
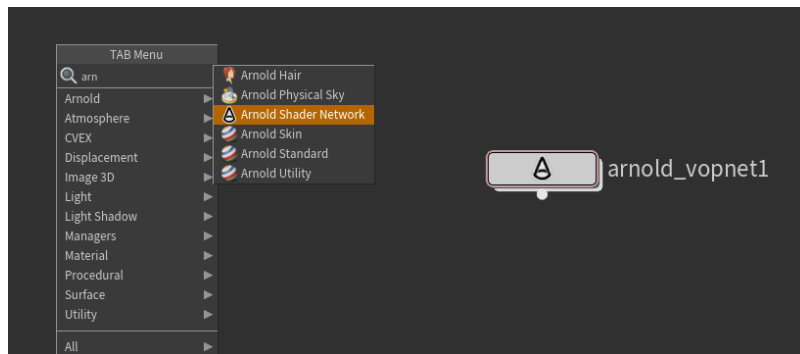



# Shaders

 You can right-click shaders, Arnold VOP networks and ROPs to upgrade them to the current version of Arnold and HtoA. You can also globally convert all the deprecated shaders in the scene from the Arnold menu. Conversion scripts for the deprecated `standard` and `hair` shaders, the Arnold ROP from a previous version of HtoA and the discontinued `alSurface` and `alHair` shaders are provided. For shaders, the conversion will attempt to match the look as closely as possible with a `standard_surface` or `standard_hair` shader.

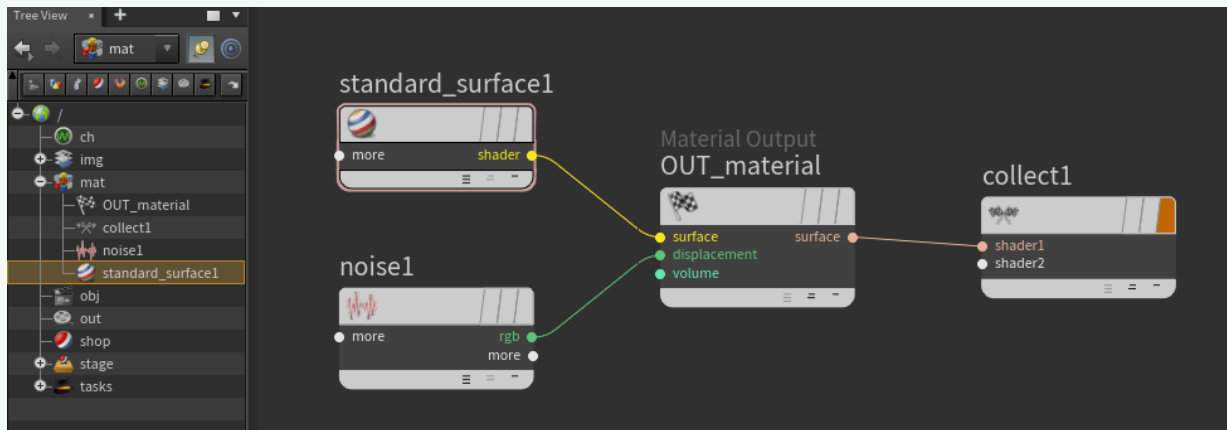
Shaders are created within Arnold Shader Network nodes. These nodes are also used for [Light Filters](#) and [Environment Effects](#).

In the SHOP context **Tab > Arnold > Arnold Shader Network** to create an `arnold_vopnet1` node. Double-clicking or pressing "I" enters a custom VEX builder context where Arnold shaders and nodes can be networked together.



 You can create shaders in `/mat` in the `arnold_materialbuilder` vop network as well as the old way in `/shop` in the `arnold_vopnet` vop network. Use the material flag to enable picking of the vopnet as a material at the object level.

The `OUT_Material` node allows collecting the surface/displacement/volume Arnold shader graphs into one material. You can then connect the output to a collect node for example in the `/mat` context.

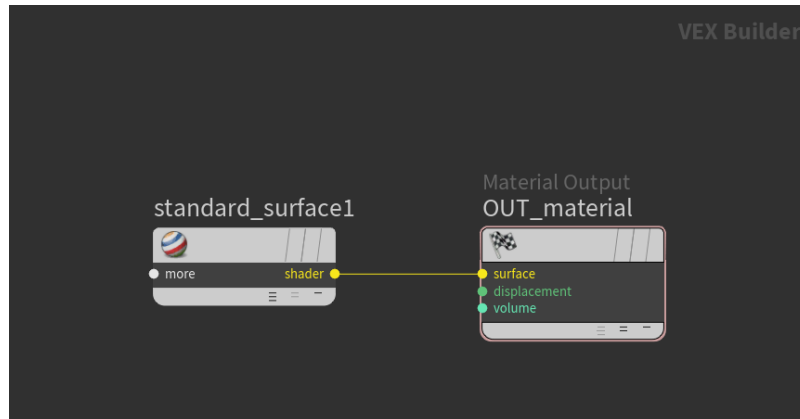


Arnold shaders created in `/mat`

## Output Nodes

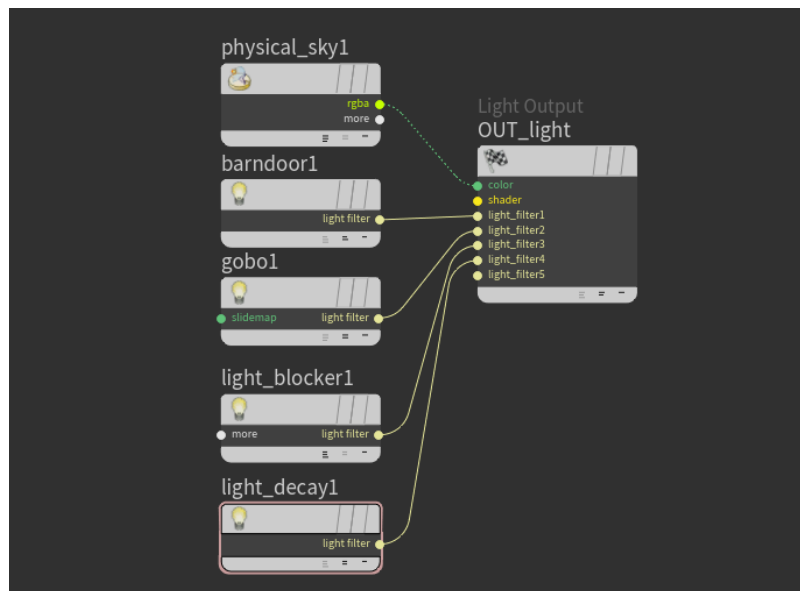
There are three types of output depending on the purpose of the shader network; *Material*, *Light*, and *Environment*. The result of the shader network must be connected to the input of the relevant output node. Multiple Outputs of different types can be contained within the same shader network. If multiple outputs of the same type exist, Arnold will use the first one translated by HtoA.

### Material Output



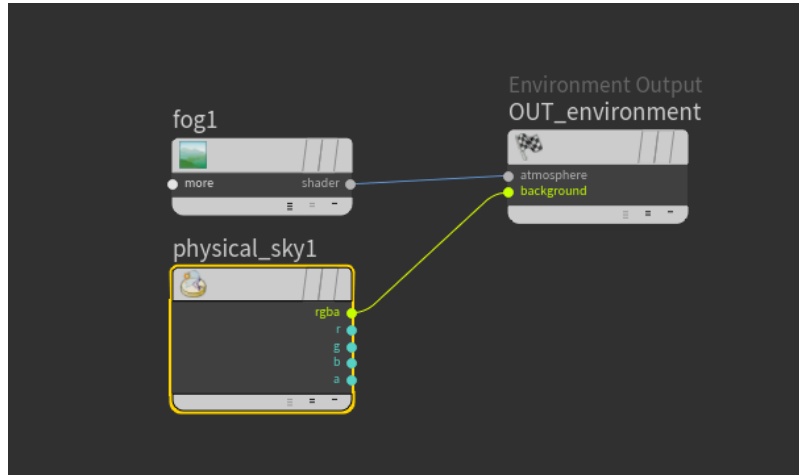
- **surface**: To connect the color output of a surface shader like **Standard Surface** or **Lambert**.
- **displacement**: Connect a bitmap texture (image node) or procedural texture, like noise, to displace the geometry surface.
- **volume**: For connecting a **Volume Collector** shader.

### Light Output



- **color**: For attaching a shader (like a **physical\_sky**) to be used as the color of a light.
- **light\_filter #**: Light Filter nodes are connected here. As each is connected another input will appear.

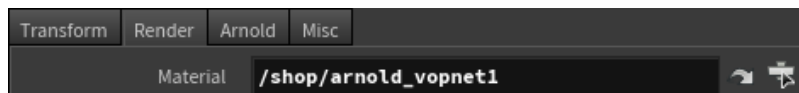
### Environment Output



- **atmosphere:** For connecting the atmosphere nodes *fog* and *atmosphere\_volume*.
- **background:** For connecting background shaders like *sky* or *physical\_sky*.

## Shader Assignment

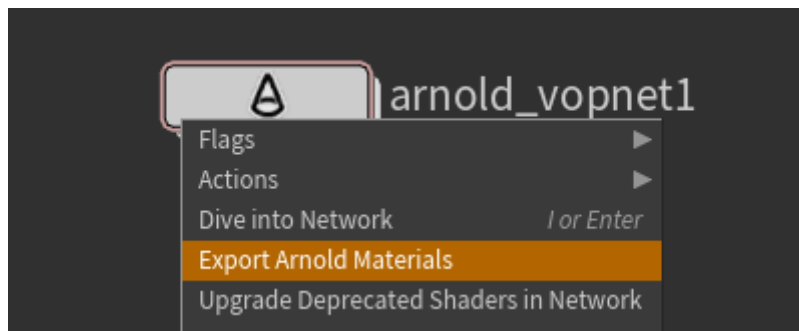
To assign a vopnet drag it from the SHOP onto the geometry or use the **Operator Chooser** in the geometry **Material** tab.



The 'jump to operator' in the material tab will go to the Arnold shader network, rather than jumping to the shader inside. This is Houdini's behavior because you're assigning a SHOP and thus it brings you to it, not inside.

## Export Arnold Materials

You can export materials in an arnold\_vopnet or from an OUT\_material VOP by right-clicking with the context-sensitive menu. This exports all of the shader graphs connected (surface, displacement, and volume).



## Shader List

These are the available shaders and nodes.

Core Shaders	Math Shaders	Utility Shaders
<ul style="list-style-type: none"> <li>Ambient Occlusion</li> <li>Bump2d</li> <li>Bump3d</li> <li>Flat</li> <li>Hair</li> <li>Image</li> <li>Lambert</li> <li>Motion Vector</li> <li>Noise</li> <li>Skin SSS</li> <li>Standard</li> <li>Ray Switch</li> <li>Utility</li> <li>Wireframe</li> </ul>	<ul style="list-style-type: none"> <li>Abs</li> <li>Add</li> <li>Atan</li> <li>Cache</li> <li>Complement</li> <li>Cross</li> <li>Divide</li> <li>Dot</li> <li>Exp</li> <li>Fraction</li> <li>Invert</li> <li>Is Finite</li> <li>Ln</li> <li>Log</li> <li>Max</li> <li>Min</li> <li>Mix</li> <li>Modulo</li> <li>Multiply</li> <li>Negate</li> <li>Normalize</li> <li>Pow</li> <li>Reciprocal</li> <li>Sign</li> <li>Sqrt</li> <li>Subtract</li> <li>Trigo</li> </ul>	<ul style="list-style-type: none"> <li>Cache</li> <li>Clamp</li> <li>Color Correct</li> <li>Compare</li> <li>Fetch</li> <li>Length</li> <li>Linearize</li> <li>Passthrough</li> <li>Random</li> <li>Range</li> <li>Space Transform</li> <li>Switch</li> </ul> <p><b>Data Types</b></p> <ul style="list-style-type: none"> <li>Color Convert</li> <li>Int To Float</li> <li>Float To Int</li> <li>Float To Matrix</li> <li>Float To RGB</li> <li>Float To RGBA</li> <li>RGB To Float</li> <li>RGB To RGBA</li> <li>RGB To Vector</li> <li>RGBA To Float</li> <li>Matrix To Float</li> <li>Vector To RGB</li> </ul> <p><b>Volumes</b></p> <ul style="list-style-type: none"> <li>Blackbody</li> <li>Standard Volume</li> </ul> <p><b>Environment</b></p> <ul style="list-style-type: none"> <li>Fog</li> <li>Atmosphere Volume</li> <li>Sky</li> <li>Physical Sky</li> </ul> <p><b>Light Filters</b></p> <ul style="list-style-type: none"> <li>Light Decay</li> <li>Light Blocker</li> <li>Gobo</li> <li>Barndoor</li> </ul>
<p><b>AOV Shaders</b></p>		
<ul style="list-style-type: none"> <li>AOV Read Float</li> <li>AOV Read Int</li> <li>AOV Read RGB</li> <li>AOV Write Float</li> <li>AOV Write Int</li> <li>AOV Write RGB</li> </ul>		
<p><b>User Data</b></p>		
<ul style="list-style-type: none"> <li>User Data Float</li> <li>User Data Int</li> <li>User Data RGB</li> <li>User Data RGBA</li> <li>User Data String</li> </ul>		
<p><b>Shading State</b></p>		
<ul style="list-style-type: none"> <li>State Float</li> <li>State Int</li> <li>State Matrix</li> <li>State RGB</li> <li>State Vector</li> </ul>		
<p><b>Ramp Shaders</b></p>		
<ul style="list-style-type: none"> <li>Ramp RGB</li> <li>Ramp Float</li> </ul>		