Nuke 17 3D system beta guide

Introduction

This document aims to serve as a guide for new artists using Nuke's new 3D system. It will provide information on each of the nodes in the new 3D system, context behind design choices and the philosophy of the new system and answers to frequently asked questions.

Welcome to the new 3D system from the Nuke team

Hi everyone, thank you for joining us in testing and exploring the new 3D system. Hopefully this document and the other training resources being created will help walk you through all the changes that have been going on to the New Nuke 3D system.

My name is Chris Wetherly and I'm one of the Product Managers on the Nuke team and am helping to lead this new 3D system. First thing I want to say right off the bat is to thank all the artists who have been leaving feedback, both on our forums, in meetings with us, or even on places like Reddit, whether that feedback is positive or critical. It all helps shape the work as what I am passionate about is making this system a place for Nuke artists and for it to serve comp artists needs.

Hopefully this document will help share the understanding of what our goals are, how we're trying to take on board your feedback and can act as a call to action to say, please give us more. Though I do ask that you provide constructive criticism, as if you don't like something I want to know, but also want to know why, so I can address it correctly. An example from past feedback has been that the system doesn't feel as intuitive. This is important for us to know, but even more helpful is knowing which parts don't feel intuitive, the way you pick masks, the naming of nodes that might break muscle memory, as then we can address this (one example was the mask path defaulting to last modified not being what artists expected so we updated this to be All Roots, to better match the classic 3D system).

The Why

So why did we decide to build a new 3D system in Nuke? This has been a long standing request from a lot of companies and artists who want to work with larger assets in a more performant manner. For basic setups the classic 3D system is tried and tested, but when you start to bring in denser meshes, larger scale scenes, and try to play them back in real time while adding projections, or animated textures moving up and down UVs this is when you would start to hit performance issues. It would also be limiting in larger fly through shots, where you would have to break up assets, or simplify them specifically for comp, so comp artists weren't always working with the same assets as other departments.

A second question might be, "why USD?" For us this choice came down to a number of factors but key among them was that USD allowed us to improve on what comp artists do most in 3D, by allowing you to bring higher detailed, more complex meshes in a more performant manner for projection or matte painting workflows without having to pay a high conversion cost. Beyond that there are new workflows that USD will enable in the future especially when we consider technologies like Hydra allowing you to easily swap out your render delegates inside of Nuke.

The other reason is to embrace open standards and allow for workflows that not only provide better performance and scalability, but also allow for greater collaboration between artists. Having a non destructive workflow that means you can work more seamlessly with the lighting artists, FX TDs, layout department, will mean less time waiting for asset handovers and an ability to see exactly what artists in other departments see. Also quite frankly as compers you often have to find a fix last minute, when there's no time or budget to push it back to FX, or no time to get a new spec render pass from lighting and having a more powerful system that allows you to cheat these things in comp is a must with the amount of work artists are asked to do these days.

There's also been a shift in understanding of where a comp artist sits in the pipeline in recent years. Yes of course we're at the end and that will still be a major part of where our work sits, but more and more comp skills are finding value in being in earlier stages of the production, whether that's helping to author cameras, create new non photoreal looks and essentially do lookdev, or be a part of helping to visualise assets on a virtual stage. These workflows, and more we will likely discover, really need a 3D system that allows a comp artist to leverage their skills at any part of the production that needs it.

The Approach

That's the why of it, now what has our approach been. Well, central to our approach has been to try to put the artist first in how we build this new system. It's the reason we put the system into beta really early, as we knew it would be really important to get feedback on both the architecture and workflows. Now hands up we've stumbled a bit here with how we've communicated with you and I apologise for that, as we put the system out in Nuke 14.0, to really focus on feedback on the USD architecture which helped shape how we make sure our implementation is pipeline friendly and helped us discover some key areas under the hood to improve. To do this we also put out a bunch of nodes in a very early state to help testers visualise how the new architecture would work. This helped achieve that goal, but meant post 14.0 any artist jumping into the new system to test were jumping into a beta system that was improving piece by piece, but if you had none of the context of what was just highlighted you would just see a number of nodes not functioning to the level expected.

In retrospect we needed to better separate the architectural focused feedback from the artist workflow focused feedback so that we could better showcase how the tools are being developed for you in the context of a workflow you care about. So now I'd like to take this opportunity to reintroduce artists to the 3D system at a point where workflow focus is the main goal.

The Philosophy

So before I jump into going through the nodes, I just want to share what our philosophy has been for the new system. The driving philosophy has always been that this is a 3D system for comp artists and their specific needs and it should feel like home to them, to you. This means that this 3D system is not intended as a replacement for other DCCs, and even though it's based on a USD architecture, you shouldn't have to know much about USD to use it. It's meant for a comper who needs to do projections for cleanup work, someone who needs to set up matte paintings and relight things at the last minute. For a comper who might need to add some additional particle simulations to complement a hero effect from FX, or even just add small background details to help guide the eye, to better compose the image and tell the story.

For any artist put off by the idea of it being USD, we get that and want you to know that for you this system should pretty much feel like 3D in Nuke always has and that USD is just the underpinnings of greater performance, new collaborative workflows and better rendering. We have worked hard to make this the most comper-friendly approach to USD, so that you can focus on doing what you've always done, using the same workflows you've always known.

We want you to be able to bring in the assets you need no matter the size, say King Kong breaking through a city, and be able to place cards and particle setups within that space to help add dust clouds, smoke trails and more. Grab the eye geometry of King Kong so that you can place a card in front of it with an actresses performance and then place a reflective shader on that eye geometry to render out a new pass that it was too late in the day for another department to give to you and allow you to tell the story through the image how you see best.

Okay thank you for indulging me in explaining what we see for this new system. There's much more beyond this with new technologies to support and better workflows to build that we've had to deal with in the classic system (ModelBuilder comes to mind). But we can talk about that after getting everyone up to speed on where things are now when you open Nuke 17.0.

The Nodes

USD terms and concepts to know

To help with explaining the new nodes in the new 3D system I will at times use some USD terminology. While the new system is designed for all artists and meant to be intuitive for artists not familiar with USD, having some basic USD knowledge will still help your understanding of the new system and will also help when working with other departments working with USD.

This won't be a detailed explanation of all USD terminology, but rather a quick guide for basic USD terms and concepts to help you jump into the new 3D system.

The first USD term to understand is a **Stage**. A stage can basically be thought of as a scene. It holds the entire scene graph and associated data. When working in the new 3D system every node network is essentially a stage

and if you have two node networks with their own 3D data each network will have their own stage. If you then merge those node networks together you merge them to create one stage.

The next USD term to know is a **Prim**. A prim is short for “primitive” and represents an individual object or entity within the scene hierarchy. A piece of geometry is a prim, a camera is a prim, really anything in the scene graph is likely a prim.

For the most part knowing what a stage and a prim is will be enough to understand the new 3D system. Some other concepts that can help though are **Layers** and **Sublayers**.

A single .usd file is a layer. A layer can represent a “piece” of a scene such as a prop, a character, part of a rig, a lighting setup, and so on. It can also represent “defaults” that you can reference in, to create a baseline, for example a layer containing settings for a shot sequence, which you reference in the files representing each shot in the sequence.

Sublayering overlays the contents of the imported file (or layer) over the existing content. So essentially you are combining layers together, some on top, or some underneath to build up your stage.

When combining layers, you might sometimes combine layers which have the same prim in both layers, but with different information, say a Camera with different position or animation data. To decide which layer’s data we pull from, USD has a concept known as “opinion strength”.

As you might expect, the layer with the strongest “opinion strength” will determine the data that is used for this prim that exists in both layers.

This is really powerful and at the heart of why USD can be so collaborative. It allows you to non-destructively make changes, as your changes can live on their own sublayer, while allowing other artists’ work to live on a different layer and when you combine these layers together to build your stage you get the result based on the strongest opinion with no data loss.

USD has two different approaches to “importing” the contents of another USD file: **Sub-layering** overlays the tree of the imported file over the current tree, while **Referencing** attaches the contents of the imported tree as a branch on the existing tree.

You can think of sublayering as composing different versions of the whole scene (for example, overlaying the lighting department’s version of the scene with final lighting over the layout department’s version of the scene with scratch lighting), and referencing as adding a part to the scene (for example, importing props to begin layout).

When importing you might also notice the option to load or not load payloads. In USD a **Payload** is essentially a reference you can choose not to load, making it easier to just bring in the parts of the scene you care most about.

In the scene graph you will also see columns for Type, Kind and Purpose.

The **Type** defines the type of primitive a prim is. Common types are Meshes (your usual geometry), Cameras (your usual cameras), Xforms (like a group that can be transformed) and Scopes (a group that can’t be transformed).

The **Kind** represents a way to classify your prims, and in Nuke can help drive different selection groupings.

The **Purpose** determines the render purpose set for the prim, allowing you to work with geometry or prims only intended for previewing that don’t get rendered, as they have not been set for a render purpose.

There is much more to USD and if you would like to know more then we have USD-focused content on the Foundry Learn page, with a short form video on USD called Foundry's Guide to USD and while the following video series is due a refresh the content is mostly still relevant for Nuke 17.0 albeit out of date in some parts: Introduction to the New USD 3D System.

I also think there are great training resources from other DCCs that help give more information on USD and even though it is many years old at this point I still really like the 'What is USD: A Primer | Rob Stauffer | SIGGRAPH 2019' video on YouTube.

Nodes for creation

Create: GeoCard

GeoCard is pretty much what you'd expect for the most part, though we have been updating the UI over the course of development. Originally a number of new USD attributes were displayed which increased the size of the properties box making it harder to access the transform controls. These have now been organised and collapsed to ensure the thing you do most is front and center, but if you are interested in the new knobs these are what they do.

Path: sets where your card lives in your scene graph hierarchy and defaults to the name of the node so you can copy and paste, or place nodes how you want without having to worry about naming or organisation. If you do however want to organise your card in your scene graph you can create parent Xforms in the Path knob and those will be automatically created with the 'Create missing parents' knob checked on and you can decide if you want those to be Xforms or Scopes.

- An Xform is like a group that you can put transformation data on and a Scope is a group that you can't put transformation data on.

The Primitive attributes gives you access to some USD attributes, which you don't have to worry about if you just want to use Nuke like you always have, but if you want some extra control then the knobs work as follows:

- **Kind** lets you determine the selection category for this card. This can be used to refine your selection based on kinds in the 3D viewer selection controls:
    - **Assembly**: an important group model, often a published asset or reference to a published asset
    - **Group**: models that simply group other models.
    - **Component**: a "leaf model" that can contain no other models
    - **Model**: base class for all model kinds. "model" is considered an abstract type and should not be assigned as any prim's kind.
    - **Subcomponent**: an identified, important "sub part" of a component model.
- **Purpose** sets how your prims are displayed in the viewer allowing you to define the viewer render quality:
    - Default: Assigns the purpose based on the Layer settings, otherwise defaults to Render
    - Render: Assign to prims intended for final high-quality rendering.
    - Proxy: Assign to prims intended for lightweight proxy rendering.
    - Guide: Assign to prims intended as markings, lines, arrows or interaction guides.
- **Visibility** sets if your geo and any descendants are displayed as hidden in the viewer:
    - Inherited: When enabled the visibility of the selected prim is the same as the prim's parent.
    - Invisible: the selected prim and its descendants display as hidden in the viewport

Under Mesh Attributes you have the following controls:

- **Subdivision**: sets how meshes are rendered in the viewer based on different subdivision models
- **Boundary**: specifies where the boundary conditions needed for interpolation are defined
- **Double Sided**: lets you shade both sides of your geo

And **Display** concerns itself with how your card is displayed in your 3D viewer

- **Display Color** allows you to colour your geo and **Display Opacity** sets how transparent it is, but know these are just for viewer display not rendering

Also new to GeoCard and a number of other nodes is the updated Snap menu. This was redesigned to make it a bit more discoverable for new users and with some added functionality for experienced comp artists. You can now switch between snapping Geo or snapping Pivots. The position, orientation and size naming convention has been updated to T,R,S to better align with other naming conventions in Nuke and are now checkboxes so you can toggle on and off what you need and when.

The first icon in "Geo to" mode is the 'snap to selection' knob and will snap your GeoCard to another piece of geometry where you have selected a vertex point. Please note that for each checkbox you have enabled you

need an additional vertex selection to help calculate the correct snap. So if just T is checked I only need one vertex selection, if T + R is checked I need two vertex selections and with T + R + S I need 3 vertex selections.

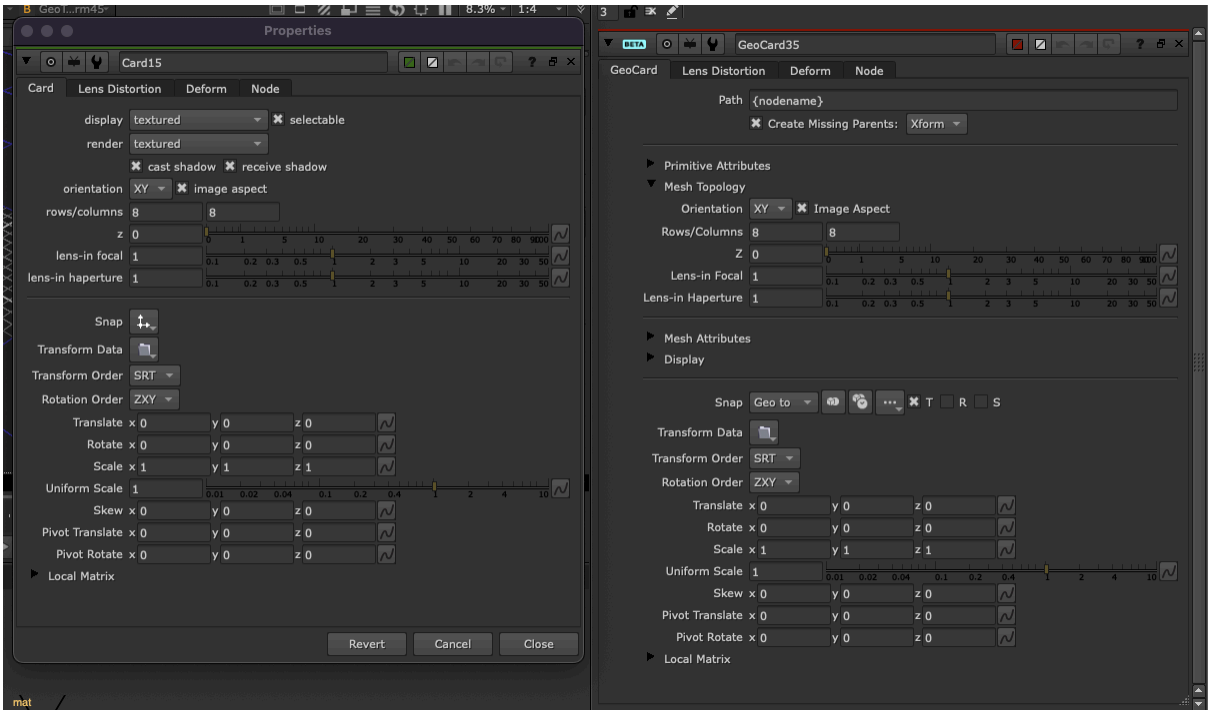
The second icon in “Geo to” mode is the ‘snap to animated selection’ knob and will allow you to perform your snap over a desired frame range and create a new key at each frame so your Card will stay snapped to a moving piece of geometry. Over large frame ranges this won’t be the most efficient method and is where using the new geoConstrain node might be more beneficial.

The third icon with the three dots is a new custom knob-and can be populated as TD’s see fit.

Switching the Pivot to mode provides the same options for your cards pivot control, but also includes an additional knob that allows you to set your pivot to the bounding box of a piece of geo in your scene. This is great when a pivot appears at the root of a scene but you want it one the geo you have selected. Just click the geo you want the pivot on and select where in relation to the bounding box you want the pivot to be.

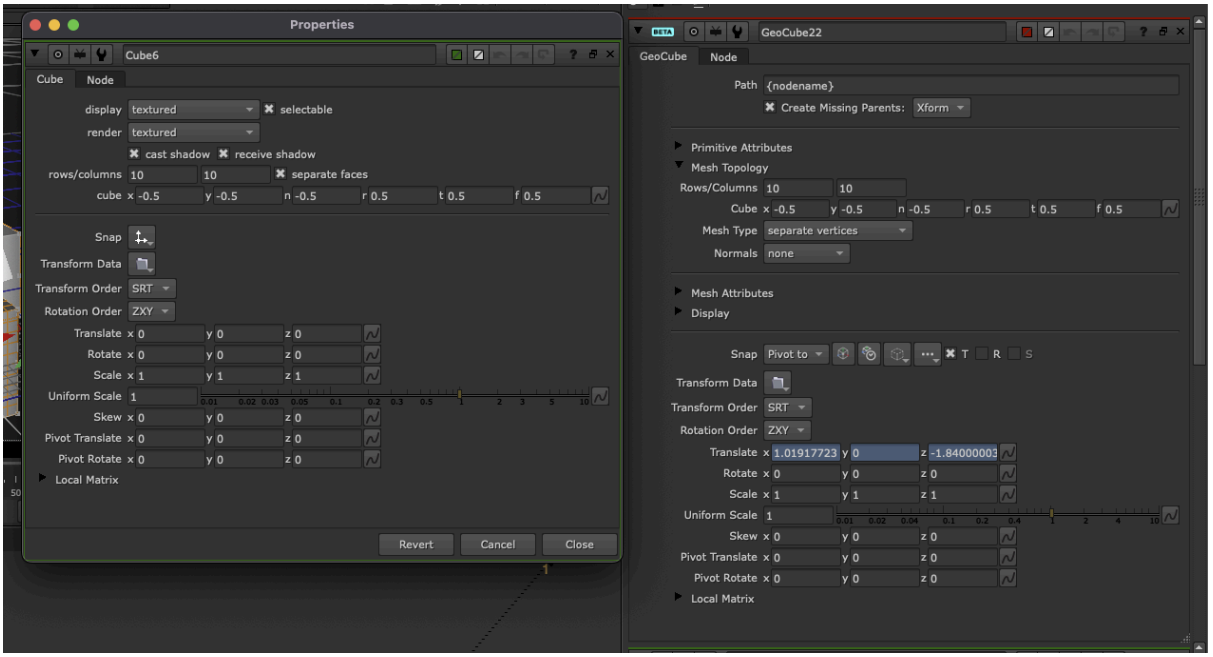
The Mat input is the equivalent to the img input, but you can shaders and materials directly in also.

The display, render, cast shadow and receive shadow knobs have been removed from the nodes at this point in time, but new lighting and shadowing options will be introduced in an upcoming open beta as will the selectable control.



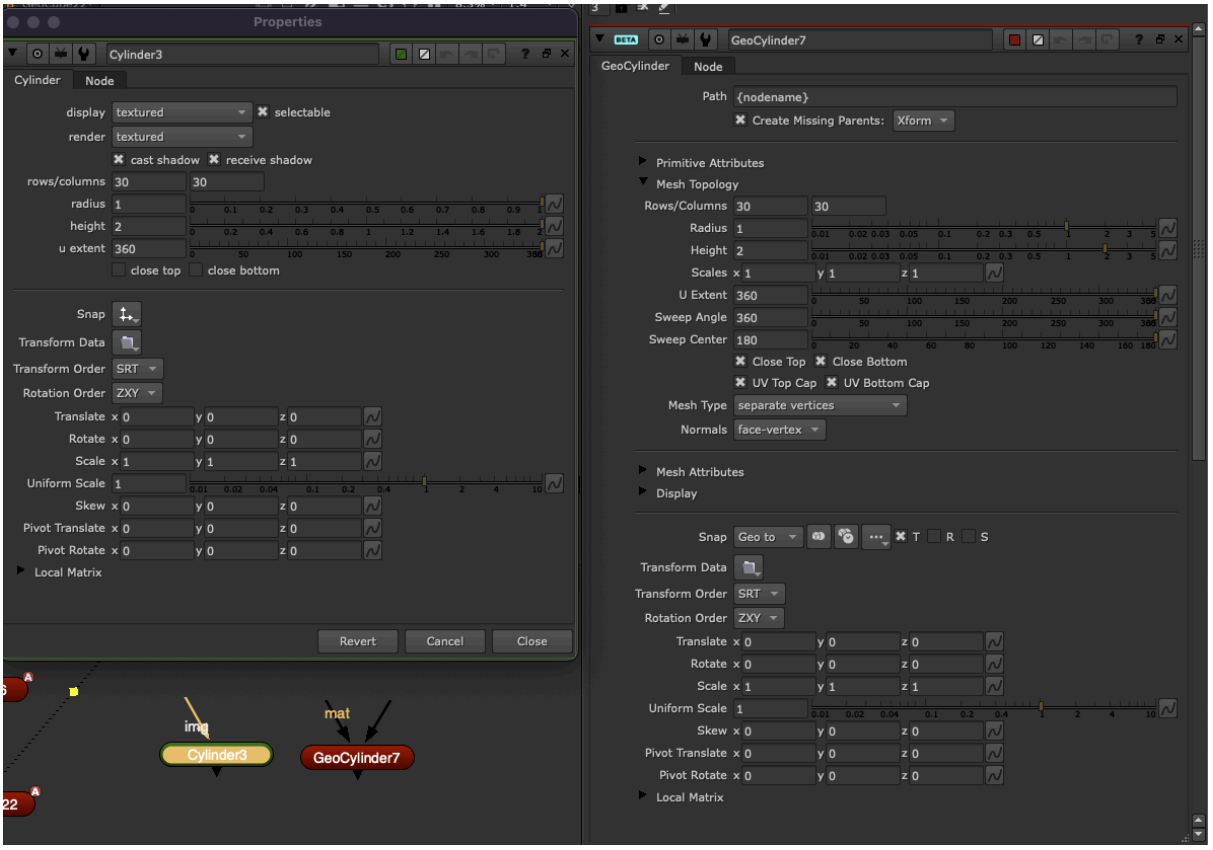
Create: GeoCube

The new knobs are predominantly the same as GeoCard with a few additional USD attributes that determine how the cube is created via Mesh Type and the Normals knobs.



Create: GeoCylinder & GeoSphere

Likewise GeoCylinder and GeoSphere have pretty similar knobs, but with some additional USD attributes that help determine how the Cylinder and Sphere are constructed and give a bit more control compared to the classic nodes.

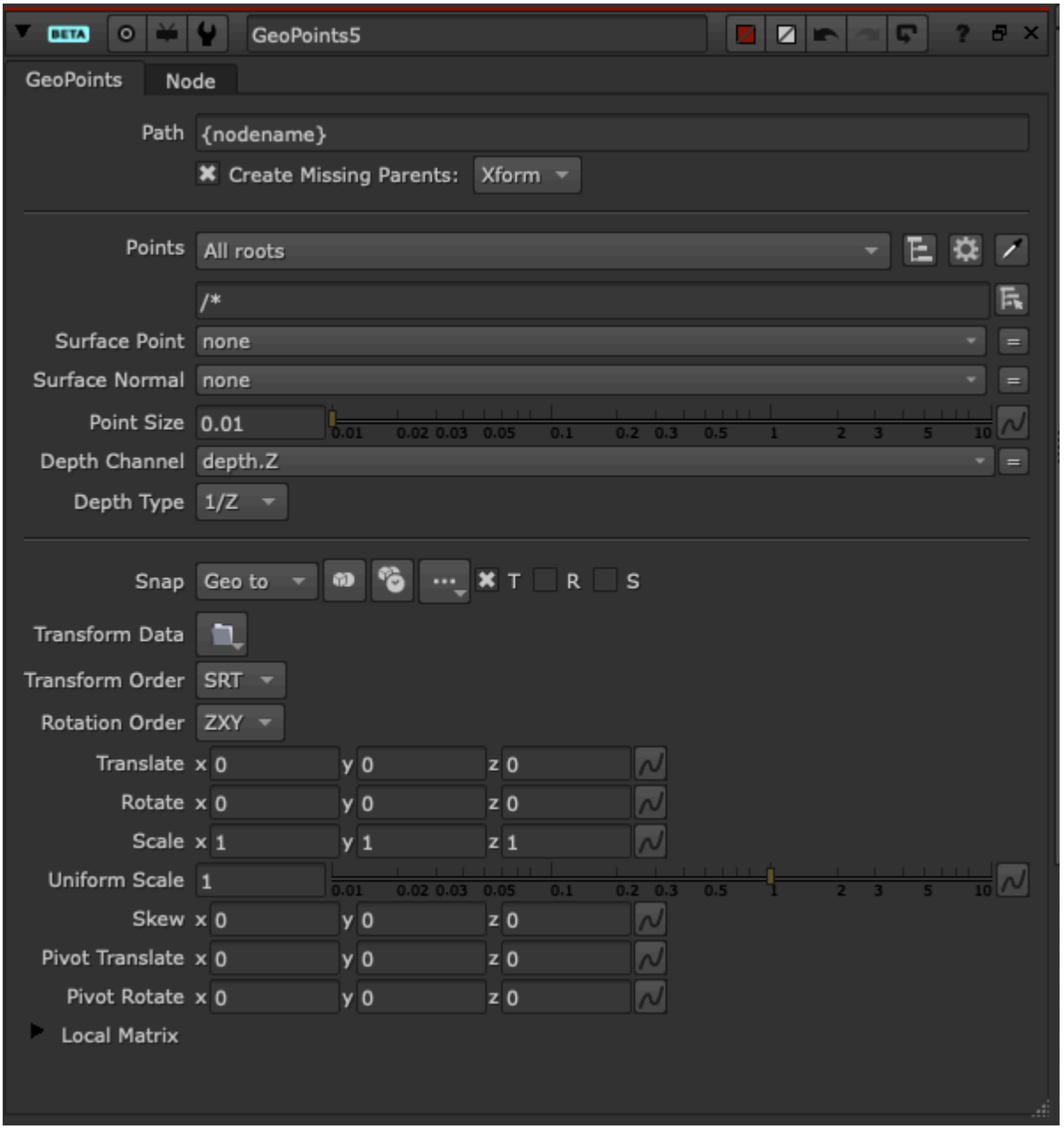


Create: GeoPoints

GeoPoints creates a point cloud from its input and performs similar functions to the classic 3D system DeepToPoints and PositionToPoints nodes, but wrapped into one. You can also create points from other types of input, such as meshes and particles, giving even greater options for comp artists and will better tie into future particle workflows.

The Points knob currently defaults to last modified which can lead to nothing being generated if the node above is say a GeoTransform, so will look to change the default behaviour to either All Roots, or All Meshes in order to turn your input into a point cloud. The surface point and surface normal knobs are for inputting channels that contain either position or normal information and you can also populate this information using the two additional input pipes on the node, though these are currently in need of some additional work, as it can be confusing to know which input pipe to use.

You can then adjust the point size to your liking and often you will need to as the default point scaling can get blocked by the mesh making it look as if no points have been generated.



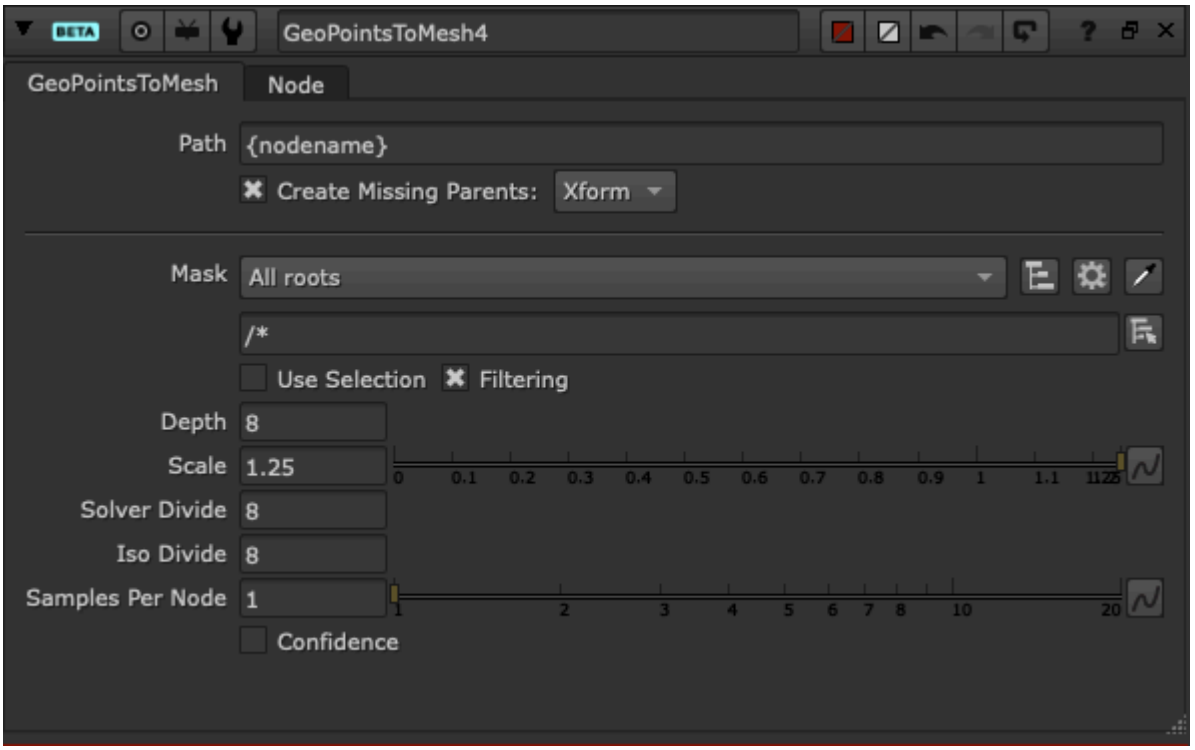
Create: GeoPointsToMesh

The GeoPointsToMesh node uses geometry normals to generate a mesh from a point cloud using Poisson Surface Reconstruction and is the equivalent of the PoissonMesh node in the classic system.

This node requires further development and does not function fully at the moment, so will be looked at for an upcoming release.

The attribute controls essentially allow you to select what point data you want to turn into a mesh and has the equivalent controls as PoissonMesh for controlling how that geometry is output.

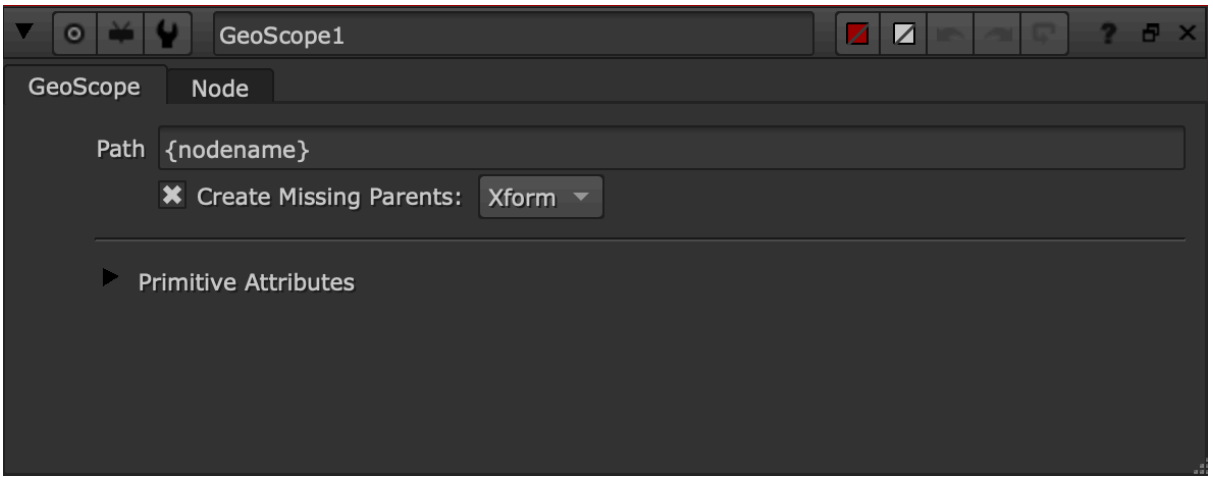
This node requires more work and in general geometry creation workflows such as these, a replacement for ModelBuilder, or even newer workflows around geometry generation are all areas we want to build towards to give comp artists greater tools to turn image or point data into geometry that can be used to project on, or for light bounces etc.



Create: GeoScope

GeoScope introduces the ability to create the simplest form of grouping, which is called a Scope. Unlike an Xform a Scope does not have a transform. Scopes can be useful for organizing the scene tree.

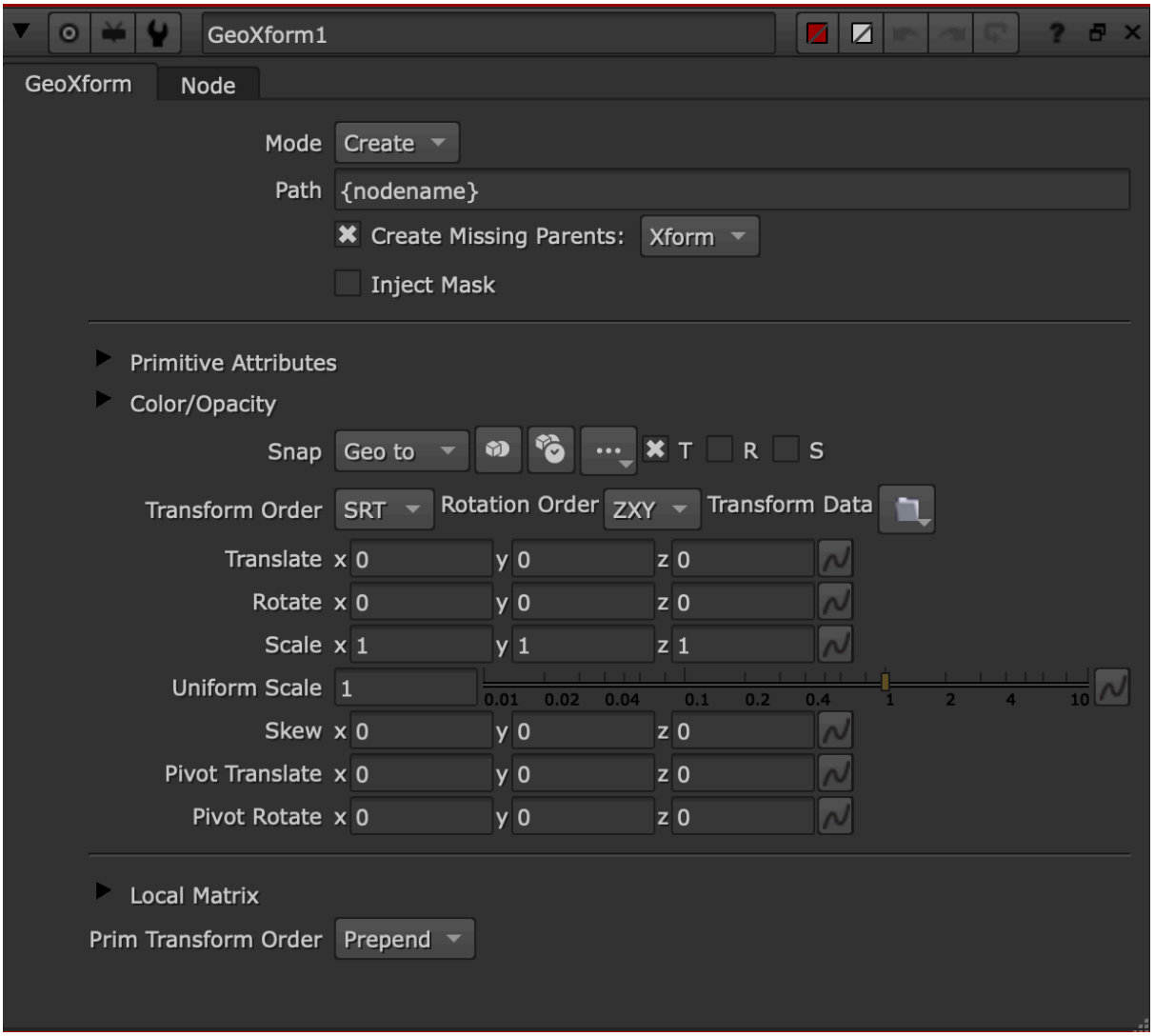
There is a little bit more work we would like to do here to introduce the create and edit mode functionality that we have in the GeoXform node.



Modify: GeoXform

GeoXform introduces the ability to create Xforms which are a form of grouping and these Xform prims that are created can also be transformed themselves. This is great if you want to nest a bunch of geometry, say cards under one group and then transform that top level group Xform so that all the geometry that are nested underneath move also, but retain their own unique transform data.

There's also a new knob we haven't come across on other nodes yet called 'Prim Transform Order', this allows you to specify at which point the transform order is specified to

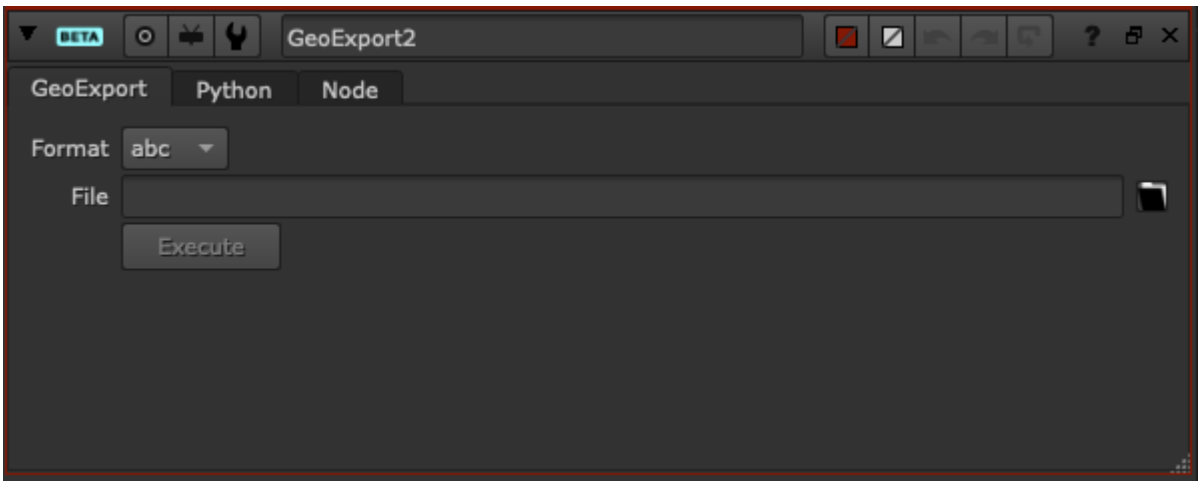


Nodes for importing and exporting

Create: GeoExport

GeoExport is currently a very basic implementation at this point in time and exports either a flattened USD layer, or an alembic file.

We're aware of some inaccuracies in the USD file created and upcoming work on this node will include the ability to export sections of your script as sublayers, the ability to define how Nuke materials in your node graph are exported, ability to just export the changes made on the USD stage to a new layer and greater control over how your comp layer is exported from Nuke in general.



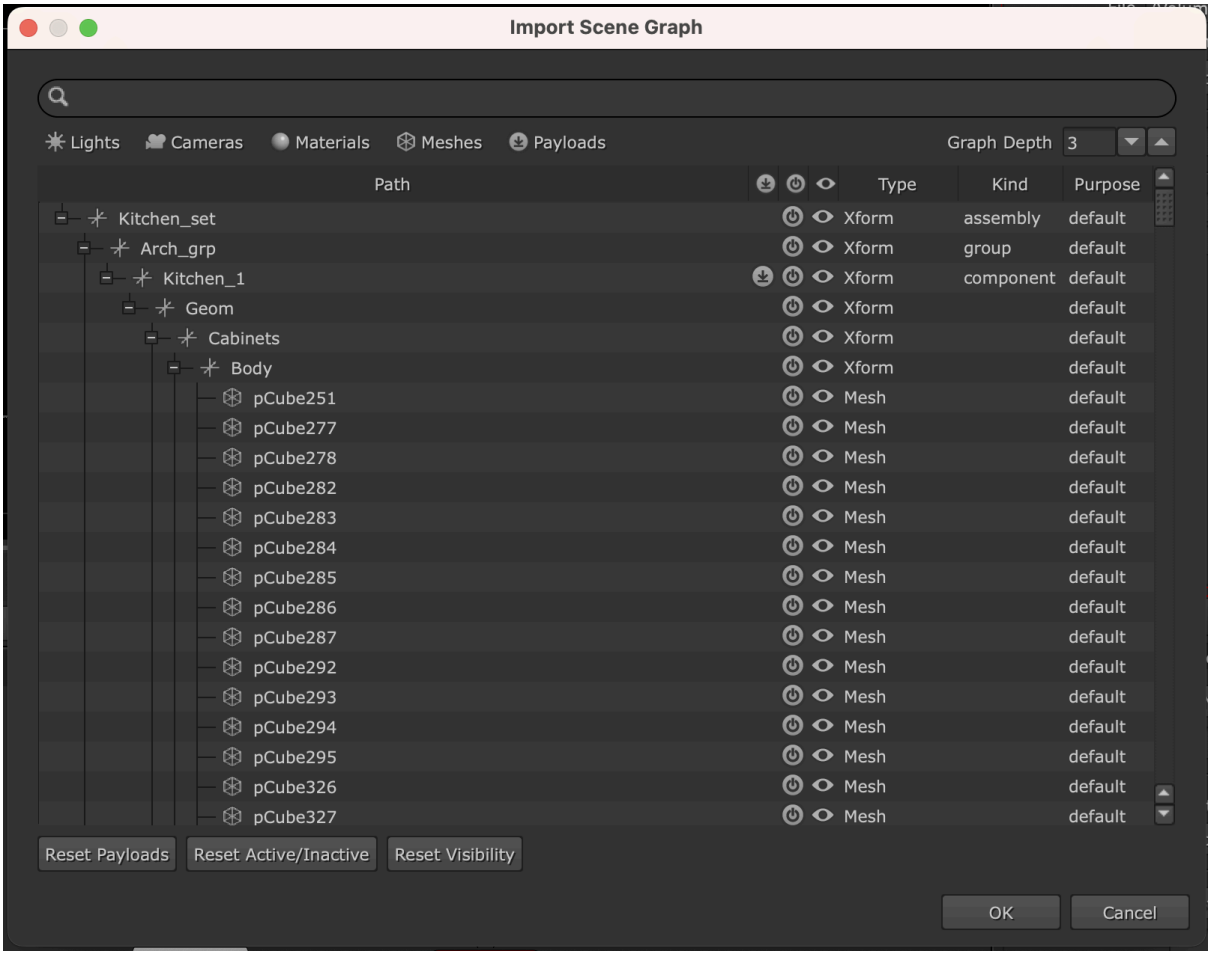
Create: GeoImport

The GeoImport node has had some major revisions recently, allowing artists to easily search a scene's continents, filter it, enable or disable prims, and control payloads, all before paying any viewer render costs.

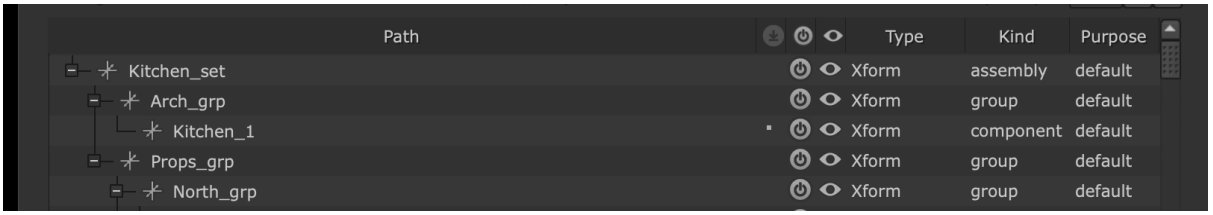
Currently there is support for USD and Alembic files (though we are aware of some alembic bugs and have been addressing them for open beta builds).

The starting point for many of these workflows will be importing the data you need to work with, whether that be geometry, cameras, lights, materials or a combination of these elements and doing so efficiently.

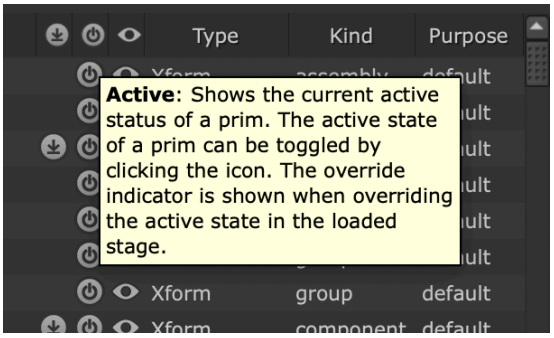
With this node you can now make import selections in a new pop up dialogue box, which means you don't have to pay the cost of interpreting imported data in the 3D viewer like in previous releases, drastically speeding up import workflows.



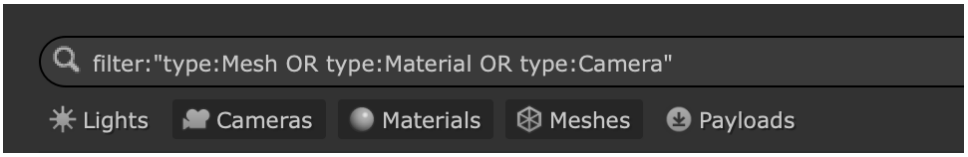
From this pop up dialogue you have the ability to load and unload payloads that have previously been set up, making it easier to load and unload large chunks of a scene to help artists just load the parts they need.



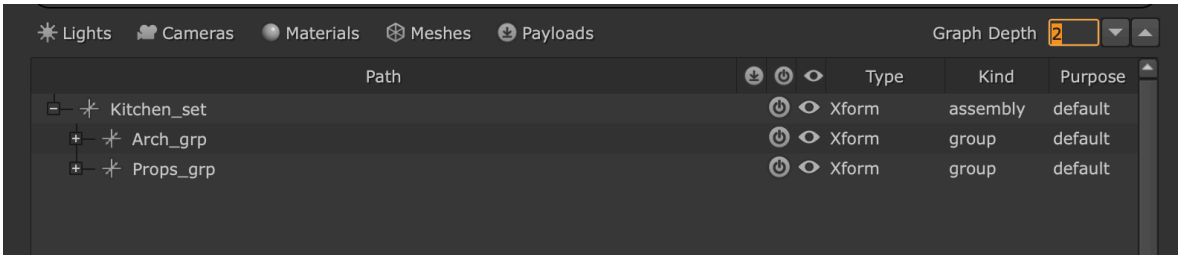
Individual or groups of prims can also be activated or deactivated, meaning you can quickly search through your scene, find any geometry, lights, or cameras you don't want and deactivate them. Or combine this with the payload workflow to grab the payload of a character you want to work with, but deactivate any materials associated with that character as you won't need them for your shot.



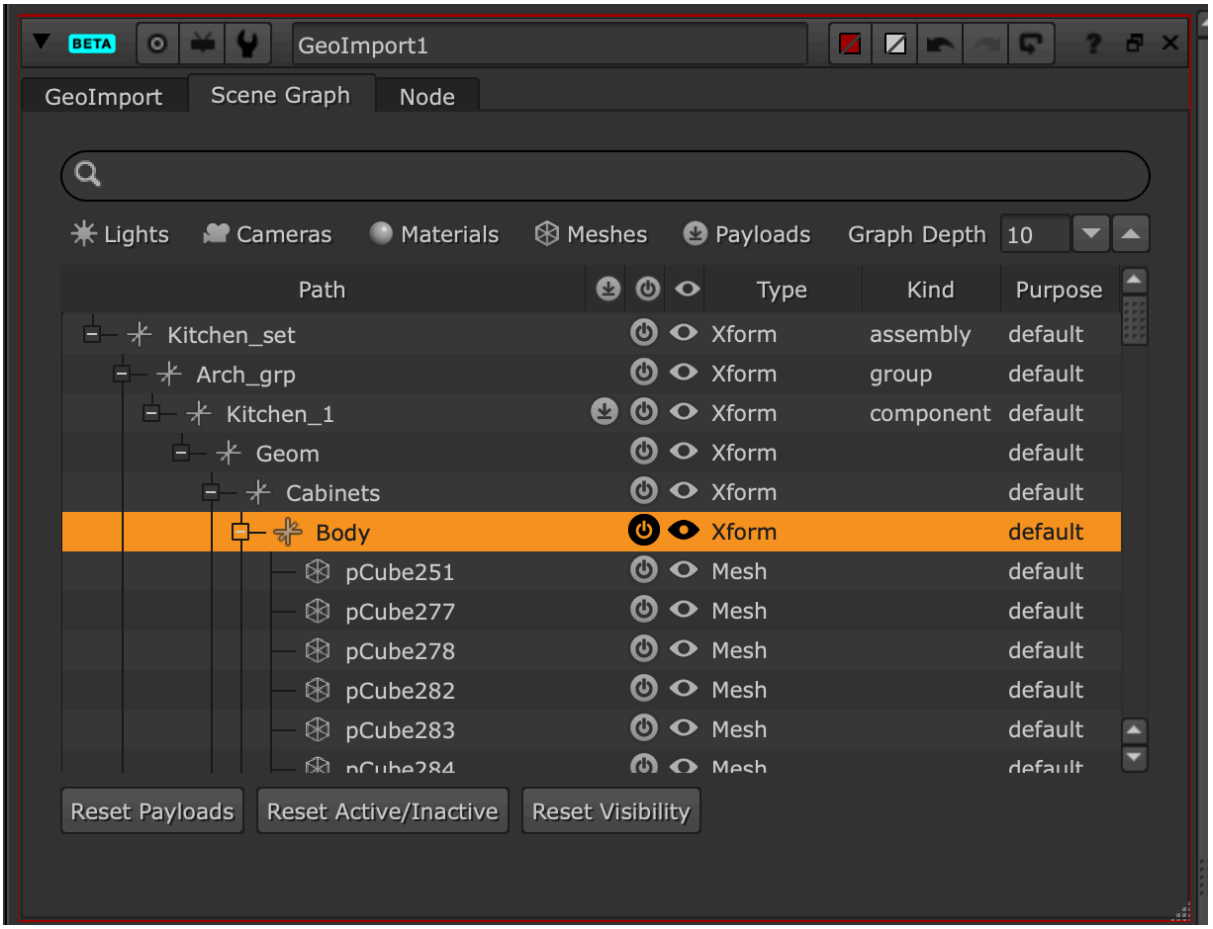
To make searching easier, you can filter your pop up scene graph by searching for keywords, or you can use the new filter types to quickly show only the cameras or materials in a scene and then activate or deactivate them all at a click of a button using the column controls.



As the new 3D system allows you to bring in even larger, more complex scenes than ever before, navigating through them can sometimes be time consuming when having to scroll past hierarchies with lots of children objects inside. To help navigate the scene graph you can quickly set the graph depth that is displayed helping you focus on the upper hierarchy levels where organisation tends to happen and then quickly increase the depth as you choose to navigate further into the scene.



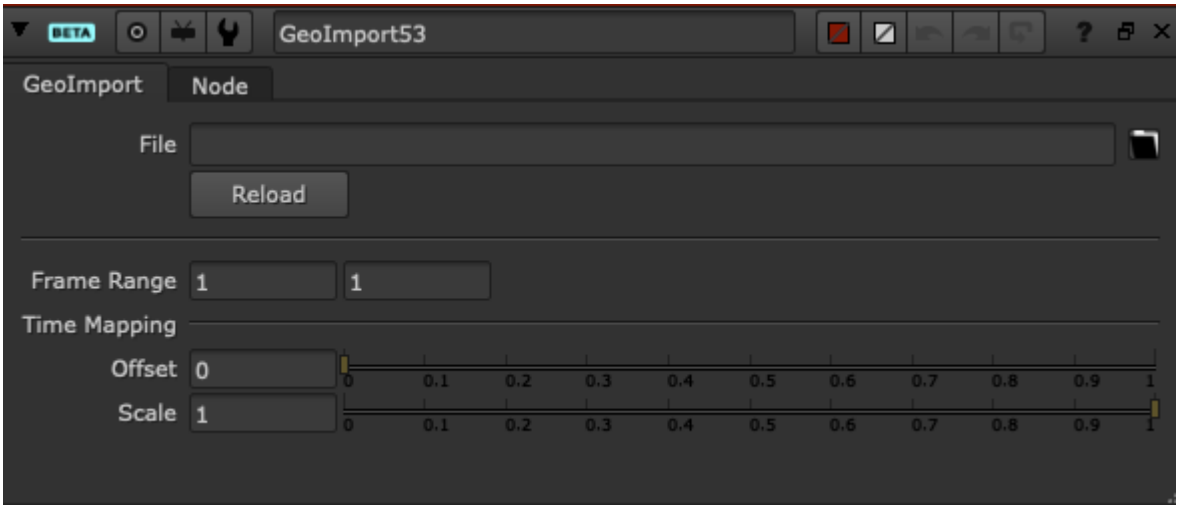
Once you have made your selections for your import you can always revisit and update them in the new built in scene graph on the GeoImport node. This gives you the flexibility to quickly make changes to what you have imported and the results can be easily visualised in the 3D viewer helping you to make informed choices when exploring a scene which may have some unknown elements



Upon load you can set frame range values to be non zero so each frame is evaluated and for animation data within an imported scene you can

Offset - Offsets the USD prim animation by this amount
Scale - Scales the referenced prim animation by this amount. Values between 0 and 1 speed up the animation and values higher than 1 slow it down.

For those familiar with USD GeoImport in essence acts as a node for importing Sublayers, allowing you to merge multiple layers together to build up a stage in Nuke. If you want to instead import data as a reference then the GeoReference node is the one to use, as this will allow you to add onto the stage.



Create: GeoReference

GeoReference is a companion node to GeoImport, but it is different in that it lets you import the same prim more than once.

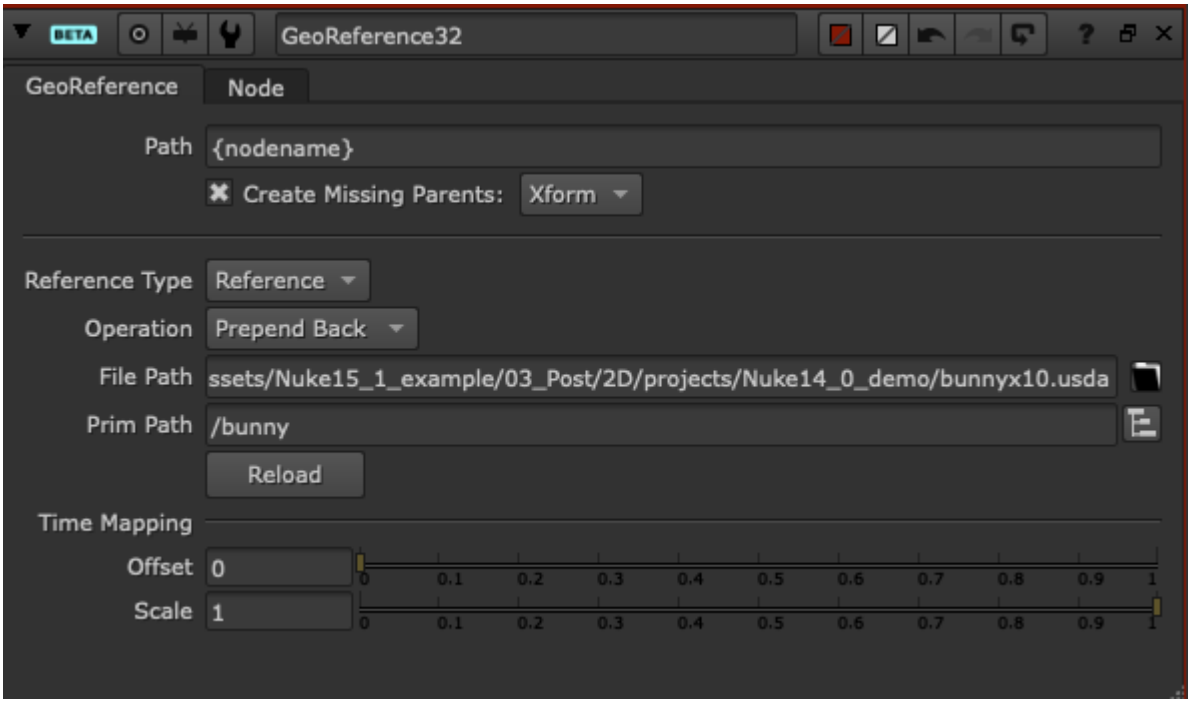
USD has two different approaches to “importing” the contents of another USD file: sub-layering overlays the tree of the imported file over the current tree, while referencing attaches the contents of the imported tree as a branch on the existing tree. GeoImport does the former while GeoReference does the latter.

You can think of sublayering as composing different versions of the whole scene (for example, overlaying the lighting department’s version of the scene with final lighting over the layout department’s version of the scene with scratch lighting), and referencing as adding a part to the scene (for example, importing props to begin layout).

Some of the knobs on this node are ‘Reference Type’ which determines the type of composition arc you want the reference you are importing to be. A composition arc is a USD concept that defines different ways to combine layers or parts of layers to construct a stage. The most common arc types are: Sublayer, Reference or Payload.

The Operation knob lets you set where to place the reference in the reference list. Essentially you are attaching your referenced layer to the rest of the stage.

The File path lets you pick the file you want to load and then the prim path needs to be populated so you are specifying what geo, or prim data you want loaded.

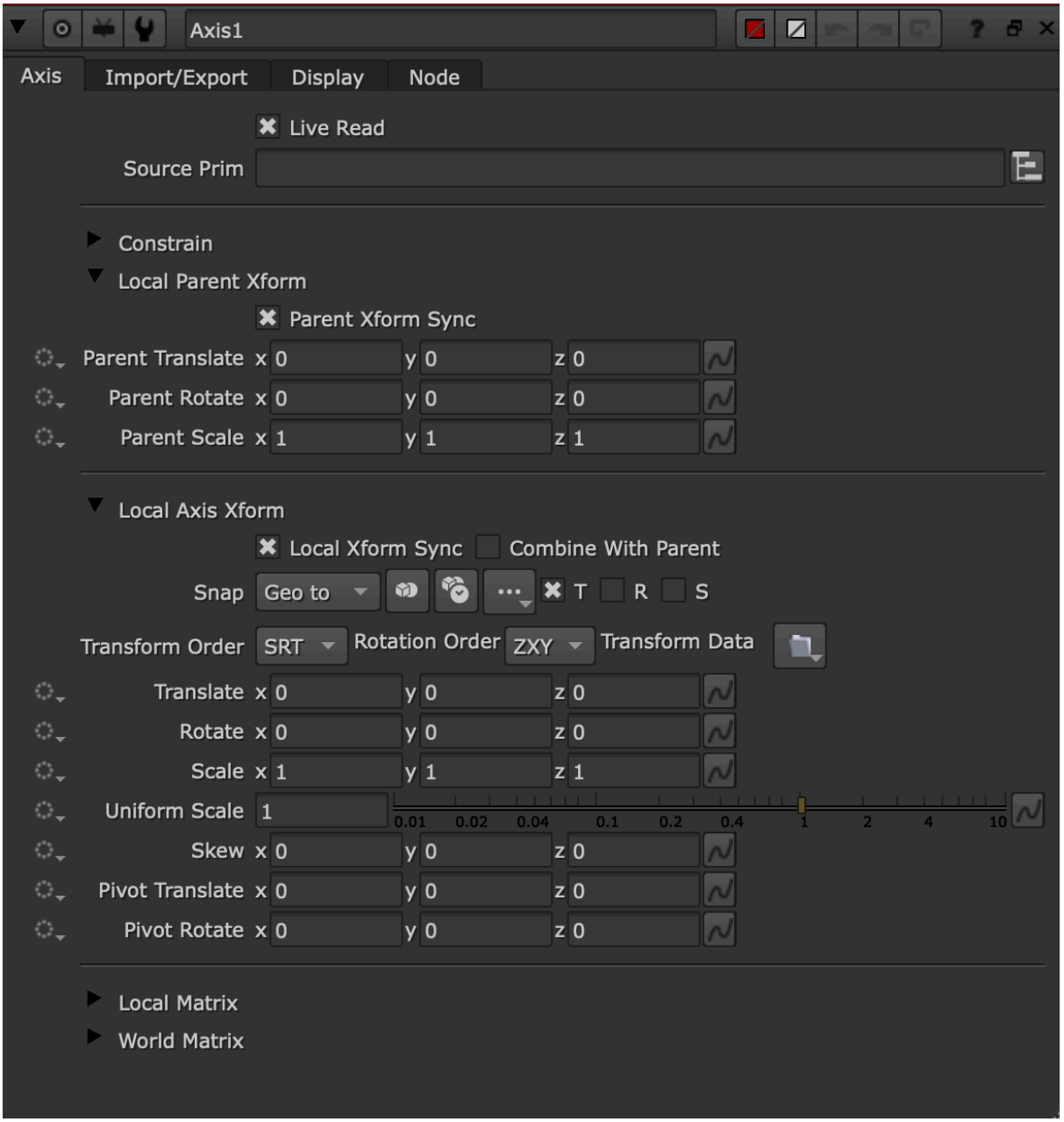


Camera and Axis workflows

General: Axis

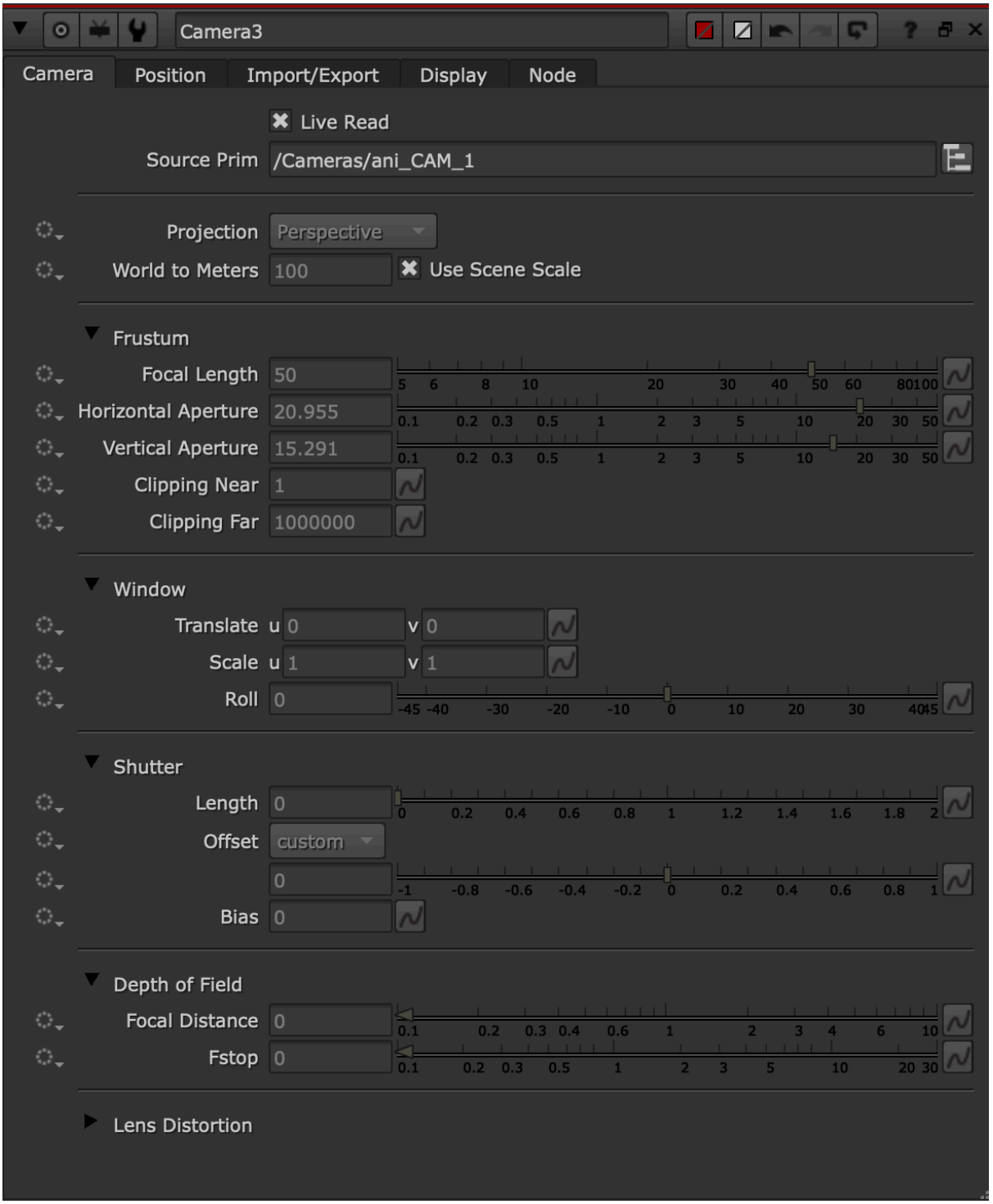
The Axis node has seen some expansion allowing for greater control on top of all the functionality you are used to.

A recent update has been the ability to pass USD prim data into the Axis node to allow the live data of that prim (such as its animation data or transformation data) to be passed through the Axis node to be used anywhere in Nuke that can connect to an Axis node. This means you can drive transformation data of particle simulations using the classic Particle nodes with animation data within your stage, allowing you to fully use all 3D nodes in one node graph.



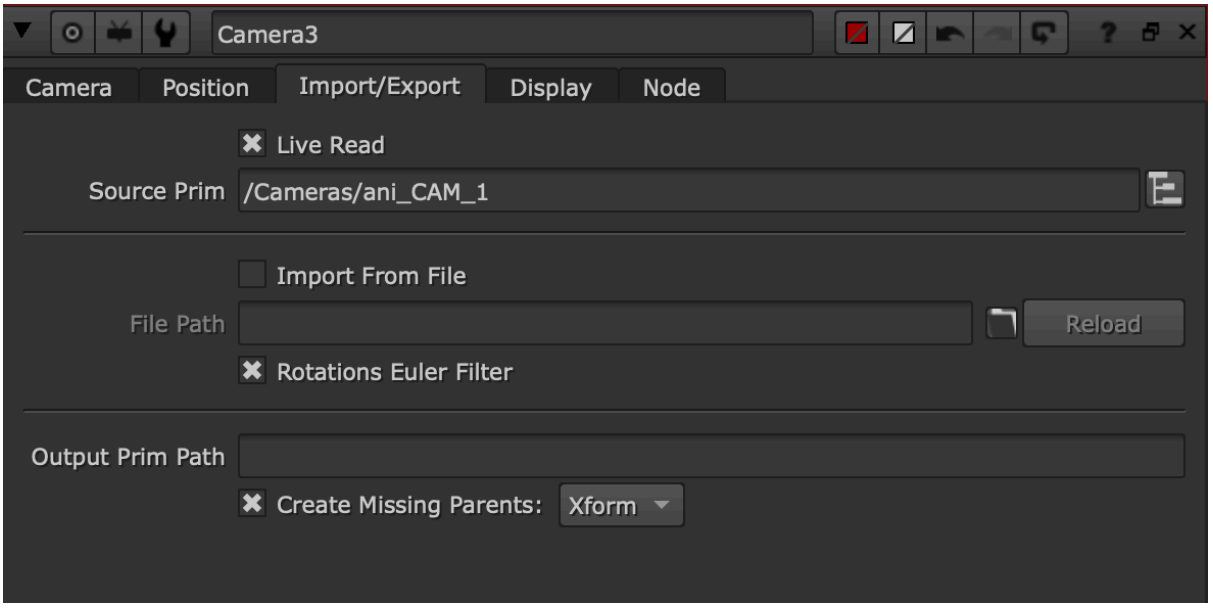
General: Camera

The Camera node now has been updated to be able to pull in live attribute data from a USD camera prim. Previously this data had to be baked which limited what Nuke's Camera node could do with USD camera prim data, whereas now all of the data can be pulled live from the USD file allowing artists to simply pass the data through the node, or to go in and start editing the data.



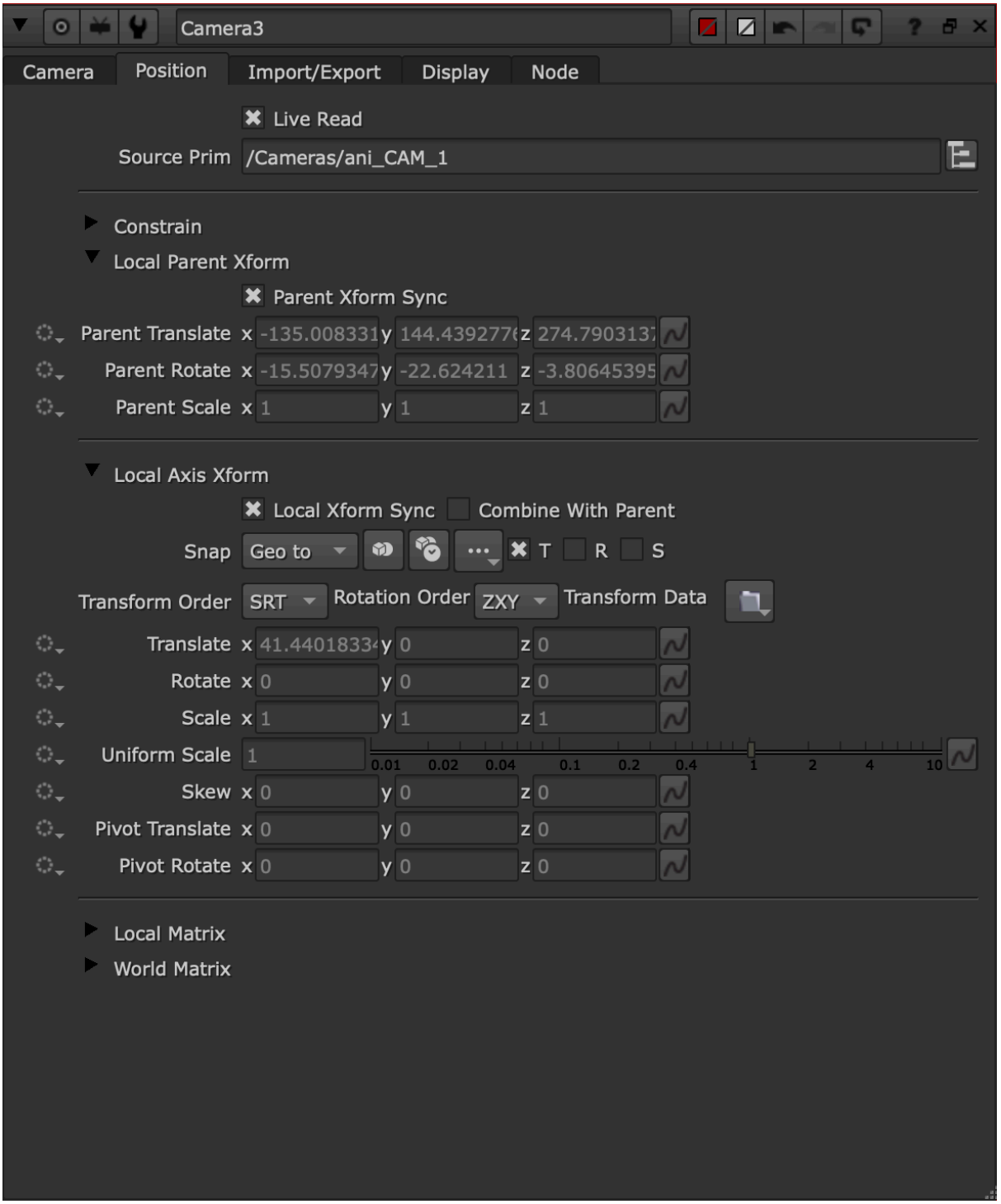
The Camera node like Axis has had some additional controls added to give users even greater functionality when working in the new 3D system.

For bringing camera data into the node, you have two workflows in the Import/Export tab, you can either import a file from disk using the ‘Import from file’ knob, or you can connect the scene input to your stage and populate the ‘source prim’ knob with a camera that is already in your scene.



When importing camera data, you may be importing a camera which has transformation data on the camera prim itself, but also has transformation data on the Xforms that the camera lives under. The new Position tab contains ‘Local parent Xform’ knobs that can contain the transformation data that is separate to the camera.

The Local Axis Xform section is essentially all the same data you are used to from the classic Camera node.



Camera workflows between classic and new

Hopefully a lot of the work that has gone into the Camera node will be invisible to users, but the work that went into rearchitecting the camera node so that it could pull in ‘live’ USD camera prim data and be able to connect to all the existing 2D and 3D nodes that have a ‘cam’ input we felt was too valuable to lose for the new 3D system. Likewise retaining all the unique projection modes that only exist on a Nuke camera node we also felt was massively important, as this would have been lost if we only supported USD camera prims, as they do not support all the projection modes that Nuke's Camera node offers.

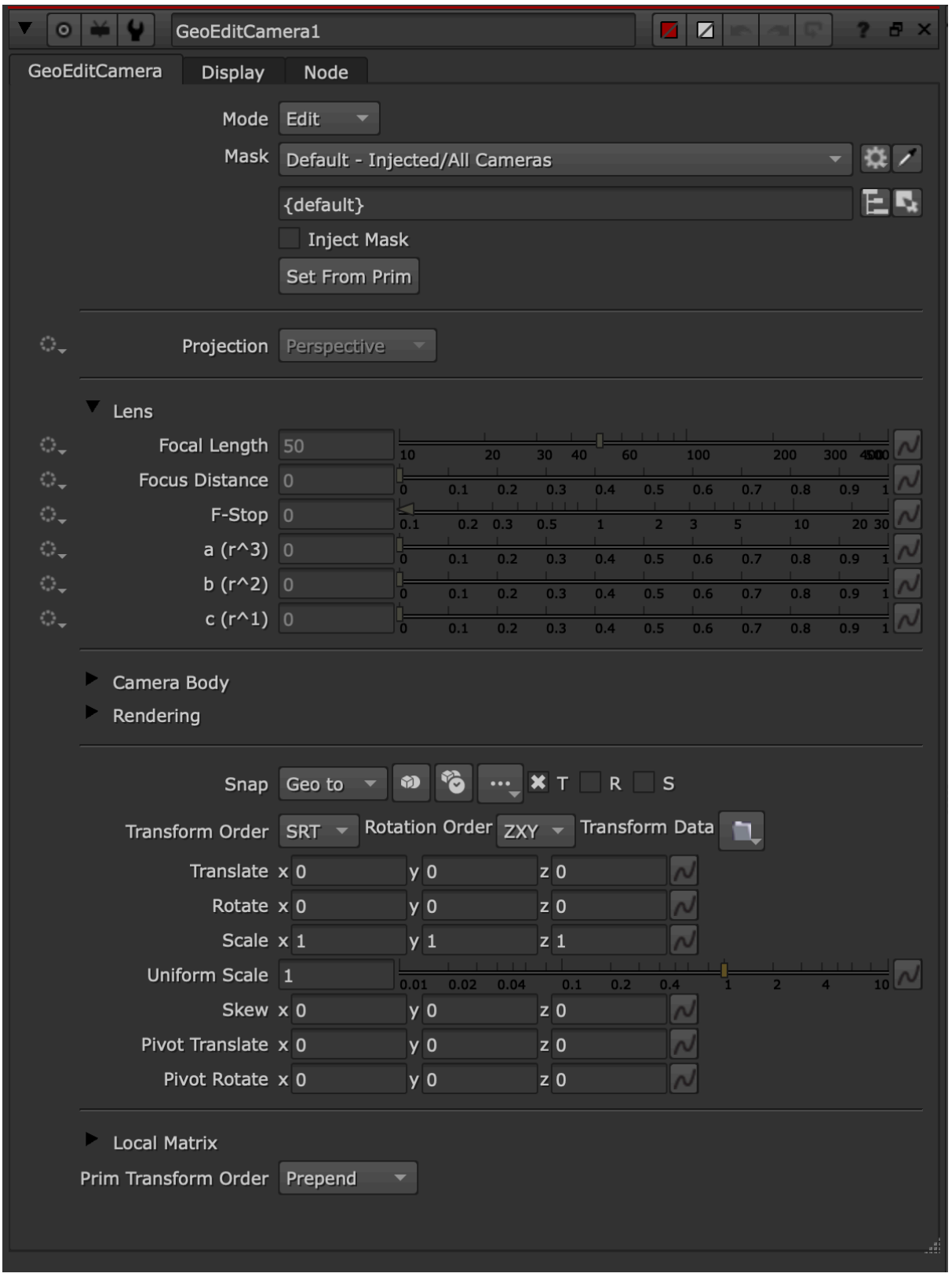
Another benefit of this work is that all of your Camera and Axis nodes are compatible between both the classic and new systems in the 3D space, allowing for easier transfer of data.

However one thing to note is that when working with the Camera node, you are essentially always going to be creating a duplicate camera of the one that exists in your USD stage.

If ever you want to work directly on a USD camera, to transform it, change its attribute data then this is where the GeoEditCamera node comes in.

GeoEditCamera

For workflows where you want to directly edit a USD camera being imported to your stage, we have introduced the GeoEditCamera node. This node allows you to load an imported camera from your stage and edit the values of that USD camera directly inside of Nuke, so you no longer have to duplicate it in a Camera node.



Lights: Overview

The light nodes have been on a similar architecture journey as the Cameras and Axis nodes, as they are all based off of the same node type, with the same connection functionality to the rest of the nodes in Nuke via, ‘cam’, ‘axis’ and ‘light’ inputs.

As such we have also worked hard to rearchitecture the Nuke light nodes to retain all the workflow features comp artists are used to, while introducing new functionality and allowing for new workflows that allow editing of the same USD light types across DCC's.

You now have access to Nuke light nodes (Direct, Spot, Point and Environment) which all have been updated to reflect the attributes of USD lights, while still retaining all the functionality of what makes Nuke lights unique (unique render controls for ScanlineRender2 with falloff controls, or ability to connect to 'axis' and 'light' inputs for driving other geometry or relight workflows). This means that when you connect them to your stage, they will generate USD lights, but will also allow for unique rendering options for ScanlineRender2. The only thing to be mindful of when working with these lights is that like Nuke's classic lights, these won't be affected by a GeoTransform node.

But now you also have access to new light nodes that create USD light prims directly (GeoDistantLight, GeoDiskLight, GeoSphereLight, GeoDomeLight). Each of these lights are 100% USD light types with the same attribute data that you will find in other DCC's meaning that you can create and edit the same lights across multiple DCC's. These nodes can be switched between create and edit modes, allowing you to import a stage with USD lights and then edit these lights directly in Nuke.

In an upcoming release we will be introducing a GeoEditLight node which will unify to the editing functionality into a single node for easier use cases

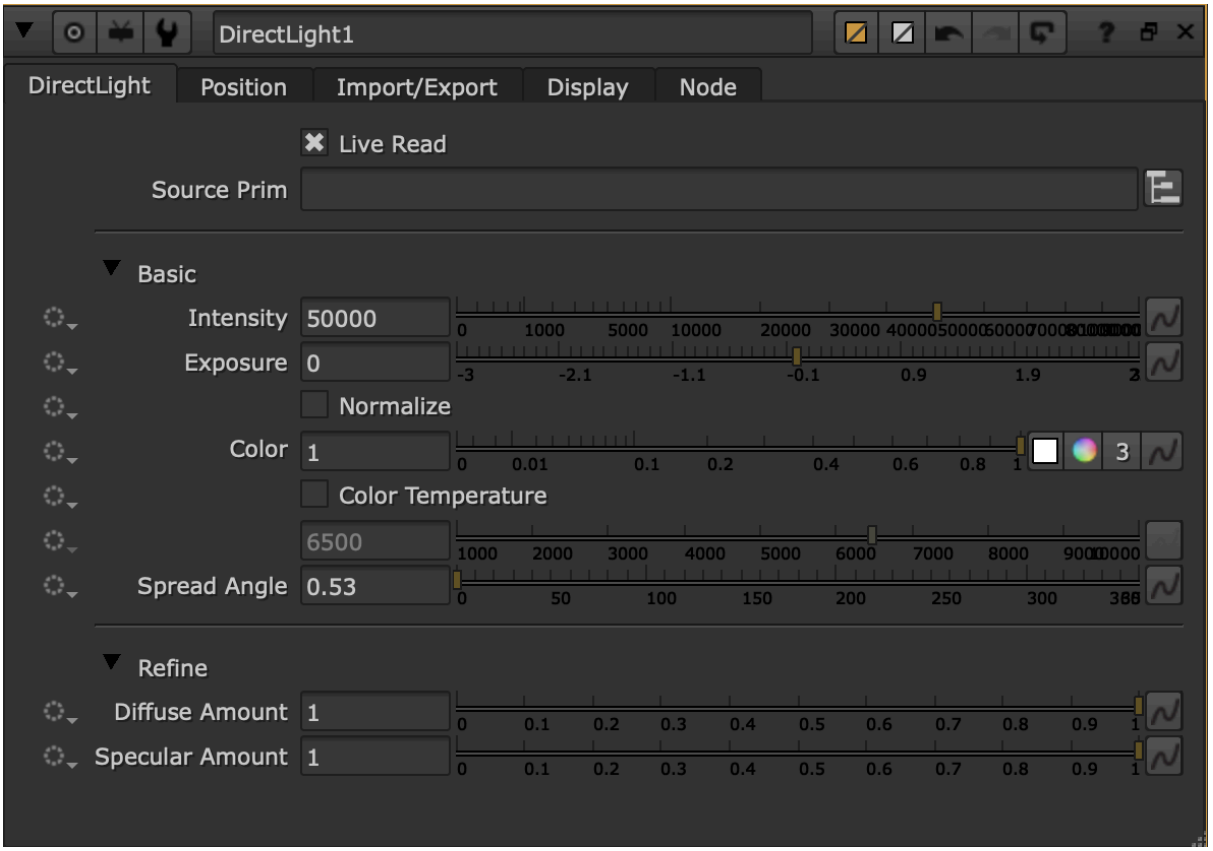
DirectLight

Looking at the DirectLight there are knobs consistent with the classic DirectLight and some new ones added also. You can set the intensity of the light, which scales the power of the light linearly and the colour of the light, just as you could in the classic node. But now you can also set an Exposure value which scales the power of the light exponentially as a power of 2, similar to an FStop control. You can choose whether you want the exposure to be connected to the scale of the light in the scene or not with the 'Disconnect Area/Power Tracking' knob, so by default with this off scaling the light does increase the power of the light. However not all lights respect this knob and DirectLight is one of them.

The 'Diffuse Weight' knob is a multiplier for the effect this light has on diffuse response of materials, so changing this essentially makes a diffuse shader respond differently to the light. Likewise the 'Specular Weight' knob does the same for the specular response of materials.

Users can now set their light colour based on colour temperature and control spread angle for helping define the shape of shadows.

As you can see that as an axis op it shares many of the same new knobs as the Camera node, as it is based on the same Axis op class. Many of these knobs will make sense to keep, but others we will look to either remove, update, or minimise their impact on the UI so the properties are as intuitive as possible for artists.



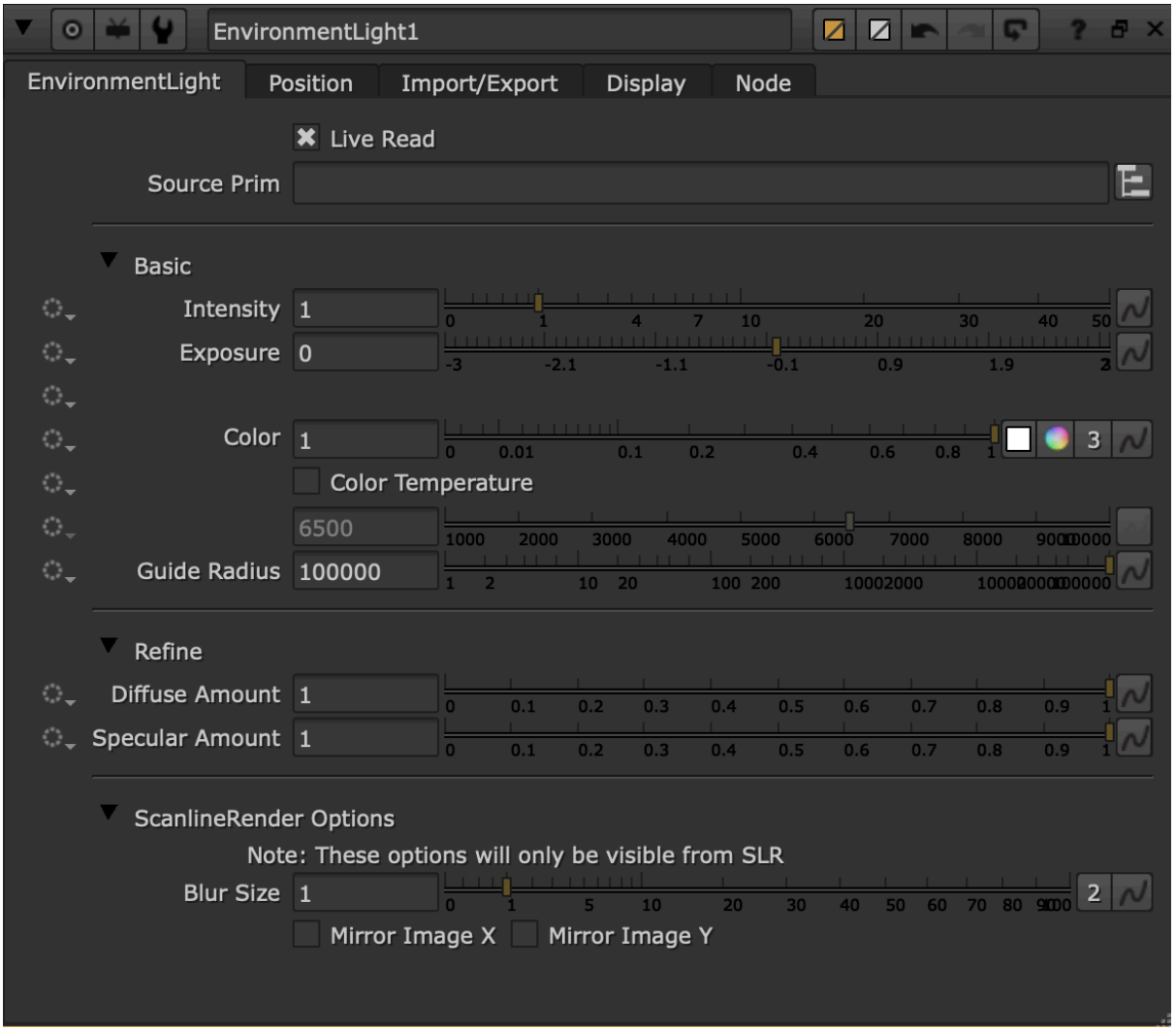
EnvironmentLight

The environment light is there to produce a meshes specular colour, so essentially anything plugged into the environment light will show up in the reflections of a mesh.

The ‘Guide Radius’ scales the locator in the viewer.

We then have a section for ScanlineRender2 specific options. In this section these changes will not be visible in the 3D viewport via HdStorm as HdStorm doesn’t provide controls for these yet, but will be visible in your ScanlineRender2 output. Blur allows you to set how much you want to blur the image you are mapping to the light, and mirror X and Y will let you flip and flop the image being mapped. You can always opt to create a blur or mirror nodes to do this in the node graph also.

The rest of the knobs are the same as what was outlined in DirectLight.

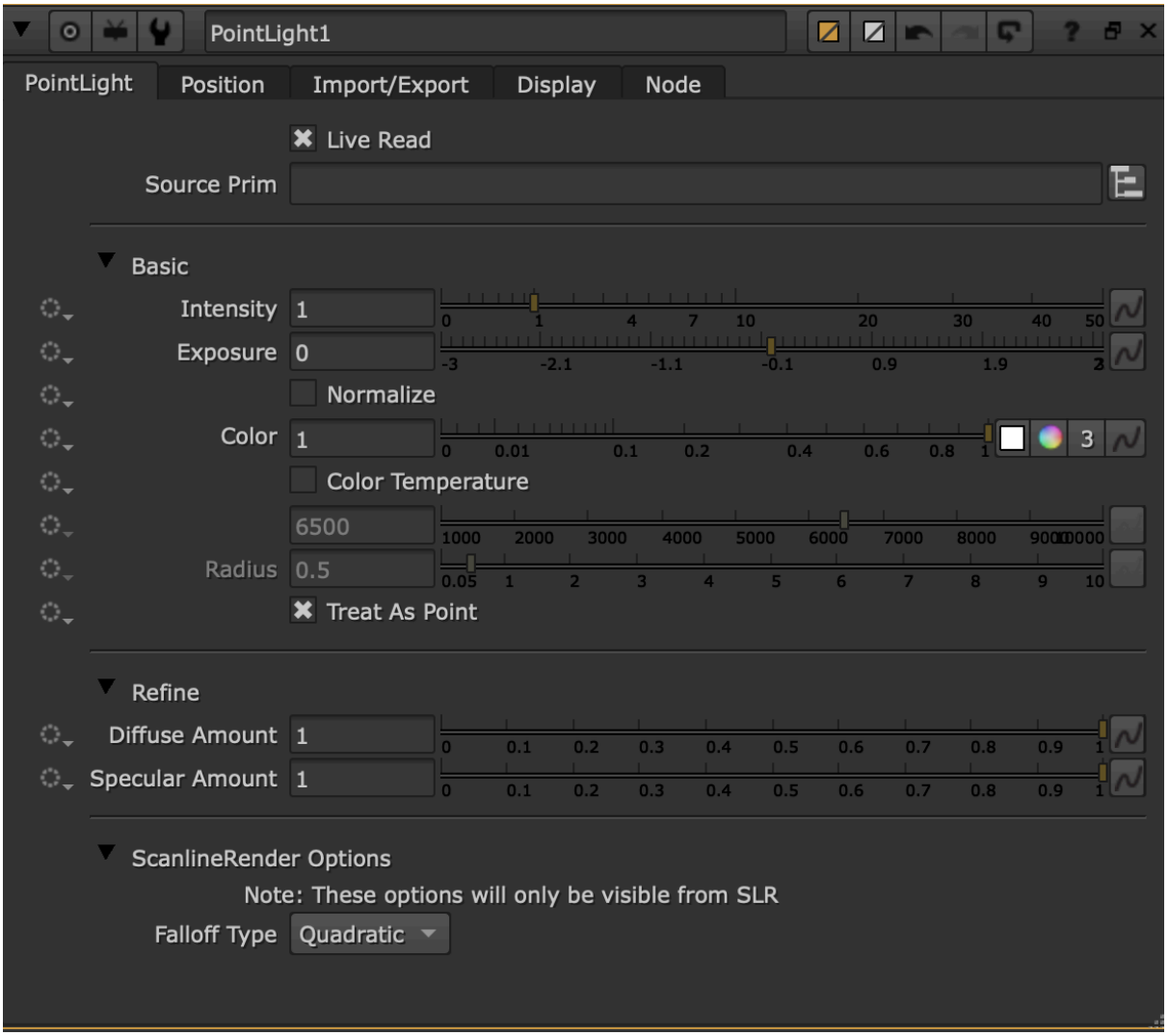


PointLight

The PointLight node lights your scene as you would expect, but some knobs have been updated to better match with USD equivalents for easier compatibility.

For instance the radius knob controls the radius of the light, where a larger radius increases the overall reach of the light. However with a PointLight, there is an additional knob called ‘TreatAs Point’ and this essentially reduces the radius to the equivalent size of a point. Part of the work here is to grey out the Radius knob when this Treat as point is active as you have no need at that point to change the radius.

As for the other knobs, these are the same as the DirectLight.



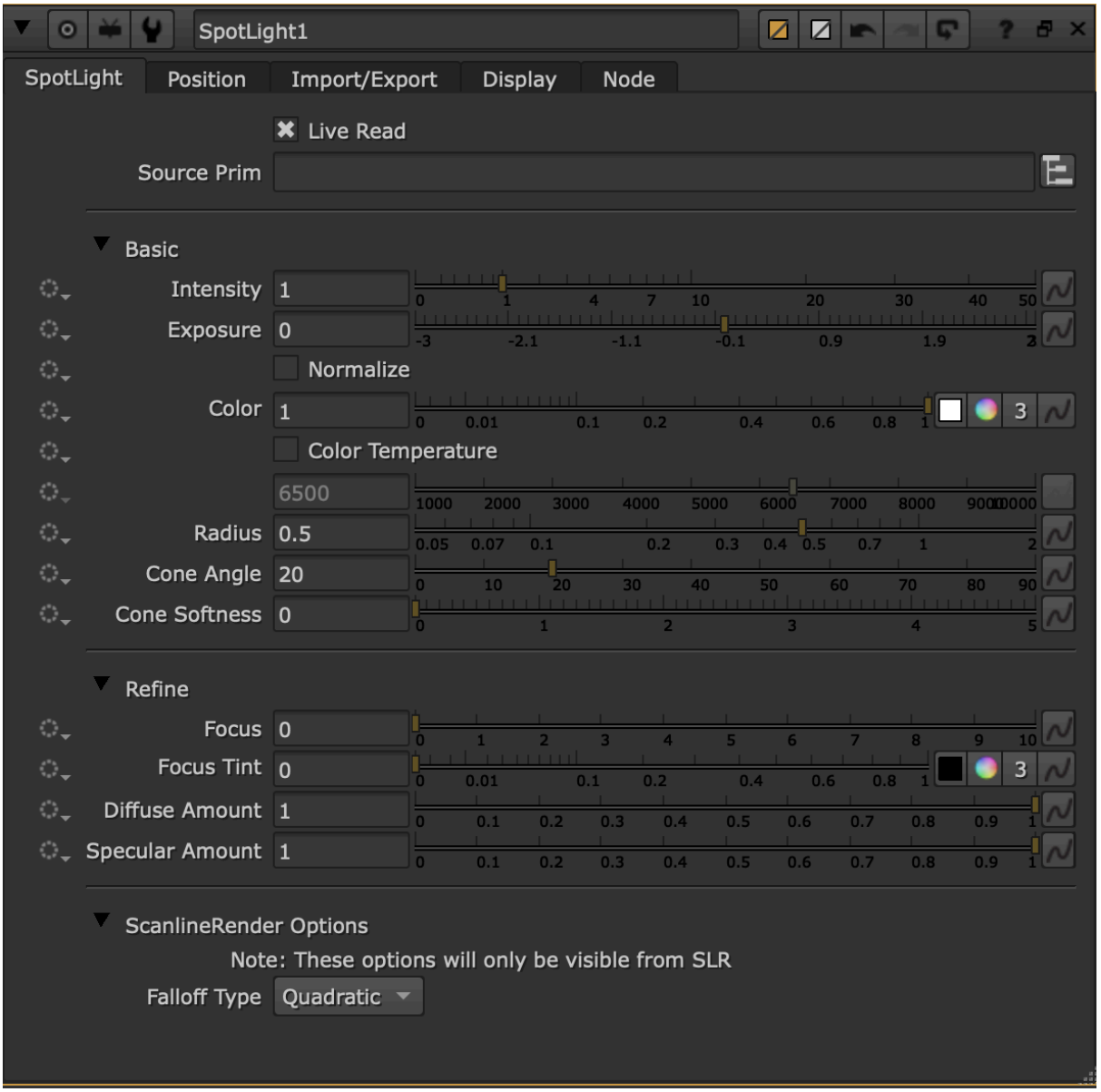
SpotLight

The SpotLight node lights your scene as you would expect, but some knobs have been updated to better match with USD equivalents for easier compatibility.

For instance the radius knob controls the radius of the light. A larger radius increases the overall reach of the light. The Cone Angle controls the spread of the light.

The Cone Softness, Focus and Focus Tint will closely recreate the behaviours of the penumbra and falloff knobs of the classic SpotLight node.

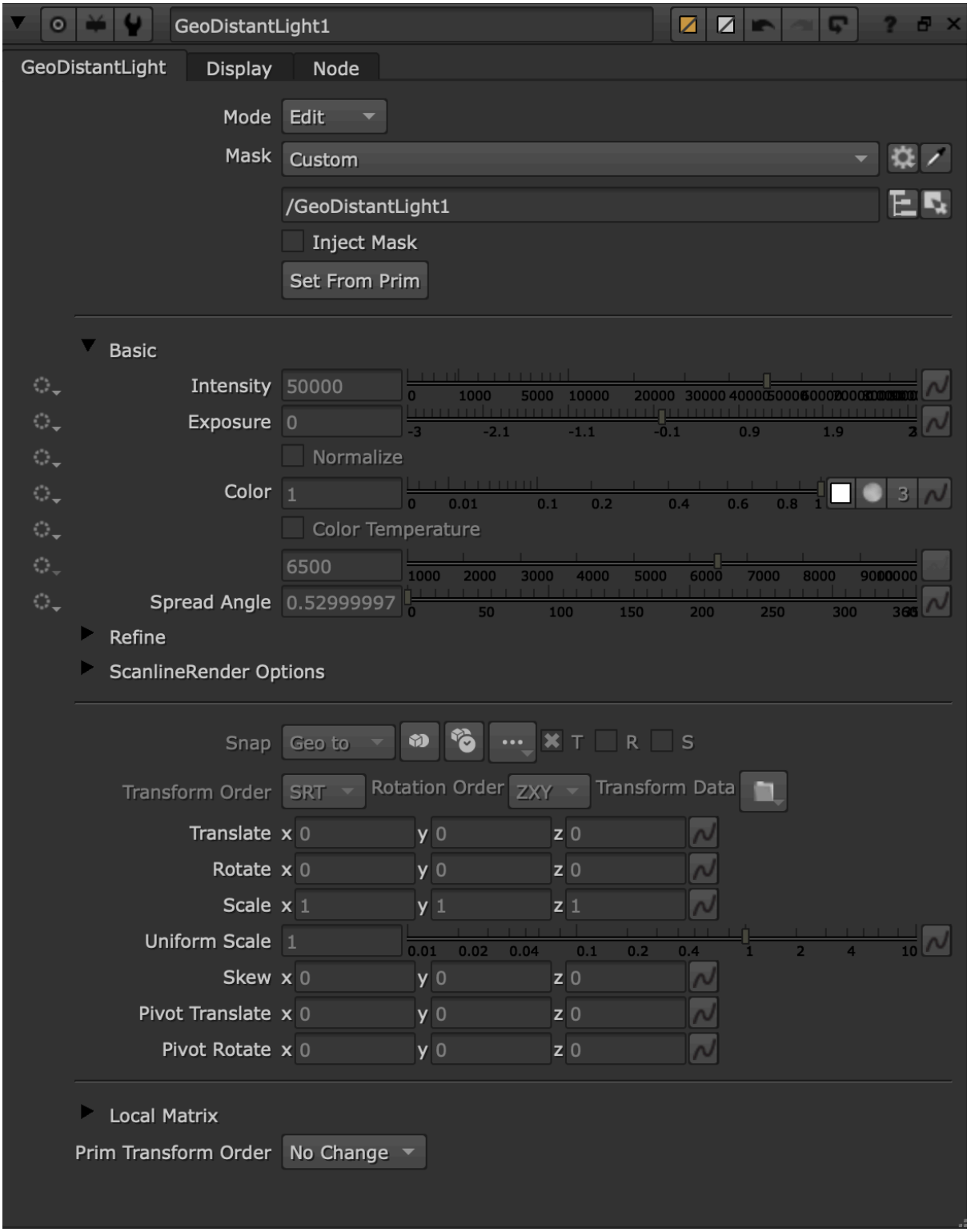
As for the other knobs, these are the same as the DirectLight.



GeoDistantLight

The GeoDistantLight node is the equivalent of the DirectLight in Nuke and is used for typically recreating the sun as a light source.

It defaults to create mode in this build, but its primary purpose is in edit mode where it allows you to directly edit a USD Distant light type in your stage. Once in edit mode you can connect a light in your stage to the Mask knob and then hit ‘set from prim’ to update the knob values with the values in your light node.



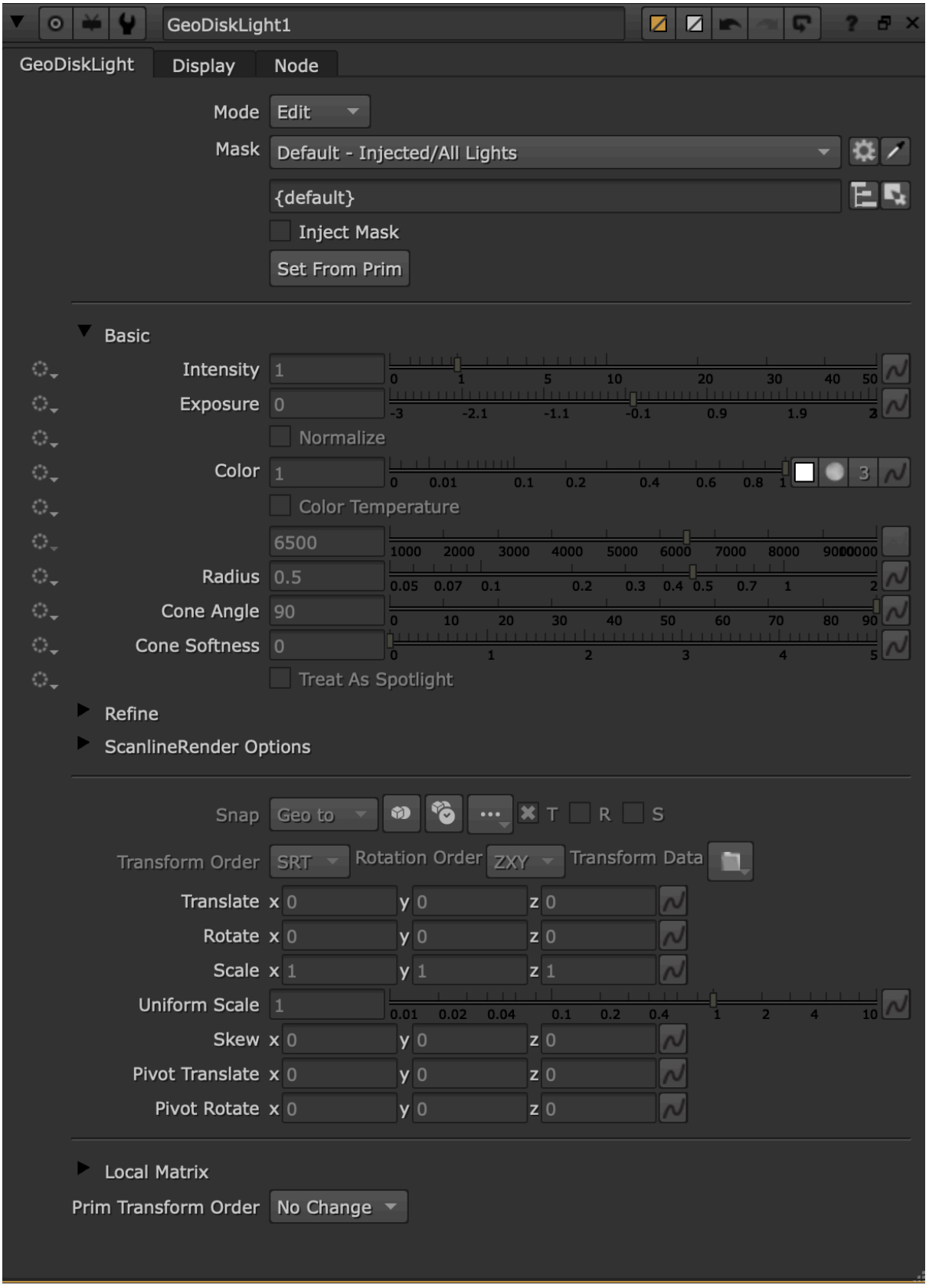
GeoDiskLight

The GeoDiskLight node is the equivalent of the SpotLight in Nuke and is used typically for creating light sources in close proximity to an object and allowing for shadows for softer falloffs.

It defaults to create mode in this build, but its primary purpose is in edit mode where it allows you to directly edit a USD Disk light type in your stage. Once in edit mode you can connect a light in your stage to the Mask knob and then hit ‘set from prim’ to update the knob values with the values in your light node.

The Disk light however is not a 1:1 equivalent of Nuke's traditional SpotLight, so we have also included a ‘Treat as SpotLight’ checkbox that sets the Disk lights emitter source to be a single point,

Note: the results of the Cone Softness attribute are only visualised via the locator as HDStorm doesn't support cone softness (it does support cone angle) because it's expensive to do and Storm is only a preview renderer, but ScanlineRender2 will be able to render the angle and softness absolutely fine.

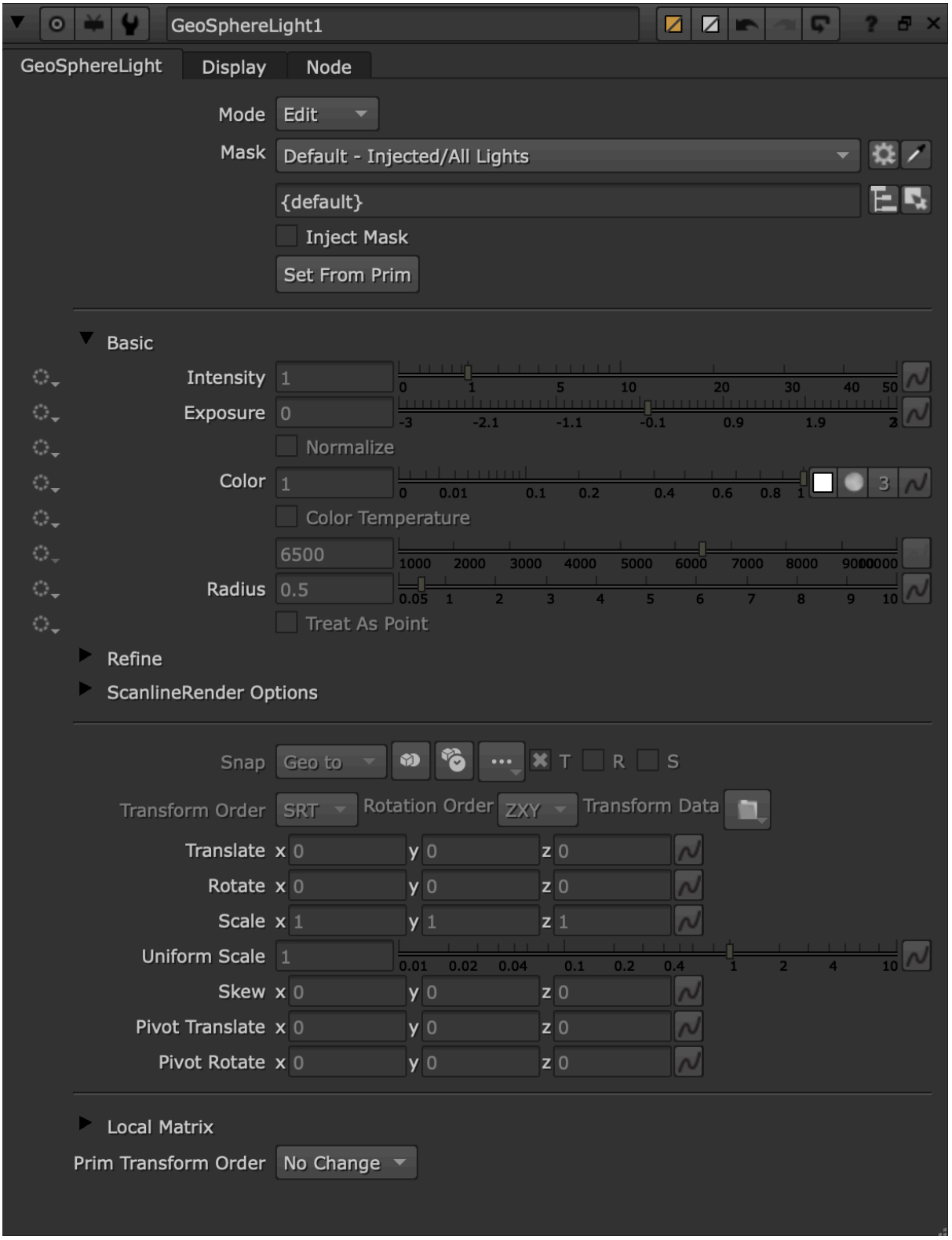


GeoSphereLight

The GeoSphereLight node is the equivalent of the PointLight in Nuke and is used for typically recreating light sources in close proximity to an object and emitting light in all directions.

It defaults to create mode in this build, but its primary purpose is in edit mode where it allows you to directly edit a USD Sphere light type in your stage. Once in edit mode you can connect a light in your stage to the Mask knob and then hit ‘set from prim’ to update the knob values with the values in your light node.

The Sphere light however is not a 1:1 equivalent of Nuke's traditional PointLight, so we have also included a ‘Treat as PointLight’ checkbox that sets the Sphere lights emitter source to be an infinitely small point. USD lights try to light at real world scales, so this functionality does break that, but helps give artists control on how they want to use their lights.

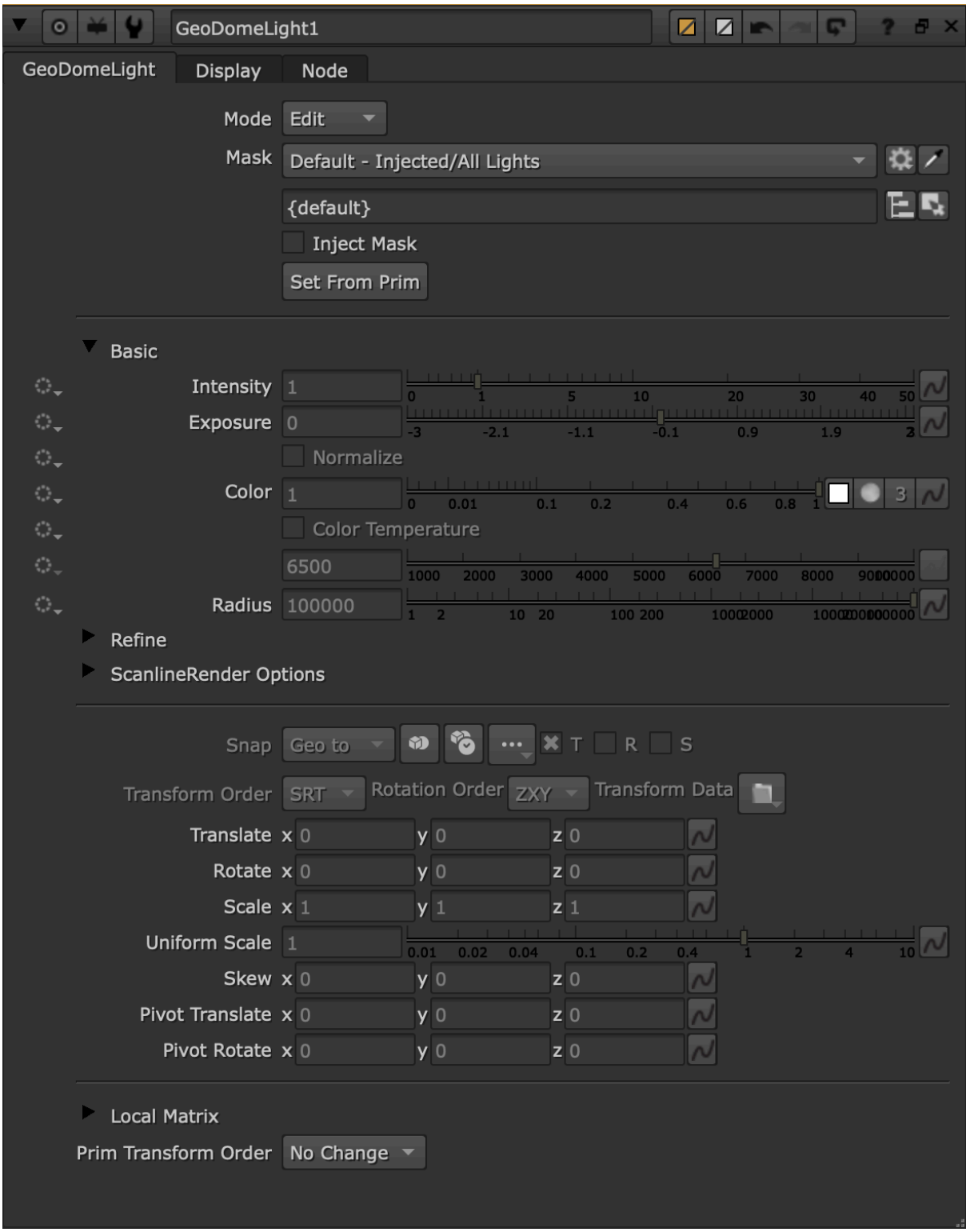


GeoDomeLight

The GeoDomeLight node is the equivalent of the EnvironmentLight in Nuke and is used for lighting an entire scene evenly from all directions and in Nuke for contributing to reflections.

It defaults to create mode in this build, but its primary purpose is in edit mode where it allows you to directly edit a USD Dome light type in your stage. Once in edit mode you can connect a light in your stage to the Mask knob and then hit ‘set from prim’ to update the knob values with the values in your light node.

We don’t currently support global illumination rendering and as such the GeoDomeLight does not provide illumination from the light's texture source.



Nodes for modifying your 3D data

Modify: GeoTransform

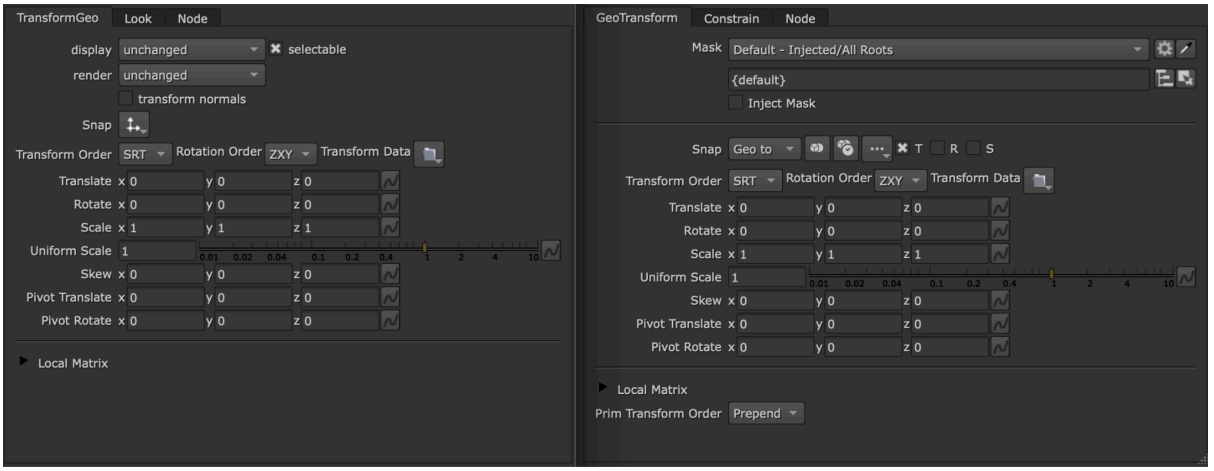
GeoTransform is as you would expect the equivalent of TransformGeo and allows you to specify the prims you want to move in your stage.

By default the mask will default to All Roots, which will move everything in your stage, similar to how the classic TransformGeo would work. The Mask menu has a number of preset options with a design aimed to match that of the 2D merge. Depending on the node use case, the default preset may be different (typically a node will default to All Roots, but some nodes will default to 'All Meshes'). You can add custom masks in this field to target specific prims inside of your stage, so that the effects of that node only target the prim you specify.

The new snap menu that was covered in the GeoCard section is also a new addition, but the main new knob to be mindful of is the 'Prim Transform Order' knob at the bottom. This knob allows you to specify the point at which this transform is applied to the masked prims, whether GeoTransform should be applied before (prepend) or after (append) the upstream transforms. We believe that Prepend matches the most closely to the classic system behaviour, but because of the new scene graph hierarchy, prims transformations could be set up in a way where append might make more sense in some scenarios and prepend in others. As an artist if your transform is not functioning as you were expecting, it is worth switching between these options and exploring how the transforms for the prim you want to move are setup in the scene graph hierarchy.

The other new addition to the GeoTransform node and many others is the evolution of the look input and look tab, to a constrain input and constrain tab. Now you are able to swap between several different constrain types, such as LookAt, Parent, Transformation and you have the ability to maintain an offset from the prim you are constraining to and the ability to set the strength. Please note that the constrain input only works with Axis ops currently and we are exploring allowing you to connect this input pipe to other geo ops, but doing so would likely require an additional mask path and we want to be mindful about how much we are asking users to specify mask paths vs having node readability in the graph.

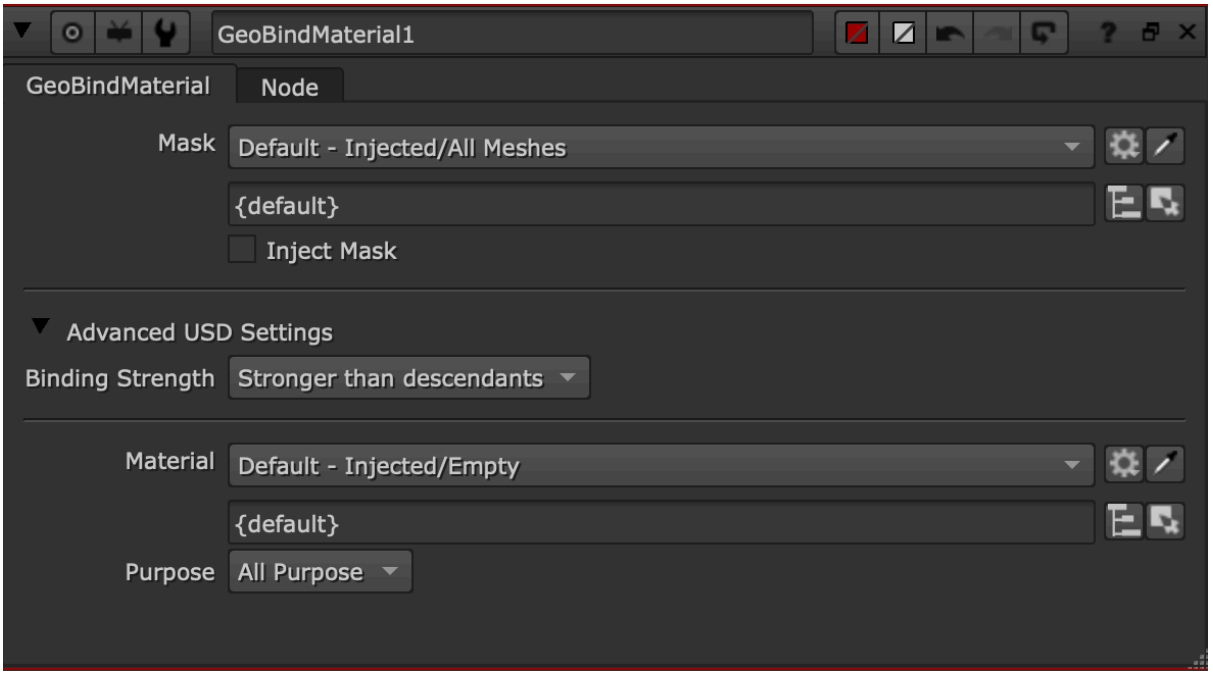
xform_op_order
Specifies the point at which this transform is applied to the prims (selected by the mask). USD prims support multiple transforms. This setting determines whether the GeoTransform should be applied before (Prepend) or after (Append) the upstream transforms.



Modify: GeoBindMaterial

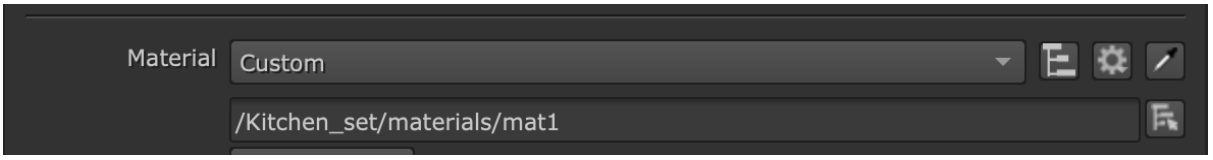
The GeoBindMaterial node is the equivalent of the ApplyMaterial node and allows you to bind a material to geometry in your scene.

The Mask knob defaults to ‘All Meshes’ essentially allowing any mesh that exists in the stage to have your desired material bound to it. The reason we did not opt for ‘All Roots’ for this node is that setting a material binding at the root would mean bindings later on in the node graph would not show as the root level binding would take priority.



Bind a material imported from your scene

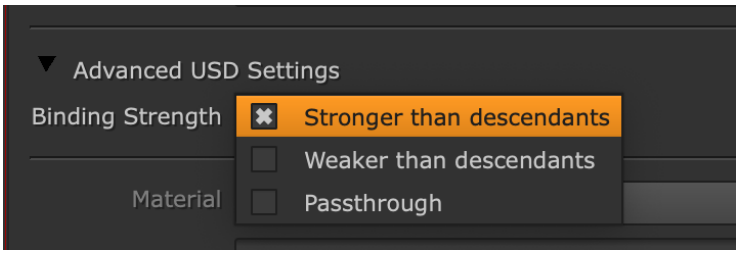
A really nice addition to binding workflows is that you can now assign a preexisting material in your stage using the Material mask knob. This is perfect for when you already have a material setup that you quickly want to reference, or even when you have imported a material in the stage that you want to grab and use somewhere else helping you take advantage of all the materials that exist in a scene.



Set binding strength

One of the changes with working in USD is that material bindings have their own purpose strengths and if you import geometry with existing material bindings, how do you want materials bound?

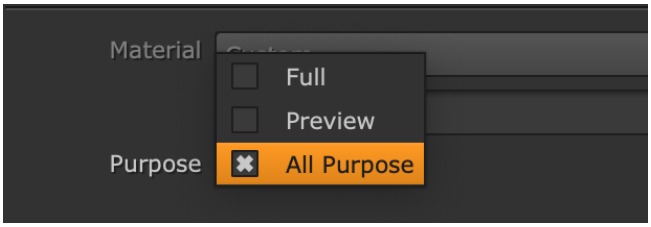
With the new scene graph hierarchy at your disposal you can have geometry that is a child of geometry you might want to bind a new material to. When you do, how do you want the children geometry to be affected? Do you want them all to share the new material binding of the parent geometry you are targeting? Do you want them to keep any original material assignments they had? All of this can be controlled using the new binding strength knob, giving you greater control over how you bind materials to more complex geometry structures.



Set your material purpose

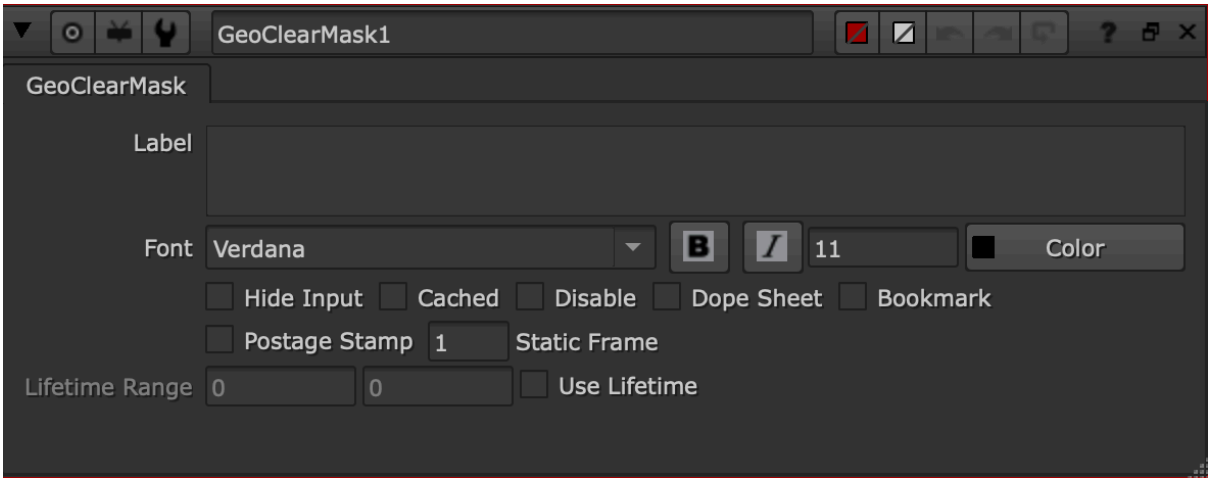
Purposes are a new workflow to help define how a material should apply in different contexts, namely your 3D viewer context where you are previewing the shot and your render context where you are generating final pixels.

With the new purpose control you can set what your binding purpose will be, giving you the flexibility to bind higher fidelity materials and shaders for rendering, while seeing a more efficient lower fidelity version in your 3D viewport. This also allows you to control how your material binding should affect geometry when an existing material binding with its own purposes exists.



Modify: GeoClearMask

The GeoClearMask node is the companion node to GeoMask and allows you to clear any masks that have been set upstream by the GeoMask node.



Modify: GeoCollection

The GeoCollection node allows you to place a group of prims into their own collection, which can be used to then be a mask input for future nodes down the stage.

As it stands the GeoCollection node isn't the most intuitive and takes some setting up in order to use. As such we want to improve this node's usability. I'll explain how it works currently and then explore what we would like to do.

The Mask knob is the same as on other nodes, but what is confusing initially is that you already need to have created a collections Xform or Scope to point to using either a GeoXformPrim, or GeoScopePrim node (technically you can create a collection of any prim type, which will be useful for future workflows like light linking, but for now its best to start with Xforms).

Once you have created a collection you can now set your Mask to this and this is where your GeoCollection will be saved to. This behaviour already feels slightly at odds, because Masking is typically used to limit a modifier nodes behaviour to a set of prims, whereas here it is technically doing that, but it feels more like it is setting where the behaviour should exist.

Once you have specified where you want your collection to live, you then give the collection a name.

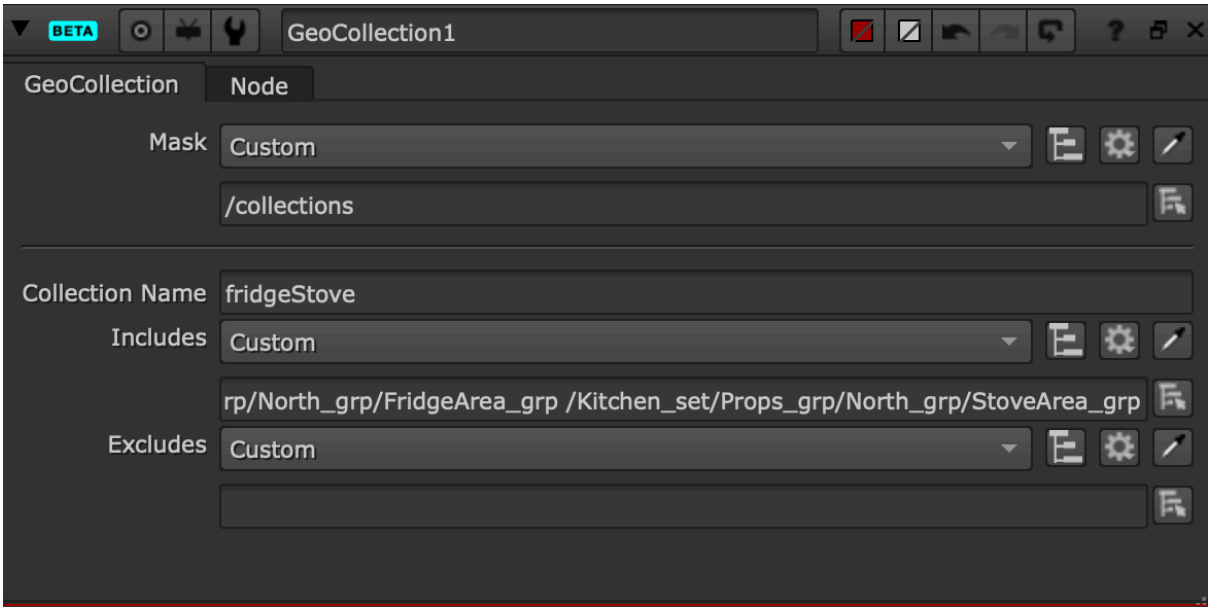
With the collection set up though you can now specify what prims you want in the collection. You can use the include knob to set what prims you want in the collection and the exclude to remove prims from the collection.

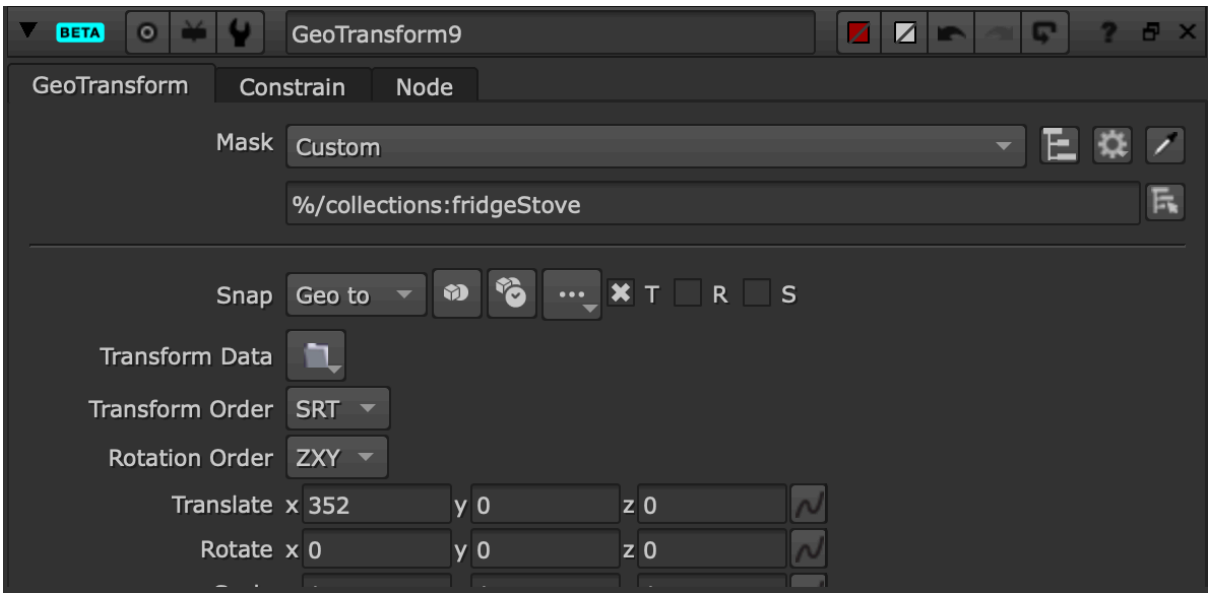
Typically you'll just leave exclude empty, but if you are using the expression language to make complex selections across a large stage, then having the exclude can be really useful to say things like include all the lights in my scene, except this particular one.

Now you have set up your collection, to use it you need to specify the Collection Name from the collection prim (see images below). This is not currently intuitive for users, and requires an updated UI for collections to be more easily seen, potentially as children in the scene graph and therefore pickable via the path mask knob.

This workflow isn't intuitive and is from an early exploration of playing with what collections could be. To improve this node ideally you could create collections directly from this node without having to set up a previous Xform, then have the collection name actually be a representation of the collection that you could see in the scene graph. Having this in the scene graph would then also mean you could just specify the collection name in order to use it in other modifier nodes.

This node is available in the closed beta, but this node and similar ones which still need further development will either remain with the beta tag for Nuke 17.0, or be hidden from the menu and only accessible via the TCL command window.





Modify: GeoConstrain

The GeoConstrain node is really cool and a nice jump forward in workflows. This node was initially an exploration ground for how we could reintroduce the lookAt pipe to the new system with greater functionality and just kept extending to have even more use cases.

To use GeoConstrain you will need the prims you want to constrain to and from already in your stage. This is something we'd like to improve where you could plug in a prim from a different stage input so that you don't have to set masks and paths within the node itself.

To set a constraint up in the current workflow you first set what you want to constrain via the Mask knob i.e. I want to make this cube look at this moving axis. Then you set the Target Path to the prim you want to constrain to i.e. the moving prim you want your original prim to look at or move with.

The Point knob lets you set where on your target prim you want to constrain to, by default this is set to the origin, but you can change this to be a specific vertex or face on the geometry and then use the selection system in vertex or face mode and the new picker knob to pick a relevant point. Note that if you are constraining to a prim selection that is not a mesh then you may only be able to constrain via origin as the target object needs to be a point-based prim.

The Time knob also lets you set a time offset for your constraint by a relative amount in the past of future, or an absolute value which basically acts like a FrameHold which you can key as needed.

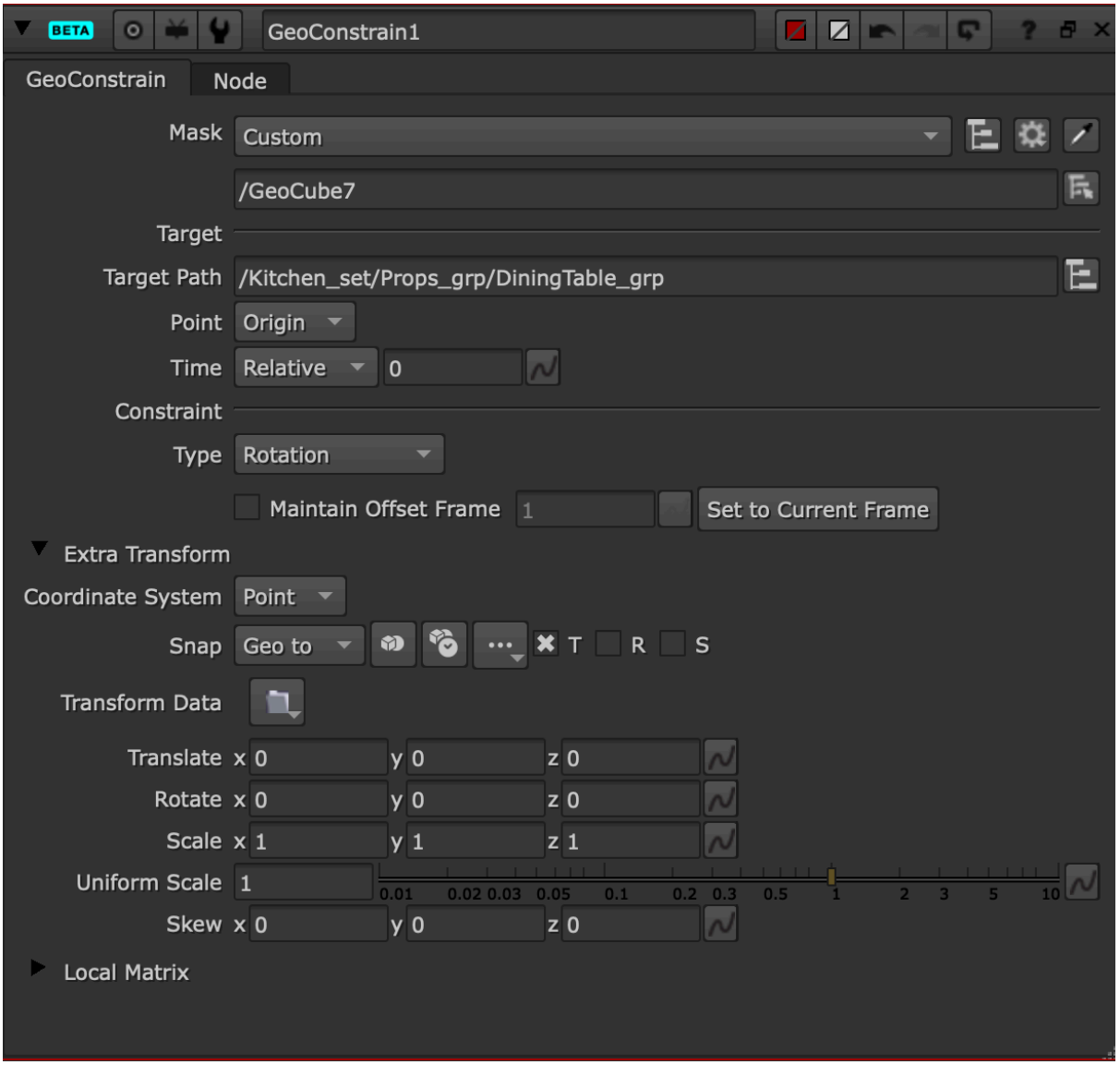
The constraint type can be changed between a few different options and defaults to the LookAt constraint, which has been a staple for Nuke artists for years, allowing your masked prim to look at the target prim as it moves. You can set which axis you want to be able to follow on and the strength, which is useful when you want to blend camera movements together.

The transformation constraint matches the location, rotation and scale of the prim to those of the target, so whatever the target prim does, the masked prim will also do. You can toggle the transformation, rotation and scale attributes on and off independently and by default we have scale turned off because most often we found artists are just trying to match the translation and rotation of a prim. If you ever just want to do one of those constraints individually then you can either deselect these knobs, or just select the transformation, rotation, or scale constraint options from the dropdown. The maintain offset knob allows you to keep a relative offset between itself and the target prim. This basically means the prim either constrains from its starting position, or goes exactly to the position of the target prim.

You can also constrain using the parent constraint which modified the prims transformation as if the target was the parent hierarchically. Most of the time this will work the same as the transformation constraint, but it is easiest to tell the difference when you start rotating your target prim, as the hierarchical nature will mean the masked prim will rotate around the target prim like planets in a solar system, rather than matching the targets prim like a Transformation constraint would do.

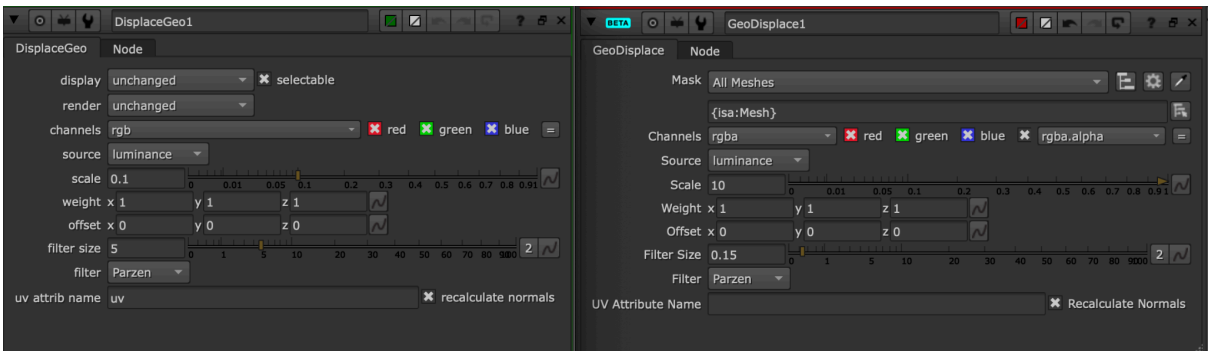
The extra transformation section does as the name implies, allows you to add any additional transformation offsets that you would like, but with the addition of a coordinate system knob, which lets you specify whether you would like the extra transformers to exist in a point, prim, or world coordinate system.

A lot of the work put into the GeoConstrain knob was also key for reintroducing the 'LookAt' input pipe but updated to be a 'constraint' input pipe which could use the same constraints found within this knob but with a simplified feature set intended for faster use cases, with GeoConstrain here to support more advanced constraint workflows.



Modify: GeoDisplace

The GeoDisplace node is pretty much a 1:1 recreation of the DisplaceGeo node from the classic system, but with the addition of the mask knob to allow you to specify the geometry you want to displace.



Modify: GeoDrawMode

GeoDrawMode is a new node that allows you to draw your geometry as bounding boxes, as cards, as origin points. Note though that GeoDrawMode only works on prims that have their Kind attribute set. This is something

we’re exploring to see if we can set on attributes without a kind set, or if we need to give artists the ability to set a Kind attribute for draw type operations on this node.

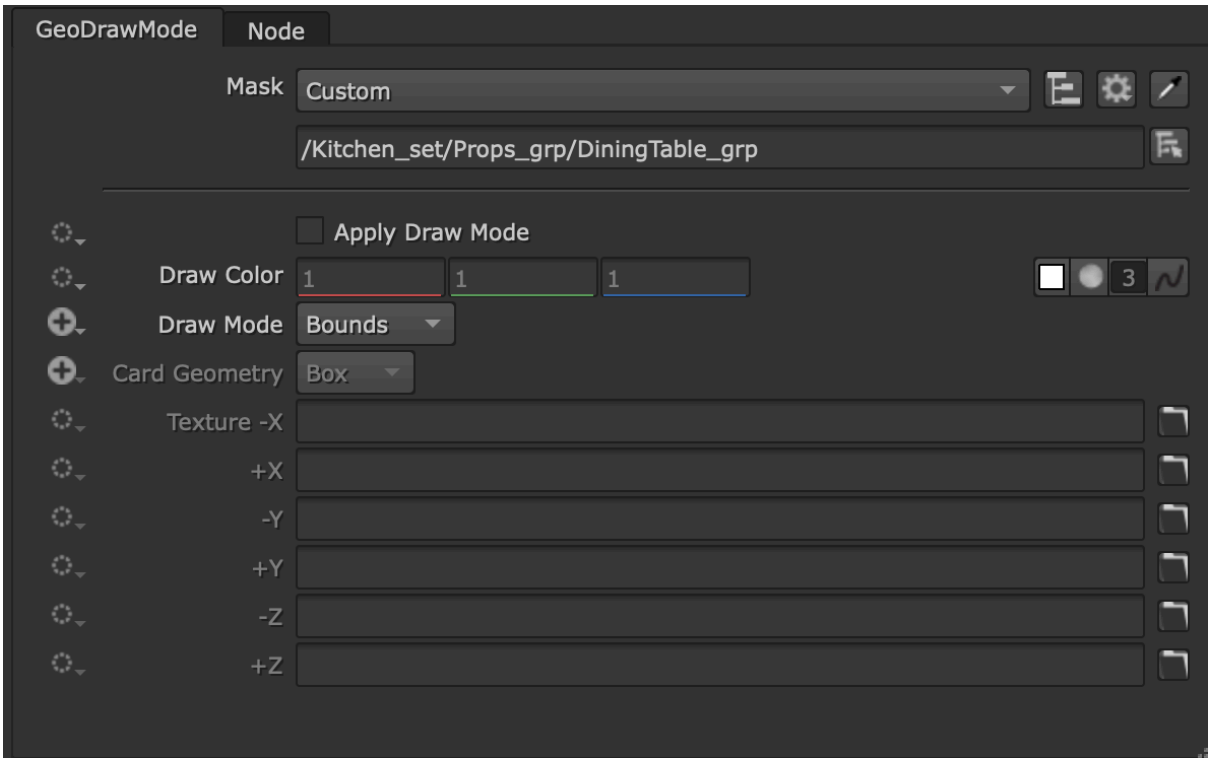
The apply draw mode knob allows you to essentially set the draw mode at the level your mask specifies, so if I mask the kitchen table group and want to show the overall bounding box I can check on apply draw mode. However if I want to show the bounding boxes of all the prims that are children of the kitchen table group I would leave apply draw mode off.

The draw colour knob allows you to set the colour of the bounding box or cards you are creating, but in order to visualise this you have to have ‘apply draw mode’ checked on, and use the authoring knob to the left of the knob name and change it to ‘set’. This would allow you to change and visualise this knob.

The draw mode is what allows you to change between the different draw options and when you set it to the Cards mode, you can visualise the cards as either a box or a cross and are able to use the texture inputs to add custom textures to these cards.

Overall this node was an early exploration of what authoring knobs could be in the new 3D system and the workflows will need to be refined to make them more intuitive.

apply_draw_mode
Sets the model:applyDrawMode attribute on the prims.
If it is not set (neither from the incoming stage nor through the knob), the underlying USD subtrees will be further traversed.
Prims in the subtrees are drawn with proxy geometry when the prim has kind component, and/or applyDrawMode is set, and the prim's draw mode is not default.



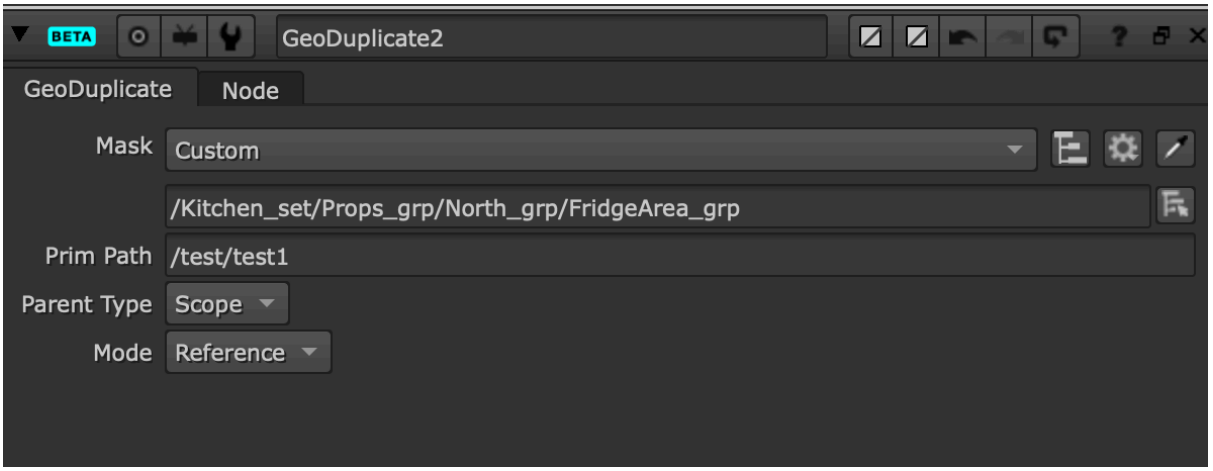
Modify: GeoDuplicate
GeoDuplicate allows you to duplicate a prim or a number of prims. By default it will name the duplicated prim by the name of the prim you are masking + the name of the GeoDuplicate node. You can update the duplicate prims name however you like though, by modifying what is in the Prim Path knob and any parent groups you can set to be either Scopes or Xforms.

The Reference mode also allows you to decide if you would like to duplicate the prims by one of two modes. You can either use Duplicate mode which makes a flattened copy of the prims. This is the default behaviour as it is the most similar to classic Nuke. Or you can duplicate the masked prims as a Reference. This essentially makes an instance of the original prim data, so you are using the source prim data as a live input and any changes you make to that data will also apply to the instance.

The best way to think of it is if a want a wholly new duplicate of the geo that only changes when I make specific changes to it, use Duplicate mode, if however I want to make changes to the original geometry and have those changes also show up in the duplicate, then use Reference mode.

Currently there are a few bugs with this node with most notably reference mode not functioning as intended and the node itself colour not updating.

mode
The duplication method:
Duplicate: Make a full flattened copy of the prims. This can be expensive.
Reference: Make instances using references. This is cheaper, but has different behaviour as the source prim is still 'live' and downstream changes to it will affect the instances.



Modify: GeoInstance
GeoInstance like GeoDuplicate allows you to create multiple copies of a prim, however all the copies from this node will be a form of instance, so any changes to the original prim will affect the instanced prims.

First you set what you want to instance and then determine the instance mode. The default mode is reference, which basically references the source prims for all the new instance copies being made. The second mode instanceable reference is a cheaper method of referencing the source prims but does not allow the results to be edited. Think of reference as a read and write mode and instanceable reference as a read only mode. The point instancer mode ..

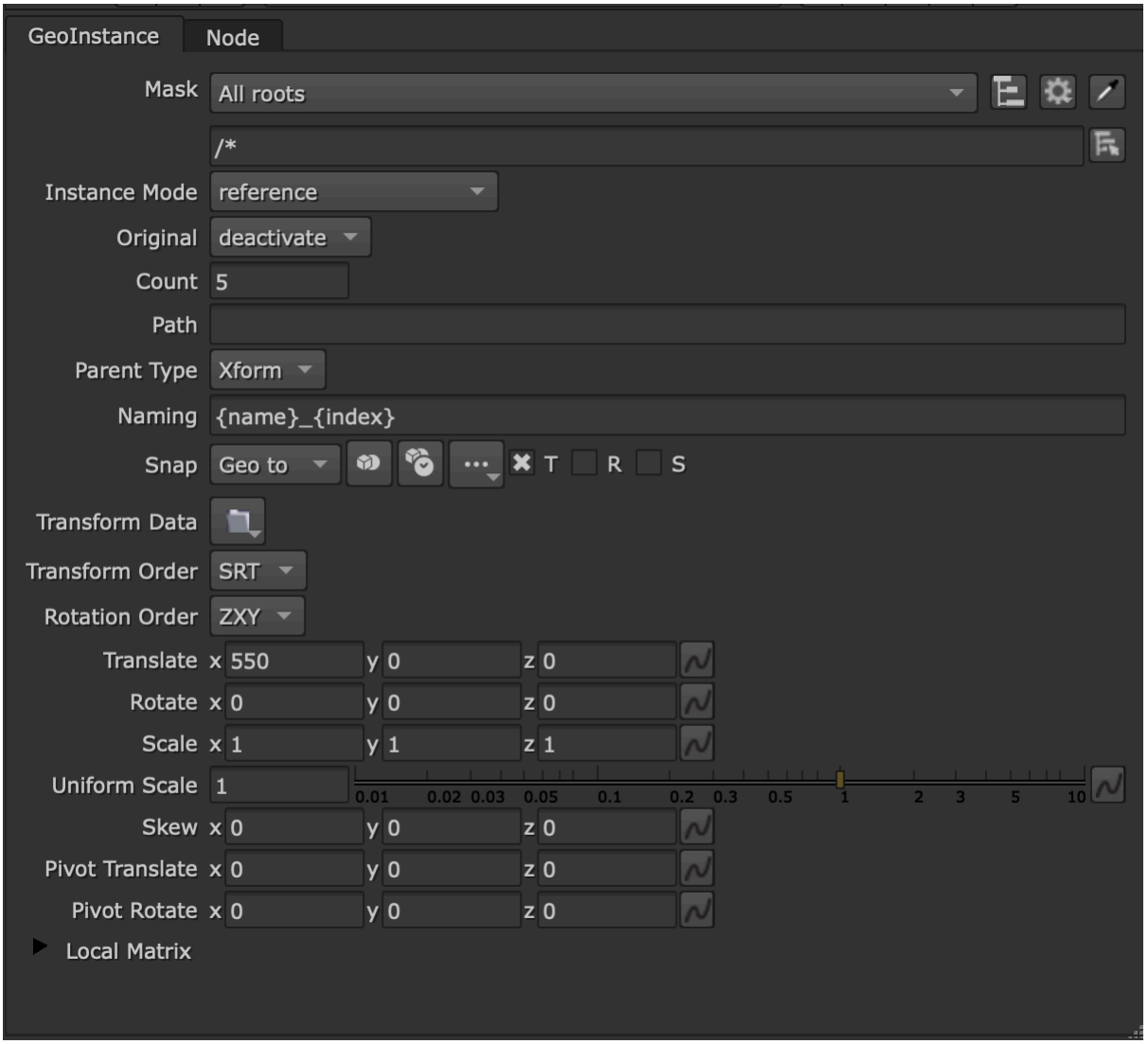
You can then set if you want the original prim you are instancing to be active or not in your scene. By default this will be set to deactivated, as the first instance will usually sit on top of the original prim data which is not always ideal. However it would be good to get feedback on how you would like this to work.

The count lets you set the number of instances you want to generate and the path knob lets you set where you want your instances to live. By default it is left blank and the instances will live in the same location as the originals. The Naming knob also lets you set how you want your instances to be named, with the index token adding the instance number to make it an intuitive default name.

GeoInstance has some more work we want to do in order to better connect it to future instancing workflows once instancing is fully supported in the new 3D system and to allow for some really cool setups with an updated particle system.

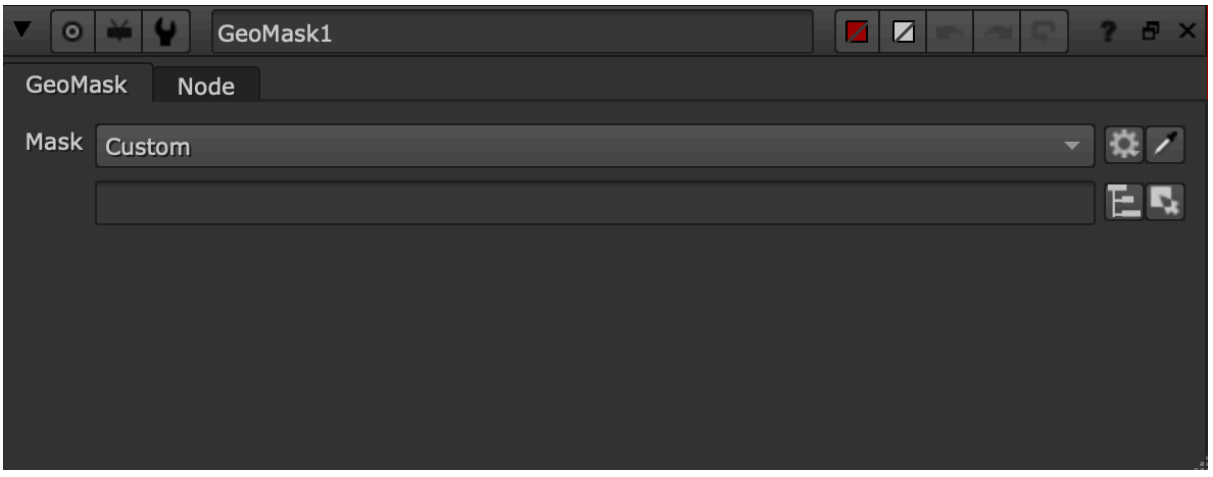
instance_mode
Specifies how to instance prims:

reference: use a reference to the source prims
instanceable reference: use a instanceable reference to the source prims. This is cheaper but the results cannot be edited.



Modify: GeoMask

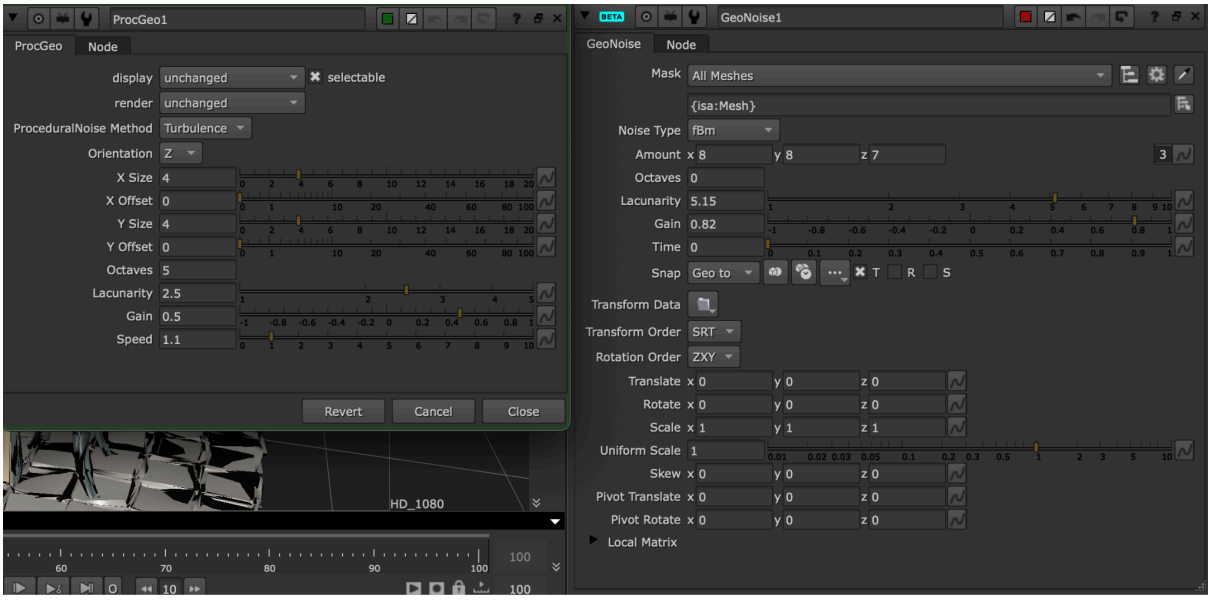
The GeoMask node allows you to set a custom mask that will be inherited by nodes downstream of it so that you don't have to continually copy and paste mask paths, or enable the inject checkbox.



In order to stop the GeoMask from affecting further nodes, simply place a GeoClearMask node at the desired end point and the mask within GeoMask will no longer be passed to nodes below the GeoClearMask node.

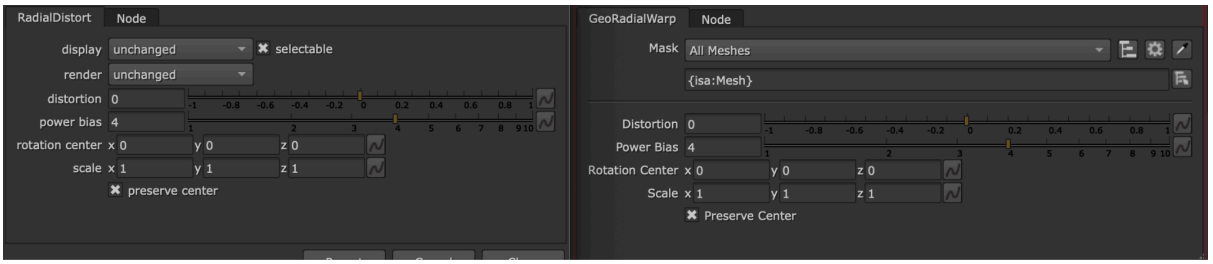
Modify: GeoNoise

GeoNoise is the equivalent to ProcGeo and while knob names might be slightly different they all have corresponding counterparts. Simply put this node allows you to deform meshes based on a noise pattern.



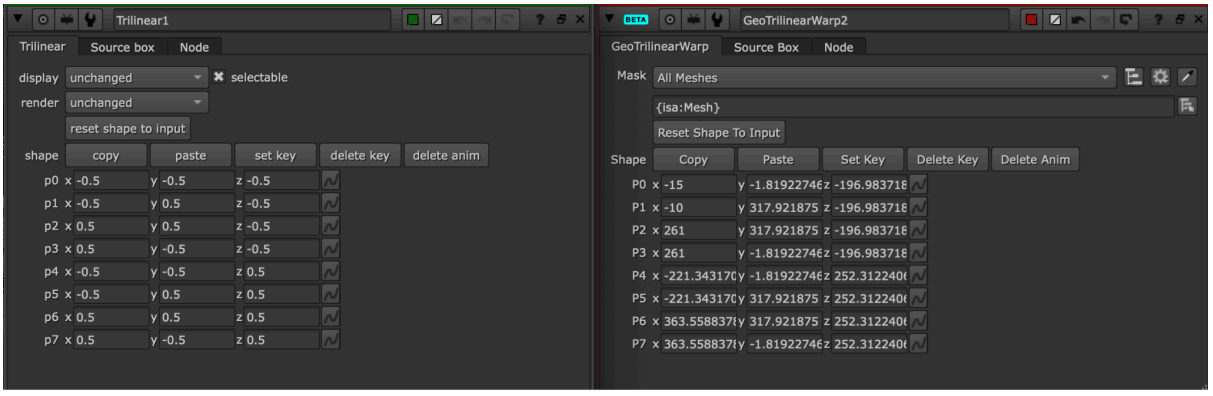
Modify: GeoRadialWarp

GeoRadialWarp is the equivalent to the RadialDistort node and functions the same as before. The main thing to be mindful of though is the different scales of geometry that can be brought in and how much the values will change and impact your geometry.



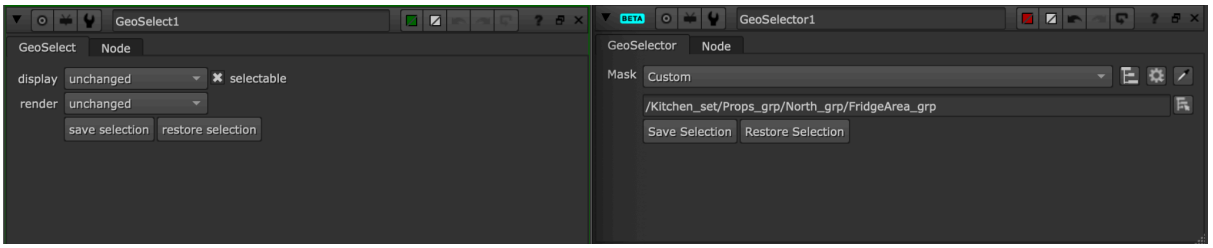
Modify: GeoTrilinearWarp

GeoTrilinearWarp is the equivalent to the Trilinear node and likewise functions as you are used to.



Modify: GeoSelection

GeoSelection is the equivalent to the GeoSelect node and allows you to store a face or vertex selection in the Viewer and then reapply the selection. To use the node you simply create a face or vertex selection, hit save selection, then if you make a different face or vertex selection and want to get back to your saved selection, you simply hit restore selection.



Modify: GeoProjectUV

The GeoProjectUV node is the equivalent to UVProject, but has had some nice feature additions to its toolkit. Workflow wise you still bind a material to your geometry and then create a GeoProjectUV node afterwards connected to a camera, which will result in the material being applied based on this projection.

The majority of knobs are the same just with some name updates for better USD compatibility and updated properties ordering. There is a new knob called ‘Cull Behind Projection Source’ which allows you to stop any projections occurring behind the projection plane.

The attrib name knob from the classic node, has been updated to include a generate perspective knob which generates an ‘stw’ attribute containing the perspective (w) component of the uv and the ‘W Attribute Name’ allows you to specify the name for this.

A really nice addition to this node is the Reference Frame knob, which essentially allows artists to perform sticky projections. By enabling this knob the projection will only be applied at the frame being referenced, making it easy to set up projections that stick to moving prims.

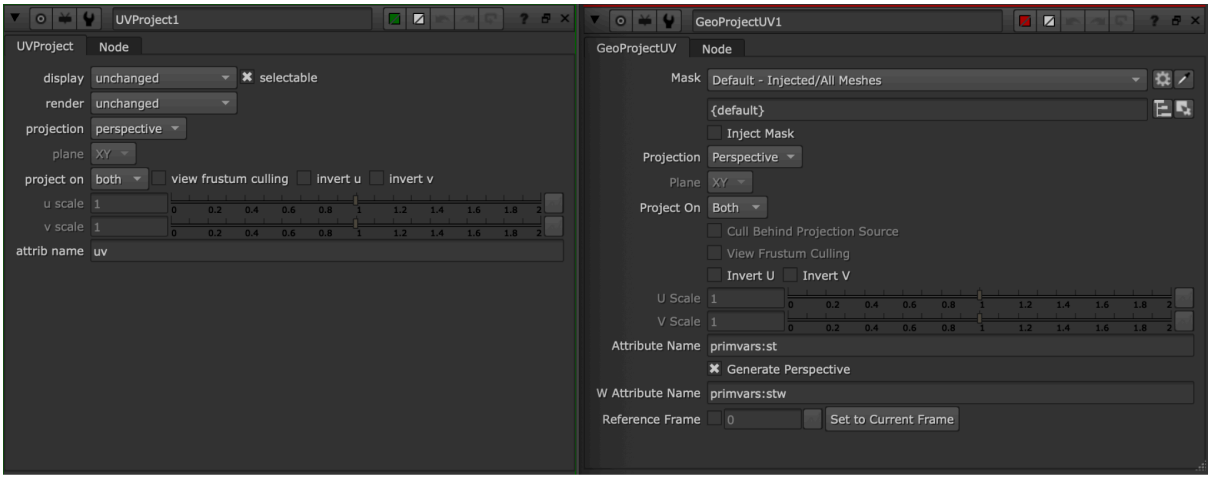
Currently there are some bugs when exploring the different ‘Project On’ modes and similarly so with the culling knobs which we are in the process of fixing.

Please note that GeoProjectUV projects onto vertices so the tessellation of the mesh is important especially when viewing in 3D Viewer as the standard texture mapping in Storm, especially with UsdPreviewShader, does not handle perspective projections and so you see distortions across a surface with large triangles/quads. The smaller the triangle/quad, the more vertices are projected on and the lower the visible distortion.

This is the same behaviour of the classic UVProject node, but the classic Shaders had built in support for perspective-projected texture coordinates, but for modern shaders this needs to be explicitly programmed into, which USD’s stock shaders have not added support for.

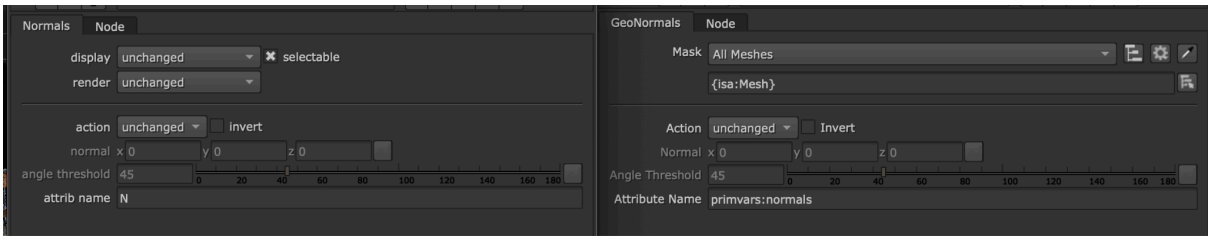
proj_plane_culling
Do not project uvs onto points and vertices behind projection plane.
Turn this on if the renderer being used does not support the perspective (w) coordinate for uvs.
By default this is off because removing uv(w) coordinates from vertices behind the projection source can break the uv interpolation across the polygon's surface when some vertices end up behind the plane while the rest are in front.

generate_w
If on, generates an 'stw' attribute containing the perspective (w) component of the uv.



Modify: GeoNormals

GeoNormals is the equivalent to the Normals node and fundamentally offers the same user experience.



Nodes for merging 3D data

Modify: GeoMerge

The GeoMerge node functions like its counterpart in the classic system, allowing you to merge geometry etc together, but this node also highlights some differences when working in a USD scene graph based system compared to the classic system which require some changes in workflow habits.

If we take two prims and combine them using a GeoMerge, you can see this works as it always has and allows you to modify both prims at once. You don't have to change anything on the node itself as the default options should reflect the classic behaviour.

However the difference in behaviour occurs when working within the same stage. For example, if I separate out a prim from the stage using GeoActivation and masking just one item, transform that separated prim and then try to merge it back. I just get the separated prim and not the rest of the stage. This is because USD is merging the activation and visibility states of the stage based on opinion strength (the A pipes opinion being strongest). So down the A pipe with the entire stage active and visible, but down the B pipe we only have the fridge active and visible. When we do the merge, the A and B pipes compare to see who has the strongest opinion and show that pipes activation and visibility settings (in this case pipe A's).

If I want to work with a prim in a stage but not have it affect the rest of the scene, then I either need to use the masks to only specify that prim in the modification I want to do, or I need to separate that prim out from the stage and bring it in as a new prim via something like GeoReference. A third workflow is to simply use the GeoMask and GeoClearMask nodes to operate only on the masked prim you want to in a specific node graph pipe.

Modify: GeoScene

A change that has been made for the new 3D system is that Camera, Light and Axis nodes only connect to the GeoScene node and not the GeoMerge node. Originally the Scene node in the classic system was intended to represent that you are bringing a scene together based on many different data types.

Given that the Camera, Axis and Light nodes are not nodes that directly affect the USD stage, we felt it was appropriate to highlight that these nodes are slightly different from the other 3D nodes and that they are best connected via their own node.

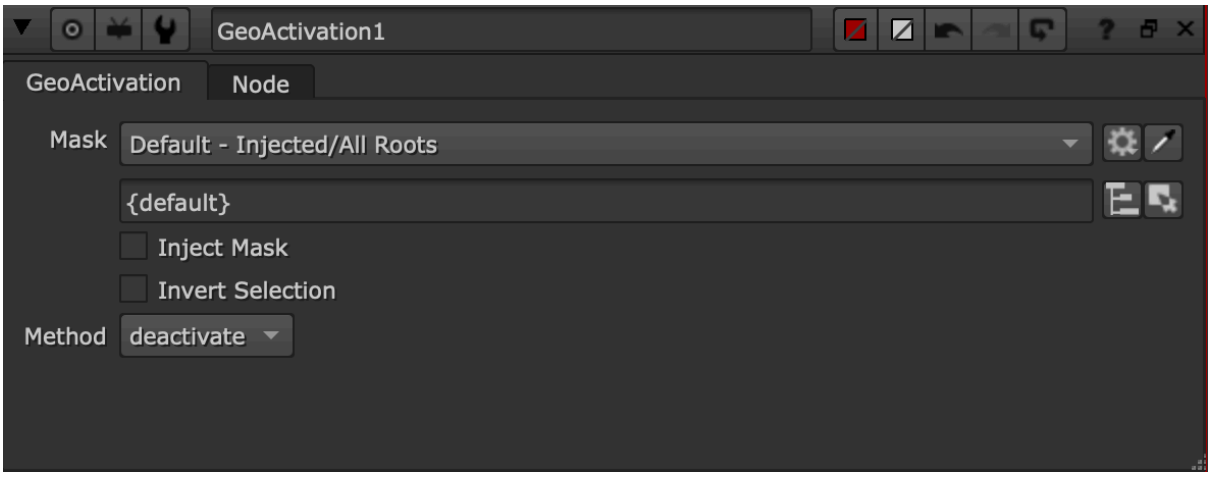
Modify: GeoActivation

The GeoActivation node is a new node that allows you to set if a prim or selection of prims in your stage is in an active or deactivated state.

By default the node will deactivate everything in the stage, so it is often good to disable the node and specify a prim using the mask picker and upon reenabling, only that prim will now be deactivated.

The invert selection checkbox is also helpful when you want to quickly say deactivate everything but this one prim.

As mentioned with GeoMerge, the act of deactivating prims is useful but will have a different behaviour than users might expect when merging a deactivated prim from a stage, back to the same stage. This is because you are not just merging back the item you have deactivated, you are merging back the entire activation attributes of the stage. So if you merge this branch back to the same stage you branched from, depending on if it is the A or B input will determine which pipes activation attributes will have the strongest opinion and will then show that branch's activation. It does not combine the activations of one pipe with the activations of another.

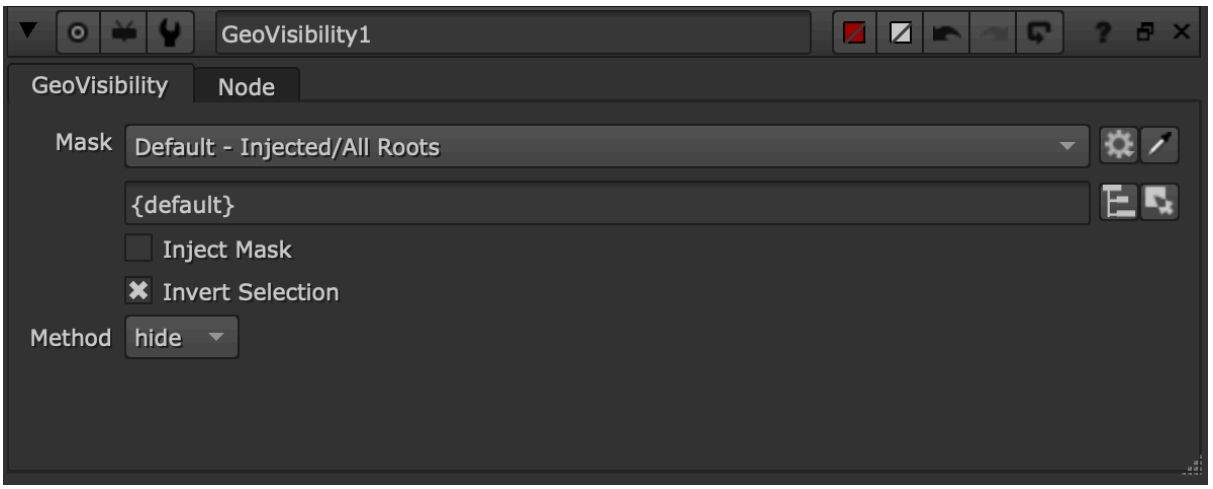


Modify: GeoVisibility

The GeoVisibility node is the counterpart to GeoActivation and lets you hide/show a prim or selection of prims in your stage, so you can work on what's left.

By default when I create the GeoVisibility node it will hide everything in the stage, which means you then need to disable the node in order to pick intuitively.

As with GeoActivation this node defaults to hiding the prim having its visibility attribute being changed, but you can swap this to show instead and invert selection if so desired.



Nodes for shading geometry

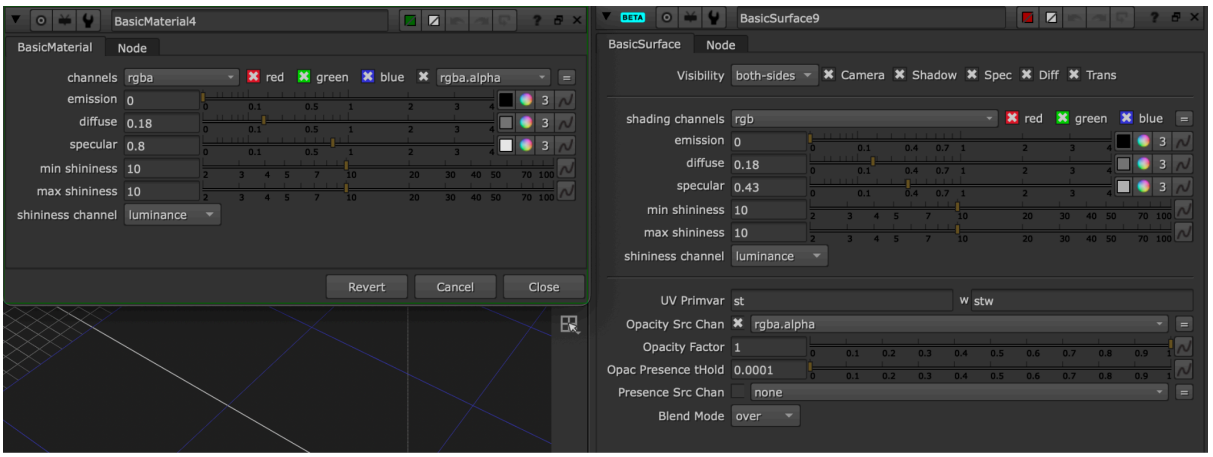
Shader: BasicSurface

The BasicSurface node is the equivalent to the BasicMaterial node, but with some additional knobs exploring new controls for raytraced workflows and new shader controls.

This node is an exploration of concepts and as such is still actively changing through development and may be replaced by an uber shader which would cover the workflow needs that this node originally fulfilled. As it stands the knobs should all be functional but subject to change (not including Diff and Trans checkboxes).

The visibility knobs allow you to set if the material is visible to each of the respective knobs, so enabling camera means the material will be visible to the render camera and disabling it would make the material invisible to the render camera which could be useful if setting up a HDR dome that you want in reflections, but not in the background.

The new knobs allow you to control the opacity of the material.

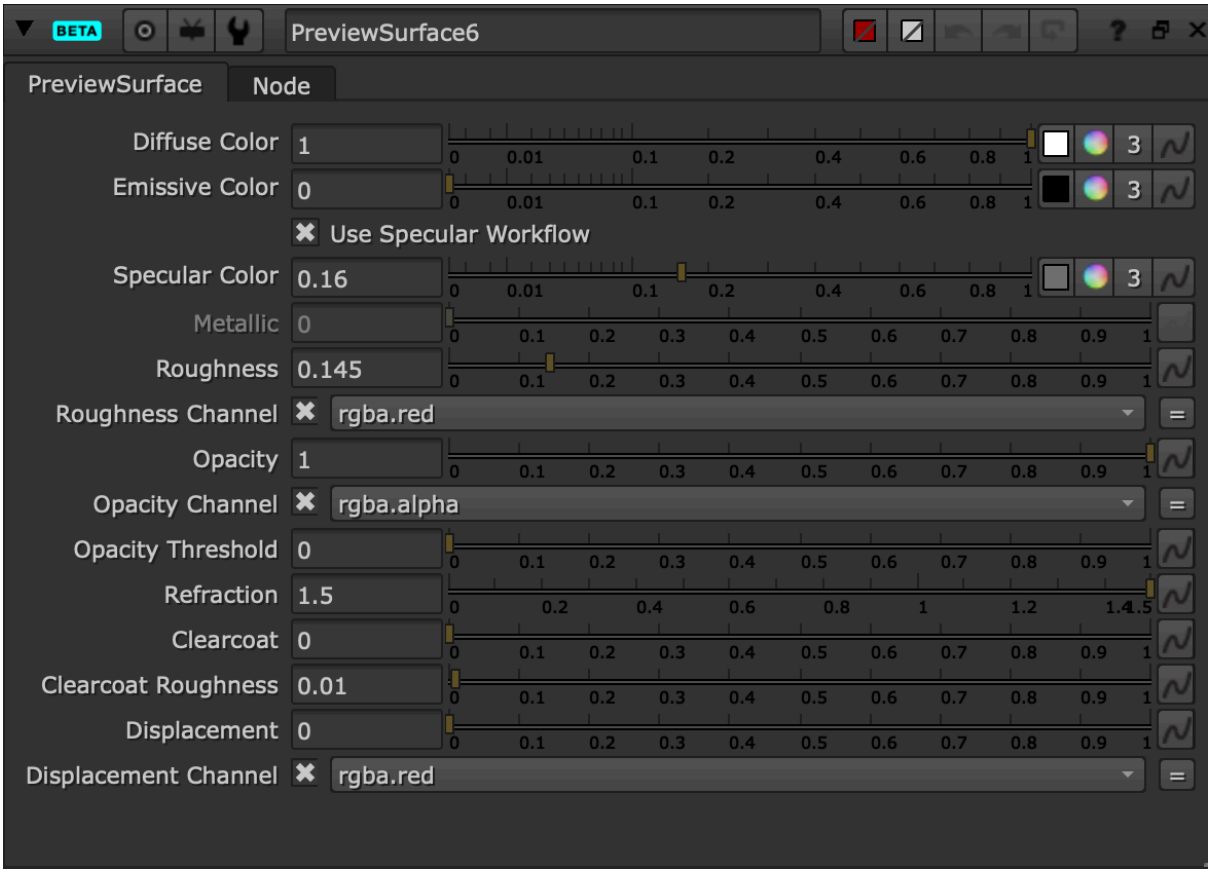


Shader: PreviewSurface

The PreviewSurface node is a representation of the USDPreviewSurface shader which is meant to model a “modern” physically based surface with both specular and metalness workflows.

The knobs while new should hopefully be pretty self explanatory and allow for you to create a variety of different shader materials.

This node still requires a number of UI improvements especially concerning the pipe inputs which get hidden behind each other currently.



Shader: ReflectiveSurface

The new ReflectiveSurface node allows you to generate mirror-like reflections, translucency, transmission and more from ScanlineRender2's new ray tracing architecture.

The reflection percentage knob lets you balance between reflection and transmission with 1 being a fully reflective surface and 0 being a transmissive surface.

Reflection tint alters the reflection colour. Transmission tint doing the same for transmission. The fresnel bias sets the amount of reflectivity you see, typically based on the angle that you look at a surface. Index of refraction lets you control how much the light is bent when going through the material. The dielectric priority knob lets you set the priority order of the material compared to others to help determine which material shader takes precedence. An example being water in a glass, both would in CG have transmissive and reflective properties, but giving the glass material a higher priority with a lower dielectric value compared to the liquid material would mean we ignore any reflections and refractions from the water and just do this for the glass, helping improve render efficiency.

The surface sides knob essentially lets you set if the renderer should render a single side or both sides. And the final knobs are the same as those described for the BasicSurface node.

dielectric_priority
Set the priority order of this material where the lower the number the higher the priority. Values 0 or lower effectively disable dielectric priority.
When multiple surfaces are intersected along a transmission ray this priority setting determines which material shader takes precedence. When the ray encounters a surface with a higher value the surface is ignored since it is at a lower priority.

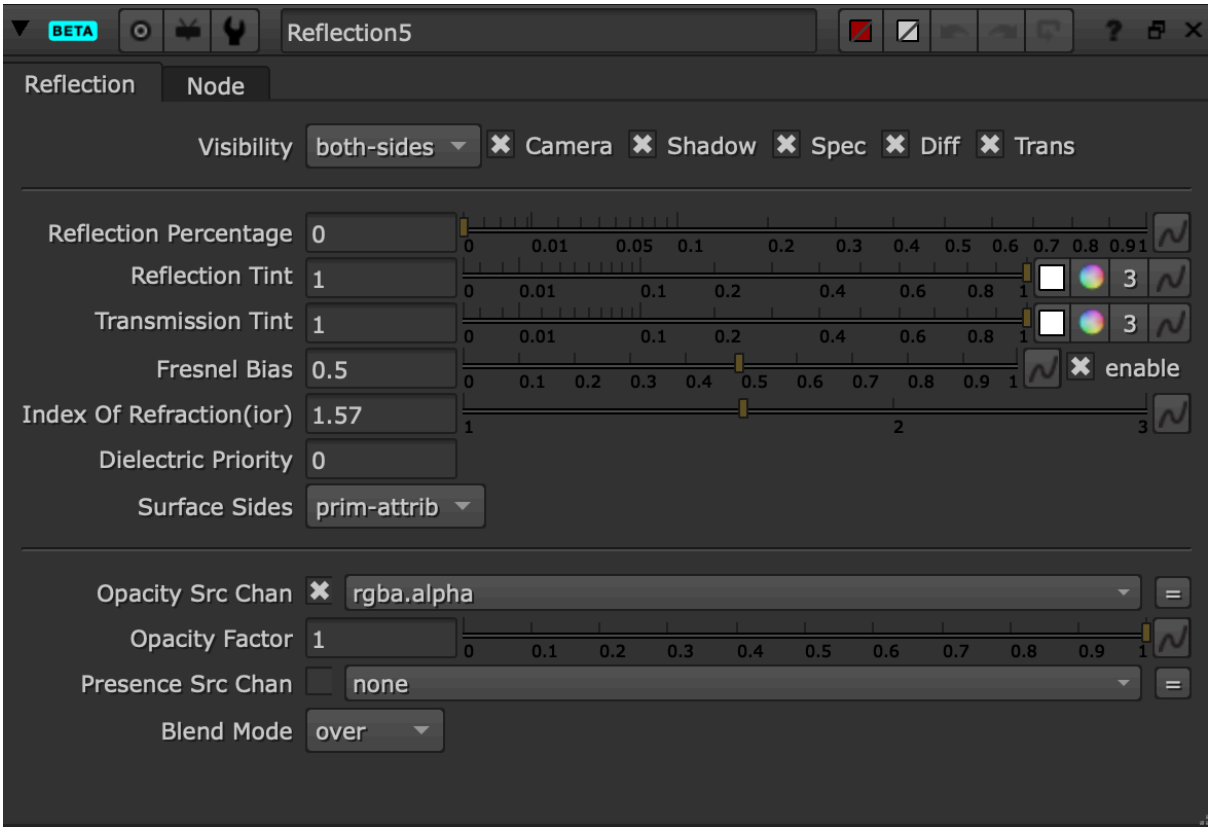
The canconical example of this is a closed mesh representing a liquid inside a closed mesh representing a drinking glass. The liquid mesh perimeter is allowed tooverlap the space inside the drinking glass wall thickness, and the liquid given a lower priority (a higher #) vs. the glass. When the transmission ray encounters the liquid mesh while evaluating transmission through the glass the liquid mesh is ignored since it has a lower priority. When the transmission ray exits the glass and enters the liquid there is no longer a conflict.

The same can be done with objects 'embedded' inside a solid, like a sphere representing an air bubble inside a solid glass cube. The sphere is set to ior=1.0(air), and a higher priority than the cube, which is set to ior=1.57(glass).

sided_mode
Hint to renderer that the surface backface should be shaded like the front, ie a thin double-sided surface. The default behavior is to retrieve the sides info from the primitive 'doubleSided' attribute, if it exists.

This hint is important for the handling of normals on the surface backface especially when determining reflection/refraction angles.

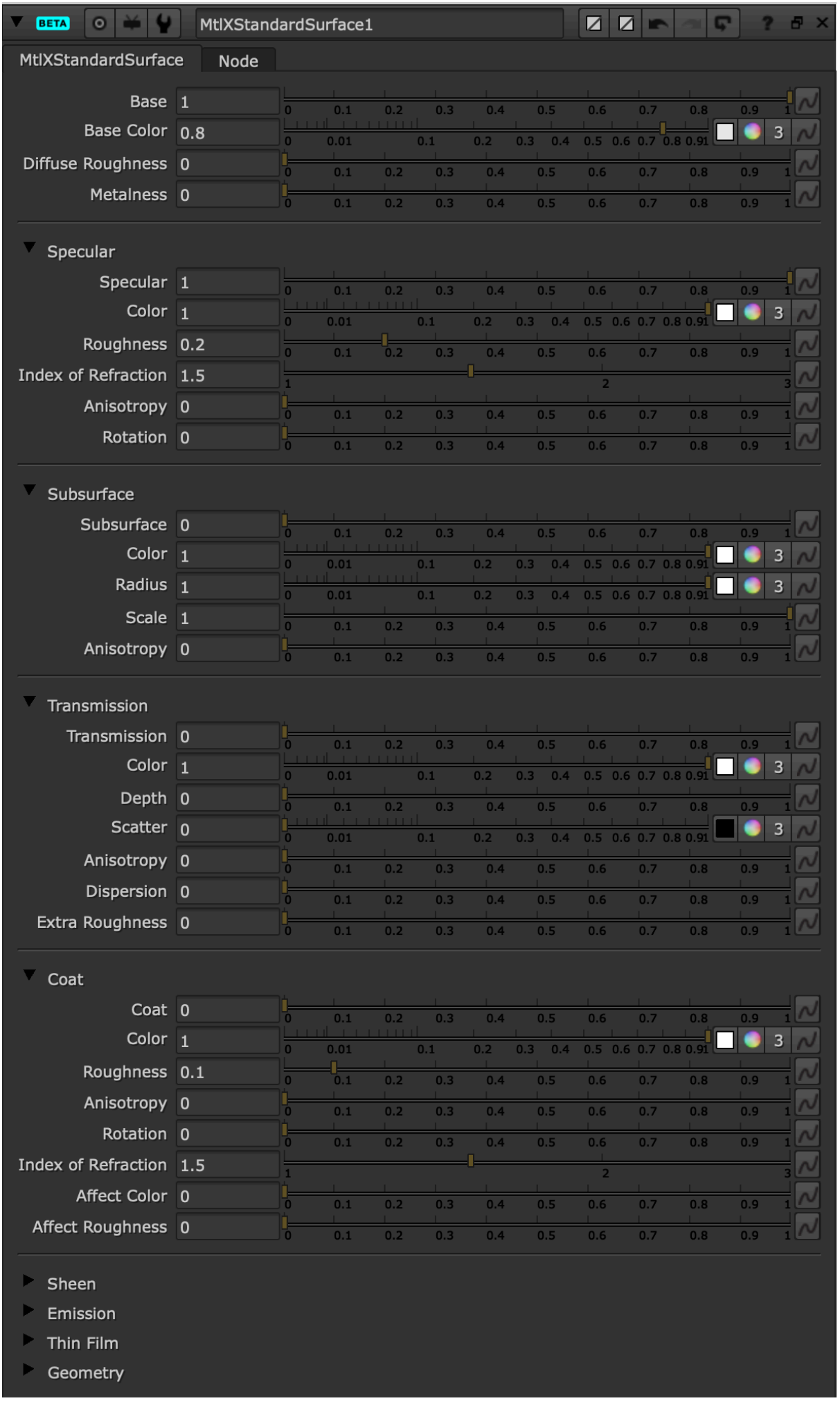
Set to 'Double-Sided' if the surface is open like a card mesh and both front and back sides should be shaded
Set to 'Single-Sided' if the surface is closed like a cube or character mesh where only front faces should be shaded and backfaces are culled.



Note: for this beta the ‘Diff’ and ‘Trans’ checkboxes are not functioning.

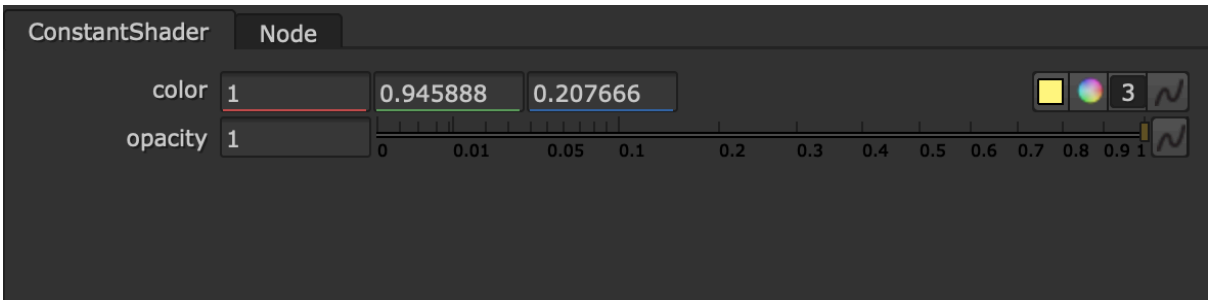
Shader: MtlXStandardSurface

The MtlXStandardSurface node is the first introduction of MaterialX shaders in Nuke and aims to act as a base shader which you can create many looks from.


BETA
MtlXStandardSurface1
MtlXStandardSurface
Node
Base 1
Base Color 0.8
Diffuse Roughness 0
Metalness 0
Specular
Specular 1
Color 1
Roughness 0.2
Index of Refraction 1.5
Anisotropy 0
Rotation 0
Subsurface
Subsurface 0
Color 1
Radius 1
Scale 1
Anisotropy 0
Transmission
Transmission 0
Color 1
Depth 0
Scatter 0
Anisotropy 0
Dispersion 0
Extra Roughness 0
Coat
Coat 0
Color 1
Roughness 0.1
Anisotropy 0
Rotation 0
Index of Refraction 1.5
Affect Color 0
Affect Roughness 0
Sheen
Emission
Thin Film
Geometry

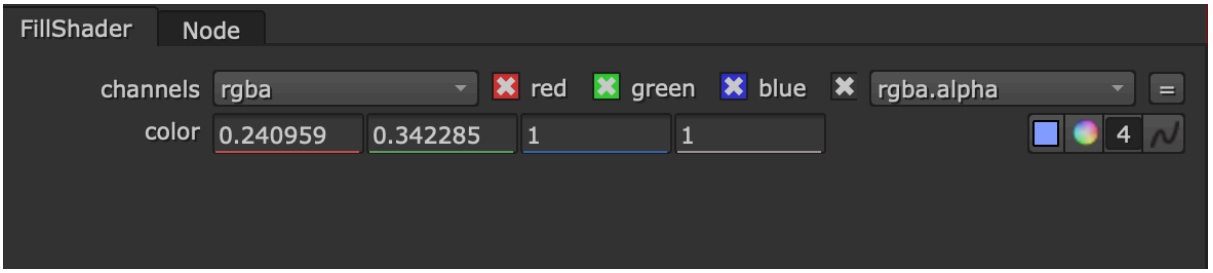
Shader: ConstantShader

The ConstantShader node lets you place a constant color across a mesh and set the opacity of the material.



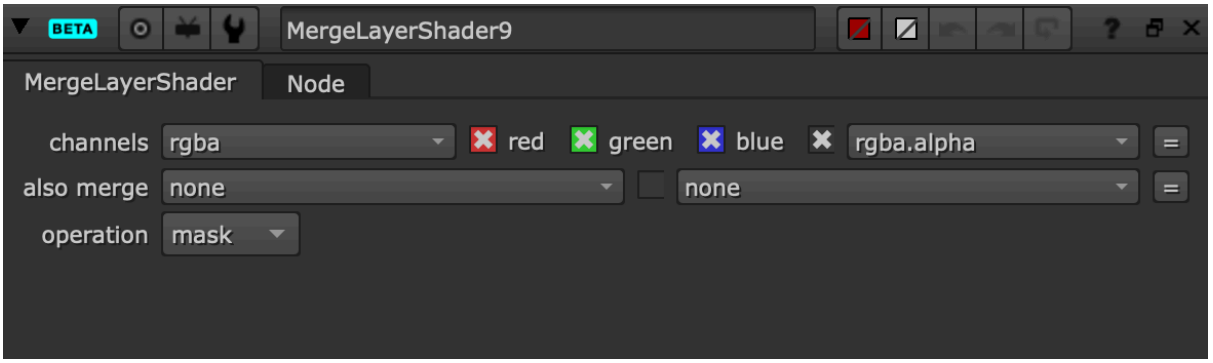
Shader: FillShader

The fill shader is similar to the constant shader in that you can populate the mesh with a colour, but replaces selected material channels with a constant color. It is the equivalent of the FillMat node



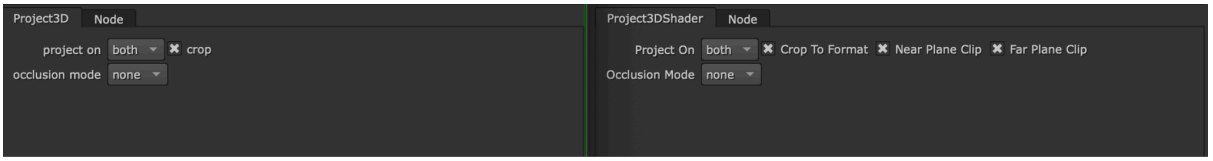
Shader: MergeLayerShader

The MergeLayer node combines two input shaders to produce a composite of the two. This node still needs some work to allow all shaders to connect to it, but otherwise functions like the MergeMat node.



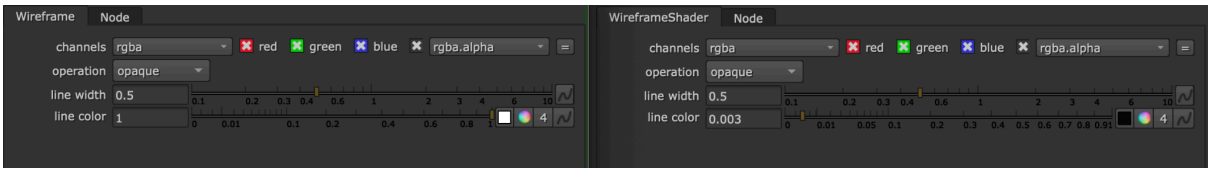
Shader: Project3DShader

The Project3DShader node is the equivalent to the Project3D node, but has two additional knobs with the ability to clip the projection beyond the cameras near and far planes.



Shader: WireframeShader

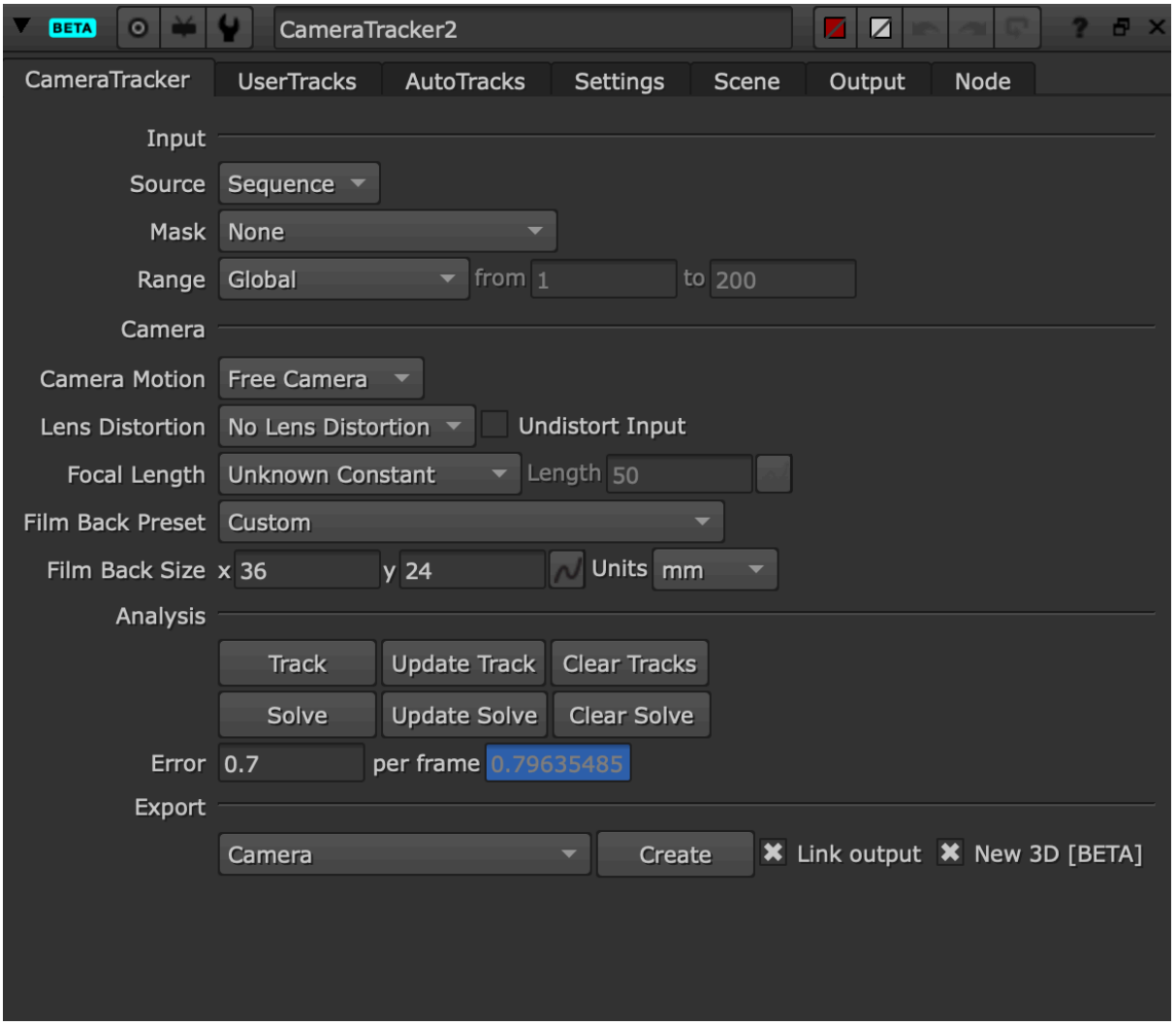
WireframeShader is the equivalent to the Wireframe node and has the same knobs, however this node has broken in recent builds so doesn't function currently.



CameraTracker & Point cloud workflow nodes

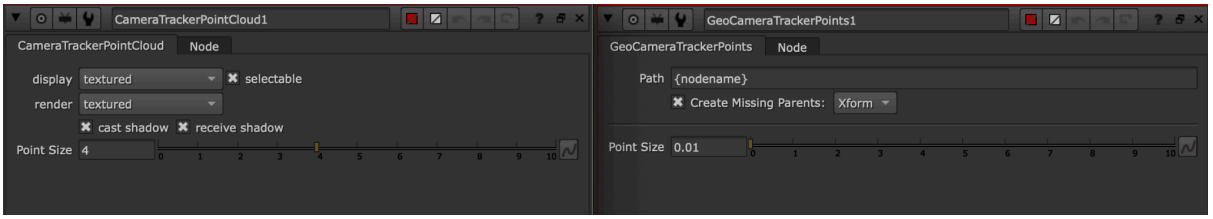
General: CameraTracker

The CameraTracker node is the same as always but you now have a knob that lets you specify if you want to create a Camera for the new system or the classic system.



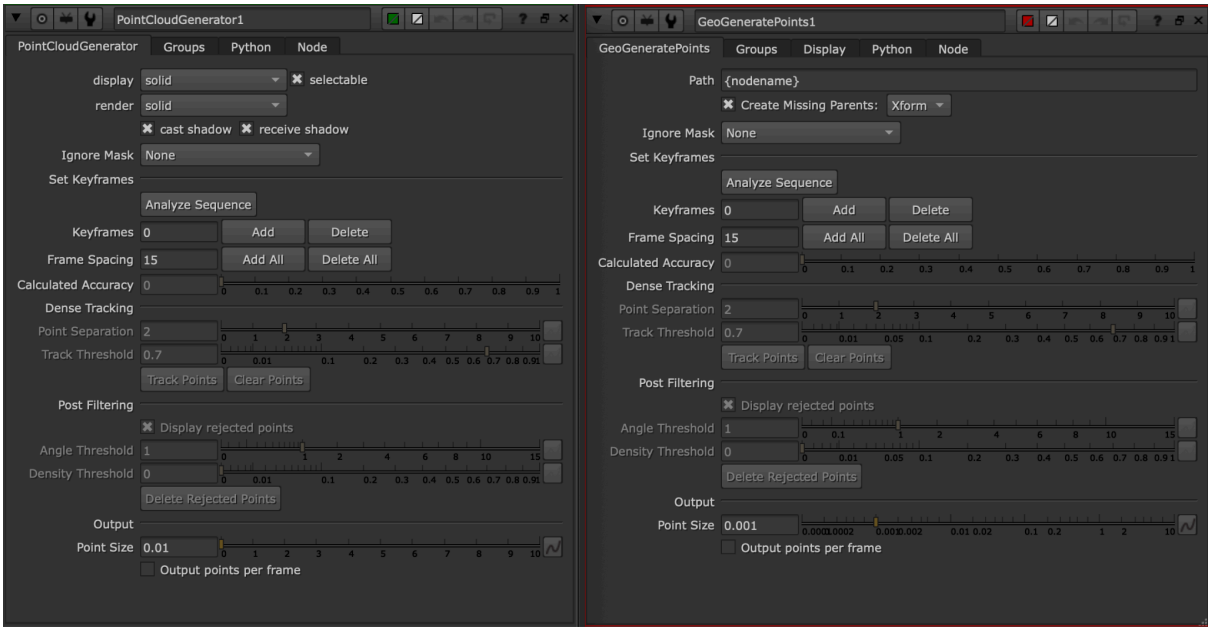
General: GeoCameraTrackerPoints

This node gets generated when creating a scene or a point cloud from the CameraTracker node and is the equivalent of the CameraTrackerPointCloud node.



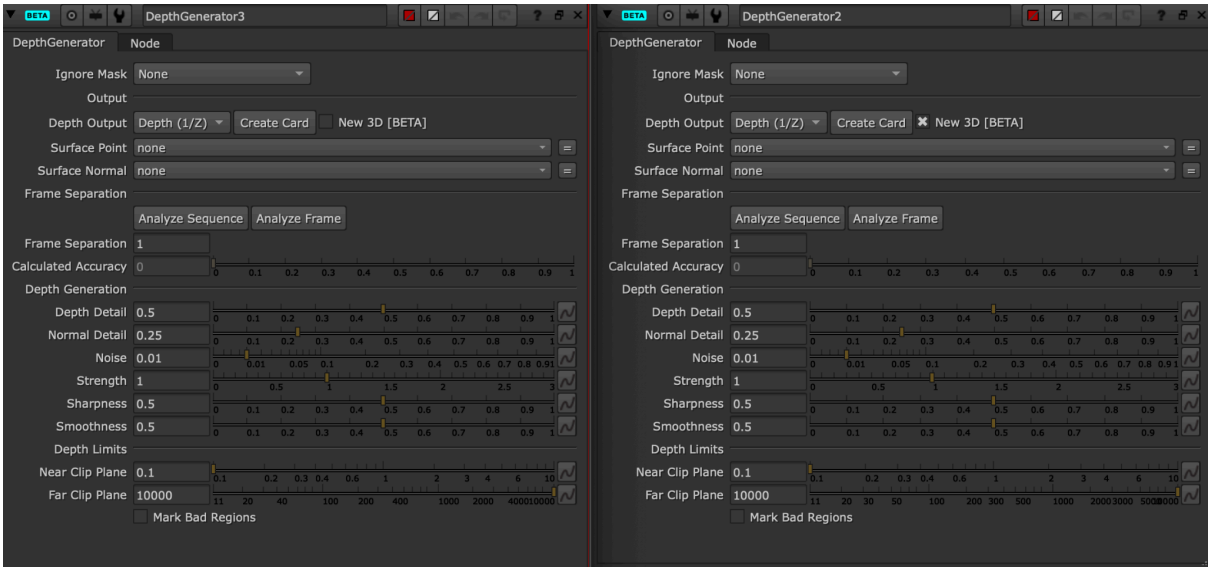
General: GeoGeneratePoints

The GeoGeneratePoints node is the same as the PointCloudGenerator node and is used to create a dense point cloud based on the information generated by CameraTracker. Knob wise it is fundamentally the same.



General: DepthGenerator

The DepthGenerator node is fundamentally the same node, but with the option for creating cards based on the depth output using the GeoCard node rather than Card node.

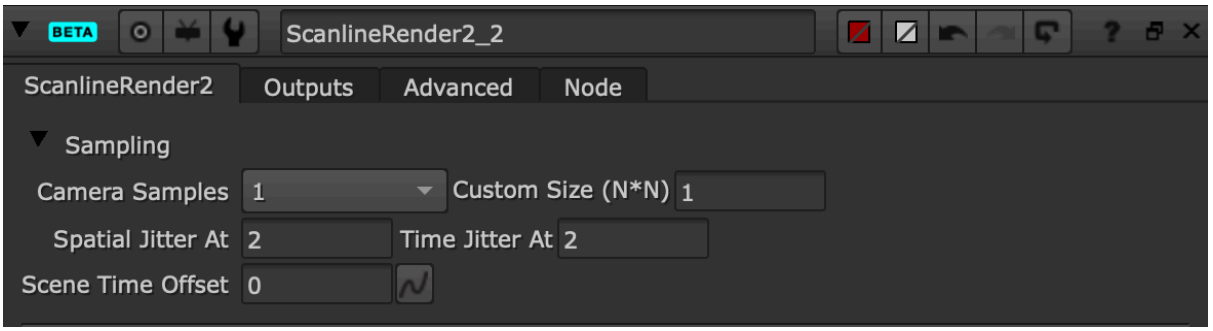


General: DepthToPosition

This is just the same node.

General: ScanlineRender2

Sampling



The sampling UI has been updated with new knobs to give you greater control over your 3D render output.

Camera Samples
The ‘Camera Samples’ knob is essentially the quality control for Slr2. More samples means smoother motion blur and less aliasing, but a longer render time. Slr1 takes a lot of samples to smoothen out, but Slr2 doesn't require nearly as many samples allowing you to get to a smoother result faster.

In Slr1 you could only control the number of samples, which were distributed between time and any kind of antialiasing. Because there was no jittering you got a "stacking" like effect for motion blur. With Slr2 you can now control the jittering and create a more natural looking motion blur.

Spatial Jitter At
The ‘Spatial Jitter At’ knob allows you to control at which sample levels Slr2 will jitter spatial subpixel locations when doing anti aliasing so that you break up the patterns for better results.

By default the location of the sampling within the pixel itself is being randomly changed at a value of 2 or higher, so only turns on when samples are above 1. When turned off only the exact centre of the pixel is being sampled when samples is 1, or in a uniform grid if greater than 1. This means you have control over which sample levels have uniform sampling (samples at a value less than the ‘Spatial Jitter At’ value) or jittered sampling (samples at a value equal to or more than the ‘‘Spatial Jitter At’ value).

Time Jitter At
The ‘Time Jitter At’ knob determines at what sample number you start randomly jittering the time for motion blur.

The default value is 2, which means jittering to produce motion blur occurs when samples are at a value 2 or higher. If samples are at 1, no motion blur will be created, but motion vectors will be produced so that you can still blur the image.

Scene Time Offset
The ‘Scene Time Offset’ knob allows you to shift the frame you are rendering and can be done in subframe increments.

This means you can slip your render in time (with subframe accuracy) without affecting the camera or the objects in the stage, which can be really helpful when you are trying to slip the camera to get vector motion blur aligned properly.

Camera/Projection

New projection modes have been added with ‘Cylindrical’ and ‘UV Unwrap’ which has been removed from the Camera node and placed as a mode in ScanlineRender2, as this is actually a special render mode rather than a projection and as such does not make sense to exist on the Camera node. Greater shutter controls also give you more flexibility.

Projection
The ‘Projection’ knob lets you set the lens projector mode you want to use to output your render data

- Render Camera: use the setting from the Camera node Projection knob
- Perspective: renders with objects in front of the camera having the illusion of depth defined by the camera's focal-length and aperture
- Orthographic: renders objects as if viewed using a parallel projection
- Spherical: renders the entire 360 degree world as a spherical map
- Cylindrical (new): renders the entire 360 degree world as a cylindrical map
- UV Unwrap: Every object renders its UV space into the output format. Used for creating texture maps
    - Note: the Camera node now no longer has UV Unwrap mode, because this is actually a special render mode, not a projection.

All the above options are now lens projector plugins, which means you can create as many as you want and add to this list (documentation and examples not yet provided). This means if you have a specific lens distortion map, you can add that as a plugin to undistort footage from a particular lens. These custom plugins can provide a custom interface (such as the UV Unwrap example)and will also appear in the Camera 'Projection' list.

Shutter
The ‘Shutter’ knob is how long the shutter should be open for when motion blurring

Shutter Offset
The ‘Shutter Offset’ knob is how the shutter behaves with respect to the current frame value.

- centered: acts as a snapshot in the middle of the frame, so motion blur would be shown as if you paused halfway through the frame
- start: opens the shutter at the start of the frame, so you see the motion blur ahead of the geo animation
- end: opens the shutter at the end of the frame, so you see the motion blur after the geo animation
- custom: opens the shutter at a time specified by the user

Nuke has always defaulted to ‘start’ and going forward in time, so at a given frame the shutter is open. However often it can be useful to change this to ‘end’ when you want a motion blur trail to an impact point on a specific frame.

Shutter Segments
The ‘Shutter Segments’ knob is the number of time 'segments' used to interpolate motion blur.
The greater the number of segments the greater the quality of rotational motion blur.

Shutter Bias
The ‘Shutter Bias’ knob moves your motion blur to be weighted more to the start of a frame or end of a frame, or one end of the shutter.

Currently this defaults to 0 to align with Slr1 and is a perfectly even distribution. However this doesn't traditionally look the most natural so we would be interested to get feedback on what users would like this to default to.

Render Region
The ‘Render Region’ knob is how Slr2 handles a bg that has a larger bbox than the render output format size.

Overscan
The ‘Overscan’ knob is the number of pixels to render beyond the format, so you can have additional pixel data if you might need to transform a render later and have more room to play with (helpful with things like camera shake etc).

Motion Blur

Greater control over the motion blur in your scene.

Camera Xform
The ‘Camera Xform’ knob enables and disables the motion blur of a camera being transformed i.e. animating a Camera through space

Please note that subsampling matrices produce artifacts currently, as this requires quaternions for the matrix interpolation method which has not yet been implemented. As a workaround increase the Shutter Segments to chop up the frame you are generating the matrix for, into multiple matrices.

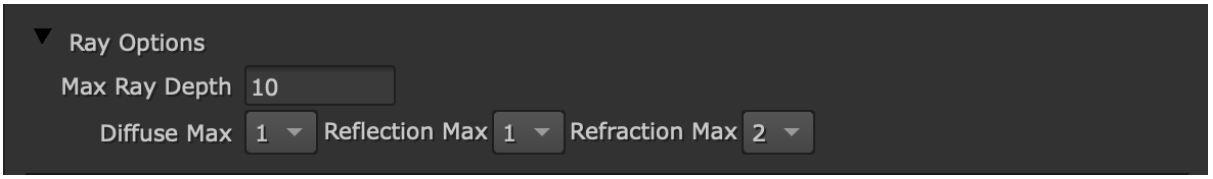
Camera Lens
The ‘Camera Lens’ knob enables and disables the motion blur of a camera projection control being changed i.e. motion blur generated from animating the focal length of the camera

Object Xform
The ‘Object Xform’ knob enables and disables the motion blur of an object being transformed i.e. animating a GeoCube through space (the mesh transform matrix is animated)

Object Deform
The ‘Object Deform’ knob enables and disables the motion blur of an object being deformed i.e. an animated radius knob on a GeoSphere (the point xyz locations are animated)

Lights
The ‘Lights’ knob enables and disables the motion blur of a light being transformed i.e. animating a light through space

Ray Options



The ability to control the ray depth across your scene at render time.

May Ray Depth
The ‘Max Ray Depth’ knob sets the number of times a ray can bounce around the scene.
The higher the number the more 'accurate' the render is, but at the cost of render time.

Diffuse Max
The ‘Diffuse Max’ knob sets at what depth count diffuse rays will stop.

Reflection Max
The ‘Reflection Max’ knob sets at what depth count glossy rays will stop.

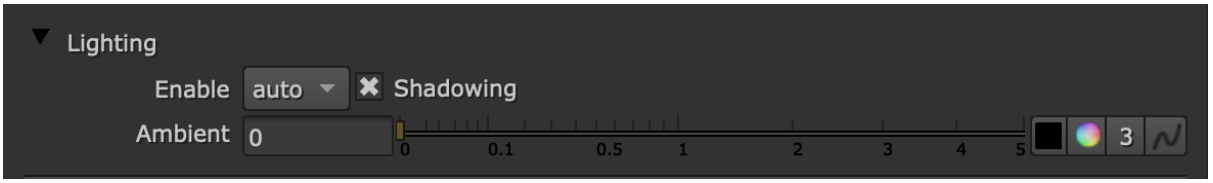
Refraction Max
The ‘Refraction Max’ knob sets at what depth count refraction rays will stop.

Material

Material controls for greater control over the output of materials at render time.

Texture Filter
The ‘Texture Filter’ knob is the default filter that texture sampling shaders will use.

Lighting



Lighting controls for greater control over the output of lights at render time.

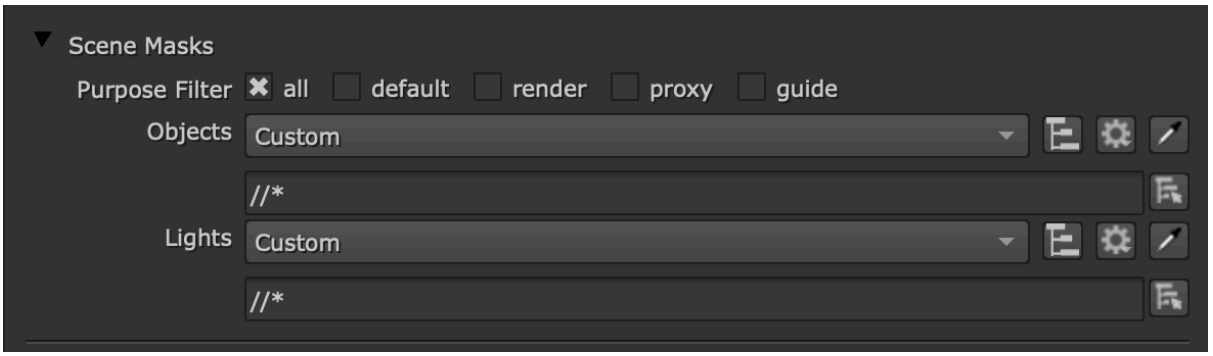
Enable
The ‘Enable’ knob sets the mode to use for rendering lights

- auto: renders with surface lighting when there are lights in the scene, or constant shading if no lights
- on: renders with surface lighting, but if no lights surfaces will appear black if not emissive
- off: renders constant shading, ignoring lights

Shadowing
The ‘Shadowing’ knob enables shadowing cast from direct lighting

Ambient
The ‘Ambient’ knob adds light globally to the diffuse surface response of surface shaders which respect this shading global.
If you set ‘Ambient’ to 1.0 with a scene with no lights, the surfaces will emit their diffuse color as if illuminated equally from all directions.

Scene Masks



Masking abilities to allow users to specify what is rendered from their stage based on purposes, or specific user masks.

Purpose Filter
The ‘Purpose Filter’ knob renders based on the purpose a prim has

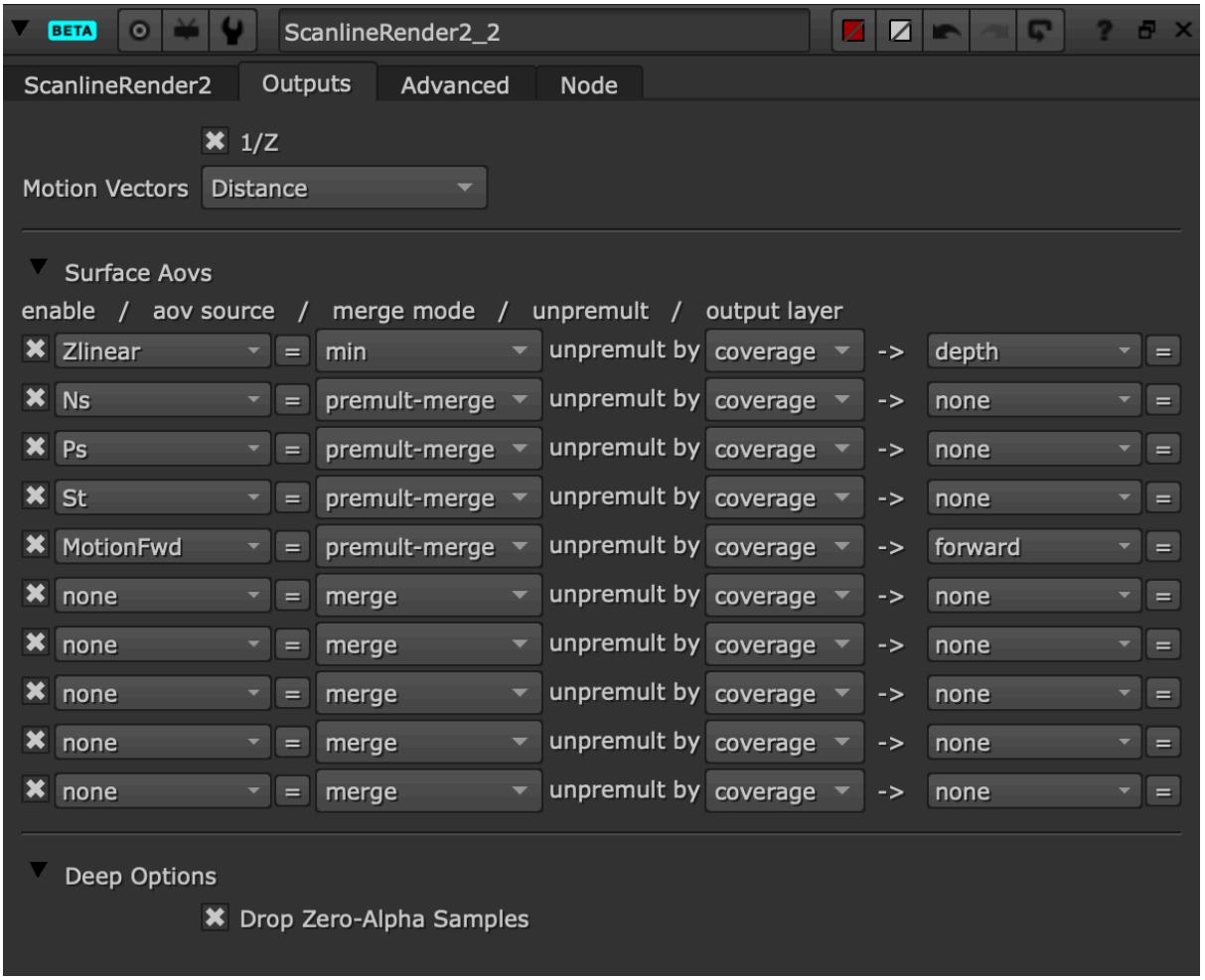
- all: renders all prims regardless of purpose
- default: renders prims based on default settings for the stage

- render: renders all prims set to render purpose
- proxy: renders all prims set to proxy purpose
- guide: renders all prims set to guide purpose

Objects
The ‘Objects’ knob allows users to set a mask for the objects they want to render

Lights
The ‘Lights’ knob allows users to set a mask for the lights they want to render

Outputs



Greater control over your AOV outputs from ScanlineRender2.

1/Z
The ‘1 over z’ knob determines how you would like to output your Z depth.
On: outputs classic Nuke Z distance, which 1/Z distance
Off: outputs absolute Z distance. This is the world space distance of the camera to the object.

Motion Vectors
The knob ‘Motion Vectors’ allows you to select the motion vector output type.

- Off: don't create any motion vectors
- Distance: stores the distance between the samples in the motion vector channel
- Distance Normalized: normalized distance between the samples in the motion vector channel

- Velocity: stores the velocity in the motion vector channel
- Velocity Normalized: normalized velocity in the motion vector channel

Surface AOVs

Enable
The ‘enable’ knob sets whether the AOV is rendered

AOV source
The ‘aov source’ knob are hard coded AOVs that are the source of the output layer

Merge mode
The ‘merge mode’ knob sets how the ‘aov source’ wants to merge with other surfaces along the same ray in depth

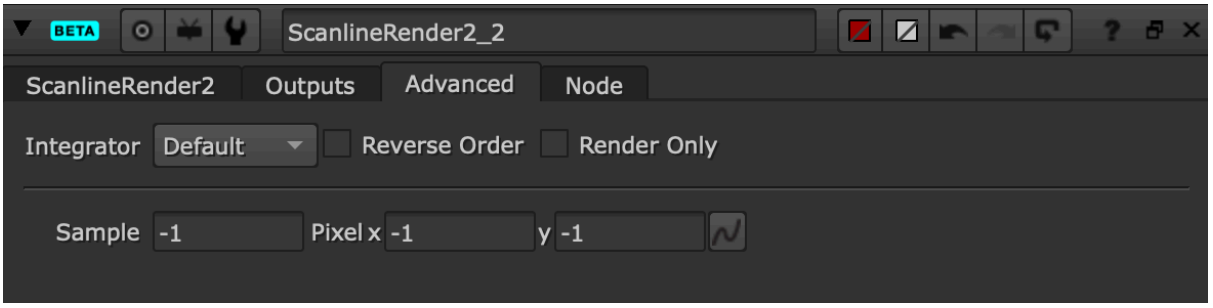
Unpremult
The ‘unpremult’ knob sets whether you would like to unpremult the final combined output.

Output layer
The ‘output layer’ knob sets the output layer I want to send this data to on render

Deep Options

Drop Zero-Alpha Samples
The ‘Drop Zero-Alpha Samples’ knob sets deep samples with an alpha value of 0 to not contribute to the output

Advanced



Advanced controls for special use cases.

Integrator
The ‘Integrator’ knob allows you to determine how you want ScanlineRender2 to take the camera rays and return different results.

- Default: render utilising the new ray tracing architecture and act as a standard utility renderer
- Debug: provides a list of modes which give you access to rendering your scene in different shading contexts, such as normals, texture coordinates and much more.
- Occlusion: renders an occlusion based result

The new Integrator plugin API can be populated with your own integrators to visualise render data for your own custom contexts. Nuke plugins named with the prefix 'slr' and suffix 'Integrator' appear in the list of available integrators to use in the Slr2 node panel.

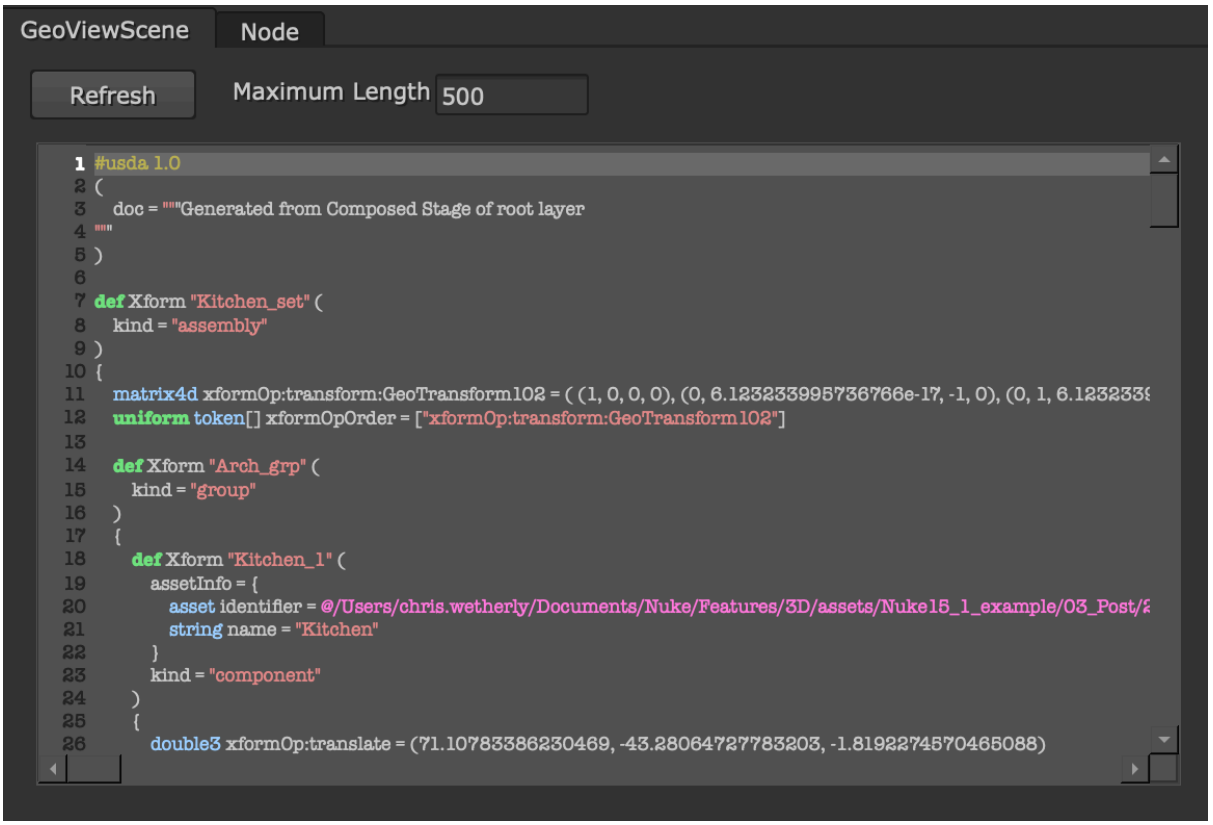
Note: Integrators are a WIP feature and only the Default and Debug plugins are expected to function correctly at this time. Documentation on setting up your own Integrators will be provided at a later date.

Default: Reverse Order
The ‘Reverse Order’ knob reverses the shading order from the farthest object to the nearest.

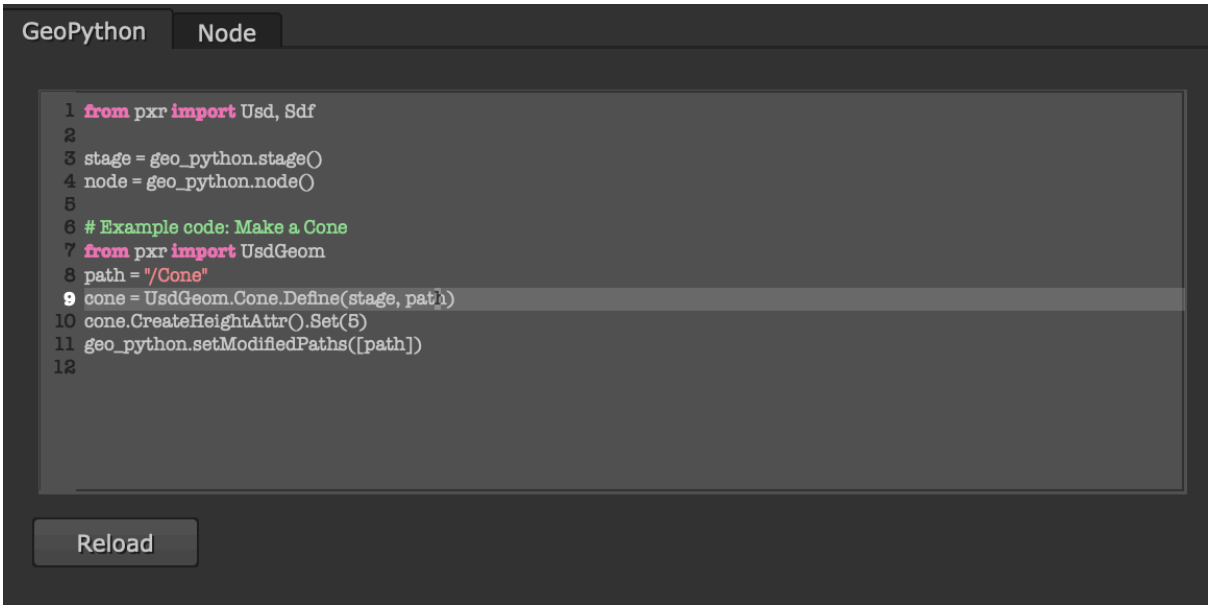
Default: Render Only
The ‘Render Only’ knob outputs the render only against black and does not take into account the bg input.

Troubleshooting

General: GeoViewScene
This is a new node that lets you view the USD stage in a read-only text form and is primarily used for debugging. Simply connect it to your stage and view. You can update the length you want to see and refresh.



Modify: GeoPython
The GeoPython node is a powerful node for artists that have some knowledge of USD and Python. While currently limited in its UI, it can be very powerful in creating new schemas and editing your USD stage.



3D features

Scene graph

The scene graph is one of the best ways to visualise your USD stage hierarchy. While we have tried to design the new system to be functional without having to display the scene graph at all times, I personally find it nice to use an updated comp view that has it in the bottom right corner so that I can easily select through and drag and drop prim paths into mask knobs if needed.

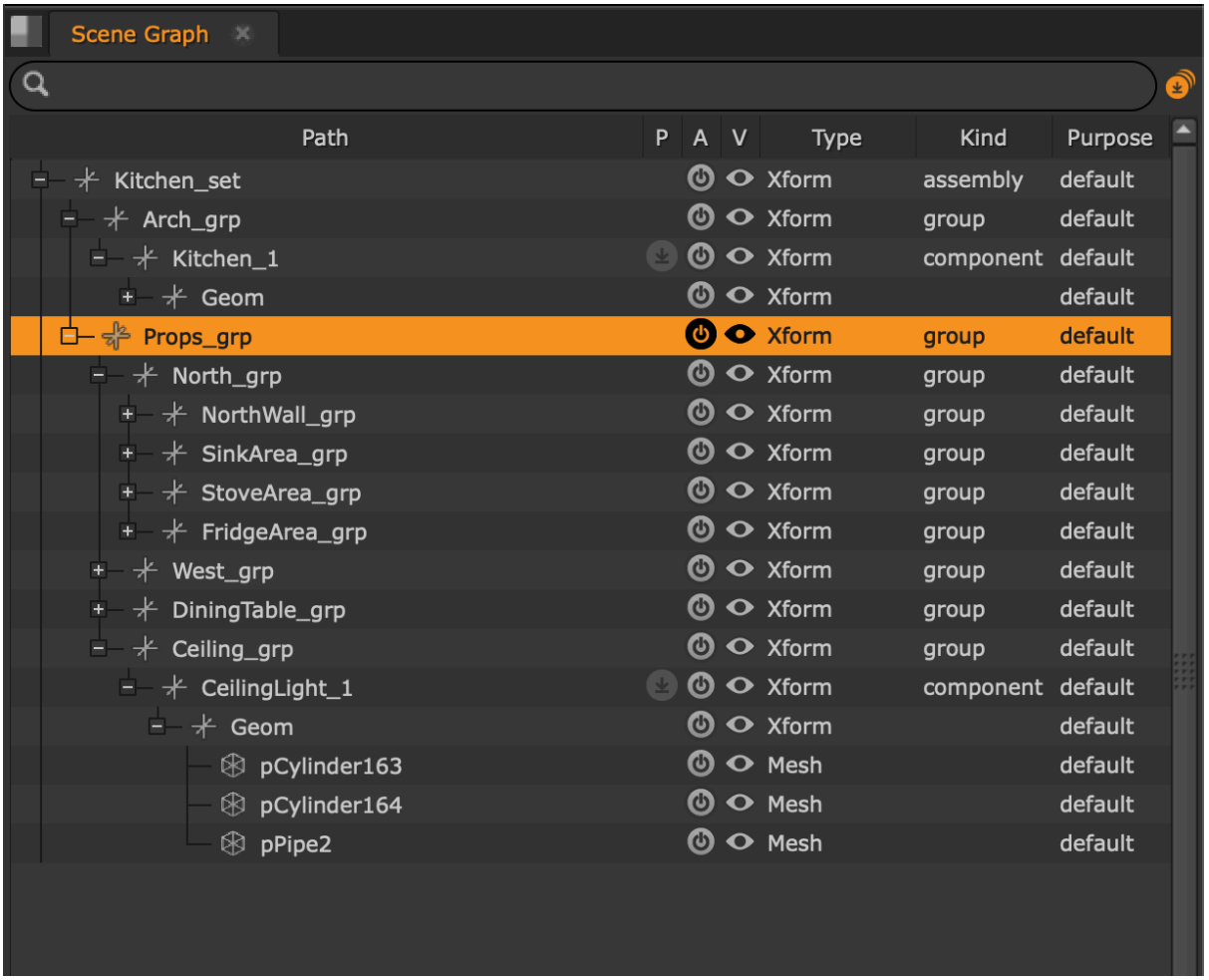
Selections in the viewer and the scene graph will sync making for easier navigation. You can search for specific prim names and hit enter to cycle through them.

The columns show the prim path, if the prim is payload, its activation and visibility status, the Type of primitive, the prims Kind value and the render purpose.

There is also a Payload button on the top right which controls whether payloads appear loaded or unloaded when opening a USD stage, as well as the ability to toggle payloads in the payload column and the ability to toggle activation and visibility of prims which will create a viewer only override designated by a small yellow icon. After playtesting these features and talking with users we have come to the conclusion that these controls need to be reworked and in some instances controlled elsewhere, as the node graph should be the source of truth for where you make artistic decisions about activating or hiding prims for instance and payload controls should really be a part of your import process.

That is not to say we can't see scene graph enabled workflows being useful, such as the ability to right click a prim in the scene graph to generate a node from it, but we will always want to balance this functionality with the node graph being the source of truth so that an artist doesn't encounter issues with a shot, looks through all their nodes to find the problem only to see that there is something set on the scene graph causing the issue.

An update we're also interested in doing is having the search functionality filter down based on what you are typing to make it easier to visualise your search criteria with the ability to toggle back to this original functionality.



Paths and masking
A new concept in the new 3D nodes is the ability to set where a prim lives inside of the scene graph hierarchy which is achieved using the path knob and the ability to have a node affect a specific prim or selection of prims using the mask knob.

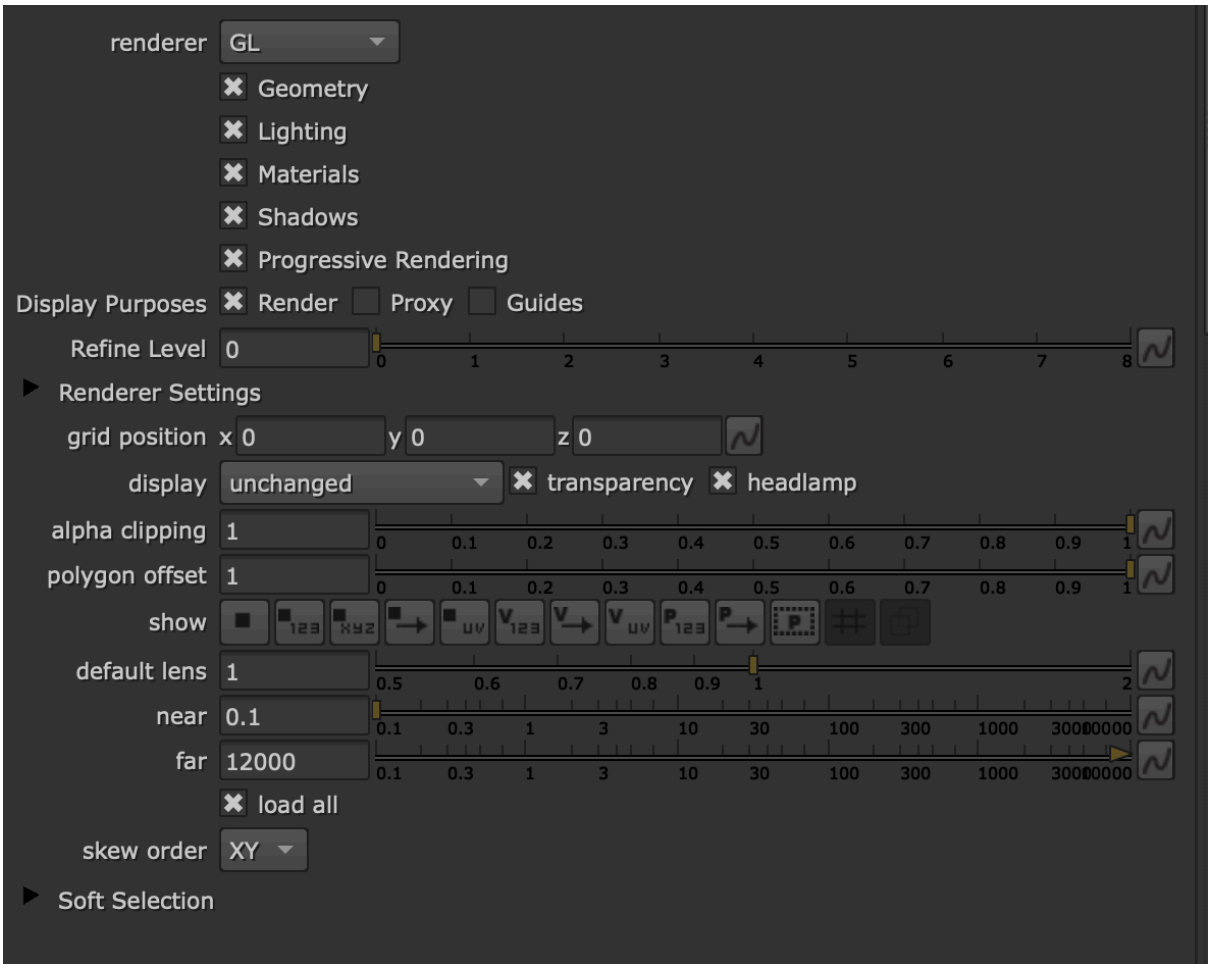
The philosophy of the path and mask knobs is that as much as possible we want to make nodes be as drag and drop as possible, where you don't have to make any edits if you don't want to, but allowing this functionality for artists who want to go deeper into working with a scene graph hierarchy and the new workflow benefits. We essentially think of paths like creating channels and masks like masks in 2D nodes which you can populate with channels to direct how you want a node to affect your output.

One thing to be wary of with certain creation nodes like GeoCube, or GeoReference, is that the prim created from this node is often named based on the name of the node i.e. GeoCube2. This is because each prim being created needs to have a unique path. This is helpful but artists then need to be mindful that if a modifier node like GeoTransform sets a custom mask pointing at a specific prim, like GeoCube2, if you then copy that cube and transform this will break in the pasted node network as that copied cube name will have updated, but the mask in the GeoTransform will not have, resulting in the node erroring.

We are exploring copy and paste mechanics that could update masks based on stage changes at paste, but that is still being investigated, so please be mindful when copying and pasting about the mask paths.

Hydra viewport
The viewport in Nuke is now a Hydra based one which currently renders using HdStorm. But the great thing about it being Hydra is that we can update the viewer to support additional render delegates so you can pick the right renderer for your viewport experience. This functionality is viable for TD's to implement themselves, but isn't quite ready yet for an out of the box user facing experience. This is something we are keen to implement to help

facilitate new workflows and pipelines where artists can have greater collaboration through a shared viewer experience and we've already seen some great examples of users implementing different renderers to use in Nuke's 3D viewport.



Two tier selection
A new feature for the 3D system is the ability to work with two tier selections now. This gives greater fidelity to make a selection of faces or vertices on one particular mesh, unlike the previous selection behaviour where you could only select everything meaning you had to be much more careful about how you perform group selections.

The two tier selection is based on the prims Kind attribute type and this can be changed in the new 3D sidebar. As every USD scene will be set up differently there is no way to default to one Kind type that will be perfect for each stage you import, but with these controls you can swap your selection type to best match the asset you are working with. A nice ease of use feature we'd like to add is a hotkey to traverse up and down the kind selection types, so you don't always have to go into the sidebar and would allow that if you make a selection but its at the wrong Kind level, then the shortcut would allow you to navigate up and down to get to the right selection level.

Please note that with two tier selection, you do need to make sure you select a prim before you can see the vertices or faces now.

Another new feature in the selection system is the ability to enable or disable occlusion culling so that you can drag select and either select vertices and faces on both sides of the mesh, or just the side facing you in the viewer.

Finally you can also toggle whether you want geometry, cameras, or lights to be selectable in your viewer.

The sidebar also has a control for letting you swap between object, screen and world space for viewer navigation.

One thing to note is that if you have been in face or vertex selection mode, it is best practice to reactivate the select nodes mode as otherwise you could be trying to select geometry in your 3D viewer with no selection happening and not knowing why.

Frequently asked questions

When is the new 3D system coming out of beta?
The 3D system is targeted to come out of beta for the next upcoming major release of Nuke, with a focus on workflows artists use most, with a main focus on projection and matte painting workflows. The goal being to give artists the ability to bring in larger, more detailed assets than ever before, set up projections and matte paintings that they need and then render with greater fidelity via ScanlineRender2. An example of this would be bringing in a large city scape asset that wouldn't have loaded previously, with a hero asset animated character like King Kong. As a comp artist you can then add a matte painting on the horizon and cards for smoke plumes across the city and projections for adding details onto buildings or performing cleanups. All of this can be rendered via ScanlineRender2 with the new comped elements being rendered via raytraced reflections in the building glass. Or if you have a shot where you need to comp an actresses performance into the reflection of King Kong's eyes to just bring in the eye mesh data you need, place the actress's performance on a card and add a reflective material to the eye mesh to render a reflection of the performance you can comp back in over the top.

Our aim is to deliver this workflow in the next major release of Nuke, but we are still working hard towards this and if you've read through this document and played with the most recent beta, then testing and feedback of these workflows is really helpful for us so we can address bugs and workflow feedback that is important to you.

Please note that not all nodes will come out of beta and there are systems such as particles which will be targeted for updated workflows on later releases.

Why did you base it on USD?
Because USD allowed us to service what comp artists do most often in the most performant way. It allowed us to bring in mesh data with a low performance cost to be used for all the comp workflows we have been doing for years.

We also try to think to the future and there are workflows that USD will afford that we feel really excited about bringing to the compositing space and making sure comp artists have access to the same data and rendering capabilities as other DCC's.

We also believe in open standards at Foundry and the ability to support pipelines built as you need.

Why are there not more USD focused features, like an attribute panel?
These are definitely workflows we're keen to get to, as giving more tools to better understand your stage will be important for unlocking the wider potential of the USD architecture, but first we wanted to focus on ensuring the new system delivered on the workflows artists rely on for their everyday productions.

Our aim is to keep introducing these workflows with each release and to try and include some USD specific updates as well each time, but our first priority is to improve on workflows and then introducing new USD specific tooling.

Why did you change the node colours?
This has been in feedback on posts and I think this was a fair criticism. The honest answer is this was a mistake on our part in the initial implementation of the new system. We wanted to help create a clear visual understanding of what are new nodes vs classic, but in doing so removed some of the visual language artists had been used to. We always planned to bring back this visual language but as a priority for each release this never rose to the top compared to other feature work. It's not down to a lack of caring as I've seen in comments, as we also didn't want to rush into reintroducing the exact same colour scheme as before, but use it as an opportunity to explore how can we look to update the node graph visual language, but we do understand that these details matter to artists and this work has been updated for this beta.

Why didn't you just add these features to the classic 3D system?
This whole project has been undertaken because a big part of feedback we got from artists and studios over the years is about wanting improvements to decades old systems, whether that be roto, tracking etc and the 3D system consistently was one of the top priorities based on that feedback.

The classic system had a ceiling on what data could be brought into it and the performance that could be achieved, such as bringing in a hero character alembic, applying textures and seeing this playback really slowly in the viewer. It also limited greater collaboration between comp artists and other departments and how instrumental comp workflows can be in different parts of the pipeline.

The roadmap aka what's next?
So the upcoming release is focusing on projection, matte painting, import, shader and rendering workflows.

Beyond that we want to provide a better export experience, support for instancing and an updated particle system. We're also keen to update to Hydra 2.0 and explore supporting additional render delegates in the viewer, but these are some of the next focal points, with others like a revamp of model builder workflows, variant workflows and new USD tools like an attribute panel also open as options. Please let us know what is important to you.

Authoring modes, what are they?
Alongside some nodes like GeoCube you will see new knobs to the left hand side that let you select from three options, set, no change and revert to default. These are authoring modes. In USD you author by setting the value on a prims attribute and this is considered 'authoring an opinion'.

Because the value you set is an opinion, there may be stronger opinions set elsewhere in your stage that modify this attribute value when the final stage is composed.

Opinions can come from a bunch of different sources: attribute default values, class inheritance, overrides in earlier layers and so on. We don't want a node to disrupt all those carefully-crafted opinions at once and need a way to say which ones we want to modify. Otherwise, for example, you wouldn't be able to use GeoEditCamera to change the focal length of a camera without changing every other attribute as well. This is more obvious in the context of editing nodes, but is also useful for the creation nodes as well: we don't want to force opinions on every attribute because later parts of the pipeline may be providing weaker opinions which would then fail.

A Nuke knob can only have 2 states, the default value assigned at knob creation and a set value, so to support USD's authoring with Nuke's knob states we have implemented these new authoring controls, which can specify 3 modes:

Set - the Knob's current value, regardless of whether it's the Knob's default or not, is set on the USD prim attribute, or 'authored'. Depending on whether the Knob's value matches the USD prim attribute's default schema value will determine if the attribute is actually 'overridden'.

No Change - do not author any value on the USD prim attribute, leave it alone.

Reset Default - reset the Knob's value back to the -Knob's- default (not the USD schema default)

Can I work with the classic and new 3D system in the same node graph?
Yes. While most nodes are independent to either the classic or new system, the Camera, Axis and Light nodes can connect to any node in Nuke and allow you pass data from one system to the other.

Early on we decided to create a new 3D system that was separate from the classic system so that the new system wouldn't be hindered by the classic one, while still ensuring that artists would always be able to use all the workflows they needed with access to the classic system and with the aim of deprecating the classic system in the future.

This is still our overall aim, but it has also become apparent that there is a lot of value for artists in being able to work within both systems until we have reached the point where the new 3D system has sufficient workflows that we can deprecate the classic 3D system.

As such we have been building ways to connect between the two systems with the GeoPoints node, the rearchitecture of the Camera, Axis and Lights node to allow data to flow between all 3D nodes and rather than treating it like two systems, have been just using all the nodes as part of one 3D system, where you switch between the two depending on your needs as an artist.