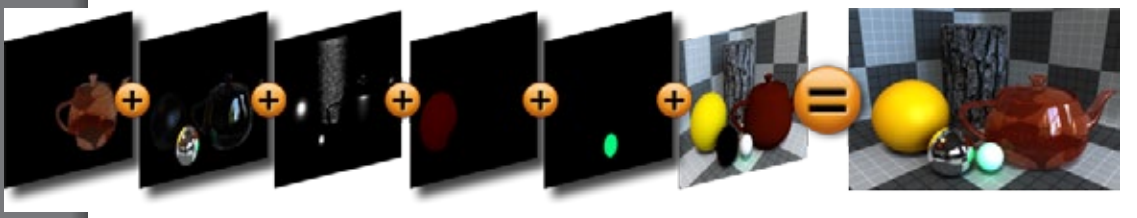
Shadow Density



The 'Shadow Density' rendering output renders a channel generated from the density of all direct light shadows in modo exclusive of color shading or texture. The darkest areas of shadow render as white white ramping toward black for the areas that would be fully illuminated. When rendering in layers, it is very useful to have shadows rendered separately giving one control of color and transparency. You could easily invert this channel in an image editing application and layer it over the unshadowed render set to 'multiply', giving control over how dark the shadows are based on the layers transparency.

Compositing Outputs

Render output are incredibly useful when users wish to compose the generated layers individually in an external application, such as Photoshop, After Effects, Nuke or Fusion. This can aide in tweaking color, among other settings as a post process without requiring additional render time. The question most often asked is "Which Render Outputs combine to equal the Final Color output?" The answer is "Diffuse Shading (Total) + Transparent Shading + Reflection Shading + Specular Shading + Subsurface Shading + Luminous Shading = Final Color", where each successive layer is added to the layer below in the composition (also called *Linear Dodge*). Users may also render out simultaneous alpha channels for each item in a scene by creating item masks for each and placing a render output set to "effect:alpha". Then each time a render command is invoked, all active (visible) render outputs will be generated simultaneously.



Render Commands

Rendering is the ultimate result of the settings in the shader tree, generated from the perspective of the 'Camera' item. The 3D GL viewports give a good preview of items in a scene, as well as any animation and deformers, but it is only an approximation and many advanced or render intensive options are excluded completely. The preview viewport offers a much better approximation, however, many settings are throttled back to keep interactive speed at maximum. The only way to view a scene, including setting as specified in the 'Render Item' are through the various render commands.

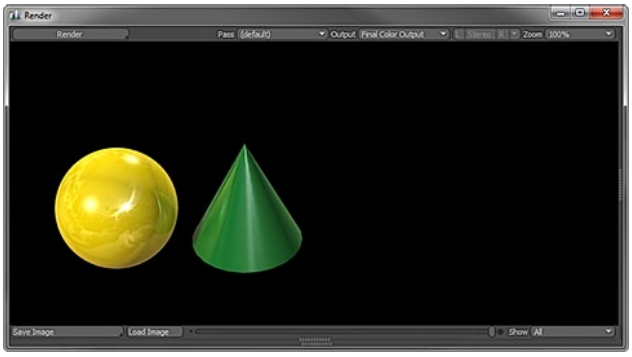
Rendering shouldn't be thought of as the final step in a linear workflow of modeling, texturing and rendering either. Throughout the process, users can render and preview their scene iteratively ensuring the proper final outcome. However, completely rendering the entire scene continuously would slow down the creative process, therefore modo has a variety of commands to facilitate various workflows, found in the menu bar under the 'Render' menu. Each is intended to give users greater control over the rendering process, ultimately providing a faster, more streamlined workflow. The various commands are especially useful in expediting the preview of large scenes, when combined with the 'Auto Visibility' settings of the Item List. The result of each

command is displayed in the 'Render Display' window which automatically opens each time a rendering command is invoked (if not currently open). Users should note that images in the Render Display are cached upon render completion, but are not *saved* to the users system. Please reference the Saving Images page of the documentation for more on how to save rendered images.

The various commands play off the visibility of items in the 'Item List' (the eye ' ' column) or as defined in the 'Display' viewport, which can be set to override item list settings. Take, for instance, the example below, there are four simple items, a sphere, a cone, a cube and a cylinder. The sphere is visible and selected, the cone is visible, but not selected and the cube is not visible but selected, and the cylinder is neither visible or selected; one of each possible *state*.

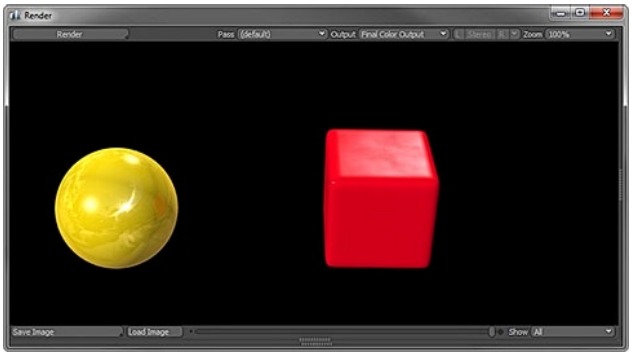
Render--

The 'Render' command is the most common of the rendering commands, it is the function invoked when pressing the ubiquitous 'F9' key. This option respects the visibility column of the 'Item List'. Items that are visible will render, regardless of selection, items not visible will not render. From the above item list, only the Sphere and Cone are visible, therefore the only items in the Render Display window.



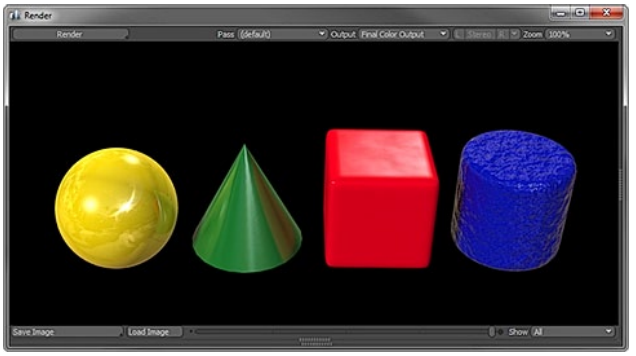
Render Selected--

The 'Render Selected' command will only render items *selected* (i.e. highlighted) in the item list, regardless of their visibility. Therefore, the Sphere and cube will be the only items in the Render Display, according to the item list above.



Render All--

The 'Render All' command will render all items in the item list regardless of visibility, therefore all four items in the list will be visible when using this command.



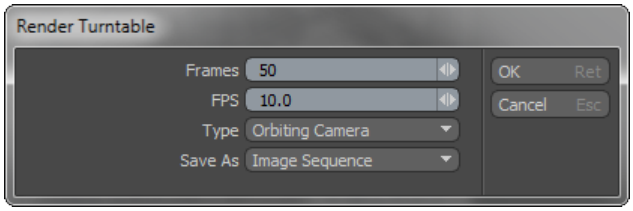
Render Current View--

The 3D openGL viewports also respect visibility of the item list, although the display style of background (unselected) items is controlled differently. Users wishing to render from the unique perspective of a 3D openGL viewport (such as any perspective or orthographic views) can utilize the 'Render Current View' command, which temporarily moves the 'Render Camera' to match the position of the viewport specific camera, rendering the scene, however, the focal length and aspect ratio cannot be matched precisely, producing a rendered image that is an approximation of the viewports view.

Although not exact, this function offers users the ability to move freely through a scene, previewing items without moving the camera item itself. If users wish to move the camera item to match the view of a viewport, use the 'Sync to View' command.

Render Turntable--

The 'Render Turntable' command provides users a simple means to generate 360° turntable animations without the need to keyframe any animations. When users choose the command, they are presented with the following dialog panel-



Frames: Sets the total length of the rendered animation in frames.

FPS: Frames Per Second defines how fast or slow the animation will play back if the user chooses to render the turntable as a video file.

Type: There are two options for the turntable type; 'Orbiting Camera' rotates the camera item around the origin and 'Spinning Geometry' which, obviously, spins the geometry itself around the origin, lights and other elements remain still. The main different between the two is in how light and shadow play over the surface as it rotates.

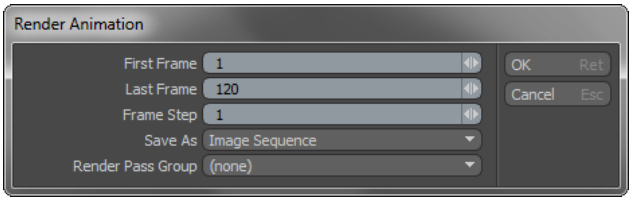
Save As: The 'Save As' option determines how the rendered turntable is saved to disc. There are three options, 'Image Sequence', 'Movie' and 'Layered Images'. Image Sequence will save a series of numbered frames, where Movie will render out a platform appropriate video file. If the users wishes additional alternative, the 'Layered Images' option will rendered out a sequence of frames, saving all the different render outputs of the Shader Tree within a single layered file. File formats supported by modo are fully covered on the 'Saving Images' page of the documentation.

Before generating a turntable, user will need to position the camera, framing the image as they wish. This will be the base position of the camera. Users should also make sure the geometry is centered at the origin, as this is the center of rotation for both the 'Orbiting Camera' and 'Spinning Geometry' types of turntables.

Render Animation--

The 'Render Animation' command instructs modo to render out a sequence of frames,

by default, determined by the 'Scene Range' values. Users can override this range by adjusting the values directly in the 'Render Animation' dialog. Users can then choose how the sequence of frames are rendered and saved to disc. The 'Render Animation' command will default to saving files to the name and location defined in the render output, if specified, asking if users would like to override or accept the designations. When no filename is specified, modo will open a OS specific dialog prompting users to choose the filename and save location.



First/Last Frame: Determines the sequence of frames rendered.

Frame Step: Determines the progression of rendered frames, useful in skipping frames to speed up a render. A 'Frame Step' of 1 would render every frame, a Frame Step of 2 would render every other frame and so on. To play back a sequence at the correct speed, divide the real FPS (frames per second) by the 'Frame Step' value to get the new FPS.

Save As: modo provides several options for saving the image sequences--

Image Sequence- Each rendered frame is saved individually, with the frame number appended to the filename as well as 'L' or 'R' for stereo pairs, if applicable.

Movie- Frames are rendered as a single animation file. Format option are dependant on platform. This option is also necessary to save still images as stereo pairs when rendering stereoscopic sequences.

Layered Images- The 'Layered Images' option acts exactly like the Image Sequence option, however, each active 'Render Output' is combined into a layered file, which can be very useful in reducing clutter when compositing images in an external application, reducing the number of files necessary to import. Check to make sure your target application can accept layered files before proceeding with this option.

Render Pass Group: Determines the 'Pass Group' to render. Pass Groups are separate iterations of a single scene that can be rendered sequentially. Users can define as many passes as they wish to a Pass Group. When that Group is specified, each iteration will be rendered and saved according to the settings of the Render Output, with naming that conforms to the settings of the Frame 'Output Pattern' option. For more

information on creating and working with 'Render Passes', please reference that page of the documentation.

Open Preview Render--

This command opens a new window that contains a Preview viewport, which uses the modo render engine in a progressive refinement mode, allowing users to see all changes to your scene as a fully rendered image. Within the default modo interface, a Preview viewport is embedded under the 'Render' tab, providing users a fast, interactive means of determining lighting and surfacing settings, albeit some throttling to the quality settings. This command calls the same display technology, but from within anywhere in modo. User may also open a Preview Render window by pressing the 'F8' keyboard shortcut.

Open Last Render--

This command simply opens the Render Display window, and displays the last successful full render without the need to invoke any of the render commands.

Render Passes

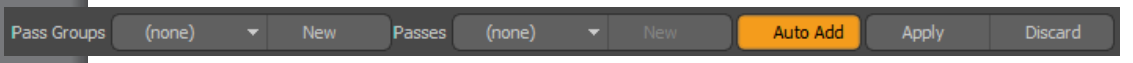
Render Passes offer users a way to control virtually any aspects of a scene that is, in-turn, used to automatically produce multiple rendered images; each rendered image reflects all the settings of an individual *pass*. In their most basic sense, Render Passes are simply containers for Channel values. Users can add a 'Pass'; subsequent edits to the scene will then be stored within the current pass or passes. When the 'Render Passes' command is invoked, each particular pass of the 'Pass Group' will be rendered in succession and saved as specified. This way of working is simple, yet incredibly powerful, providing users a means to create rendered variations that were previously very difficult, if not impossible to do within a single scene.

Understanding Passes

Passes conceptually work as containers, these containers only hold channel *values*. 'Channels' are simply any attribute of an 'Item' that can be animated (not that they *have* to be, just that they can be). Users can learn more about 'Channels' by referencing that page of the documentation. Once these Channel values are created, they sit in their Pass *container* as a layer above the scene itself called the 'Setup' layer. When a Pass is selected, any values held in the Pass will replace those values of the 'Setup' layer below it. For example, a Cylinder could be colored Red at the Setup level, and a Pass could be created that changes the color to Orange, disabling the Pass would return the color to Red. When the passes were rendered, there would be two resulting images, one with a red cylinder and another with it orange. The layering of the channel values

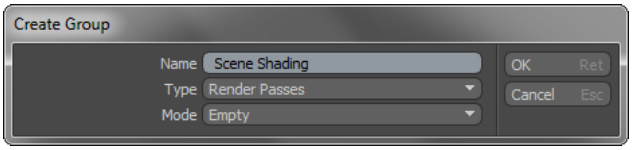
is an important aspect to understand, because there is also a layer in-between a 'Pass' and a scene called the 'Edit' layer, explained further on.

Basic Pass Workflow

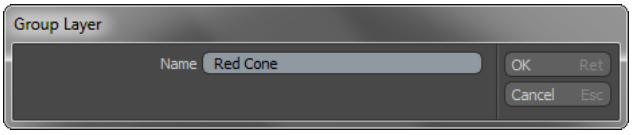


The Render Passes UI (user interface) is found in the 'Render' interface tab. It is controlled mainly by the buttons above the default OpenGL viewport. The workflow for passes moves left to right, first with the 'Pass Groups' bank with a button to select the Group and a button to create a new group. Next the 'Passes' bank, also with a selector and a new button. Next are the "Auto Add", "Apply" and "Discard" buttons that further control pass creation.

Prior to creating any 'Passes', users will need to first create a 'Pass Group'. Pass Groups are simply collections of different passes, meant strictly as a organizational device. Users will be unable to create individual 'Passes' until a 'Pass Group' is defined *and* selected. To create a group, press the 'New' button and define a name in the Popup dialogue box. Press 'OK' accepting all other defaults.



With the Pass Group created, it is automatically selected (as evidence of the named group replacing the (none) value); any passes created will be automatically added to that Group. So the next step is to create the 'Pass' itself. Pressing the 'New' button to the right of 'Passes' opens the Group Layer dialogue where the Pass itself is named. Once a name is entered, press OK to define the pass.



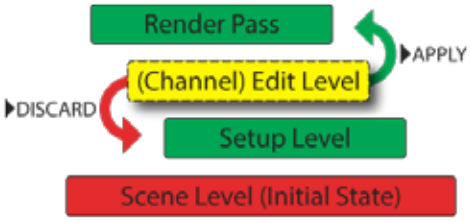
Next is the editing of the Channel values themselves. With 'Auto Add' enabled (the default state), any changes made to any attribute automatically adds that attribute Channel to the currently selected pass. What this means is users can simply edit the scene and any modifications are automatically stored/saved in the pass. Disabling

the pass, by returning the pass selection to '(none)' will return all the values to the underlying Scene state.

TIP: With the addition of Passes to a scene, complexity can increase exponentially, therefore is is important that users maintain a well organized scene, especially in relation to the Shader Tree.

Applying and Discarding

modo's pass system conceptually works in layers, you have the bottom most base layer called the 'Scene' layer, this is the inital default state of any item as it is added to the scene, including Mesh items, Cameras, Lights and so on. The next layer is the most important, it is the 'Setup' layer, all attributes from any item in the scene as they are modified are stored at this level. Each named Pass exists as a layer above the Setup layer. For any channel values stored in the pass, when that pass is enabled, simply by selecting it, the stored values replace those of the Setup level values as long as the pass is active. However, when users select a pass and begins to make changes, a new layer is inserted between the curent Pass and Setup layers called the 'Edit' layer. This is where the 'Apply' and 'Discard' buttons come in to play. When a modification is made to a scene with a pass active, the values don't go directly into the pass. They can be pushed into the pass by either exiting the pass, or pressing the 'Apply' button. Pressing the apply button allows users to continue making modifications, pressing discard will returned the values to either their previous state when the Pass was entered or when Apply was last pressed.

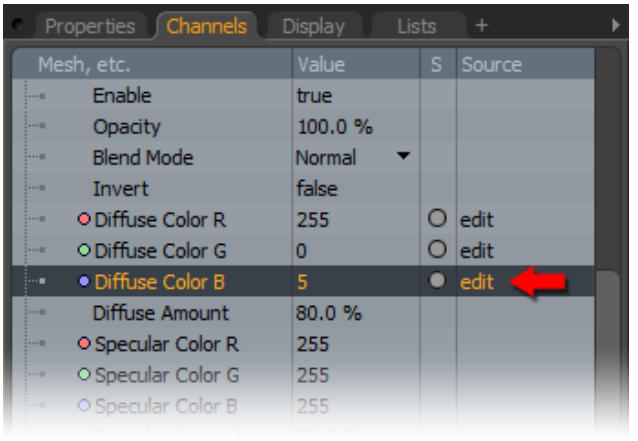


Advanced Render Passes

Users can successfully use modo's Render Passes and be blissfully unaware of what is going on in the background. For most users, simply using the 'Auto Add' function and then making normals scene modifications, modo will automatically take care of all the technical details, as long as the proper Passes are sleected when attributes are modified. However, if something does go wrong, a basic understanding of the

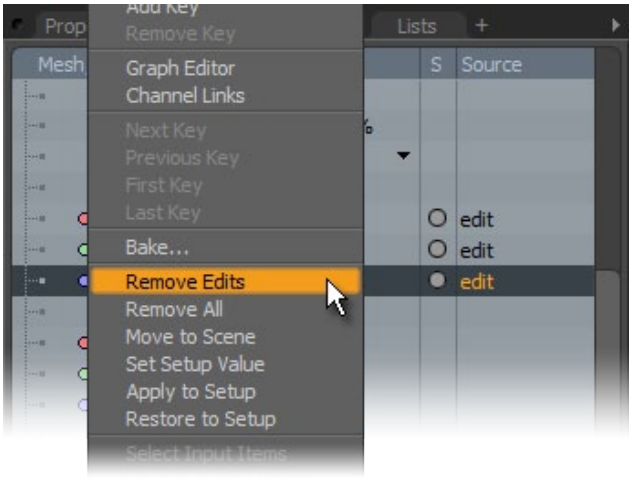
mechanics of it can be helpful in resolving any issues that may arise.

Editing scene attributes for passes can be a highly controlled *manual* process or completely *automatic*, all depending on user preference. This is controlled by the 'Auto Add' button. The button is enabled by default. In order for a Channel value to be stored in the Pass container, the channel itself needs to be stored in the pass. What the 'Auto Add button does is simply places the channel of any modified attributes in the Pass. If users were to *disable* the 'Auto Add' function, then users would manually need to add the channels that they would wish to modify. This is done in the 'Groups' palette. Passes are actually a function of the 'Groups' palette, used to store the collections of channels and their resulting values. Anyone familiar with Groups will then understand why the dialogues that pop up are called 'Create Group' and 'Group Layer'. The Channels can be added by selecting them in the Channels viewport and dragging and dropping them into the Pass Group in the Groups viewport, or by using the 'Add Items' function found in the 'Groups' viewport itself. More information on working with Groups is available on that page of the documentation.



There is also some additional information provided regarding pass states in the 'Channels' vieport. Down the right hand side of the viewport is a 'Source' column. When any pass is selected, and channel values are modified, the source column will say 'edit' telling users the current value is in the edit layer, it has not yet become part of the pass, and has been modified from the 'setup' level. When users apply those changes, pushing the values into the pass, then the source column will display the name of the pass. When the pass is disabled, the name will revert to 'setup', letting users know the current value for that channel is the base level. This can be very helpful in those instances where values were mistakenly applied in a pass that were meant for the scene, or vice-versa. By RMB+clicking on the channel to open the contextual menu, a

number of options are provided to give users some ability to push those values around between the different layers.



Remove Edits: The 'Remove Edits' option reverts the values to the last saved state, such as if a Render Pass was selected and then several edits were applied, prior to exiting the pass or pressing 'Apply', users could use the 'Remove Edits' command, returning the values to the initial state they were in when the pass was selected. This is the same function as the 'Discard' command.

Remove All: The 'Remove All' command removes all edits to the target channel reverting the values back to the initial default (Scene) state.

Move to Scene: Takes the current 'Edit' state value and pushed it to the Scene level.

Set Setup Value: Sets current channel values as the base Setup value and pushes value to render Pass.

Apply to Setup: Moves the current edit value to the 'Setup' layer, leaving any selected pass values unchanged.

Restore to Setup: Reverts the current pass value to that of the Setup.

Adding New Items

An important detail to understand is that Passes cannot contain Items, only channels, therefore any item that is added will be added at the Scene level (and subsequently becomes a part of all existing passes). If an item is added to the scene, it will need to be added at the 'Scene' level. If one were to accidentally add an item while a Pass was

active, not to worry, the Channels list viewport offers functionality to move data in and out of the 'Setup' level of the scene, so all your hard work wont get lost.

Rendering Passes

Once the various passes have been created, in order to actually render the passes, users need only to invoke the 'Render Passes' command found in the menu bar under the 'Render' menu. Users can select a Pass Group and all of the contained passes will be rendered in succession and cached into the Render Display window. Users can utilize the 'Save As' function to save the image to their local hard drive. To automatically save images when rendered, users can utilize the 'Render Animation' command and determines the pass group for rendering in that dialogue where the files will save according to the settings of the 'Render Outputs' of each pass. File names will conform to the settings of the Render Items 'Output Pattern'. Additionally, Passes within a Pass Group may be enabled and *disabled* temporarily, controlling whether they will render or not. Users can *disabled* a Pass from rendering within a Pass Group in the 'Groups' palette by LMB+clicking the tiny camera icon in the 'Render State' column of that viewport. LMB+click the empty column slot to enable rendering for the Pass.

Post Render

Render Display

The 'Render Display' window provides users a variety of useful options and feedback for working with the resulting rendered pixels for any modo scene. The Render Display window opens automatically when invoking any of the Render commands, either by keyboard shortcut or through the menu bar. Users may also recall the window without rendering any pixels with the command "Render > Open Render Window". Once open, the window provides a view of the rendered scene, along with a number of useful statistics, comparing tools and image adjustment controls for fine tuning an image. The window is broken into four main areas that are all inter-related -Image Display, Control panels, Render Browser and the Network View (*minimized by default*). Each section is documented below.

Image Display

The main area is the actual image display itself. This does duty as both a final image display and provides useful visual feedback to the progression of the rendering frame. Directly above this window are the controls that determine what is actually displayed, such as a particular Render Pass or Render Output to display during rendering, stereo image display options and a zoom level control. Users may also zoom in the window directly using the 'Ctrl+Alt+LMB click and drag. Once zoomed in, use the LMB+click and drag to navigate across the image. Pressing 'Ctrl+A' will fit the entire image to the viewable area.

Control Panel

The Control panel is organized into three tabs, each with a dedicated function. Users can LMB+click on any of the tabs to view the associated attributes. Going left to right, the first tab is the 'Image Processing' panel which provides scores of non-destructive controls to adjust, correct and tone the rendered image. The second is 'Compare, Region Options' which provides tools for rendering limited regions and comparing subsequent and prior renders. The third and last tab is the 'Statistics' panel, which provides useful information regarding the frame during rendering, such as memory usage, elapsed time and the final numbers of polygons; updated live throughout the render process.

Image Processing--

The 'Image Processing' section provides all the controls necessary to tweak the look of the final rendered image coupled with advanced visualizers, users can avoid most trips to an external image editor, best of all, its done non-destructively. When adjusting the values in the panel, the Image Display updates providing real time feedback to the changes making it easy to experiment and try different combinations.

Edit: The 'Edit' menu provides some options for moving image processing settings between the Render Output and the Render Display window.

Copy From Scene- If adjustments we're made to the Render Output options after the rendered image is complete, the 'Copy from Scene' option will grab those settings, updating the options of the Render Display window to match.

Paste to Scene- Any adjustments made to the rendered image within the Render Display window will not apply to subsequent renders, as the Render Display window reverts to the settings of the 'Render Item'. The 'Paste to Scene' option copies the settings to the Render Output back to the scene, applying it to all subsequent renders.

Reset- The 'Reset' option returns all image adjustment options to their default values.

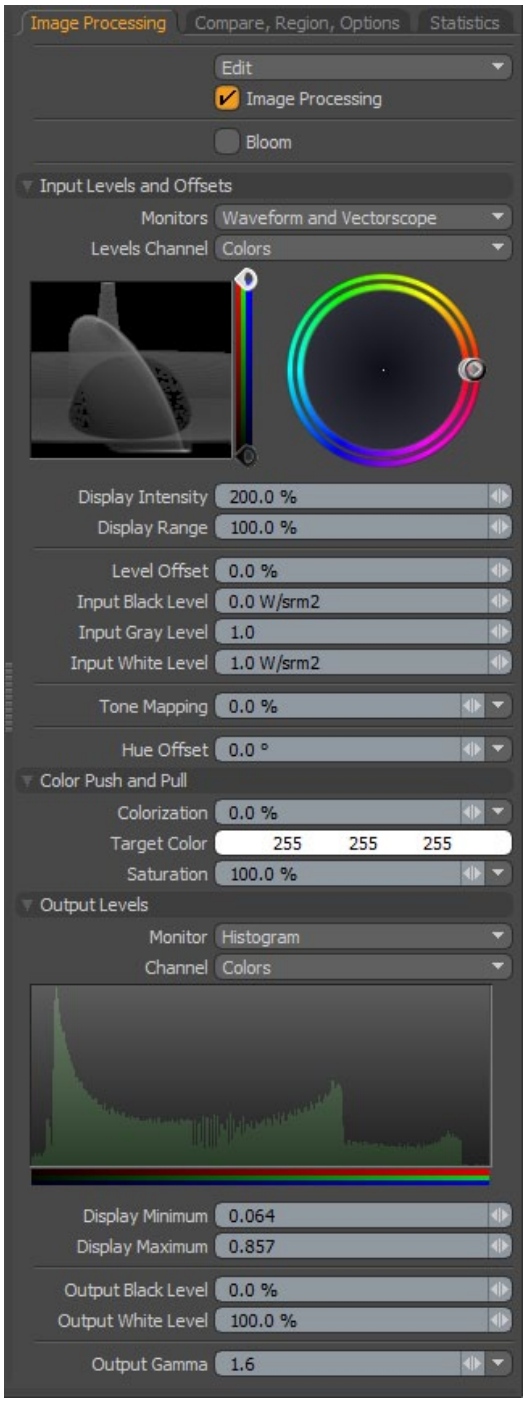


Image Processing: The 'Image Processing' toggle *disables/enables* the application of all the image adjustments options in the Render Display window only, especially useful in seeing the effects of the applied adjustments.

Bloom: In photography, when a very bright part of an image neighbors a very dark part, the bright part will appear to glow, this phenomenon is known as *Bloom*. modo's 'Bloom' simulates this effect by removing excess energy from a pixel and distributes it around to its neighbors. This checkbox enables the effect, once checked, two additional options become available. Values can be adjusted, providing near real-time feedback in the Render Display window-

Bloom Threshold: The 'Bloom Threshold' sets the lower threshold to which pixel are effected by 'Bloom'. When set to 100%, only the very hottest specular highlights will bloom, where at 0% every pixel in the scene will receive the bloom effect. (*examples are found in the* Render Output *page of the documentation*)

Bloom Radius: The 'Bloom Radius' sets the distance that energy is distributed, in effect controlling the size of the glow. 2% is the default value. Larger radius' impart and ethereal look to the rendered image. (*examples are found in the* Render Output *page of the documentation*)

Input Levels and Offsets--

Monitors: Several types of feedback displays are offered providing users with a means to analyze values in an impartial way when making adjustments-

Waveform- Displays the amount of energy at each intensity level as a series of vertical strips. Especially useful for detecting crushed shadows or blown out highlights across a wide area of an image, or for checking how evenly the overall tonal range is used in a given image.

Vectorscope- Displays the amount of energy across the hue and saturation spectrum within a unit circle of colors. Especially useful for checking overall color balance or to examine the amount of saturation for a given range of color.

Hue & Saturation Parade- A side-by-side variation of the 'Waveform' display, but for the amount of energy in hue or saturation instead of overall intensity.

Level Channel: The 'Level Channel' option allows users to choose which particular channel information contributes to the monitor displays. Entire images (Colors), or individual channels (Red, Green & Blue), or image brightness information (Luminosity).

Display Intensity: Display Intensity multiplies the displayed brightness of the monitor to make it easier to view intensity levels for under-sampled areas of the image.

Display Range: The 'Display Range' option allows users to set the range of the monitor(s) to values other than the default 0 to 1 range.

Level Offset: The 'Level Offset' option slides all color values up or down the value scale, shifting them identical amounts in the rendered image. For example, if an image had a 0% black pixel and a 50% gray pixel, offsetting the values 50% would result in the black pixel changing to 50% gray and the gray pixel changing to 100% white with all other values changing corresponding amounts.

Input Black Level: The 'Input Black Level' control specifies the radiance or luminance level that corresponds to pure black in the final rendered image (a pixel color value of 0.0). To use the input level controls effectively, 'Clamp Colors' must be disabled.

Input Gray Level: The 'Input Gray Level' is a non-linear luminance adjustment that applies a curve-like function to the pixels, modifying the mid-range the most while attenuating the adjustment amount for values moving toward the defined 'Black' and 'White' points. Values above 1.0 will lighten the midrange while values below 1.0 will darken the midrange.

Input White Level: The 'Input White Level' control specifies the radiance or luminance level that corresponds to pure white in the final rendered image (a pixel color value of 1.0). To use the input level controls effectively, 'Clamp Colors' must be disabled.

Tone Mapping: Dynamic Range is measured as the difference between the darkest shadow to the brightest highlight in an image. Most image formats dynamic range pale in comparison to that of the world around us. Much in the same way that a photographer struggles to capture all of the range our eyes can see using a camera, you may find your renders have shadow areas that are too dark, and highlight areas that are too bright. Luckily, modo renders in full floating point accuracy, providing dynamic range well beyond what any monitor is capable of displaying. This allows users to modify the overall tonal balance of the rendered image in several ways, including 'Tone Mapping', a technique that is used to compress the rendered dynamic range into something viewable; shadows will open revealing previously obscured details as will highlights. To use the 'Tone Mapping' function, 'Clamp Colors' must be disabled. You will often find it takes a combination of settings for White Level, Tone Mapping and Gamma to achieve the best results, please reference the 'Tone Mapping' page of the documentation for further examples.

Hue Offset: The 'Hue Offset' option will adjust the color values of the rendered image independently of the luminosity or brightness values, shifting them across the entire spectrum in a sequential fashion, for example when shifting a red color, moving the values in a positive fashion will adjust reds toward an orange hue, then yellow and so on. The hue values should be thought of like a wheel, where a rotation of 180° inverts all the color values and a rotation of 360° brings them back to their initial state.

Color Push and Pull--

Colorization: Users can use the 'Colorization' options to introduce an overall color tint into an image. This can be purely for artistic reasons but can also be helpful in reversing color casts introduced by Image Based Lighting among other things. Users need only to define a color with which to tint by setting a 'Target Color' and then adjusting the 'Colorization' amount determining the strength of the color into the image. A value of 0% produces no colorization attenuating toward 100% where the image is replaced fully by the target color.

Target Color: This value determines the color with which to tint the rendered image when the Colorization option is above 0%.

Saturation: The 'Saturation' options controls the concentration or amount of color in an image independent of the luminosity of brightness. At 100%, colors will be fully saturated as defined in the items material settings, reducing the value will reduce the overall color saturation down to a value of 0% producing an image only with gray shades. Since modo renders in full 32bit floating point accuracy, values above 100% will increase saturation without introducing the color banding and artifacts generally associated with oversaturating images.

Output Levels--

Monitor:

Histogram- A Histogram is a graphical display that shows the distribution of values across the spectrum of the image. Think of it like a graph of how many values each individual shade has, useful in subjectively determining wether an image is neutral, low key, high contrast or otherwise.

RGB Parade- Where the 'Histogram' shows overlays of individual channels the 'RGB Parade' shows each individual R, G and B channel side by side, proving a clearer view of each channels individual contribution to the final image.

Channel: The 'Channel' option allows users to choose which particular channel information contributes to the monitors display. Entire images (Colors), combination of all or individual channels (RGB, Red, Green & Blue), or image brightness information (Luminosity).

Display Minimum/Maximum: The 'Display Min/Max' option allows users to set the range of the monitor to values other than the default 0 to 1 range.

Output Black Level: This value determines the output value of what is considered fully black in the image. Adjusting the value upwards changes the Black point to a shade of gray. As values are adjust, intermediate values between black and white are adjusted as well.

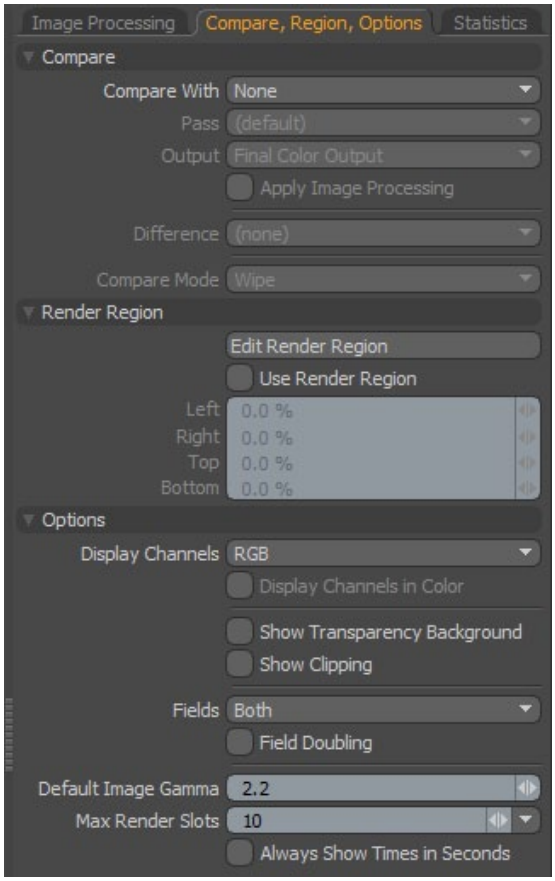
Output White Level: This value determines the output value of what is considered fully white in the image. Adjusting the value downwards changes the White point to a shade of gray. As values are adjust, intermediate values between white and black are adjusted as well.

Output Gamma: Gamma is the measure of total contrast in an image and represents the relationship of an image input to an image output. In this case, the 'input' is modo's rendered image; what modo is holding in its memory buffer and the 'output' is either the image being displayed in the 'Render Display' window (what you see) or the image that is saved. A gamma value of 1 is said to be 'linear', what modo rendered, is identical to what is displayed. Modifying the 'Gamma' value changes how the image is represented. Values higher than 1 lighten the image, while values lower than 1 darken it. Adjustments to the gamma value are not linear, i.e. they don't change all the pixels in an image the same amount, explaining why it is often referred to as a gamma curve. The default Gamma value is derived from the global preference found in the Preference Editor under 'Render'. Adjusting this Render Output 'Gamma' value overrides the global preference on a per render buffer basis.

To separate the gamma of the 'Render Display' window from that of the saved file,

open the Preferences window from the menu bar 'System > Preferences' and select the 'Rendering' options. By activating "Independent Display Gamma" you can set a gamma correction that will only apply to the image as it is viewed in the Render window. The saved image data will always use the Output Gamma value found on the Render Output in the shader tree.

Compare, Region, Options



Compare--

The Compare options provide users the ability to examine two images in a variety of ways, making it easier to contrast any differences between them.

Compare with: The 'Compare with' option determines what opposing images are compared-

None- No image comparison

"B" Slot- Two separate images are compared, 'Slot A' is determined by the current image selected in the browser, 'Slot B' is determined by holding down the 'Ctrl' key and LMB+clicking on a second image slot in the browser. A small 'A' and 'B' icon will appear beneath each image signifying which is which.

"A" Slot Output- With this option, users can compare between associated 'Passes' or 'Render Outputs' of the same image.

Pass: When the '"A" Slot Output' option is selected, users can determine which particular 'Pass' output to compare against with this option.

Output: When the '"A" Slot Output' option is selected, users can determine which particular 'Render Output' to compare against with this option.

Apply Image Processing: The 'Apply Image Processing' toggle *disables/enables* the application of all the image adjustments options in the Render Display window.

Difference:

Compare Mode: Several modes define how the two images are compared, each with additional options-

Wipe- Two images are overlaid and the top image is wiped away revealing the lower image. Additional options allow users to select 'Horizontal' or 'Vertical' wiping directions and set a position percent, or choose the 'Dissolve' option and set an 'Opacity' amount.

Spotlight- The 'Spotlight' option reveals the "B" images as a circle around the mouse pointer (move the mouse over the image to see its results) . The size of the circle is determined by the 'Spotlight Radius' control.

Side by Side- The 'Side by Side' options displays two half width images that can be scrolled with an option to Stack the images vertically.

Render Region--

The 'Render Region' option allows users to selectively render a smaller segment of the entire image making the iterative process of creating a final rendered image faster and more fluid. For users convenience, these settings duplicate the functionality of the Render items Render Region functionary.

Edit Render Region: When this option is *enabled*, users can drag directly over the rendered image to define the limited region area interactively. Subsequently, once the initial area is defined, edges and corners of the region will highlight allowing further refinement to the defined region. LMB+click and drag the highlighted border to adjust.

If edits are being made to the same scene, the area outside the defined region will be used as a background to successive renders until disabled. When switching between scenes or image resolutions, then the background area outside the region will simply render as black.

Use Render Region: The 'Use Render Region' option toggle will temporarily *enable/disable* the limited region functionality without losing any of the settings.

Left/Right/Top/Bottom: Left/Right and Top/Bottom percentage settings work in conjunction with each other to specify the total rendered region width and height.

Options--

Display Channels: All resulting rendered images are generated as standard RGB bitmap format images. For diagnostic purposes, it may be valuable for users to see the individual 'Red, 'Green' or 'Blue' color *channels* of an image.

Display Channels in Color: When this option is enabled, individual channels will display as colored versions of themselves. When disabled, individual channels will display as grayscale representations.

Show Transparency Background: When *enabled*, any loaded images that contain an alpha/transparency channel will display a grey checkerboard where the image is transparent mimicking a similar option found in many image editing applications. Note that this option doesn't apply to internally rendered alpha channels.

Show Clipping: When enabled, a diagonal black and white stripe pattern (called a Zebra pattern) will display over overexposed or blown out areas of the image providing useful feedback for adjusting image white point and tone mapping an image to compress as much dynamic range information as possible.

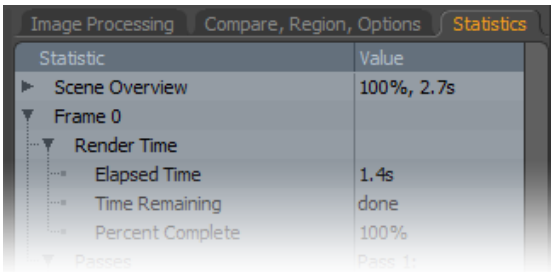
Fields: For scanline televisions (older tube televisions) there were originally two half frames that made up each whole frame, interlaced on top of each other to make a single image. Enabling the 'Field Rendering' option will properly render two discreet moments in time and interlace the resulting half-frames into a single image (in essence doubling the frame rate). The 'Fields' option allows users to independently display either the 'Upper' or 'Lower' field, or 'Both' which is the standard default for non-interlaced (progressive) frames.

Field Doubling: When viewing individual fields, enabling the 'Field Doubling' option will duplicate every other line eliminating the black screen lines from the display.

Default Image Gamma: When loading non-HDR image formats into the Image Browser, this is the default gamma value applied to the imported image. For more information on gamma, please reference the Render Output page of the documentation.

Max Render Slots: This value determines the maximum number of frame buffers (renders) saved on disc in the Image Browser. Once this value is exceeded, the oldest render will be deleted, unless locked. If all frames are locked, then they will be treated as if all unlocked once the maximum number is met. This option is not affected by images loaded into the Browser.

Always Show Time in Seconds: Displays time in total seconds instead of HH:MM:SS, example- 01:12:37 would be displayed as 4357 seconds.



Statistics

The Statistics panel provides a huge amount of statistical information related to rendering the scene. Included in this information is the elapsed time as well as an approximate remaining time for in-progress frames. For finished frames, the total rendering calculation time is displayed, along with total memory usage, total number of polygons generated. The information is cleanly organized and largely self explanatory. Navigation of the list works the same as other viewports. Users can open or close a specific section by LMB+clicking the preceding arrow (▶). Holding shift and clicking an arrow will open/close all sub-sections of the clicked on topic, making it easy to view all the information related to a single frame.

Render Browser

The Render Browser allows for an unlimited number of rendered frames (well, it is limited by the amount of hard drive space available). The browser itself serves a couple of purposes, foremost being that users can select the current image to be displayed in the display area by LMB+clicking on any of the icons. The current image selection is designated by the orange outline of the icon; only the current image can be modified using the image adjustment controls. The render order of images is also displayed using the small number icons above each image. When the mouse hovers over an icon, two additional controls appear, an 'X' that if clicked will remove the image from the browser (deleting it from disc) and a small lock icon users can click to lock items from being deleted accidentally. Simply click the lock icon again to unlock. Users can select multiple images by holding down the Ctrl' key as well allowing for gang removal/locking/unlocking of multiple images.

The Render Browser is also used to Load and Save images. When images are rendered, their existence in the Render Display is kind of in limbo, as the images are visible can be viewed and modified in the Render Display, but they can't be accessed outside

of the display and therefore will need to be saved. Using the 'Save Image' button directly above the browser allows users to define a format and location for the image. Please reference the 'Saving Images' page of the documentation for more information concerning images and formats.

Additionally, images can be loaded into the browser area, which is very helpful for when users wish to apply image adjustments, or use previous renders to compare a current render from when the render buffer no longer exists. To load an image, LMB+click the 'Load Image' button above the browser area, opening an OS specific window. There, users may navigate to the stored image location, select the target file, and press 'Open' to load the file into the browser. If any modifications are applied to the file, it must be saved again to retain the settings.

Network View

Below the Render Browser area is a minimized 'Network View' panel. Users may open the panel by LMB+clicking and dragging upward on the small group of tiny squares, revealing the panel. This viewport indicates the number of cores available on each machine, machine name, user name, mode, status and message. Further, you can tell if a machine is connected to your active host by the color of the text. Green text indicates that the machine is connected and ready for rendering. Black indicates systems that are connected, but not enabled as slaves. Any other color indicates a problem with the connection or availability.

General View Shortcuts

The Render Display provides standard modo controls to adjust size and minimize frames, all the while automatically adjusting the interface, giving users control over what can be seen. Dragging the splitters between the panels will scale the contents of each. There are also some additional controls that provides users an easy way to quickly makes certain panels within the display visible.

'Spacebar' - minimizes everything except the image pane.
'T' - toggles thumbnail panel visibility.
'N' - toggles the network view.
'S'/'I'/'C' - toggles right-hand panel tabs ('Statistics'/'Image Processing'/'Compare' respectively).

Tone Mapping

Tone mapping, for computer graphics, is simply the adaptation of contrast across a range of values. The reason for the manipulation can range from strictly aesthetic

to purely technical. Tone mapping as it relates to modo is often meant for the compression of dynamic range in the rendered image; dynamic range being the ratio between the brightest value and the darkest value.

Let's stray for a moment to propose an example to illustrate dynamic range. If you look at the world around you, there are areas that are extremely bright, such as a window with a good view of the sky, and areas that are very dark, possibly under your desk; the total range of values between the darkest area to the brightest in any scene is known as the *dynamic range*. While your eyes are able to adjust appropriately to see detail in both areas, most image capture and display devices cannot. A camera for instance; if you take a photo of this scene that has the window and the shadow area beneath the desk both visible, the shot can be exposed either for the area under the desk, *overexposing* any detail visible through the window, or exposed for the window *underexposing* the shadow area, losing any detail in total blackness. Even if the capture device could capture the entire possible range, no means of viewing it could display it properly. A photograph of the sun would never sting your eyes when viewed on a computer monitor or printed in a book the same way the real sun does.

Lets revisit this example from the Image Based Lighting page, various exposures of the scene were captured, and combined producing a new image that has an approximation of the total illumination in the scene, i.e. the full dynamic range of light to dark.

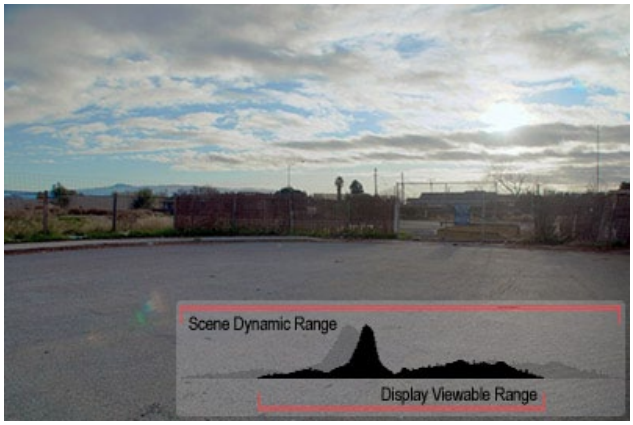
Sky exposure (-2 EV)

Default Exposure (0 EV)

Foreground Exposure (+2 EV)

Once the HDR image is created, all that extra information is still, in a way, hidden until the user tells the system how to display it properly. This is where tone mapping comes in. Most modern displays, whether CRT or LCD, are based on technology that greatly limits the visible dynamic range. As with the exposure examples above, areas outside this range are simply clipped, or cut off. Consider these histograms (a graphs of the brightness values), a normal full range HDR can only display a select range of values at any one time, producing a similar effect as the original exposures above.

Tone mapping can compress the total dynamic range of the image into a range that is viewable on your screen. Here is a simple example of the combination of the above sequence of images tone mapped with an example of the compressed histogram.



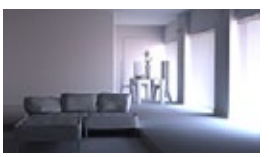
Unfortunately, it's not a simple push-button process, as the result of the tone mapping is highly dependant on the image. Each image will react differently, some becoming too dark and/or bright, and others will look dull and gray. modo offers a variety of functions that can help resolve these issues, now that you hopefully have a better understanding of dynamic range, lets visit them.

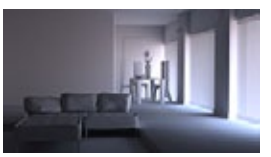
modo renders in a full floating-point precision format, that sounds very technical, but what it really means is the rendering engine is just very precise with the values it generates, including capturing the full brightness range from the scene producing HDR images with much higher degree of accuracy than those generated from multiple photographic exposures. Among other benefits, this affords users with a lot of leeway in manipulating the tones, avoiding any of the common drawback of typical low-dynamic range image manipulations such as banding due to posterization. More importantly, the changes are all non-destructive meaning users may make adjustments without compromising this extended amount of information (assuming levels aren't 'Clamped').

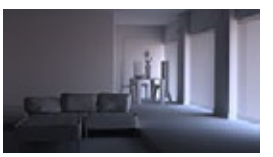
Lets consider the above example, rendered using the Physical sky feature, unclamped, with default settings. The image is of a small interior room with an outside facing window, but its pretty tough to see anything outside of the couch in this render, its pretty blown out and looks completely unusable, but hidden in those overly bright pixels is the detail we want.

The first setting to consider is the 'White Level', specifically this sets what pixels are pure white in the image producing a result similar to how exposure functions in a camera. Lower values would seem overexposed, though increasing values wont underexpose the shadow areas. Below, raising the white level value increasingly reveals the details in the windows and rear of the room, but as the other pixels adjust accordingly it has the unfortunate side effect of darkening the midtone values. These can be adjusted with Gamma.

White Level 2.0

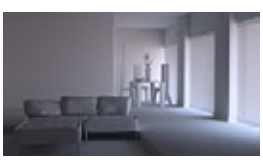

White Level 4.0


White Level 6.0

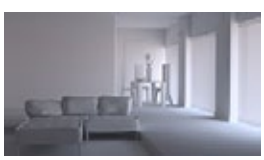

White Level 8.0

The gamma function will adjust the distribution of tones, modifying the mid-tone values to a much greater degree than the lightest and darkest values in the image. Increasing the curve of the gamma (by raising the value) you can see it lighten up those mid-tones

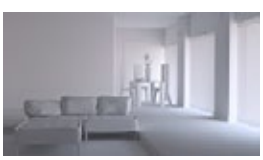
while keeping the darkest shadows *and* the lightest areas produced by the white level adjustment made earlier.



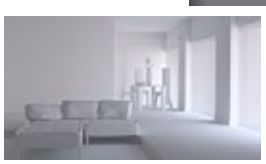
Gamma 2.2



Gamma 3.0



Gamma 3.8



Gamma 4.6

Now if we take a quick step back to the default render above, we can see the addition of just the Tone Mapping setting to our raw render. When set at 100%, this function compresses the full dynamic range of the unclamped render into the same limited range that's visible on our monitors, revealing all the detail of the blinds in the window and the rear of the room, but unlike setting the White Level, the midtones are relatively untouched. Only problem is, at 100% the room also looks relatively flat. Including some of the contrast in the initial render will help create a more interesting image, it will take adjustments from all three settings for the best overall result.

Tone Mapping 25%

Tone Mapping 50%

Tone Mapping 75%

Tone Mapping 100%

The perfect image will always be a matter of personal preference, but below I've adjusted the white level to 3.0, given the image a gamma of 2.2 and applied an 80% tone mapping amount for this result. It keeps some of the subtle coloration introduced by physical sky, resolves a lot more detail in the vertical blinds and windows while retaining a pleasing shadow under the foreground couch. All of this without a single second of additional rendering time either, all these settings are applied effectively in real time. Compare the image below to the raw render output above and you'll quickly realize the power of these three seemingly simple settings.

Tone mapping isn't just for global illumination, image based lighting or Physical Sky either, it can help in lots of instances. It won't solve all lighting problems, but it can add finesse and refinement to seemingly difficult situations, and save you a trip to your image editor. The best way to build familiarity with the settings is to experiment, after time anyone can become proficient at producing consistently high quality images with ease.

TIP: Many modern image editing programs are beginning to include functionality for working with HDR images, with full complements of tone mapping functions (The variety of options available is almost absurd). If you prefer to work in another tool, simply save the rendered image unclamped (and undithered!) to a format that preserves the extended information, such as EXR or HDR and open in your favorite HDR image editor.

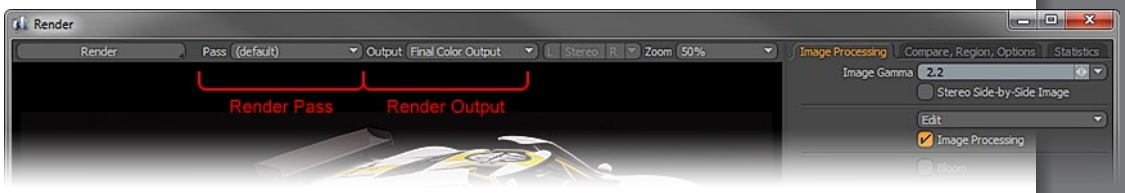
Saving Images

modo saves images in various ways, depending on how the image is generated and/or applied. Understanding the individual instances will ensure that no image goes unsaved, or is inadvertently saved either.

Saving Rendered Images

When rendering by way of the 'Render' command (menu bar option "Render > Render" or by pressing 'F9') modo obviously renders the image, but once finished, the image is in a limbo state within the Render Display window. It is cached to disk temporarily for you to preview; this is useful for the iterative process of adjusting scenes. However, the image itself is not truly *saved* (i.e. accessible to other applications) until the user

specifically tells modo to do so using any of the commands necessary for saving; the 'Save Image' and 'Save Layered Image' functions. Each time any 'Render' command is invoked, modo will calculate all render outputs and render passes. Once rendered they appear in the 'Pass' and 'Output' lists of the Render Display (*highlighted below*).



When using the 'Save Image' option, the currently selected (i.e. visible) render output and/or pass will be saved. Pressing the 'Save Image' button found in the upper left of the render browser area of the Render Display opens an operating system appropriate dialog window, allowing users to specify a location and filename for the image, and most importantly a format. modo supports a wide variety of bitmap image formats, each with their own benefits and limitations. By itself, modo support the following formats--

Format Name	Extension	Bit Depth per channel	Compressed	Type
Flexible Precision Image Format	.FLX	32 bit	Yes	Lossless
Targa	.TGA	8 bit + Alpha	No	
Windows BMP	.BMP	8 bit + Alpha	No	
Radiance High Dynamic Range	.HDR	HDR	Yes	Lossless
JPEG 2000	.JP2	8 bit	Yes	Lossy
JPEG 2000 16-bit	.JP2	16 bit	Yes	Lossy
JPEG 2000 16-bit Lossless	.JP2	16 bit	Yes	Lossless
JPEG	.JPG	8 bit	Yes	Lossy
Portable Network Graphic	.PNG	8 bit + Alpha	Yes	Lossless
Portable Network Graphic 16 bit	.PNG	16 bit + Alpha	Yes	Lossless
PSD (Photoshop Document)	.PSD	8 bit	No	
SGI RGB	.SGI	8 bit	No	
Tagged Image File Format	.TIF	8 bit + Alpha	Yes	Lossless
16-Bit TIF	.TIF	16 bit + Alpha	Yes	Lossless
16-Bit Uncompressed TIF	.TIF	16 bit + Alpha	No	
Open EXR	.EXR	HDR	Yes	Lossless

When using the 'Save Layered Image' option (found by a LMB+click and hold on the 'Save Image' button to open the pop-up menu), modo will save all available Render Outputs simultaneously, when pressed an operating system appropriate dialog window opens allowing users to specify a location and filename for the image(s), and a format as well. The following options are available--

Format Name	Extension	Bit Depth per channel	Compressed	Type
Layered PSD	.PSD	8 bit	No	
Multi-File PNG*	.PNG	8 bit	Yes	Lossless
Multi-File PNG 16 bit*	.PNG	16 bit	Yes	Lossless
Layered Open EXR	.EXR	HDR	Yes	Lossless

*Multi-File PNG saves all outputs with appropriate output name appended to each

Saving Render Passes

modo's flexible Render Pass system provides users the ability to control virtually any aspects of a scene, and then render the results consecutively, producing multiple frames that can then be composed in an external image editing application if so desired. The Render Display offers two additional commands allowing users an automatic way to save multiples of Render Passes simultaneously. The 'Save Passes as Images' and 'Save Passes as Layered Images' commands. Both work identically to their single image counterparts, but will save all passes automatically using the naming conventions determined by the 'Output Pattern' options found in the 'Frame' subtab of the Render Item. Selecting either option will open a OS specific dialogue allowing users to specify a location, file name and format. The output pattern naming options will be automatically appended to the specified name when the user presses 'Save'.

Saving Animation Sequences

Saving an entire animation of still images individually would be pretty tedious, thankfully, there are several options for saving sequences automatically. Utilizing the 'Render Animation' command from the menu bar (''Render > Render Animation') presents three options for saving-- 'Image Sequence', 'Movie' and 'Layered Images'. Choosing 'Image Sequence' is highly recommended; this allows users to render entire animations of individual frames that can be assembled in a video editing application that accepts them. It also provides an easy means of updating frames without having to re-render the entire sequence, should the need arise (additionally, it offers the ability to recover from system crashes, allowing user to pick up where the system left off, without the fear of a corrupted video file). When invoked, an operating system appropriate dialog window will open allowing the users to, again, specify the File Name, Location and File Format. The 'Image Sequence' and 'Layered Images' options offer identical file format options as their respective commands from the Render Display window (listed above). The 'Movie' option provides choices specific to the users platform and operating system.

Render Outputs--

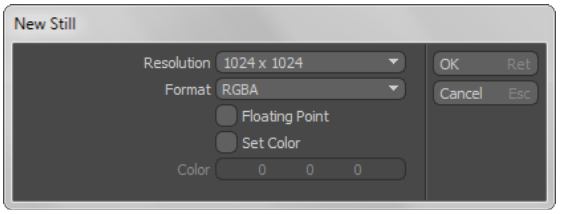
Individually, file location, name and format can also be specified within each Render Output. This is useful as it is saved within the file and is therefore portable and persistent across modo sessions. modo only utilizes the 'Output Filename' option when the 'Render Animation' command is invoked. Once specified, and the command initiated, a pop-up requestor will prompt the user if they wish to use the filename as specified in the render output. Select 'Yes' to use it, and 'No' to specify an alternate name.

Important: Users should note that pressing 'F9' or utilizing any of the other various render commands will not automatically save images as specified in a 'Render Output'. Only the 'Render Animation' command utilizes this option.

Saving User Created Images

modo offers users the ability to create bitmapped images directly in modo eliminating the need for an external image editing application. Within nearly every image loader and image browser viewport, users are offered options to create a 'New Image' or 'Image Sequence' (Alternatively, these commands are also available in the menu bar under the 'File' option). When utilizing either command, a operating system specific

dialog will open prompting users to choose a filename, location and format. Once specified, another dialog opens allowing users to choose various options related to the image.



Once the user clicks 'OK' or presses 'Return', the image is saved to disc. It's important for users to understand that this file is not imbedded into the modo file itself, but acts like any other imported image remaining linked to the saved external file as specified. This newly created image is effectively a blank canvas for users to use for painting, sculpting and baking (or any combination thereof). The image appears within the images viewport and is immediately available as a texture layer for placement anywhere within the shader tree. However (!!!) any modification made to the image once created will not be saved unless the user specifically requests modo to do so. Files can be saved several ways. First, by way of RMB-clicking the image in the image viewer palette, and choosing either the 'Save' or 'Save As' command. Secondly, by way of the menu bar commands 'File > Save Image' or 'File > Save Image As'. In the case of scenes with multiple images, the highlighted or selected image in the image viewer palette is the image saved when utilizing either command. Lastly, if any modifications have occurred to images present in modo when exiting the application, users will be prompted if they wish to 'Save' the changes, overwriting the files previous saved state.

Saving 3D Images

PNG Anaglyph Optimized mode

PNG Side By Side mode

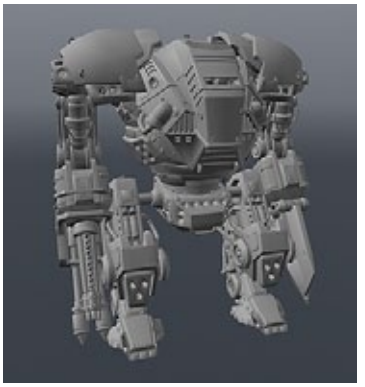
modo has the ability to save scenes rendered in stereo into any of the supported bitmap formats. In order to do so, the 'Stereo' option must first be enabled for the 'Camera' item in the properties viewport under the 'Camera Effects' subtab. Adjustments can also be made to the 'Interoccular Distance' and the 'Convergence Distance' if necessary. The 'Stereo Composite' option then determines how the two rendered images are combined to produce the final stereo image, covered fully in the 'Stereoscopic' section of the Camera Items documentation. Once *enabled*, each time a render command is invoked, the two images will be rendered consecutively representing the left and right eye views. These two images are combined as defined and displayed as a single slot in the Render Display window. Using the controls above the display window, users can choose to view the 'Left', 'Right' or combined 'Stereo' image. As with the other options, whatever is currently visible to the user will be the image that is saved.

TIP: Users often ask which is the best format to save in. Unfortunately, there is no one correct answer as it is highly dependant on the intended use; particular workflows impose specific requirements. Visual effects work is enhanced by the additional information afforded by HDRI formats such as HDR and openEXR, but these formats also require the greatest amount of storage space. Advertising and visualization are often further edited in imaging applications such as Photoshop. The layered PSD format could be a good choice, but the 16 bit flavors of TIF and PNG can afford additional editing leeway with their extended bit depth. Any of the compressed formats will help reduce the amount of disc space necessary, especially when storing long image sequences, but lossy formats may also introduce compression artifacts that can reduce quality of the final deliverable. For best results, users should thoroughly test their chosen format to ensure they receive the intended outcome.

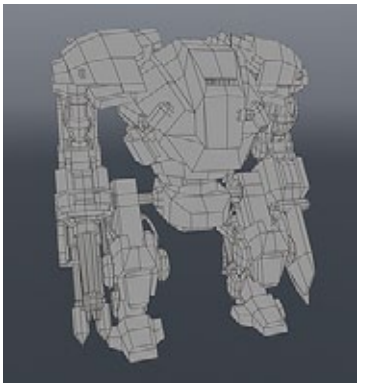
Baking

Image Baking Workflow

Render baking is used to pre-compute texture and lighting information. The result is often used for real-time applications (such a for games) or for exporting models into other software packages with complex shading intact, or simply as a way to eliminate rendering overhead in elaborate scenes. In this way shaders and lighting are computed by the modo rendering engine and *baked* directly into an image that maps perfectly to an existing UV texture.



Hi-Res Geometry



Lo-Res Geometry

Lo-Res Baked Result

Depending on what your intended purpose is, there are a couple of methods for baking images. If your intent is to simply merge (flatten) multiple shader tree layers, then all that is needed is an image map added to the shader tree set to the appropriate 'Effect' type before baking. When baking maps such as Displacement and Normal maps, two layers of geometry are necessary, typically one is the high resolution model comprised of all the details of the final model, and a second low resolution model that approximates the shape of the high resolution model (and the baking target). The low resolution geometry requires a proper UV map with a unique name (unique from any materials applied to the high res version). Additionally, users will need to make sure the applicable layers are selected in the item list, and that the appropriate UV map is selected. When Baking to Textures, suitable image maps must be placed and selected in the Shader Tree (Don't worry, it only sounds complicated, each set-up is detailed below). For information regarding adding layers and setting their 'Effect', please reference the 'Shader Tree' page of the documentation. All of the baking commands can be invoked from the menu bar within the 'Render' menu. They are as follows-

Bake to Render Outputs--

The command 'Bake to Render Outputs' provides a means to render a texture layer utilizing the functionality of the Shader Tree's 'Render Outputs'. This allows the baking of Ambient Occlusion maps, or baking a Final Color output that computes all surfacing and shading of the final rendered item. Once rendered, images can be saved out from the Render Display viewport, then imported and applied to the geometry.

When invoked, the command will render into the currently selected UV texture map so it is important to make certain you have the correct mesh item layer and UV map selected prior to activating the bake operation. Render Baking will use the settings from the current 'Render Item' to determine quality and frame size settings as well

as the intended contributors to the shading of the bake. By default, a baked render will contain all textures and shaders contributing to the RGB final image (as the default render output is set to Final Color). Baking can easily be redirected by simply retargeting the 'Effect' output setting on the Render Output prior to invoking the command.

Bake from Object to Render Outputs--

The 'Bake from Object to Render Outputs' command is similar to the 'Bake to Render Outputs' command above, however it allows the additional ability to render the surfacing and shading from a high resolution mesh onto a low resolution mesh. Baking from objects requires one high resolution model layer to extract the geometrical, shading and lighting information from and a low resolution *proxy* model layer with a UV map to apply those baked maps to. The low resolution layer should be in the foreground (selected and visible) while the high resolution model in the background (visible but unselected). In the Shader Tree, render baking will use the settings from the current 'Render Item' to determine quality and frame size settings as well as the intended contributors to the shading of the bake. Once specified, Bake from Object will proceed to use the low resolution layer's selected UV map to "Bake" out the user defined Image Maps. Once rendered, images can be saved out from the Render Display viewport, then imported and applied to the geometry.

Bake to Selected Texture--

The command 'Bake to Selected Texture' can be found in the menu bar under 'Render > Bake to Selected Texture', but is is more commonly invoked from the Shader Tree RMB context menu (opened by clicking on the target layer). This command offers the ability to generate a texture map image of one or more shader tree layers. Baking textures from inside the Shader Tree is useful for baking a single 'Effect' such as Diffuse Color into an image whereas the "Bake" command from the Render menu bakes all lighting, shadows and texture contributors into a final shaded image. It is reasonable to think of the Shader Tree level of baking as similar to *flattening* layers in Photoshop. Only layers of the same Effect setting *underneath* the texture layer that is triggering the bake will contribute to the baked layer. Once baked, this texture can remain in the Shader Tree as a pre-computed result of all contributors which results in a savings of render time and provides a real-time view of the effect within the viewport and when rendered. The 'Bake to Texture' command requires a UV map on the baked surface as well as a blank image map in the shader tree to bake into. Users should note that baked images are not automatically saved by modo. Once generated, images can be saved using the menu bar command 'File >Save Image'.

Bake from Object to Selected Texture--

This command offers the ability to use the UV map of a low resolution model to bake a texture map from a higher resolution model. Maps can be used for any of the various 'Effect' settings, including Diffuse Color, Displacement and Normal (This is how Tangent Space Normal Maps are created). Like the 'Bake to Selected Texture' command above, a UV map is required and a blank image map is necessary to bake into. Both surfaces must have different material name and cannot share the same UV map (meaning the name of each must be unique). When calculating a displacement map, the 'distance setting' from the popup requestor should be used as the 'Displacement Distance' setting in the Material dialog box when applied. Once baked, this texture can remain in the Shader Tree as a pre-computed result of all contributors which results in a savings of render time and provides a real-time view of the effect. Users should note that baked images are not automatically saved by modo. Once generated, images can be saved using the menu bar command 'File >Save Image'.

You have to make sure that the normal Map layer in the Shader Tree is not in the same mask as the item you are trying to bake from.

TIP: Depending on the type of geometry, and the resolution of the baked image, UV border seams may be visible. These can be eliminated by the using the 'Bake UV Border Size' option of the Preferences (menu bar 'System > Preferences', listed under Display > Rendering section). In order to take advantage of this functionality, users *must* have an active 'Alpha' render output as well, as modo uses the information of the alpha to compute the border offset.

Frequently Asked Questions

Have a problem that's perplexing you with modo? It's likely others have had the same problem, take a look through the Frequently Asked Questions chapter for a possible answer.

Documentation--

Why does Internet Explorer give me an error every time I open the documentation?

> For security purposes, Internet Explorer disables the execution of scripts that are served locally (i.e. from your local system, not the internet). If you are viewing the documentation from your local drive, you will need to 'Allow Blocked Content' each time the documentation is accessed. Advanced users may wish the enable the "Allow Active Content to Run in Files on My Computer' setting in the Advanced Internet Options dialog.

I can't see the inline help videos, what can I do?

> The documentation relies on Quicktime to play back embedded videos, therefore it is required that Quicktime be installed. Quicktime can be obtained from http://www.apple.com/quicktime/

File I/O--

Why wont my decompressed file open under Mac OSX?

> I could be because the folder that the file is residing in has a an .LXO extension in its name (i.e. a folder named "modomodel.lxo"). If this is the case, renaming the folder to something else (or just dropping the .lxo part) will resolve the issue. We're looking for other resolutions in a future update.

Interface--

How can I customize my interface?

> Users can customize their interface easily, but before doing so, understanding the functions of the 'Thumb' and 'Widget' are essential. Those functions are covered on the 'Layout Controls' page of the documentation. Additionally, users can follow a walk through of building a custom interface on the 'Customizing Your Layout' page.

How can I map my own keyboard shortcuts?

> Keyboard shortcuts can be assigned within the Input Editor found in the menu bar under "System > Input Editor". An easier way to accomplish the same task is to utilize the 'Command History' viewport. To map a command to a custom key, users can simply fire the command in modo, by either invoking the command from the menu bar or within the modo UI itself, then within the 'Command History' viewport, locate the appropriate command and RMB+click and choose the 'Map Command to Key' option from the contextual menu. This opens the 'Key Assignment' dialog where users can define the particular key they wish to use. Press 'OK' to set the new definition. Users should note that this new value is saved in the modo

configuration file (modoX01.cfg). Removal of this file will also remove custom keyboard shortcuts.

How can I change my viewport colors?

Viewport colors are changes first within the Preferences section "System > Preferences" under the "Display > Colors" section. Users can modify the colors to their liking and then use the 'Save to Scheme' button and assign a new name to the color scheme. To actually *apply* the scheme to a viewport, users need to LMB+click within the target viewport to select it. Selected viewports are denoted by the orange highlighting of the Thumb icon. Then in the menu bar, select the command "View > Viewport Color Scheme" and choose the named scheme defined earlier in the Preferences.

How can I change the default lights in the 3D viewports for modeling?

modo offers users the ability to adjust the default lighting in 3D viewports using the options found in the menu bar under "View > Edit Viewport Lights". Once the function is active, users can LMB+drag in the viewport to change the current lights position. Additional properties are available in the Tool properties form for 'Color' and 'Intensity'. The option 'Current' defines the particular light being edited. Users can use the commands "View > Add Viewport Light" and "View > Remove Viewport Light" to modify the number of lights visible.

Why does my 'X' keyboard shortcut not work?

Certain keyboard shortcuts require modo to be in a specific state or mode, such as the timeline shortcuts require modo to be in "items' selection mode or the brush tip pie menu requires a paint tool to be active. Make sure the proper tools is selected, if necessary, and the correct selection mode is active before applying the keyboard shortcut.

After I've changed an option, how can I return its setting to the default value/state?

Each animateable attribute value is preceded by a small dot or circle that defines its keyframe status. Users can RMB+click on the option to open the contextual menu and select the 'Remove All' option returning any attribute back to its default state.

Modeling--

What is a bounding box?

The bounding box is the virtual rectangular volume encompassing all the geometry in any Mesh Item layer, defined by the maximum and minimum vertex values on all three axes.

What size should I make my object?

In most cases models are best created at real-world scale. The way lighting attenuates in a scene is all based off of real world values. To get the most realistic results building to the correct size will produce the best results, However, there are situations where working to scale will have benefits, such as when modeling anything smaller than about a millimeter in size should be scaled up, or anything larger than a few city blocks should be scaled down, items getting beyond these sizes push the boundaries of modo's accuracy. You defiantly don't want to be modeling sub-atomic particles or entire planets to actual scale.

How do I insert an image into my viewport for modeling?

Users can insert custom background images into a viewport using the Backdrop item added in the Item list viewport. Within the Background items properties, specific images can be applied and the settings determine how it is displayed in the 3D viewport. For more, please reference the 'Modeling with Images' page of the documentation.

How do I change the default behavior for a tool?

Introduced in modo 501, tools will remember most user settings between uses, where the last used settings become the defaults for the next application. If users wishes to change the system default values for a tool this can be done within the 'Tool Pipe' viewport. With the tool active and the appropriate new settings defined, RMB+click over the command name in the Tool Pipe and choose the 'Save Preset' option. Keep the default name value in the resulting save dialog to overwrite the default behavior (Make sure only the tool you wish to modify is active, having additional Action Center and Falloff settings will be included if also active).

Why do my OBJ files import as all triangles/why can't I edit my imported OBJ file?

modo has an option to import OBJ files as a 'Static Mesh', which makes it much easier to deal with multi-million polygon objects. This option limits the modifications available to only Items mode transforms. However, the process also triangulates the geometry when generating the 'Static Mesh' from the imported geometry. Users that wish to further modify imported OBJ files should make sure this option is disabled (the default state) before importing any OBJ format geometry.

When I duplicate an object it changes size, Why?

Understanding the difference between modo's component and Item selection modes is very important to successfully using modo. Users that aren't sure of the differences can read more about it here. Basically, what often happens, users will

scale an item layer accidentally while in Items mode, not realizing it. Subsequent duplicates will default to the zero base transform values, changing the size of the duplicated objects. If this happens, users can use the "Freeze > Scale" option in the items properties panel before duplicating the item layer.

Why are my Replicators positioned away from the Point Source when I assign the Prototype?

Replicators base their position information in relation to the Prototype geometry layers 'Center' position. If the geometry was created at an arbitrary position away from the Center, users can use the 'Center to Bounding Box' commands to easily reposition the Center, or simply Center the geometry using the appropriate command.

How do I get files from SketchUp into modo?

A SketchUp (.SKP) importer was developed for modo 302, but is no longer compatible with newer .SKP files. Starting with modo 501 we have moved to a COLLADA importer for interfacing with SketchUp integrated with the 'Open' and 'Save' functions.

Accuracy--

How do I find out my models dimensions?

Absolute dimensions for the overall bounding volume of any selection can be found using the 'Dimensions Tool' command found in the menu bar under "View > Dimensions Tool". Invoking the command will display the values over a ruler display directly in the 3D viewport. To confine the display to a specific section, simply select it, using the appropriate selection mode. When nothing is selected, the Dimension will default to the overall bounding box volume of all visible items.

How do I scale my model to a certain size?

Users can scale objects, based on the selections bounding box volume using the 'Absolute' palette found at the bottom of the 'Snaps' tool palette. Use the 'Grab Size' command to find the current size, then enter the new value in the appropriate input field and press enter to apply the new dimension.

How can I snap items in a scene?

modo has a snapping function that is based on the snapping of tool handles. Proper snapping requires an understanding of Action Center settings. The snapping functionality is covered on the 'Snapping' page under Precision in the modo Toolbox section.

Alignment and Positioning--

How can I accurately align my object onto the surface of another object?

For simple geometry in the same item layer as the target surface, users can use the Tack tool with the 'Move Connected Vertices' command enabled. For more complex object, it will be easier to use the 'Drop Action' functionality found in the Animate interface tab. Both the source and target need to be separate item layers. Then users can set the 'Center' of the source item to the appropriate position in relation to the geometry; typically using the command "Edit > Center to Bounding Box > Bottom". Next, set the 'Drop Action' function to the 'Place and Align' option. Finally, with no tool active, simply LMB+click on the source mesh and drag it over the target surface and release the mouse button setting the object to the surface.

I can't get a background constraint to work, how should I set it up?

Background constraints foremost need some background geometry. The target geometry needs to be in the foreground, meaning in the Items List the layer is visible and selected. If you're not familiar with foreground, background and visibility states, please reference the Items List page of the documentation. To enable Background Constraint, press the 'F11' key and choose 'Background' from the Constraint Mode options. By default, the Background constraint only constrains tool handles. To constrain geometry as well, on the Tool properties form choose either 'Screen Axis' or 'Vector' options under Geometry Constraint. More about these options can be found in the 'Constrain to Background' section on the Snapping page of the documentation.

How can I specify specific positions for elements?

This is dependant on the particular selection mode. While in Items selection mode, users can define the position of individual layers in the Items properties panel, this sets the Center position to the specific location with the associated geometry remaining in proper relation to the Center element. Individual vertices can be explicitly positioned in the 'Info and Statistics' viewport. For Edges and Polygons, transforms are applied as an offset from the present position using the Move tool, to assign a specific positon, simply 'Center' the selection prior to applying the offset.

How can I tell what size my object/selection is?

See "How do I find out my models dimensions?"

Animation

My Item isn't rotating/scaling from where I want it to?

This is likely because the Center isn't positioned correctly in relation to the geometry. Users can use the 'Center to Bounding Box' commands to easily reposition the Center, or Simply Center the geometry using the appropriate command.

Why don't my deformers show up?

Users can define how [MDD, Morph and Weight] Deformers display on a per viewport basis. Be default most viewports are designated to *not* display animated deformers (only the 'Animation' and 'Rendering' interface tabs do). Users can open the options panel for a particular viewport by hovering the mouse over the target viewport and press 'O' on the keyboard. On the viewport options popover, users need to set 'Enable Deformers'. If they still aren't working after doing so, make sure that users are working in 'Items' selection mode when adjusting the deformer values.

Shade/Rendering--

Why do my new scenes suddenly appear washed out in 501?

The default 'Gamma' option has been changed from '1.6' to '2.2'. All new render outputs will utilize this settings. The higher gamme will require images invert this value (de-gamma), or otherwise look light or washed out. This is covered on the Image Map page of the documentation.

I've applied some preset materials to an object, but when I duplicate it, the clone is just gray, why does this happen?

Material Presets default to limiting their application to a single mesh item, by way of an Item Mask. Disabling the Item mode masking function in the material Group item will alleviate this problem. Simply change the item name listed under the 'Item' to '(all)' to resolve the issue, allowing the textures to globally affect the scene.

Why do my surfaces turn black when I enable 'Conserve Energy'?

The 'Conserve Energy' option of the 'Material Item optimizes the 'Diffuse Amount' setting based on the amount of reflectivity. If the amount of reflectivity is at 100%, then Diffuse amount is reduced to 0%, producing a black surface. If reflections are minimal in a scene or very dark, the resulting surface will likely appear black. To have a colored reflective surface, try adjusting the 'Reflection Color' amount instead of 'Diffuse Color'.

How can I get rid of those terrible splotches in my render?

Splotchiness is generally an indicator that the number of Irradiance Rays is

too small to accurately depict the surrounding environment when calculating Irradiance Caching. Try increasing the value to better capture the surrounding lighting. Lowering the Irradiance Rate and increasing the Irradiance Ratio can also help in certain scenes.

Why is my 'X' surface so noisy in a final render?

Noise that is confined to a specific area, be it a shadow, a reflective surface or a transparent object are typically the result of the number of samples related to the particular option. modo offers a extremely flexible shading solution providing user the ability to adjust the number of rays used to calculate individual effects. The default values will provide speedy renders but on occasion don't provide enough sampling latitude to produce smooth results. Lights that generate soft shadows individually have a 'Samples' setting that when raised increases overall rendering quality. The Material item offers a 'Samples' setting when rendering soft reflection as does the Refraction Roughness setting. Volumetric light do as well. When trying to reduce noise in a scene, typically there is some investigation required to determine the likely culprit. Noise related directly to shading can be reduced by reducing the Shading Rate and noise produced by geometry (such as using replicators of fur) can be reduced by increasing the amount of Antialiasing. Keep in mind that when increasing these settings, render times will almost always increase as well.

Where did my Bump Maps go/why do Bump Maps render differently in 501?

The way in which modo calculates Bump Maps has been completely overhauled for modo 501. Instead of using the now deprecated 0%-100% input value, users define bump strength using specific offset values. However, opening older scenes, the old way needs to be translated to the new way, and well, depending on the scale of the scene this may or may not have worked well. If Bump Maps are too strong or seem to have disappeared, users will need to adjust the 'Bump Amplitude' value to an appropriate level manually to obtain optimal results.

Why, when I add a Replicator, do my Irradiance pre pass times go through the roof?

When rendering lots of very tiny detail using Global Illumination, the default calculation method, 'Irradiance Caching' (IC) will calculate the irradiance value every time a surface normal changes. When there is a lot of detail due to Replicators, Fur or Displacements, the expensive IC calculation will take much longer to generate. Users should assign affected surfaces to use the less expensive Global Illumination option 'Monte Carlo'. To do this, create a 'Selection Set' of all the affected geometry. Add a Material Group item to the Shader tree and place it above the 'Base Shader' item. In the Material Group properties, set the 'Polygons Tag Type' to 'Selection Set' and under 'Polygon Tag' select the previously created

Selection Set. Next, add a Shader to new Material Group "Add Layer > Special > Shader" and in the Shaders Properties, set the 'Indirect Illumination Type' to 'Monte Carlo'. Fur by default will utilize Monte Carlo shading. The Fur Item itself as an option 'Use Irradiance Cache' that disables this behavior.

My scene won't render, what can I do?

The number one reason modo fails to render a scene is the user has run out of available memory, especially in 32 bit installations that are limited to 2Gb of ram. Often times this is due to simply too many, often unnecessary polygons. To reduce the memory overhead, users can reduce the number of subdivisions for Subdivision Surfaces geometry in the Item properties panel for certain high res objects. If 'Adaptive Subdivision' is enabled, try to disable that and set the subdivision level manually for each item. When using Displacement, users can set the 'Displacement as Bump' option enabled in the Render items under Settings, and reduce the number of subdivisions. The resulting image will have far fewer polygons but be visually similar to the higher level render. Also Fur items offers a 'Fur Frustrum Culling' option the removes all fur outside of the camera view. If there are no reflective surfaces in the scene this can be a huge memory saver. Other options include rendering multiple 'Limited Region' sections and assembling them in an image editor.

Index

A

Absolute Morph Map 455, 458
Absolute Scaling 505
Absorption Amount 966
Absorption Color 966
Absorption Distance 956, 959
Action Centers 418
  Automatic 419
  Element 420
  Local 420
  Origin 420
  Parent 420
  Pivot 421
  Pivot Center Parent Axis 421
  Screen 420
  Selection 419
  Selection Border 419
  Selection Center Auto Axis 419
Action Center Types 419
Actor 919
Adaptive Sampling 1546
Adaptive Subdivision 1640
Add
  Add Image 522
  Add Image Sequence 523
  Add Texture Commands 525
  Add User Channel 759
  Channel Modifier 769
  Color Texture 525
  Displacement Texture 525
  Fur Texture 526
  Mask Texture 525
Add Loop 406
Advanced Color Navigation 62
Advanced Handles 504
Advanced OpenGL 94
Airbrush 426
Alembic 223
Align
  Align Polygon 416
  Align to View 148
  Align Vectices 398
  Align Work Plane to Selection 510
Allen Hastings 952, 1631, 1649
Alpha 1734
Alpha Channel 1510
Alpha Channels 1618
Alpha Type 1567
Alternate Buttons 60
Ambient Intensity 1643
Ambient Light 1643
Ambient Occlusion 1561
Amplitude Ratio 1575
Anaglyph 1722
Angle of View 1715
Animate 190
Animated Booleans 1705
Animation Modifiers 769
  Channel Color HSV 770
  Channel Color Kelvin 770
  Channel Noise 771
  Channel Oscillator 771
  Dual Joint Planar IK 780
  Dynamic Parenting 781
  Linear Blend 783
  Logic 772
  Math 774
  Math Multiple 774
  Matrix Blend 775
  Matrix Compose 776
  Matrix Vector 776
  Measure Angle 778
  Measure Distance 777

Revolve 784
Simple Kinematics 785
Trigonometry 775
Vector 779
Vector Maths 779
Animation Setup Tools 786
Align 791
Center Position 789
Center Rotation 789
Center to Pivot 793
Extract 788
Hierarchy 786
Insert 788
Insert at Parent 788
Item Position 788
Match 790
Match and Align 788
Match Rotation 789
Match Scale 790
Mirror 793
Parent/Unparent 787
Pivot Position 789
Pivot Rotation 789
Position 792
Rotation 792
Set 791
To Work Plane Position 791
To Work Plane Rotation 792
Anisotropy 949
Antialiasing 1635
Antialiasing Filter 1636
Application Menu Bar 179
Apply 73
Applying Constraints 761
Applying Presets 165
Apply Morph 466
Apply to Sub-Group 944
Array 363
Assign Up Vector 781
Autodesk 3DStudio 224
Autodesk dxf 224
Autodesk fbx 224
Autofocus 55, 1719
Auto Visibility 111
Average 790
Axis Drill 388
Axis Rotate 277
Axis Slice 381

B

Backdrop Item 513
Background to Morph 467
Baking 1775
Bend 320
Bezier Clone 355
Bezier Extrude 348
Binding 797
Bitmap Image Formats 225
Blobs 1700
Bloom 1562, 1757
Bloom Radius 1563, 1757
Bloom Threshold 1562, 1757
Blur Length 1721
Blur Offset 1721
Boolean Buttons 62
Booleans 387
Boundary Edges 444
Bridge 344
Brush Tips 526
Hard Brush 527
Image Brush 529
Inks 531
Procedural Brush 528

Smooth Brush 526
Sphere Brush 529
Text Brush 530
Bulge 335
Button Popups 59
Buttons 59

C

Camera 1710
Camera Effects 1718
Camera Safe Overlay 1724
Camera View 1712
Caustic 1697
Cellular 1573
Cell Width 1574
Cel Shading 94
Center Commands 284
Center Selection Mode 439
Channels 756
Characters 919
Clear Image 612
Clipping 964
Clipping Plane 1724
Clone 362
COLLADA 224
Collapse 399, 408, 417
Color Maps 452, 457
Color Picker 195
Color Theme Editor 618
Common Liquids 958
Constrain Mode 497
Constrain Modes 499
Constrain to Background 501
Constrain to Primitive 502
Constraints 763
Direction 764
Distance 764
Edge Normal 765
Edge Position 765
Geometry Constraints 765
Intersection Constraints Types 767
Path Normal 766
Path Position 766
Polygon Normal 765
Polygon Position 765
Position 764
Rotation 764
Scale 764
Vertex Normal 765
Vertex Position 765
Context Help 69
Contextual Popups 64
Contour Color 1563
Contours 1563
Contour Width 1563
Convergence Distance 1723
Convert Selection 442
Convert to Quadrangles 416
Convert to Vector Displacement 613
Copy 392
Copy to Morph 615
Crease Threshold 1564
Curve Clone 352
Curve Extrude 345
Curve Slice 377
Cut 391
Cycle Component Selections 434
Cylinder 424

D

Defining Selection Sets 446
Deformers 874

Bend Effector 877
Falloffs 902
General Influence 889
Item Influence 891
Lag Effector 880
Magnet Effector 882
MDD Influence 893
Morph Influence 895
Slack Effector 885
Soft Body Influence 897
Vortex Effector 887
Deforming Geometry 286
Delete/Remove 392
Delete Selection Set 449
Density 967
Depth 1561
Depth of Field 1637, 1719
Diffuse 1696
Dispersion 960
Displacement as Bump 1642
Displacement Rate 1641
Displacement Ratio 1642
Dissolve Amount 961
Drag and Drop 68
Drag Snap 503
Drag Snap Rigid 503
Drawing
Arc Tool 260
Bezier 258
Curve 257
Pen 254
Sketch 259
Dropping Selections 442
Duplicate Commands 374
Duplicating Geometry 338

E

Edge Bevel 401
Edge Extend 402
Edge Extrude 402
Edges 401
Edge Selection Mode 435
Edge Slice 378
Edge Slide 403
Edge Weighting 1720
Edge Weight Tool 473
Edit Commands 211
Edit Commands - Cut/Copy/Paste 391
Edit Commands - Delete/Remove 392
Edit Geometry 376
Edit Selection Set 448
Effects 1707
Element 427
Element Move 286
Encapsulated PostScript 225
Environment Material 1655
Environments 1653
Environment Shading Rate 1636
Expand UV Seams 612
Explicit 505

F

Falloffs 421
Capsule Falloff 657
Falloffs Definition 656
Linear Falloff 659
Radial Falloff 661
Spline Falloff 690
Falloff Types 423
File Commands 209
Filler Alpha 1574

Fill Quads 409
Film Back 1716
Film Fit 1718
Film Offset X/Y 1717
Film Width/Height 1717
Final Color 1562
Flare 338
Flatten Layers 613
Flex 289
Flip 416
Floating a Window 69
Focal Length 1715
Focus Distance 1714, 1719
FPS 1747
Frame 1629
Frames 1747
Frame Step 1748
Freezing 723
Frequencies 1575
Frequency Ratio 1575
Front Weighting 962
F-Stop 1719
FX Item 1707

G

Gemstones 956
Geometry Snap Mode 498
Geometry to Brush 613
Global Illumination 1643
Gooch Tone Shading 94
Gradient
  Basic 1514
  Custom Material 1519
  Fur 1519
  Gradient Editor 1531
  Particle Effects 1522
  Sample Parameters 1522
  Shader Control 1524
  Special Effects 1526
  Surface Gradient 1514
  Surface Parameters 1526
  Surface Particle Generation 1529
  Surface Shading 1529
  Volumetric Effects 1530
Grid Snap Mode 498
Group Masks 1618
Grow Quads 409
Grow Selection 442
Guide Snap 499

H

Hair 582
  Comb 585
  Curl 587
  Hair Guides 591
  Move 586
  Push 583
  Root Puff 590
  Separate 588
  Set Guide Color 591
  Smooth 584
  Stretch 589
  Tip Edit Mode 590
Hauling 68
Heat Binding 798
Hierarchies 729
Hierarchy 739
HPGL Plotter File 225

I

Image 430
  Image Adjustment Commands 612

Image Based Lighting 1657
Image Processing 1757
Implicit Selections for Component Modes 440
Import/Export 222
Import Reference 220
Indirect Bounces 1647
Indirect LOD 1643
Indirect Range 1648
Indirect Rays 1646
Inks
  Image Ink 532
  Parametric Ink 534
  Randomized Ink 535
Inline editing 61
Input Mapping 72
Inset 413
Instance Array 365
Instances 717
Interface Layout Tabs 180
Interocular Distance 1723
Inverse Kinematics 902
  Full Body IK 908
Invert Image 612
Invert Selection 443
Iris Blades 1719
Iris Rotation 1720
Irradiance Caching 1649
Irradiance Rays 1649
Item Masks 1619
Items 437
  Add Item 651
  Add Points 651
  Add Weight Container 650
  Backdrop Item 695
  Bend Effector 639
  Blob Item 680
  Camera Item 632
  Converting Items 629
  Creating Items 626
  Duplicating Items 628
  Fractal Gasket 672
  Gear Item 674
  Ground Plane 684
  Group Locator Item 664
  Instance Item 693
  Item Reference 630
  Lag Effector 641
  Locator Item 663
  Magnet Effector 644
  Mesh Item 634
  Morph Influence 646
  Particle Cloud 666
  Particle Generator 669
  RealFlow Particles 671
  Remove Item 651
  Remove Points 651
  Removing Items 629
  Renaming Items 627
  Render Boolean 687
  Replicator Item 652
  RPC Mesh Item 676
  Scene Item 630
  Static Mesh Item 691
  Weight Container 650
Item Selections 60

J

Jitter 300
Join 398, 408
Join Averaged 398, 408
Julienne 382

K

Keyboard Shortcuts 73
Key Toggles 71

L

Lasso 429
Lasso Select 441
Layer Blend Modes 1623
Layer Effect 937
Layer Effects 1620
Layering Surfaces 1619
Layer Masks 1618
Layers 1620
Layout Controls 204
Layout Editor 189
Lens Distortion 1715
Lighting 1660
Light Material 1694
Light Types 1660
  Area Light 1676
  Cylinder Light 1682
  Directional Light 1660
  Dome Light 1680
  Photometric Light 1686
  Point Light 1671
  Portal 1692
  Spot Light 1666
LightWave Object 222
Linear 423
List Columns 67
Lists viewport 454
Locking 89
Loop Slice 379
Luminosity 963
Luminous Color 963
Luminous Intensity 963

M

Managing Vertex Maps 455
Masking Surfaces 1618
Material Selection Mode 438
Material Tags 924
Matte Clipping 964
Matte Color 964
Maximum Depth 962
McNeel Rhino 223
Menu Bar 58
Merge 400, 416
Merge Tool 396
Mesh Cleanup 393
Micropoly Displacement 1641
Minimum Edge Length 1642
Minimum Importance 1639
Mirror 342
Model 196
Model Quad 186
Modes 435
Modifiers 769
modo modes 181
Morph Map 451
Morph Maps 456
Morph Tool 474
Motion Blur 1637, 1720
Move 271

N

Navigation 88
Nesting & Hierarchies 61
Network View 1765
New Sequence 130

New Vertex 397
Noise 428
Normalization 754

O

Occlusion Range 1562
Occlusion Rays 1561
Other Maps 453

P

Paint 197
Painting 520
  Airbrush 565
  Blur 578
  Clone 570
  Customizing Brushes 521
  Eraser 569
  Fill 573
  Lasso 581
  Line 572
  Linear Gradient 576
  Paintbrush 567
  Particle Paint Tools 521
  Radial Gradient 574
  Sharpen 580
  Smudge 577
Painting Tools 564
Paint Select 441
Paint Tab 188
Parent/Child Hierarchies 112
Parenting
  3D Viewport Parenting Actions 741
Particle Density 1708
Particle Dissolve 1708
Particle Maps 453
Particle Size 1708
Particle Tools 598
  Dissolve 611
  Flatten 608
  Move 603
  Paint 598
  Pinch 604
  Size 609
  Smooth 601
  Smudge 602
  Spin 607
  Tangent Pinch 605
Part Tags 926
Paste 392
Paste Tool 373
Patch Curves 266
Path 429
Pen Clone 358
Pen Extrude 350
Pen Slice 379
Photons 1651
Pie Menus 66
Pivot Selection Mode 439
Pivot Snap Mode 499
Planar Rotate Snap 503
Planar Rotate Snap Axis 504
Polygon 924
Polygon Bevel 410
Polygon Extrude 411
Polygon Reduce 414
Polygons 409
Polygon Selection Mode 436
Polygon Set Material 929
Polygon Set Part 930
Polygon Set Smoothing Group 930
Polygon Tags 924
Popover Window 67

Poses 921
Pose Tool 799
Preferences - Input 213
Primitives 238
Capsule 245
Cone 242
Cube 238
Cylinder 244
Ellipsoid 249
Sphere 240
Teapot 253
Torus 247
Tube 250
Project Directories 219
Projection 1611
Box 1612
Cubic 1612
Cylindrical 1611
Front Projection 1612
Implicit UV 1613
Light Probe 1613
Planar 1611
Projection Camera 1613
Solid 1611
Spherical 1612
UV Map 1613
Push 327

Q

Quantize 302

R

Radial 425
Radial Array 369
Radial Sweep 339
Ray GL 91
Ray Threshold 1639
Ray Tracing 964, 1638
recoil 805
Anchor 855
Baking Dynamics for Rendering 871
Box Constraint 836
Constraints 827
Dynamic Item 815
Falloff Box 867
Falloff Cylinder 868
Falloff Plane 870
Falloff Spherical 865
Force Drag 859
Force Linear 856
Force Radial 857
Force Turbulence 860
Force Vortex 862
Hinge Constraint 844
Make Constraints 850
Motor 851
Pin Constraint 829
Plane Constraint 833
Point Constraint 831
recoil Falloffs 865
recoil Forces 856
Slice Constraint 842
Slide Hinge Constraint 847
Soft Body Item 822
Solver 811
Spring 853
Universal Constraint 839
Usage Overview 805
Wakeup Triggers 873
Refine Bucket Borders 1637
Refinement Shading Rate 1636
Refinement Threshold 1636
Reflection 95

Reflection Depth 1638
Refraction Depth 1638
Refraction Rays 960
Refraction Roughness 960
Refractive Index 959
Refractive Index Table 956
Relax Mesh 614
Remove 399, 408, 417
Render 191, 198
Render Boolean 1704
Render Browser 1764
Render Display 1755
Rendering 1727
  Alpha 1734
  Ambient Occlusion 1733
  Compositing Outputs 1744
  Depth 1727
  Diffuse Coefficient 1732
  Diffuse Shading (Direct) 1735
  Diffuse Shading (Indirect) 1736
  Diffuse Shading (Total) 1737
  Diffuse Shading (Unshadowed) 1736
  dPdu Vector 1728
  Dpdv Vector 1728
  Final Color 1735
  Geometric Normal 1729
  IC Positions 1731
  IC Values 1731
  Illumination (Direct) 1738
  Illumination (Indirect) 1738
  Illumination (Total) 1737
  Illumination (Unshadowed) 1739
  Luminous Shading 1739
  Motion Vector 1740
  Reflection Coefficient 1732
  Reflection Occlusion 1734
  Reflection Shading 1741
  Render Animation 1747
  Render Commands 1744
  Render Current View 1746
  Render Output Items 1727
  Render Passes 1749
  Render Turntable 1747
  Segment ID 1729
  Shading Normal 1740
  Shadow Density 1743
  Specular Coefficient 1733
  Specular Shading 1741
  Subsurface Shading 1742
  Surface ID 1730
  Transparent Shading 1742
  UV Coordinates 1730
Render Item 1628
Render Region Tool 1634
Replica Array 366
Replicators 163, 652, 719
Rotate 274
RPC Browser Window 679

S

Scale 280
Scale Absolute 504
Scattering Amount 965
Scattering Color 966
Scattering Distance 962
Screen 426
Sculpt 328
Sculpting 518
  Attenuate 561
  Carve 542
  Center Pinch 556
  Emboss 560
  Flatten 545
  Fold 548

Image Base Sculpting 519
Inflate 550
Lock Borders 564
Mask 563
Mesh Sculpting 519
Move 555
Multi-Res Sculpting 519
Push 538
Sculpting Commands 613
Smooth 541
Smudge 553
Spin 559
Tangent Pinch 557
Sculpting Tools 537
SDS Subdivide 389
Search Filter 63
Select Between 443
Select Close Loop 444
Select Colinear 445
Select Connected 443
Select Coplanar 445
Select Diagonal 445
Selection 431
Selection Modifiers 442
Selections 71, 434
Selection Sets 446
Selection Styles 441
Select Loop 443
Select More/Less 443
Select Ring 444
Set Position 397
Set Reference System 111
Set Resolution 613
Setup Editor 189
Set Vertex Normals 474
Shaded 93
Shaded Texture 93
Shader Tree 934
Cel Edges 967
Cellular Procedural 1572
Cel Material 969
Checkerboard Procedural 1578
Constant 1512
Dots Procedural 1581
Enhance:modo Textures 984
Fur Material 1542
Gradient 1513
Grid Procedural 1585
Hair Material 971
Halftone Material 976
Image Map 1507
Matcap Shader 1606
Material 945
Material Group 942
Material Trans 955
Noise Procedural 1589
Occlusion 1534
Process 1539
Render Outputs 1556
Ripple Procedural 1595
RPC Texture 1598
Shader 1564
Skin Material 979
Surface Generator 1569
Texture Locator 1609
Weave Procedural 1599
Weight Map Texture 1540
Wood Procedural 1602
Shear 296
Shift 412
Show Vertex Normals 475
Shrink Selection 443
Simplified 196

Size Absolute 504
Skeleton Tool 793
Sketch Extrude 341
Slice 376
Smooth 299
Smooth Positions 1642
Smooth Shift 412
Snapping 494
Snapping Mode 496
Snapping Modes 498
Snapping Options 496
Snaps Palette 503
Soft Drag 295
Soft Move 292
Soft Selection 432
Soft Select Move 305
Soft Select Rotate 308
Soft Select Scale 310
Soft Select Transform 313
Solid 92
Solid Drill 388
Solid Sketch 261
Spectral Samples 1639
Specular 1696
Spikey 413
Spin Edge 407
Spin Quads 415
Split 399, 408, 415
Statistics 1764
Stereo Camera Volume 1725
Stereo Composite 1722
Stereo Eyes 1722
Stereoscopic 1721
Subdivision Rate 1640
Subsurface 1697
Subsurface Amount 961
Subsurface Color 962
Subsurface Scattering 961, 1648
Surface Clipping 964
Surface Generators 721
Swirl 325
Symmetry 493
Symmetry Tool 304

T

Tablet Nozzle 535
Tabs 58
Taper 331
Temperature 1697
Text 263
Texture 93
Texture Layers
- Agate 1042
- Art Deco 1108
- Basket 1231
- Bath Tile 1236
- Blast 1419
- Blister 1111
- Box 985
- Bozo 1045
- Branches 1114
- Bricks 1241
- Camo 1183
- Caustic 1117
- Cellular 1121
- Checks 1246
- Cheesy 1125
- Circular 989
- Concrete 1129
- Coriolis 1423
- Cornerless 1250
- Corners 993

Counter 1 1373
Counter 2 1377
Crackle 1132
Cruddy 1049
Crumpled 1187
Cubes 1254
Cubic 997
Dash Line 1258
Dented 1052
Diamond Deck 1263
Dimples 1001
Dino Skin 1191
Dirt 1135
Disease 1195
Disturbed 1139
Drip Drop 1469
Easy Gradient 1415
Easy Wood 1142
Electric 1146
Etched 1055
FBM 1058
Fire 1150
Fire Wall 1153
Fish Scales 1267
Flare 1427
Flow Bozo 1062
Frog Skin 1199
Gas Giant 1431
Grainy Wood 1203
Granite 1066
Grid 1005
Hard Wood 1156
Hexagonal Tiles 1271
Hurricane 1434
Hybrid 1070
Iris 1009
Lattice 1 1275
Lattice 2 1279
Lattice 3 1283
Leather 1208
Linear 1013
Lump 1074
Marble Noise 1077
Marble Vein 1081
Membrane 1159
Minky 1163
Monster 1213
Mosaic 1287
Nurnies 1438
Octagonal Tile 1292
Parquet 1297
Pastella 1217
Paving 1302
Peel 1385
Peened 1221
Plaid 1307
Planet 1442
Planet Clouds 1446
Planks 1311
Plates 1390
Polygon 1017
Radial 1021
Rain 1472
Regional HSV 1417
Ribs 1316
Ring 1025
Rings 1450
Ripples 1476
Rivet Rust 1395
Rivets 1400
RndLinear (Random Linear) 1029
Rounded Tile 1321
Rust 1405
Scar 1085
Scatter 1166
Scratches 1225

Scruffed 1089
Shingles 1326
Sin Blob 1169
Smear 1410
Spiral 1033
Spots 1330
Stamped 1335
Star 1037
Star Field 1454
Strata 1092
Stucco 1096
Surf 1479
Swirl 1457
Tacos 1340
TarTan 1345
Terra 1461
Tiler 1349
Tri Checks 1354
Tri Hexes 1358
Tri Tile 1362
UV LEDs 1381
Vector Bozo 1100
Veins 1172
Wall 1366
Waveforms 1489
Waves 1482
Windows 1465
Windy Waves 1485
Wires 1175
Worm Vein 1179
Wrapped FBM 1104
Theme Editor 195
Thicken 413
Thumb 205
Toggling Layers 938
Tone Mapping 1765
Toolbox 180
Tool Falloffs 582
Tool Handles 70
Tool Modifiers 418
Tool Properties 181
Tooltips 68
Topology Pen 383
Topo Tab 187
Track View 742
Transform 268
Transform Array 368
Transforming Geometry 267
Transform Item Tools 699
Array Items 714
Axis Rotate 706
Clone Items 711
Curve Clone Items 713
Move 702
Radial Array Items 715
Rotate 704
Scale 708
Scatter Clone Items 712
Transform 700
Uniform Scale 710
Transform Maps 453
Transform Modifiers 716
Transition Width 1575
Transparency Amount 959
Transparency Color 959
Transparent Materials 958
Triple 415
Twist 317

U

Uniform Scale 282
Unify 417
Use Selection Set 449

Utilities 506
Dimension Tool 508
Protractor Tool 507
Ruler Tool 506
UV Edit 199
UV Edit 102 200
UV Editor 188
UV Mapping 481
Align UV 487
Create UV Tool 481
Cut/Copy/Paste UVs 492
Delete UVs 492
Export UVs to EPS 493
Fit UV 488
Flip UV 490
Mirror UVs 489
Move and Sew UVs 490
Orient Pieces 491
Pack UVs 492
Relax UV 487
Rotate UV 490
Select Border 489
Sew UVs 490
UV Peeler 484
UV Unwrap 485
View Projection 484
UV Maps 451, 456, 475

V

Value Presets 64
Vertex Bevel 395
Vertex Color Map 472
Vertex Color Tool 472
Vertex Convexity Baking 471
Vertex Extrude 396
Vertex Illumination Baking 470
Vertex Map 92, 428, 450
Clamp 462
Clear 460
Copy 459
Create New Vertex Map 458
Cull 463
Delete 460
Erase 597
Jitter 462
Paintbrush 594
Paste 460
Quantize 462
Scale Down 597
Set Value 461, 597
Set Vertex 461
Set Vertex Map Name 459
Smooth 595
Smooth Vertex Map 465
Transfer Vertex Map 464
Unify 463
Vertex Map Math 467
Vertex Map Tools 594
Vertex Mapping 457
Vertex Map Types 450
Vertex Normal 452
Vertex Normals 457
Vertex Tool 264
Vertex Weight Tool 472
Vertices 395
Vertices Selection Mode 435
VideoScape 224
Viewport Maximize & Restore 69
Viewports
3D Sparse 194
3D Viewport 86, 182
Animation Viewports 742
Assembly Viewport 136
Channels 122

Channels Viewport 749
Clip Browser 128
Color Picker 139
Command History 134
Deformers Viewport 753
Display Viewport 144
Gradient Editor 151
Graph Editor 153
Groups Viewport 158
Image List 125
Info & Statistics 133
Item List 109
Preview 128
Preview Viewport 171
Properties 131
Render Display 176
Schematic Viewport 744
Shader Tree 116
Tool Pipe 176
UV Viewport 102
Vertex Map List 120
View Types 89
Visibility 89, 101
Volumes 1700
Volumetric 965
Volumetrics 1698
Vortex 322

W

Wavefront Object 222
Web3D Standard 224
Weight Container 912
Weighting 912
Adjust Weights 915
Erase 917
Isolate Weights 915
Paint 916
Scale Down 917
Set Value 918
Smooth 916
Smooth Selected 917
Weight Maps 450, 456
Widget 207
Wireframe 92
Work Plane 508
Work Plane Controls 510