FTV469077

Top Seven OSL Shaders for 3ds Max 2021

Steven Schain
Post-Production Supervisor / M & E Content Manager
4D Technologies

Learning Objectives

- Identify 3ds Max OSL shaders and third-party resources.
- Apply OSL Shaders at the scene and materials level in 3ds Max 2021.
- Customize OSL shaders to suit your needs.
- Identify 7 useful OSL shaders.

Description

Both mechanical and architectural design rely on visualization technologies to showcase designers' ideas. Increasingly, that relies on generating highly realistic materials and lighting. This class will look at my pick of the top 7 OSL shaders that can be applied to your scenes in 3ds Max 2021 to render photorealistic images and animations. Learn where to find and download OSL shaders that can be used in a visualization and how to customize your own OSL shaders. Use multiple shaders to create a unique material for your models. Get ahead of the competition by learning to create visually compelling, photorealistic renderings using these top 7 OSL shaders in a streamlined workflow.

Speaker

Steven Schain is the post-production supervisor for all CADLearning products from 4D Technologies, as well as the content development manager of CADLearning's Media & Entertainment and Design products for Autodesk, Inc. software, including 3ds Max, Maya, Inventor and Fusion 360. In 1998, Autodesk recognized Steven as one of only 16 Autodesk Training specialists worldwide. He has since contributed to Autodesk's certified courseware for 9 releases of 3ds Max, was a co-developer of Autodesk's ACI Program and 3ds Max's fundamental standards, and is currently an Autodesk Certified Instructor. As a premier Autodesk trainer, he has continued teaching end users, companies, and many others, including The Walt Disney Company, Guess, and the United States Army. As a 9-year veteran of Autodesk University and a featured speaker, Steven has taught classes ranging from creating particle fountains in 3ds Max, to classes on 3D printing and entrepreneurship.

sschain@cadlearning.com

Table of Contents

Introduction

As an architect or designer, you can use Autodesk's software to create buildable designs for anything from multi-floor high-rise buildings to complex mechanical assemblies. At any stage in the design process, you may need to share or present your design, and often, it is a requirement to render it using a method that provides photorealistic results. And, while Revit, Inventor, and other third-party design programs are capable of high-quality renderings of your design, you may decide to use 3ds Max because of its flexibility, and its ability to render both photorealistic and non-photorealistic images.

Incorporated into 3ds Max, Arnold is a rendering tool that provides an exceptional level of rendering quality and a true photographic view of your design. 3ds Max is the ideal environment for the creation of high-quality renderings of your designs. While 3ds Max can generate beautiful, realistic renderings, it's in the creation of the materials that often determines the look of the final rendered image.

Creating unique renderings can be done using standard materials and the default shaders. And, with the introduction of the Open Shading Language (OSL) shaders, you can use a much larger variety of shaders for rendering, many of which can be used to create realistic textures. Since OSL is an open-source language, there is a large community of shader developers and several repositories where additional shaders can be downloaded.

Introduced in 3ds Max 2020, OSL shaders display accurately in the viewport when using the realistic viewport setting, which provides an accurate match for the final Arnold render. In addition, OSL shaders are fully supported in the Quicksilver Hardware Renderer, which allows you to generate high-quality rendered animations when you're under a tight timeframe.

Introducing OSL

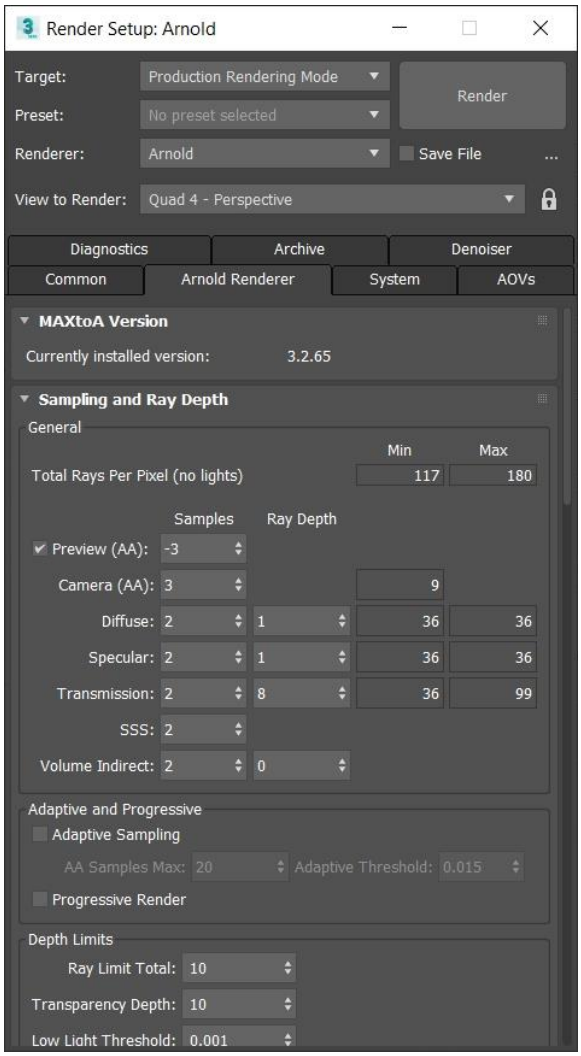
OSL, or Open Shading Language, is a programmable shading language for use in advanced renderers. The OSL language was created by Sony Pictures Imageworks as part of their in-house rendering pipeline for feature film animation production. OSL has been used on numerus feature films and has become a widely-used shading language in the visual effects industry. And, in 2017, Sony Pictures Imageworks won the Academy Award for Technical Achievement for the development of OSL.

OSL was released by Sony under the “New BSD” open source license, and the OSL documentation is released under the Creative Commons Attribution 3.0 Unsupported License. This allows you to use OSL shaders in both free and commercial productions, and you can modify the source code as you need to, with the stipulation that you keep the copyright notice.

OSL Basics

At its core, OSL is a shading language that is designed for use with today’s physically based renderers. With the introduction of Arnold 5.0, 3ds Max can make use of shaders written using OSL. One of the benefits of using OSL is that writing shaders for it is simpler than working with other methods like C++. Since OSL provides a high-level coding language, writing shaders is much easier than other programming methods, making it more accessible to individuals not familiar with advanced programming languages.

The OSL implementation within 3ds Max is available when using both the Arnold renderer and the Nitrous based Quicksilver renderer in 3ds Max 2020. One important note is that whichever version of 3ds Max you’re using, you must be using 3ds Max 2016 or newer with Arnold 5.0 or newer. You also want to make sure you have the latest version of the MAXtoA plug-in for 3ds Max. The version number can be found on the Arnold Renderer tab in the Render Setup dialog.

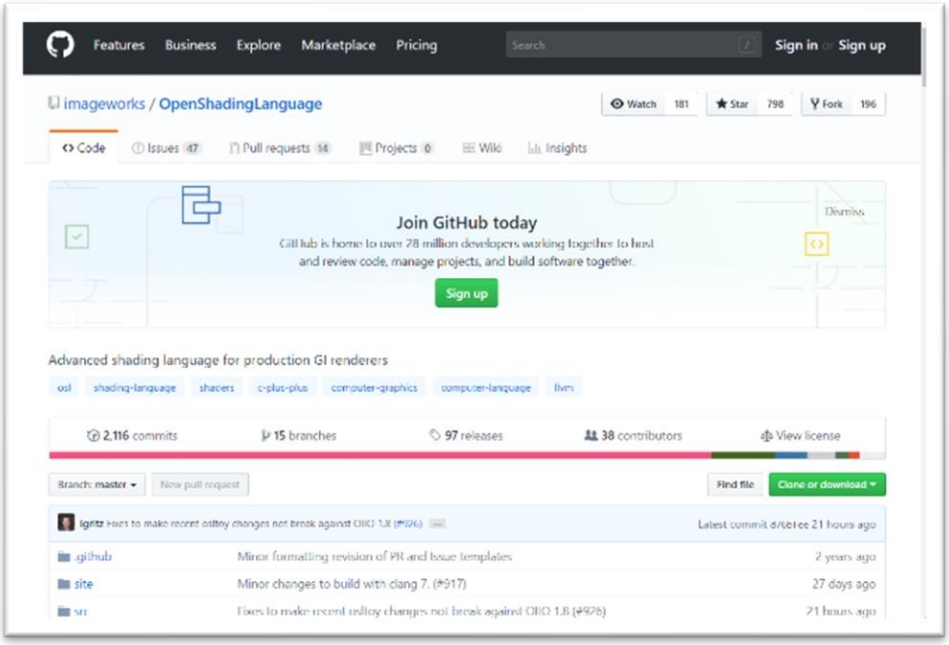


THE RENDER SETUP DIALOG SHOWING THE MAXTOA VERSION IN THE ARNOLD RENDERER ROLLOUT.

Discovering OSL Shaders

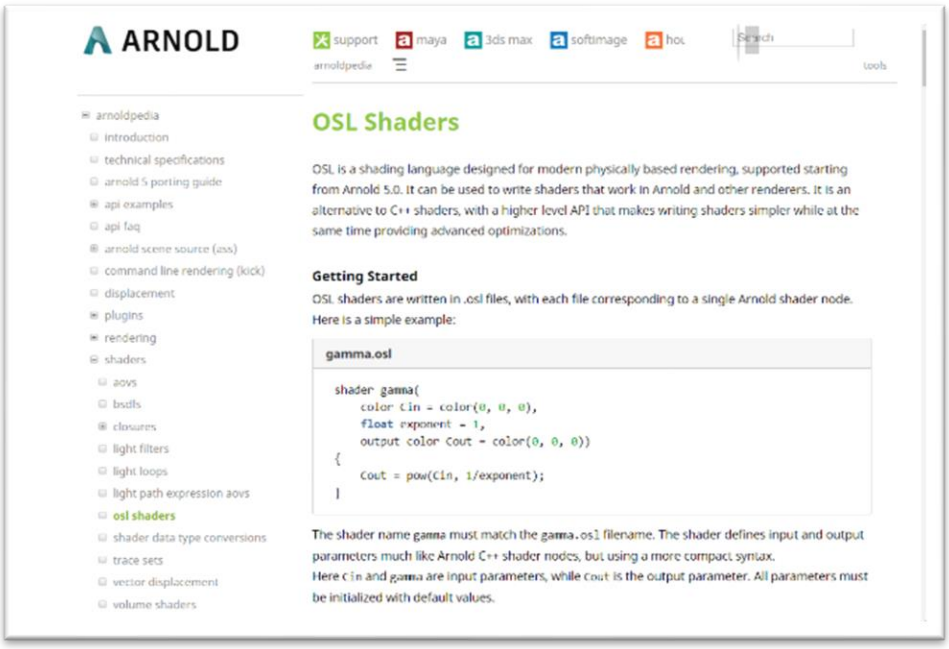
3ds Max comes with a number of default OSL shaders already available through the Material Editor. However, these are just a select few out of the numerous shaders that are available online. When looking for OSL shaders online, there are a number of repositories.

The first place to start is Sony Pictures Imageworks' Open Shading Language page on github.com: https://github.com/imageworks/OpenShadingLanguage. This is the official page put up by Sony pictures Imageworks for those who want to dive deeper into OSL shader development. Here, you can download source code, as well as a few OSL shaders.



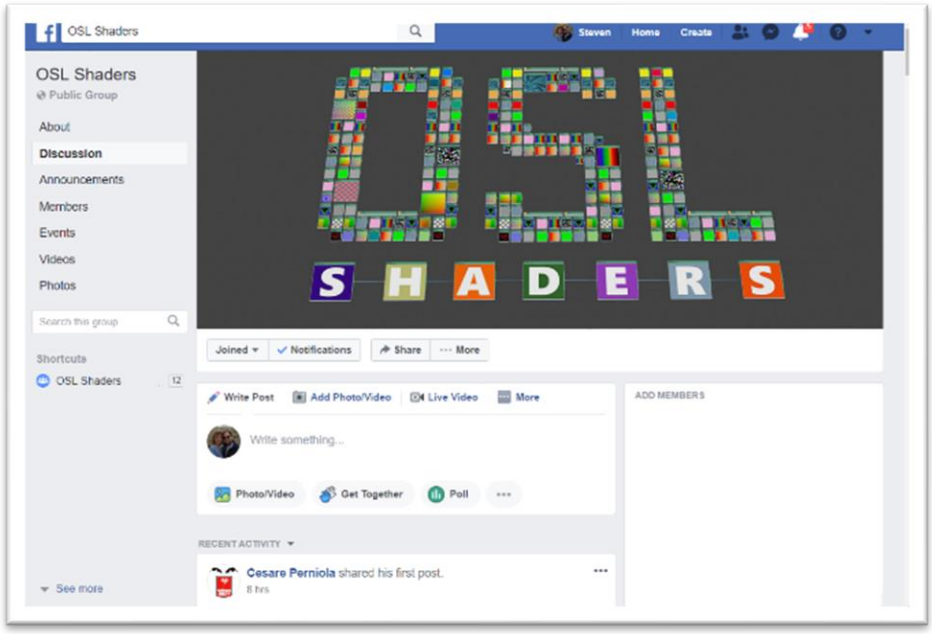
HTTPS://GITHUB.COM/IMAGEWORKS/OPENSHADINGLANGUAGE

Another valuable resource is the OSL page in the online Arnold documentation:



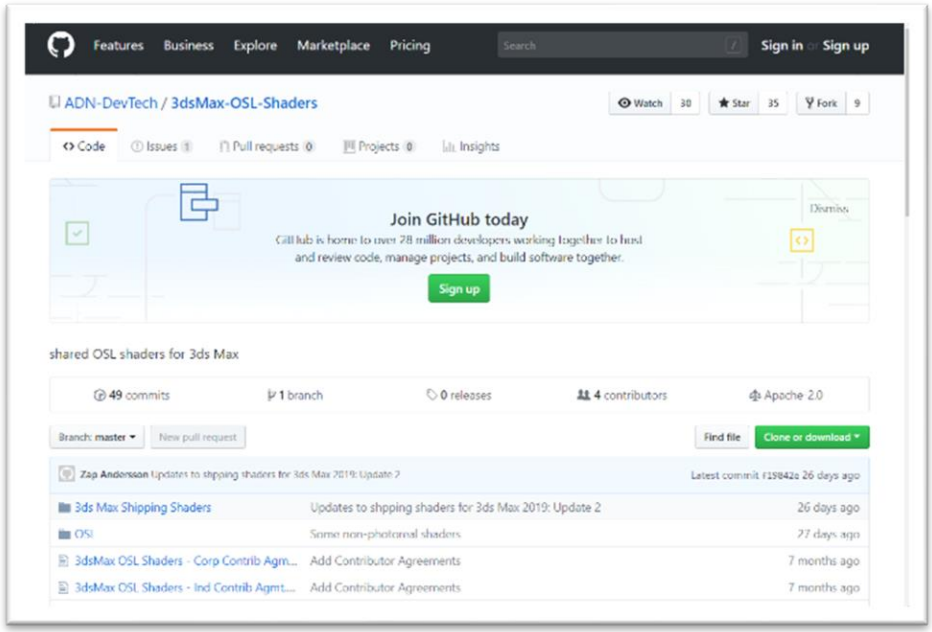
HTTPS://DOCS.ARNOLDRENDERER.COM/DISPLAY/A5ARP/OSL+SHADERS

A third place to go, if you're looking for help with writing shaders or are looking for links to shaders, is the OSL shaders group on Facebook. This is a public Facebook group with some of the top 3ds Max programmers as active members.



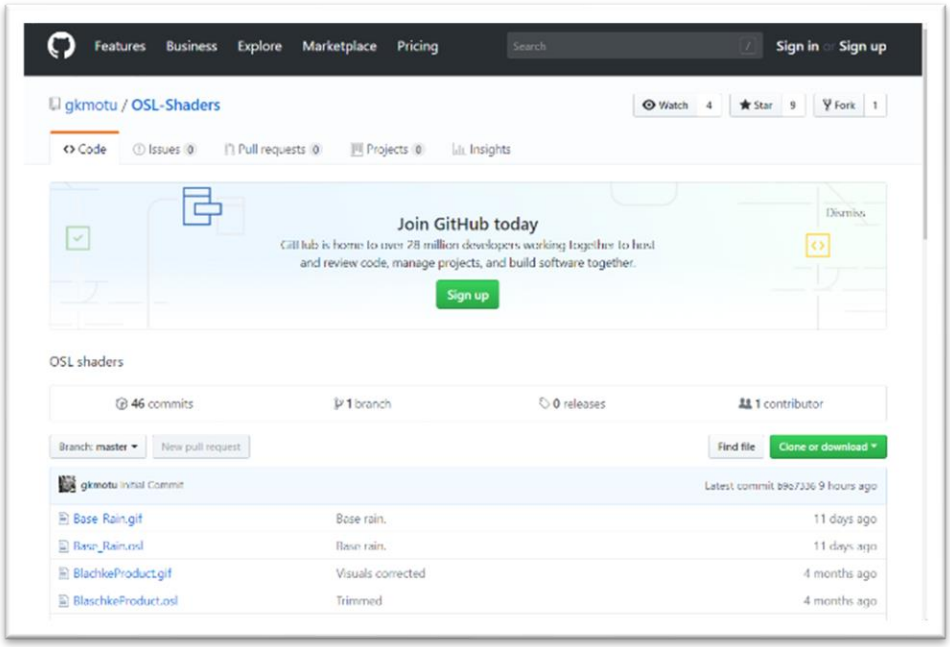
[HTTPS://WWW.FACEBOOK.COM/GROUPS/OSL.SHADERS](https://www.facebook.com/groups/OSL.SHADERS)

If you're looking for specific shaders, there are several repositories available on GitHub and elsewhere:



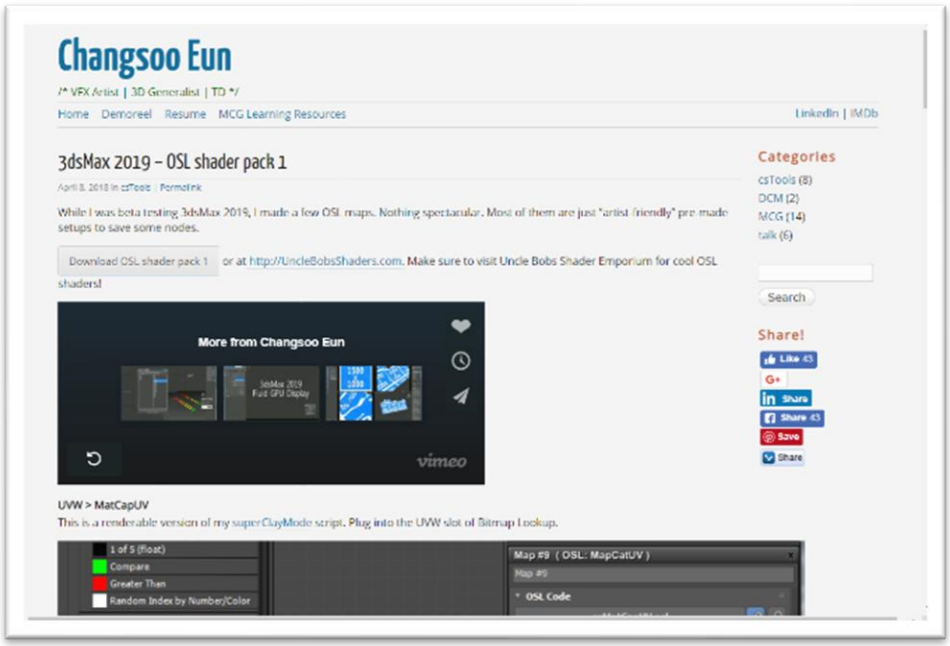
[HTTPS://GITHUB.COM/ADN-DEVTECH/3DSMAX-OSL-SHADERS](https://github.com/ADN-DevTech/3dsMax-OSL-Shaders)

Some more GitHub shaders are available at:



HTTPS://GITHUB.COM/GKMOTU/OSL-SHADERS

You can also find OSL shader packs, such as this one created by Changsoo Eun:

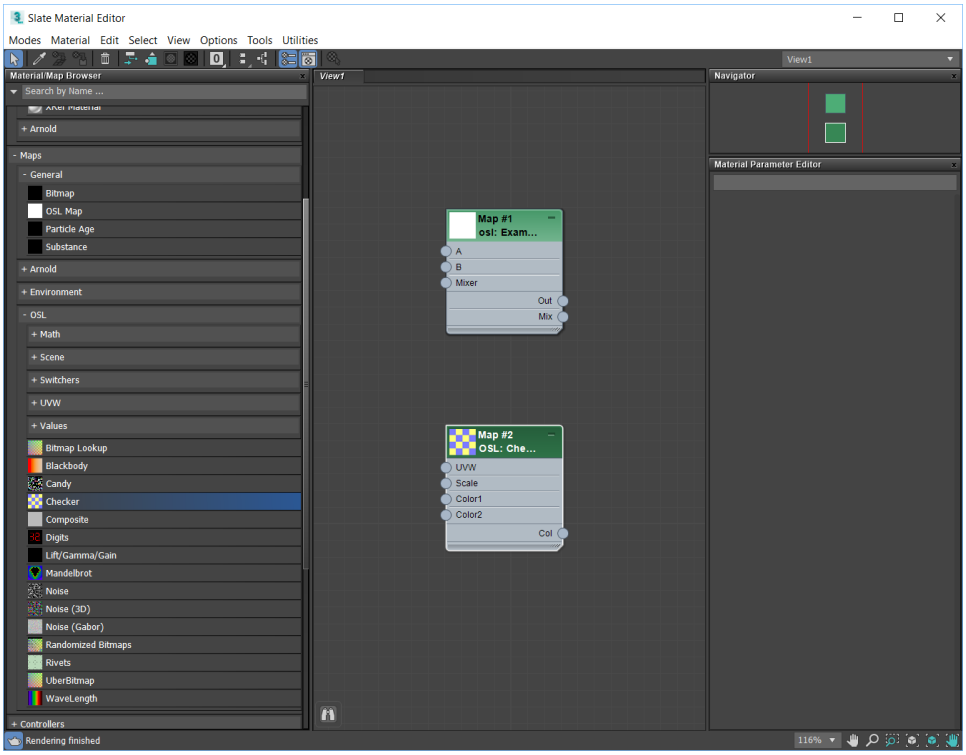


HTTP://CGANIMATOR.COM/3DSMAX-2019-OSL-SHADER-PACK-1/

While most OSL shaders you download from other locations around the Internet can be used, there are a few things to understand about OSL implementation in other renderers. This is because the implementation of OSL in 3ds Max has a few limitations. Shaders that use a *closure*, essentially a material, are not supported in 3ds Max. Neither are shaders that use an *#include* statement. If something needs to be included in the shader, find the code that needs to be included and copy and paste it directly into the shader.

Using OSL Shaders

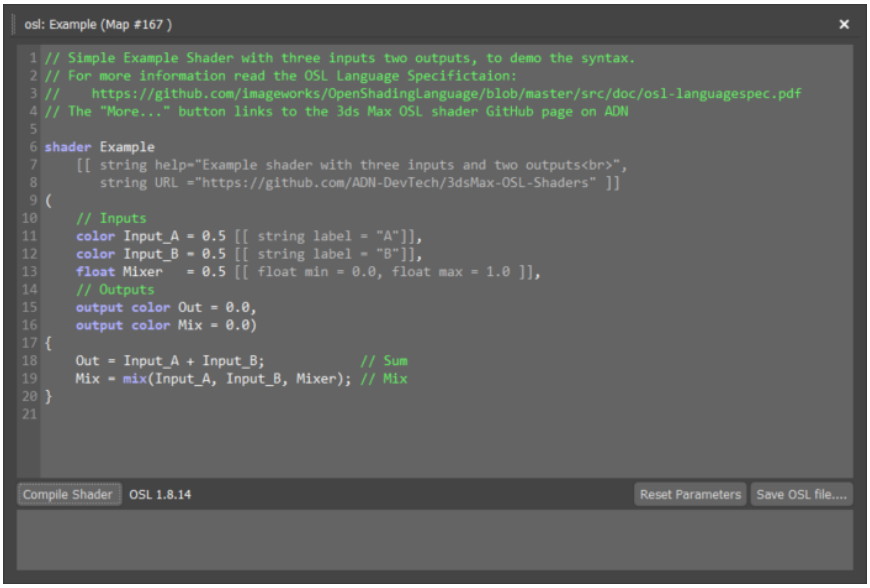
Working with shaders inside of 3ds Max can be handled two ways. They can be used directly by dragging them from the material/map browser into the Slate Material Editor (here, referred to as just the “Material Editor”).Or, you can use the OSL map and load shaders from there.



THE MATERIAL EDITOR SHOWING BOTH THE OSL MAP (TOP) AND THE CHECKER OSL SHADER (BOTTOM).

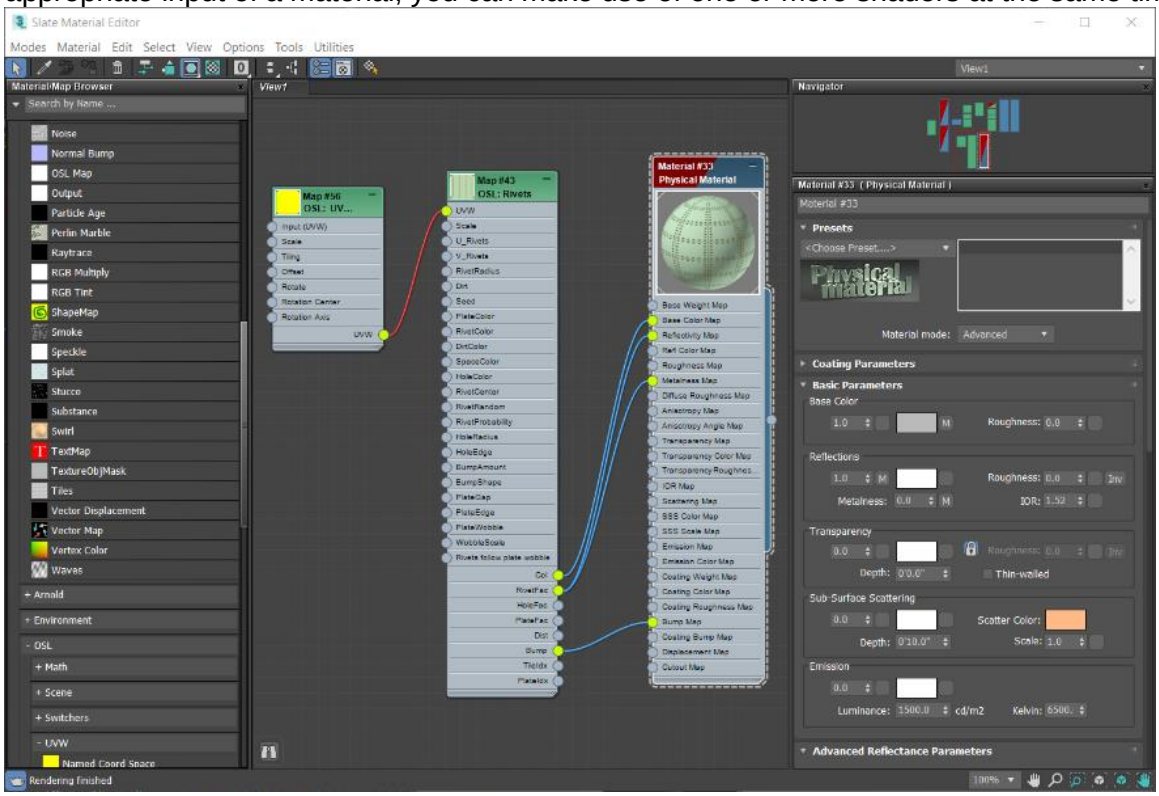
OSL shaders themselves cannot be used directly on objects and must be added as a map to an existing material. Depending on the shader that you use, some will have a single output, while others can have additional channels available to connect to a material. This is fully dependent on which shader is being used.

Once an OSL shader is loaded, whether by drag-and-drop or through the OSL map, you have the ability to edit the shader as you would like. If you are learning to write OSL shaders, you can edit them directly inside of 3ds Max. Once you write the shader, you can test-compile it directly in the editor. Once you’re satisfied, you can save your file to a new OSL file, or if you’re updating an existing file, you can simply overwrite it.



OSL shaders provide a great deal of flexibility and can be used for a broad range of applications. Looking at the 3ds Max **OSL** maps rollout in the **Slate Material Editor**, not only are there shaders for textures, but there are also shaders that can be used for other purposes, like mathematical operations, UVW modification, and more.

When applying OSL shaders to a model, you can use any compatible material, like the Physical Material or the Arnold Standard Surface Material. By connecting the output of an OSL shader to the appropriate input of a material, you can make use of one or more shaders at the same time.



THE RIVET SHADER APPLIED AS BOTH THE COLOR AND BUMP MAPS IN THE MATERIAL EDITOR.

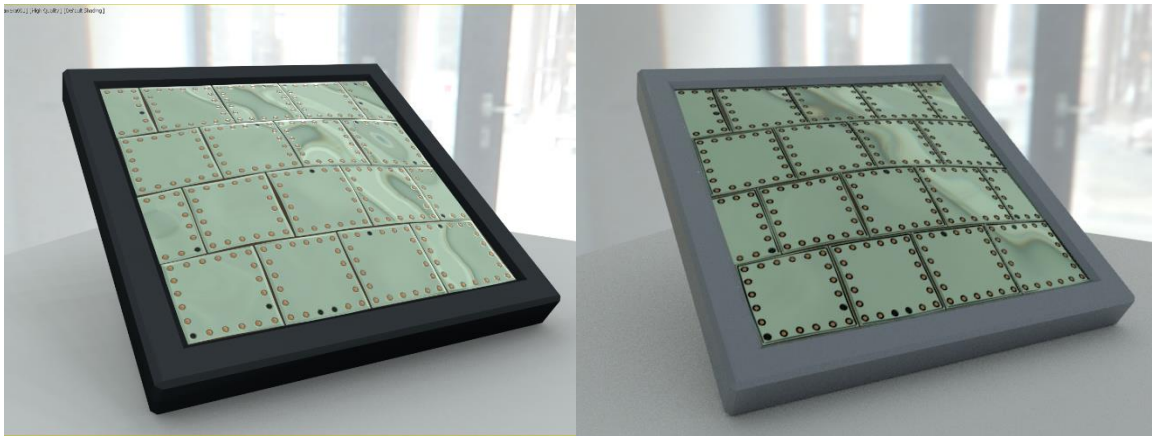
THE RENDERED VERSION OF THE OSL SHADER ON A PAINTED METAL SPITTOON.

OSL Viewport Display

New to 3ds Max 2020 is a major advancement in the look of OSL shaders within the 3ds Max viewport. When you set the viewport to use the High Quality (Realistic) viewport setting, the OSL shaders display accurately, and match the rendered version of the shaders very closely. This works by converting the OSL shader to an HLSL shader for viewport rendering. And, while this works very well, it's not always a perfect conversion for the viewport.

Because of the problem converting every shader, 3ds Max's OSL Map contains a Viewport Accuracy indicator. This indicator is located at the bottom of the OSL Map and shows the accuracy result as a percentage. In the case shown here, the result is a Viewport Accuracy of 100%.

The result of this translation process is a preview of the OSL shader in the viewport that closely represents what you will see in a rendering of the scene.



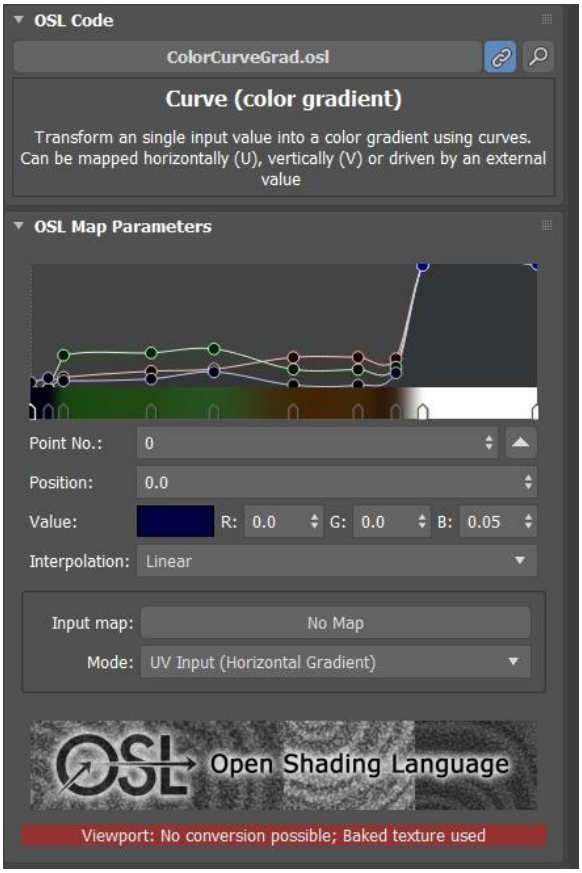
THE ABOVE SHOWS THE HIGH QUALITY VIEWPORT DISPLAY (LEFT) VERSUS THE RENDERED VERSION (RIGHT).

Top 7 OSL Shaders

In the previous sections I reviewed what OSL shaders are, and where you can find some that are not included with 3ds Max. Here, I'm going to review my top 7 OSL shaders that ship with 3ds Max. These are shaders that ship with 3ds Max 2021.2 and cover a variety of uses.

When I work with various OSL shaders I'm not always using them to create texture maps. So, I've created this list from OSL shaders that are both new to 3ds Max and ones that have been around for a few releases. I also took into account how these shaders can be used within a scene for modifying mapping, modifying other shaders, and adding animation to materials. This list is in no particular order.

Curves Shaders (Color Gradient, Color Correction, Float)



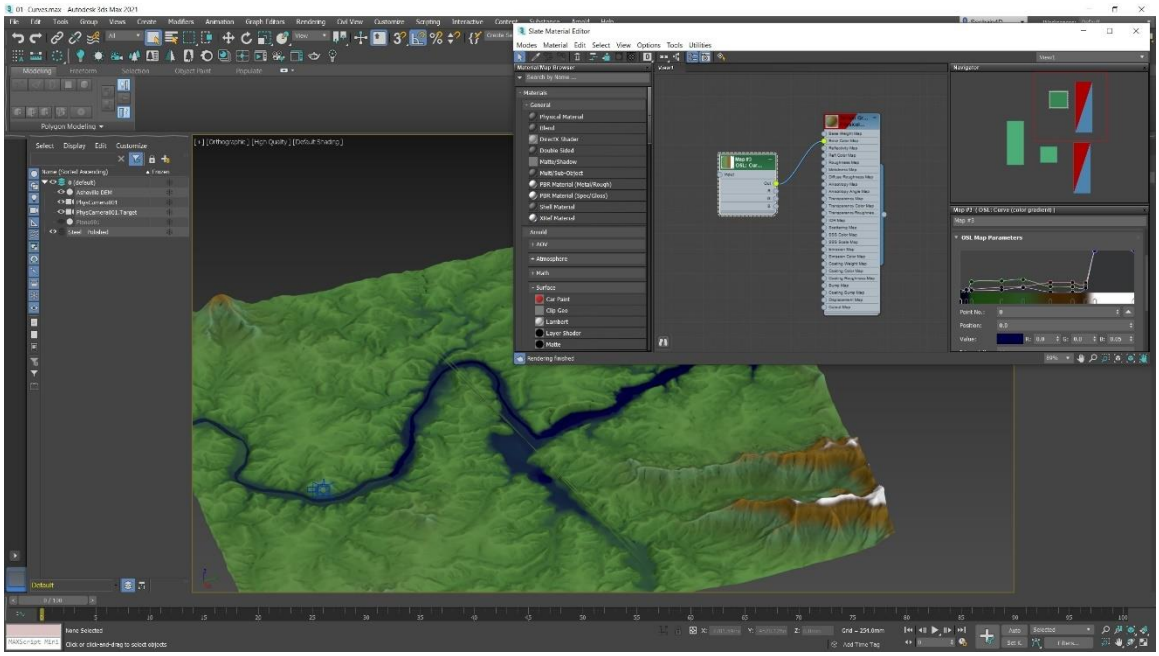
Curves shaders are used to create a color gradient, apply color correction, or transform an input float value by using values that are defined by a curve. Curves for this set of OSL shaders are edited in the shader itself. There is also an option to pop out the Curve Editor and edit your curves shaders using a larger window than the one that is provided in the Material Editor, Properties panel.

Curves (color gradient)

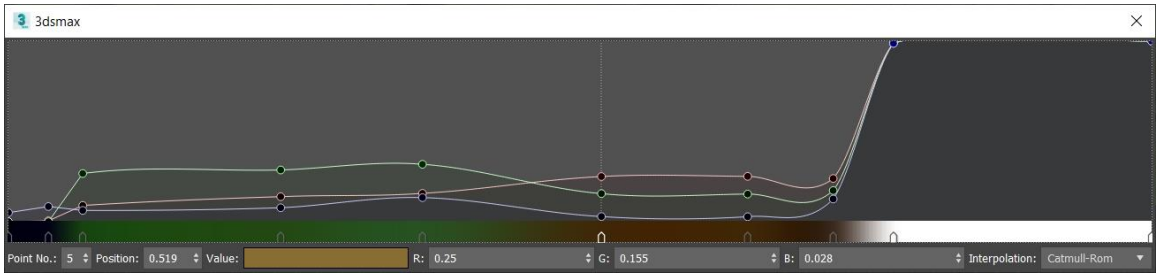
The Curves (Color Gradient) OSL Shader

The color gradient curves shader allows you to utilize a custom curve to create a color gradient. The resulting gradient can be applied to any object using standard UVW mapping coordinates. But, you can also use a grayscale image as the input to modify how the gradient gets drawn based on the gray values.

The image below shows the grayscale being used strictly as a gradient to show the elevation changes in the terrain model. This gradient goes from a dark blue through to green, then to brown, and finally to white for the higher elevations. This is based on UVW mapping that goes vertically through the model.

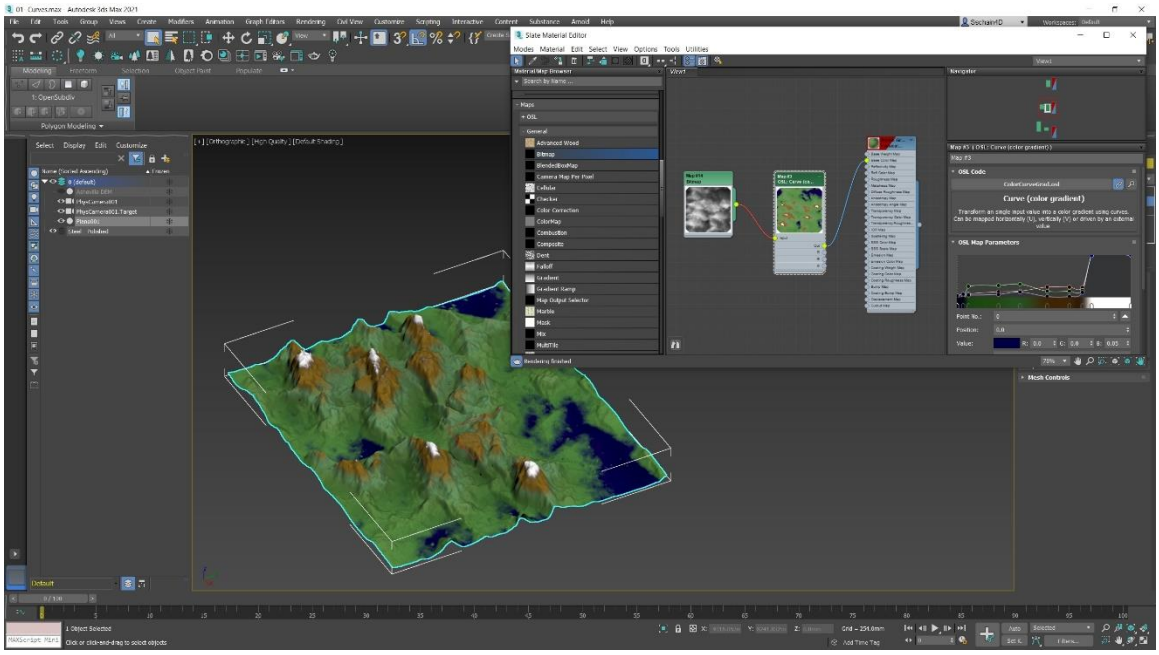


CURVES (COLOR GRADIENT) OSL SHADER APPLIED AS A GRADIENT MAP TO A DIGITAL ELEVATION MODEL.



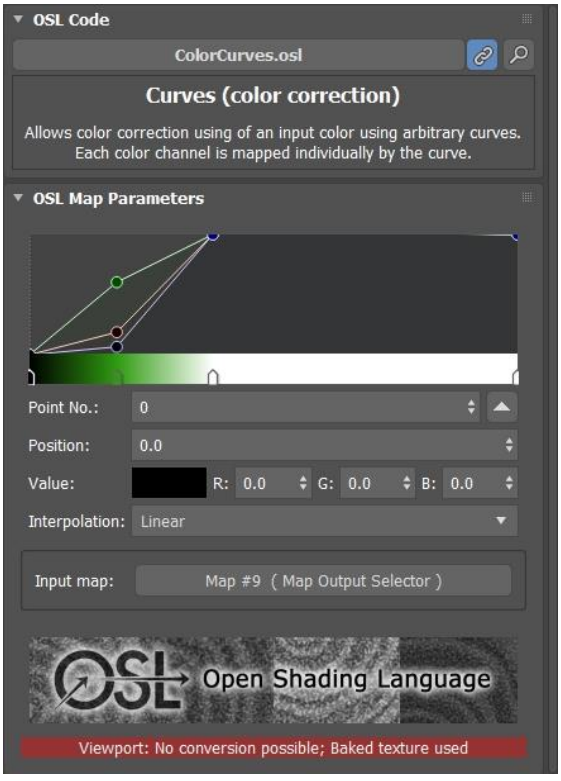
THE EXPANDED CURVE VIEW ALLOWS FOR EASIER EDITING OF CURVE POINTS, COLORS, AND POSITION NODES.

The image below is the same gradient being modified by grayscale bitmap and applied to a displaced mesh using that same displaced bitmap to set the height. You can see that this is a flexible shader that can be used in a variety of materials.



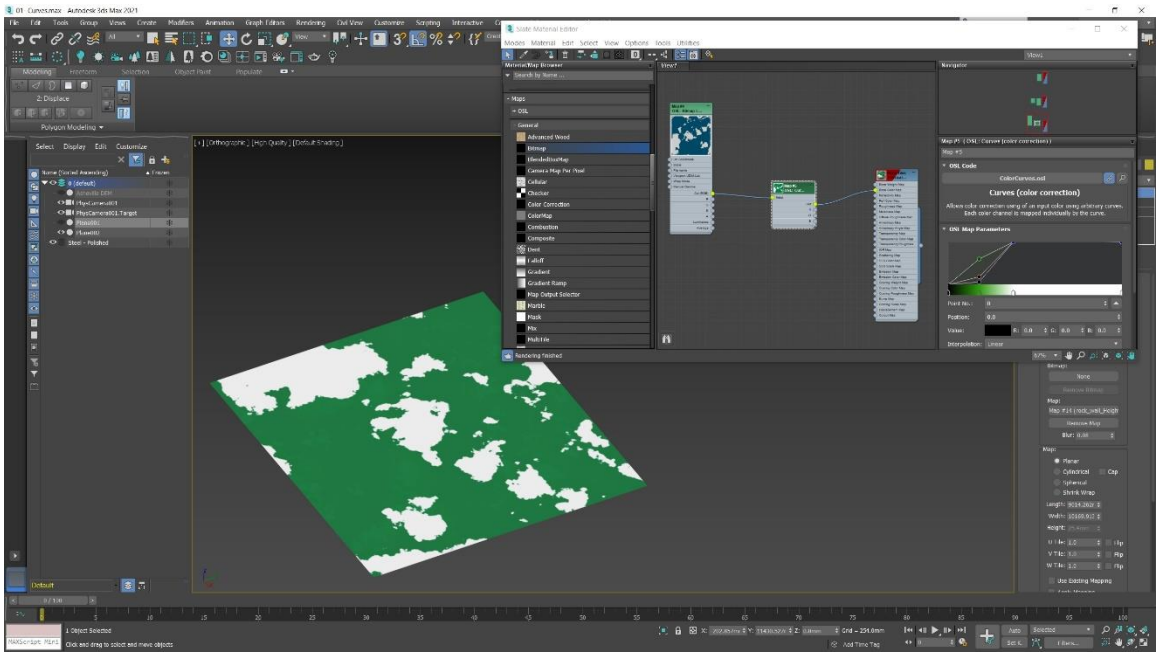
THE OSL CURVE (COLOR GRADIENT) SHADER BEING MODIFIED BY GRAYSCALE BITMAP.

Curves (color correction)



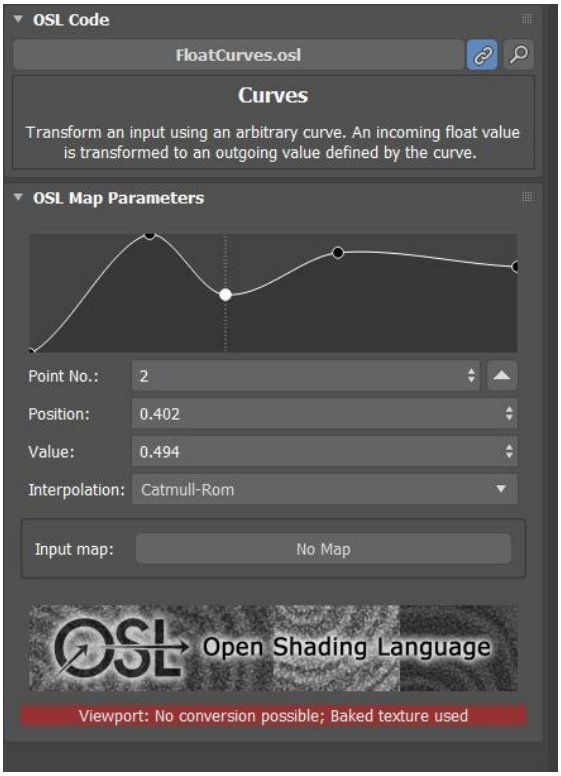
THE CURVES (COLOR CORRECTION) OSL SHADER

Another curves shader used for color correction allows you to take an RGB input and remap the colors using a curve-based approach. The example below takes the blue color from the texture map and remaps it to green.



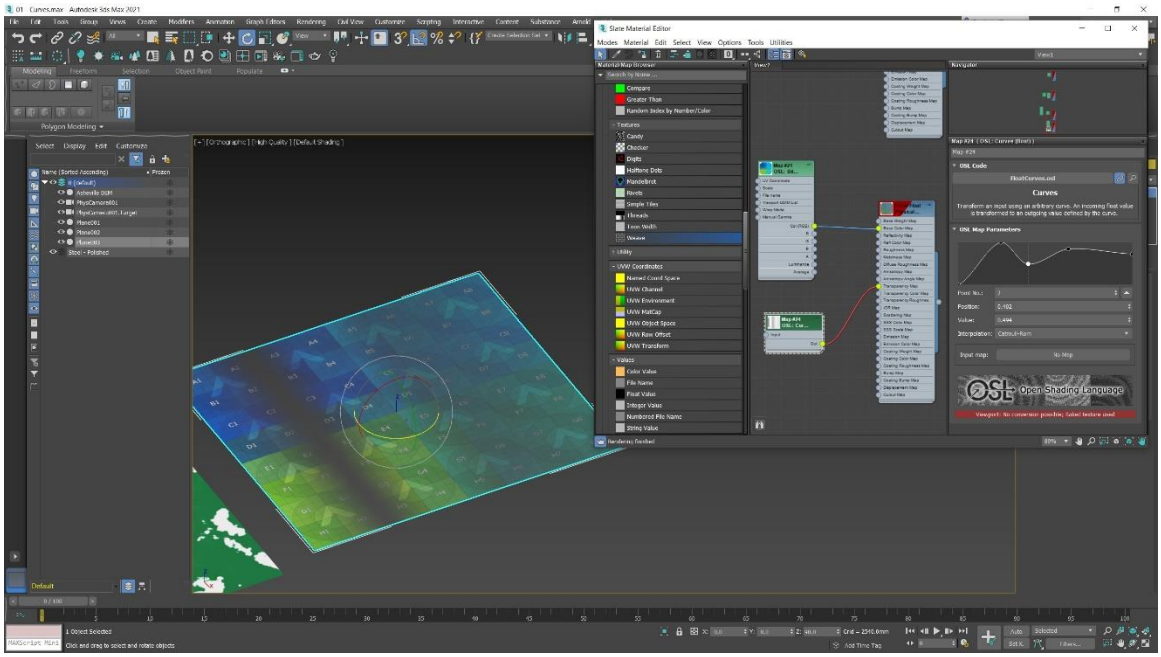
THE OSL CURVES (COLOR CORRECTION) SHADER BEING USED TO REMAP BLUE TO GREEN.

Curves (Float)



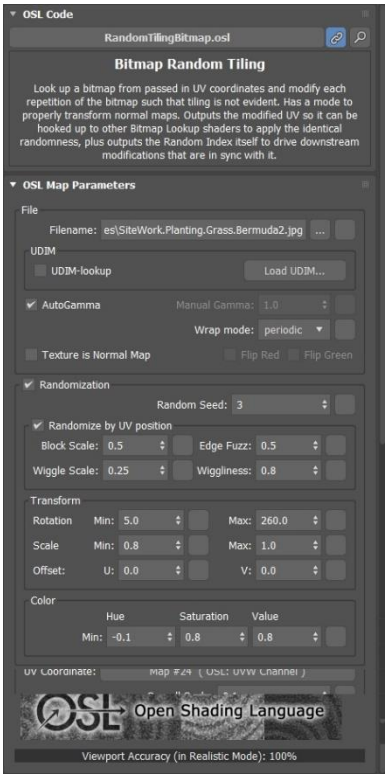
THE CURVES (FLOAT) OSL SHADER

The third curves shader uses a two-dimensional curve to output a float value. This value can be used in a material parameter, such as a transparency map where the grayscale value determines the amount of transparency. Placement can be based on UV mapping or modified by an input map.

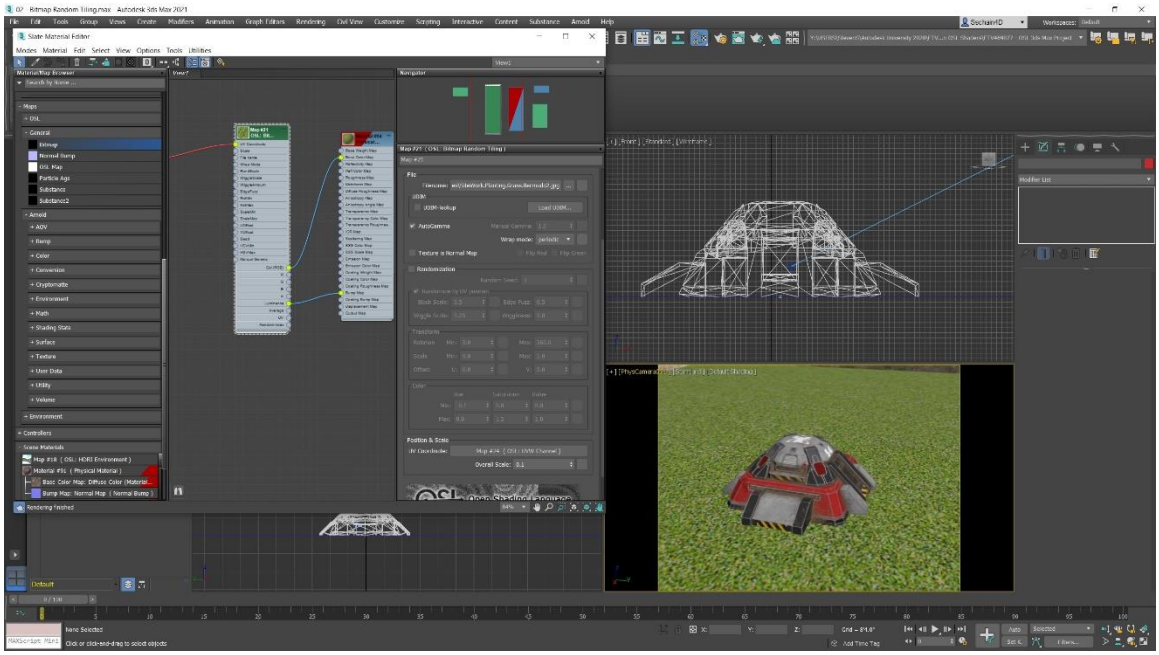


THE CURVES (FLOAT) OSL SHADER BEING USED TO MODIFY THE TRANSPARENCY OF A MATERIAL.

Bitmap Random Tiling



This shader uses a bitmap image along with UV coordinates and allows you to tile the image using randomized values. There are a range of uses for this shader, but the one I will show here is taking a simple tileable texture and allowing you to create a more random, realistic look. This is especially helpful for materials like grass, where a tileable texture may look fine for a small patch, but over a large area tends to look fake because there is little variation.



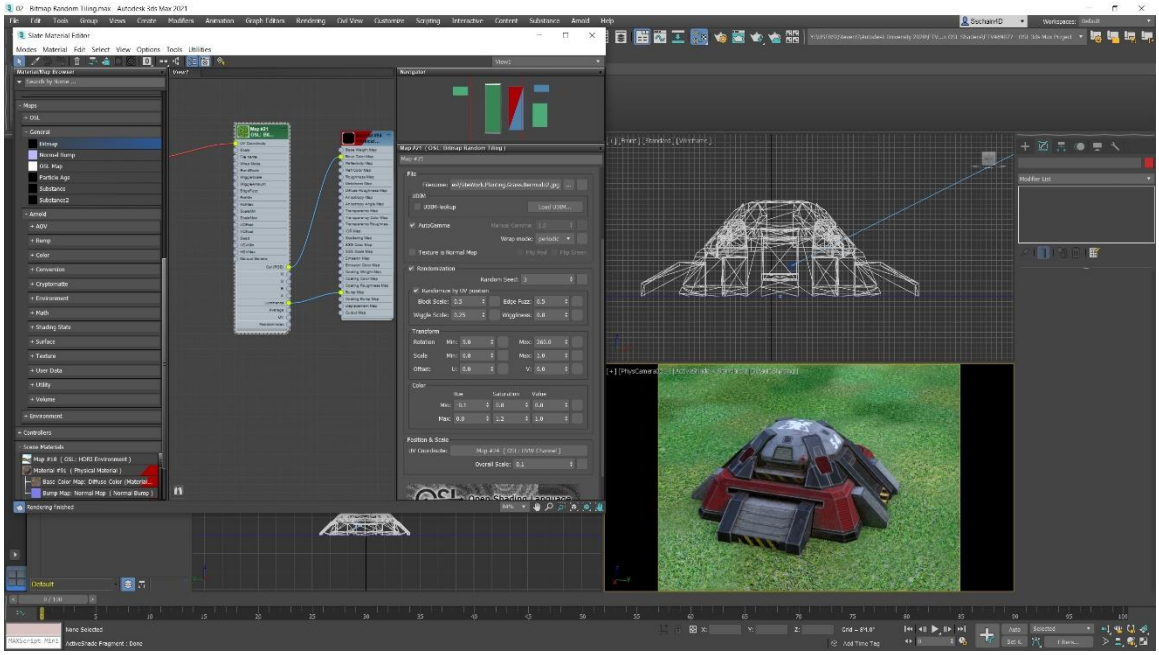
THE BITMAP RANDOM TILING SHADER WITH RANDOMIZATION TURNED OFF. NOTICE THE REGULARITY OF THE GRASS TEXTURE.

Working with this shader requires that you know what you want out of the texture that you put into it. For this grass texture example, I wanted something that was going to be a little bit more random, not only in the size of the grass patch itself, but also in the rotation and color values.

The first thing I did was set the Block Scale, Edge Fuzziness, Wiggle Scale, and Wiggliness values. This helped to randomize the placement of the texture map across the plane of the grass. Without doing anything else, this would've been just fine, however, I wanted to change the rotation, scale, and color as well.

To accomplish the rotation and scale, I set the Rotation minimum to 5° and the maximum to 260°. Then I set the Scale to a minimum of 0.8 and a maximum of 1.0. With these two options set, I now had variation in both the orientation of the texture and the size of the texture. By limiting the maximum scale to 1.0, I made sure that I didn't blow up the image too far for the texture.

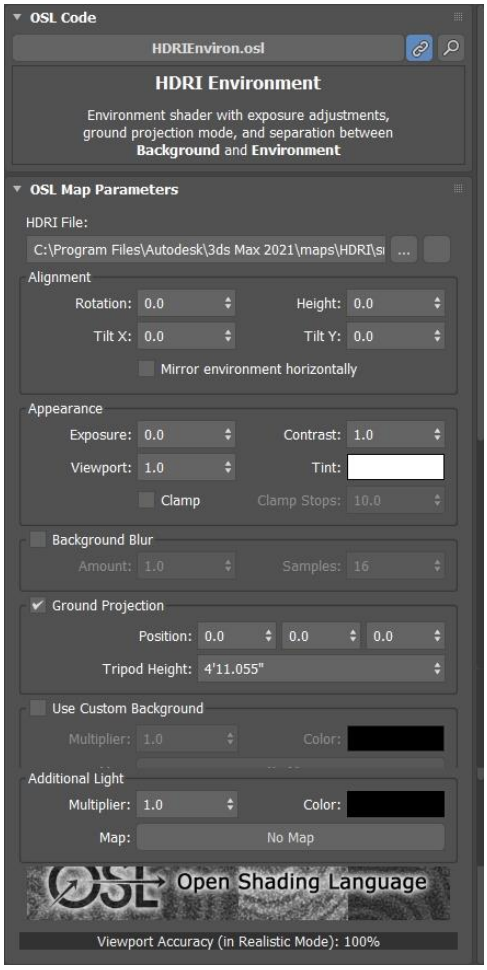
To modify the color, I adjusted the minimum and maximum values for the Hue, Saturation, and Value options in the Color group.



THE BITMAP RANDOM TILING SHADER WITH RANDOMIZATION TURNED ON. NOTICE THE CHANGE IN THE COLORING OF THE GRASS TEXTURE.

HDRI Environment / HDRI Lights

HDRI Environment



THE HDRI ENVIRONMENT OSL SHADER

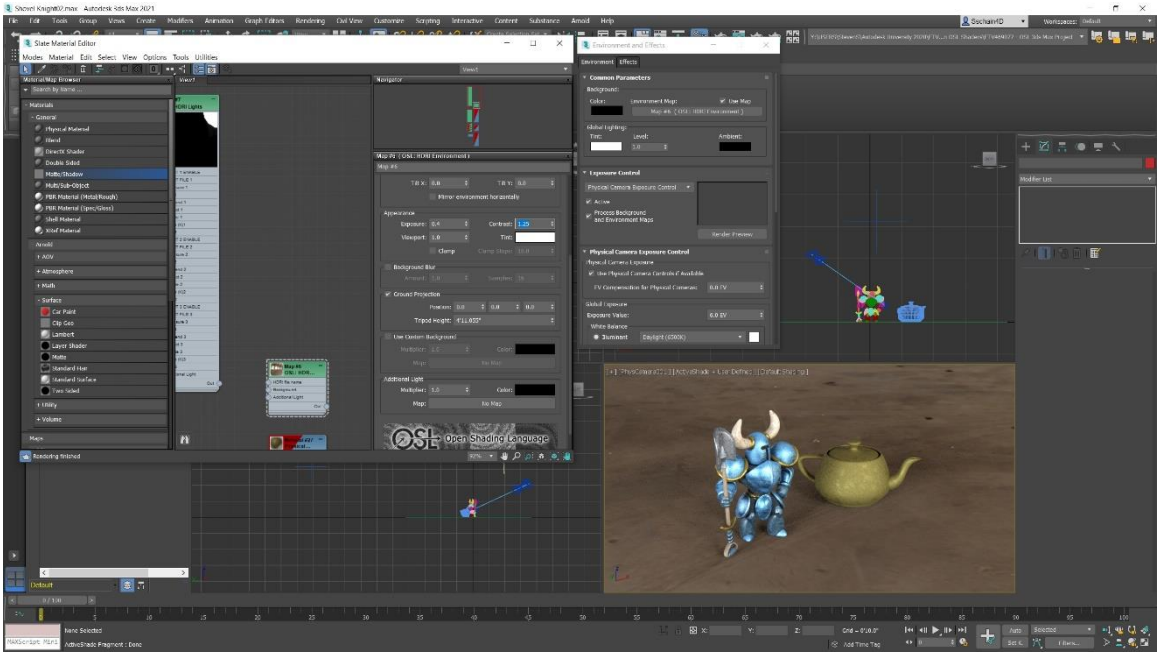
Lighting a scene using traditional lights can take time and expertise in order to get a highly realistic look. However, 3ds Max has been able to use high dynamic range images to light scenes for many releases. Using HDR images provides an easy way to get photorealistic rendering with a minimum of effort.

An HDR image consists of multiple exposures that are then combined into a single image. This image is represented using 32-bit floating point values for red, green, and blue. When used to render a scene, HDR lighting pulls the color value from this image and uses it as the light color and intensity.

AN EXAMPLE HDRI IMAGE SHOWING 5 EXPOSURES

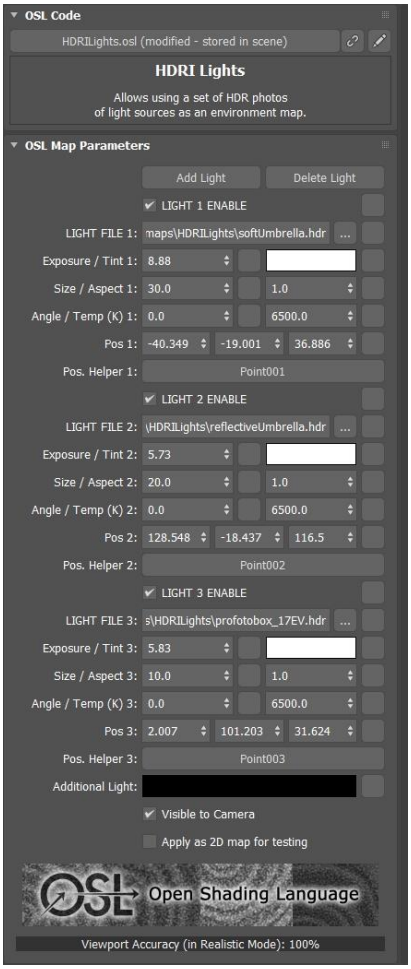
Utilizing the OSL HDRI environment is a simple process. All you need to do is instance the HDR environment shader into the background environment map. Once there, you can adjust the alignment, appearance, and other parameters of the OSL shader in the Material Editor. Getting the right exposure is a combination of adjusting the exposure value in the shader and adjusting the exposure value in the Physical Camera parameters.

When making the adjustments, I always start with the parameters in the camera first. I make sure that the exposure looks good, then I can make any tweaks I need to in the appearance parameters of the shader. Once I get it looking the way I like it, I can render the scene using that configuration.



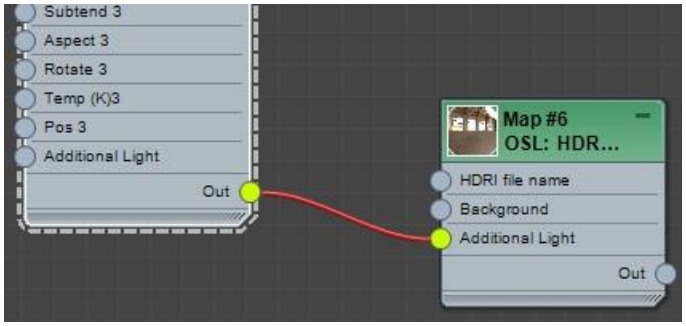
SHOWING THE HDRI ENVIRONMENT USED TO RENDER THE SCENE.

HDRI Lights



THE HDRI LIGHTS SHADER WITH THREE LIGHTS ENABLED.

The second HDRI shader available that I use quite often is the HDRI Lights shader. This shader is similar to the HDRI Environment shader in that it uses HDR images to light the scene. The difference is that this shader uses high dynamic range images of individual lights. This shader is often used in conjunction with the HDRI Environment shader and is added to the additional light input.

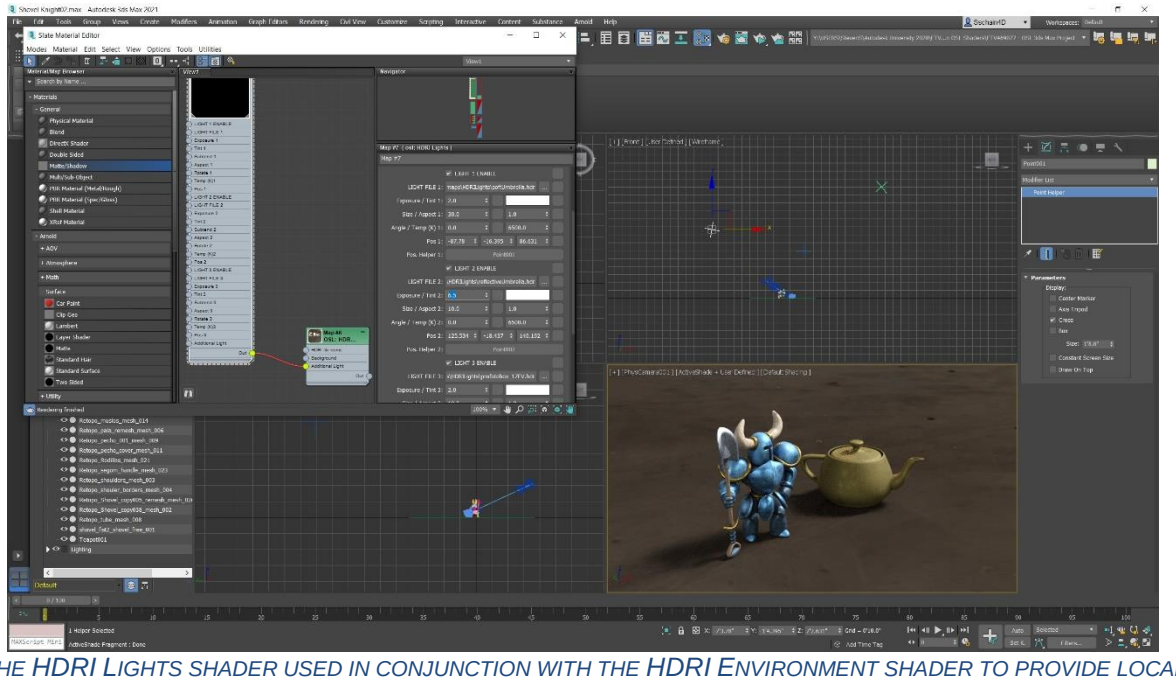


THE HDRI LIGHTS CONNECTION INTO THE HDRI ENVIRONMENT SHADER.

Once the connection is made, the lights in the shader represent the local lights that are in the scene. Initially, the shader starts off with one light. The light file option allows you to select which HDR light you want to use. There are several examples that ship with 3ds Max. For this one, I used the soft umbrella HDR file. Then, I added two additional lights, a reflective umbrella, and a pro photo box.

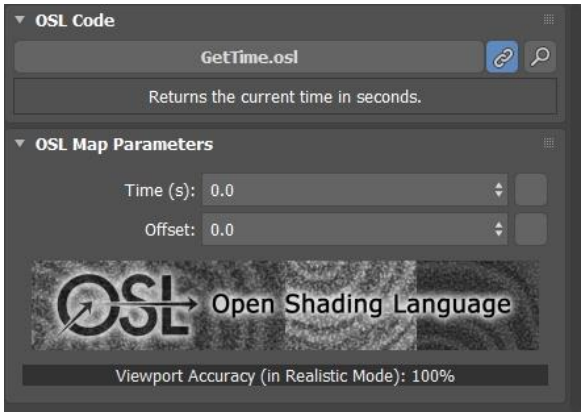
Enabling the Position Helper option creates a helper point within the viewport that you can use to move each light around the scene. Keep in mind that each light is only being moved in relative position and is always pointed towards the center of the world. By positioning the lights and adjusting each light's parameters, you can set the relative brightness for each light.

An easy method to create a final rendered image using both the HDR environment and HDR lighting is to add a plane that uses a Matte/Shadow material. The Matte/Shadow material catches the shadows and transfers them to the background. This gives the impression that the objects in the scene are sitting on the background image.



THE HDRI LIGHTS SHADER USED IN CONJUNCTION WITH THE HDRI ENVIRONMENT SHADER TO PROVIDE LOCAL LIGHTING AND SHADOWS.

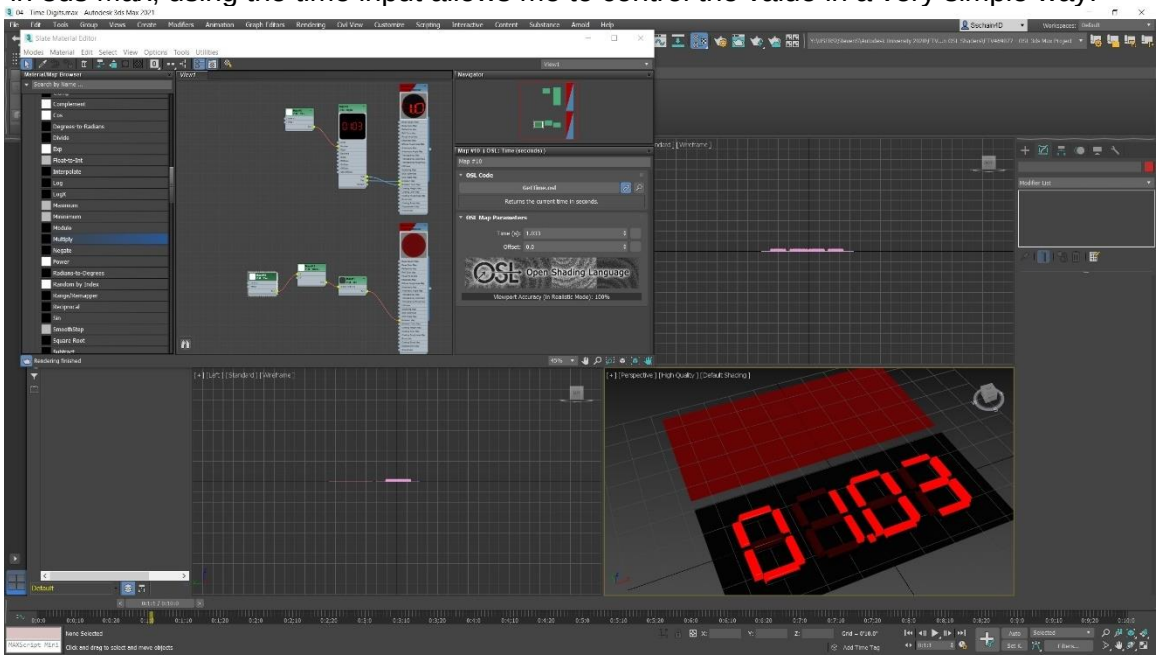
Time (Seconds)



THE TIME (SECONDS) OSL SHADER.

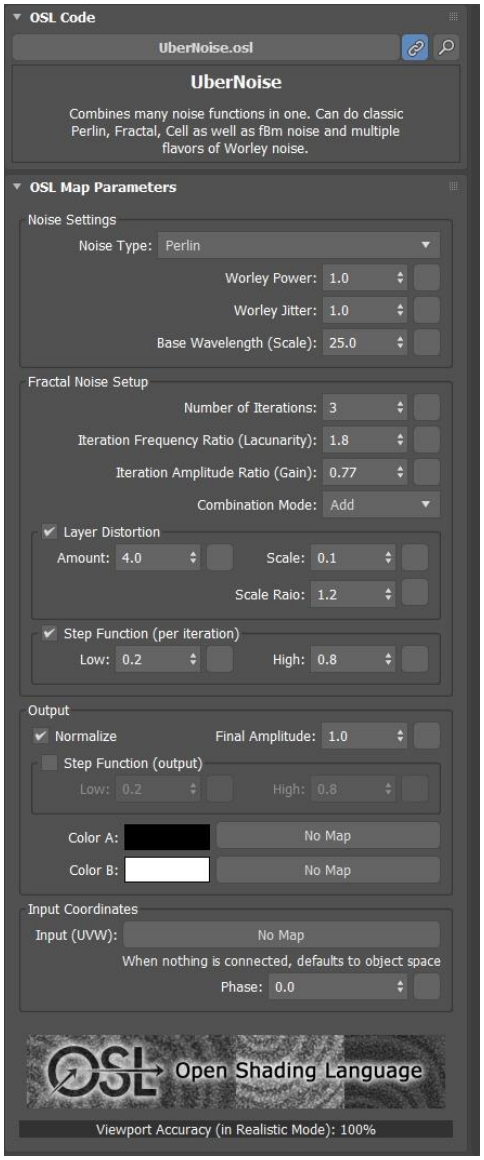
The Time (seconds) shader is a very simple OSL shader that reports back the time in seconds as you scrub the Time Slider or play the animation. There are only two parameters: time, and offset. Time reports the time in decimal seconds, and the offset value adds or subtracts from the time value.

The examples below show two uses of the counter to control not only maps but also material parameters. The first example is a setup of a counter used as a texture map. It uses the digits OSL shader as an emission map and an emission colormap. Connected to the number is the time shader. The second example uses the time along with a multiply and sin OSL shader to control the value for the emission map of a material. While there are several ways to accomplish this in 3ds Max, using the time input allows me to control the value in a very simple way.



THE TIME (SECONDS) SHADER BEING USED TO CONTROL A DIGITS VALUE OF ANOTHER OSL SHADER AND THE EMISSION VALUE OF A PHYSICAL MATERIAL.

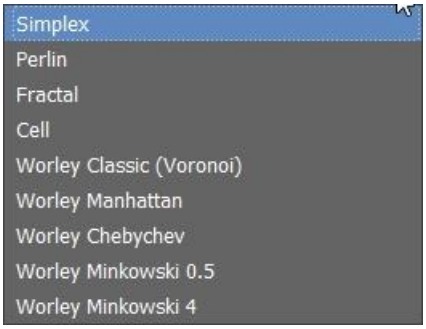
Uber Noise



THE UBER NOISE OSL SHADER

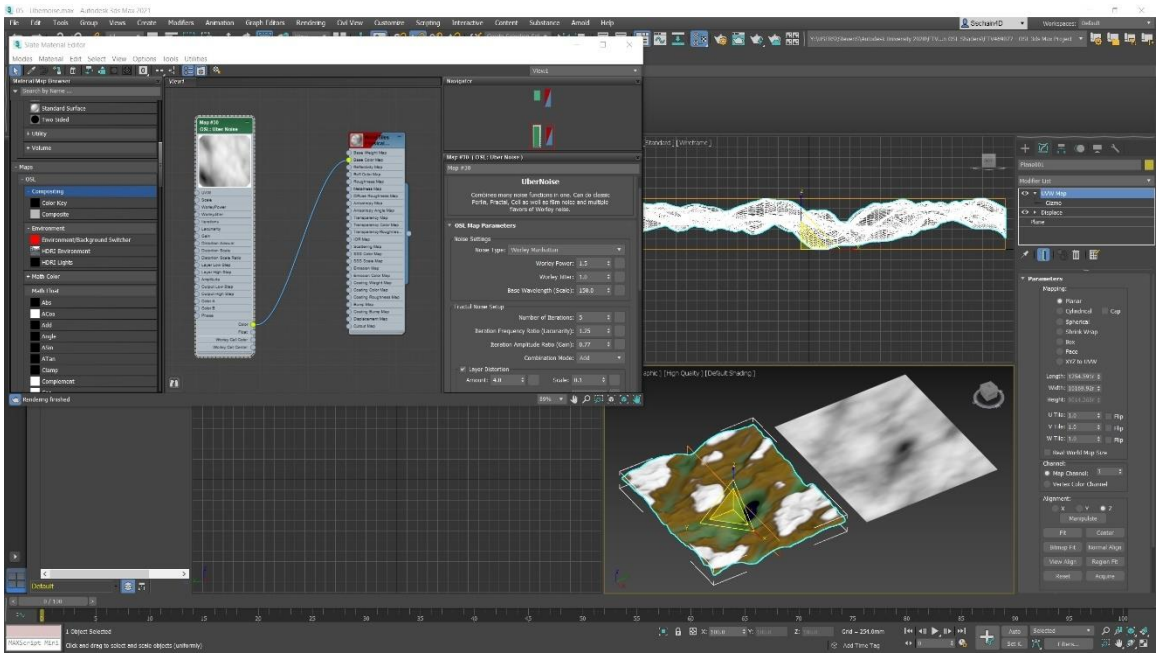
I've always been a fan of the noise map that came with 3ds Max since day one. It comes in handy when I need any kind of texture with some randomness. However, once I discovered the Uber Noise OSL shader, I've started using it in place of the noise map.

There are a few reasons I switched over to the Uber noise shader. The first is that it contains more than the three noise types available in the 3ds Max noise map. Uber noise contains four standard noise types along with five Worley noise types. These five are based on the noise functions developed by Steven Worley in 1996 and can be used to create a variety of procedural textures.



UBER NOISE OFFERS 9 NOISE STYLES, INCLUDING 5 WORLEY NOISE TYPES.

Noise is a texture that can be used in such a wide range of materials that it is indispensable as a tool in your material creation toolbox. It can be used to add detailed texture to flat surfaces like paint as a bump map, or it can be used to create unique mountain ranges.

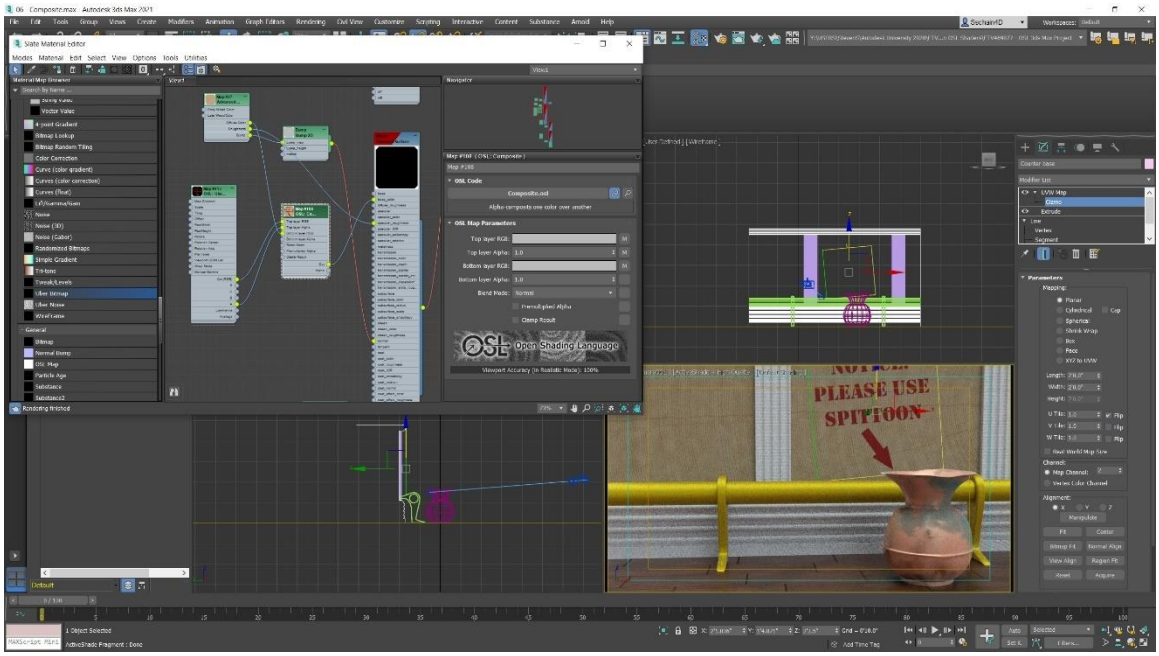


UBER NOISE USED AS A DISPLACEMENT MAP FOR AN EASY MOUNTAIN RANGE ON THE LEFT, AND THE 2-DIMENSIONAL VERSION ON THE RIGHT.

Composite

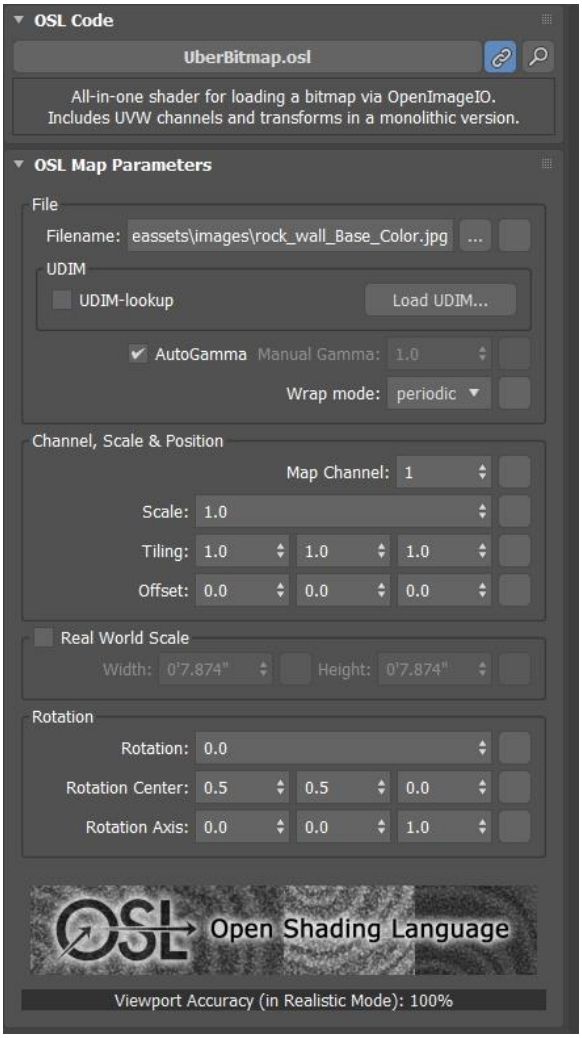
THE COMPOSITE OSL SHADER.

The composite OSL shader provides a simple way to overlay one image onto another using both alpha channel and blending modes. This shader is on my list because it provides an easy and flexible way to composite one image over another. While there are other ways of doing this, this shader offers minimal complexity. And, working in conjunction with the Uber bitmap, you can pull in the RGB and alpha channels individually and use them as needed.



THE COMPOSITE OSL SHADER BEING USED TO OVERLAY A STENCILED SIGN ON A WOOD BACKGROUND.

Uber Bitmap

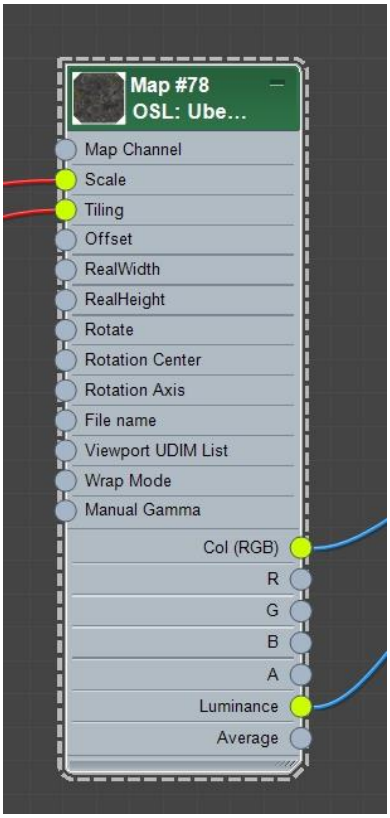


THE UBER BITMAP OSL SHADER.

The Uber Bitmap OSL shader utilizes the OpenImageIO library to load bitmaps into 3ds Max. OpenImageIO is an open-source library for reading and writing images in nearly any format. The Uber bitmap provides additional parameters for selecting the map channel, scale, and tiling, as well as rotation values. While you can use 3ds Max's bitmap texture map, the Uber bitmap shader has a few advantages.

The main advantage is that the shader has options for outputting an RGB color value, along with separate red, green, blue, and alpha channels. In addition to those outputs, there is a luminance output which simply outputs the brightness value of each pixel, and an average

output that outputs a grayscale image that represents the average of each RGB value at each pixel.

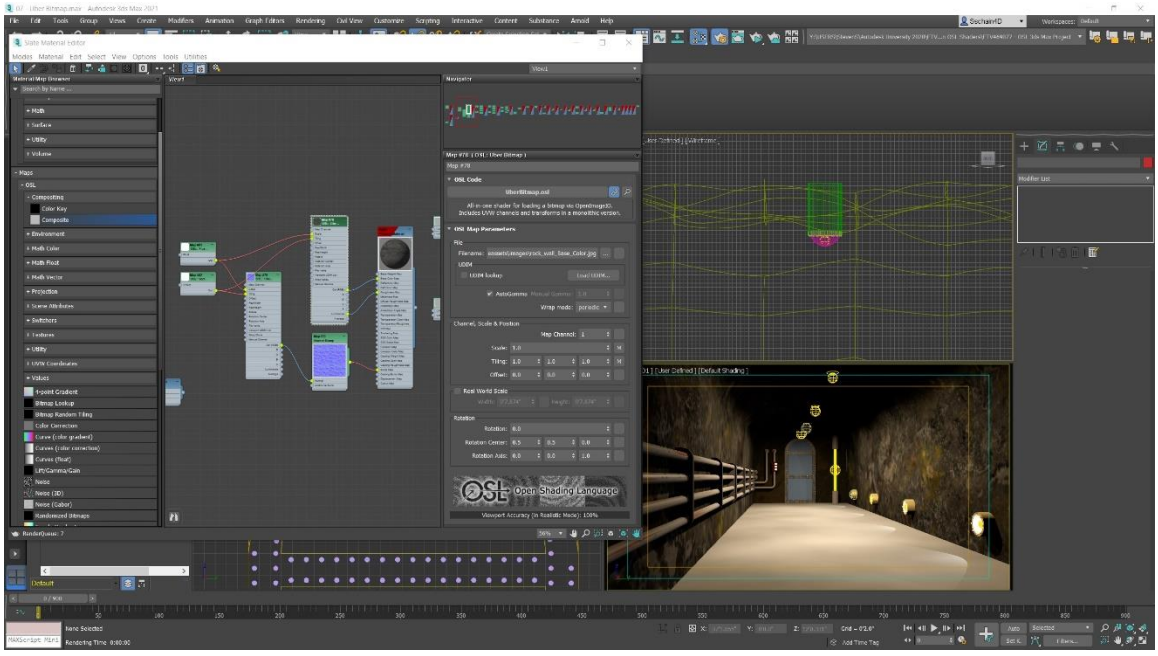


THE UBER BITMAP OSL SHADER SHOWING THE INPUTS AND OUTPUTS.

While the shader can output multiple versions of the image, it is the input values that add flexibility. By utilizing other OSL shaders to control specific values, you can link them to multiple Uber bitmap shaders and control them monolithically.

For the rock walls in the example below, the scale value for the color texture and the normal map are controlled by a Float Value shader. This allows me to adjust the scaling in both Uber bitmaps by adjusting a single value in a shader connected to the scale values.

The same goes for tiling of the normal map and colormap. Here, I used a vector value shader to control the three values for the U, V, and W tiling. It's important to understand which type of value you need for the different inputs. A single float value works for the scale, but it would not work for the tiling. Using this technique, you can synchronize multiple Uber bitmap shaders to give you the results you're looking for.



MULTIPLE UBER BITMAP OSL SHADERS USED TO CREATE THE LOOK FOR THE ROCK WALLS OF THE TUNNEL.

Conclusion

OSL provides a tool for adding unique and helpful shaders into 3ds Max. OSL shaders are compatible with multiple renderers including Arnold. The seven shaders that I reviewed here are just the tip of the iceberg when it comes to available OSL shaders. Take some time, review the shaders that ship with 3ds Max, and look on the internet for additional shaders that may be helpful in your next animation or rendering project.