

## Differences Between MaterialX Specification v1.33 and v1.34

### **Custom Types**

Custom types are now allowed to define specific named and typed member variables, making custom types more equivalent to "structs" in modern programming languages. Custom types can also now define a "semantic" to indicate how the type should be interpreted, and we have defined two such semantics: "color", indicating types that are expected to be color-managed, and "shader" to indicate a type that is the output of a rendering shader; see below for details. Custom types further can also define a semantic-specific "context" for the type, which "shader"-semantic types use to differentiate between surface, volume, displacement, and light-context shaders.

With the addition of named/type member variables, the now unnecessary "valueformat" attribute for typedefs has been removed.

### **Shaders**

Most significantly, the way "shaders" are handled has been overhauled: shaders are now just nodes, differentiated from texture-processing nodes (now referred to as "pattern" nodes) simply by declaring a "shader" semantic on its output type, indicating that the node can be referenced in a material. As such, the way that custom nodes and shaders are defined in terms of their parameter interfaces and their implementations are now completely orthogonal: what was called "shaderprogram" in v1.33 is now just "node", and the v1.33 "shadertype" is now expressed using the context definition of the shader node's output type.

Since shaders now output specific custom types, the shader-specific concepts of AOVs and "coshader"-type inputs have been removed. Since shader nodes may now be used within a nodegraph, the Standard MaterialX Shaders have also been removed, under the assumption that specific implementations will provide overloaded definitions of standard node types appropriate for the custom types used by shader nodes as needed.

Shader inputs may no longer be statically-bound to nodegraph outputs: instead, `bindinput` elements should be defined within a material to express this connection.

### **Custom Nodes**

Nodedefs for custom nodes and shaders can now specify `category`, `default/defaultinput`, and `require` attributes, just like standard nodes implicitly do. Default values for custom node/shader parameters and inputs are now provided in the `<parameter>` and `<input>` elements within the nodedef.

`Implementation` elements for custom nodes/shaders now refer exclusively to external file definitions and may optionally specify the `language` of the source as well as the rendering `target` list (not "application" as in previous versions) it applies to, while any nodegraph element may be used as the functional definition for a node/shader by adding a `nodedef` attribute (and optionally a `target` attribute as well) to the `<nodegraph>` element; the list of inputs/parameters for the nodegraph are now provided solely in the nodedef, and not replicated in the nodegraph. The `target` attribute is also applicable for custom parameters and properties ("application" is deprecated here as well).

The attribute used to select which output of a multi-output custom node to connect to has been changed from "outputname" to just "output".

## Namespaces

MaterialX now supports the definition of namespaces for included files.

## Terminology

We have renamed a few MaterialX terms to better align with common usage in industry. In particular, "opgraphs" are now called "nodegraphs".

## Parameter Curve Values

Parameters can now be given a "baked per-frame" value curve instead of a static value.

## Other Changes

- The specialized string types "opgraphnode", "opgraphname" and "shadernode" have been removed in favor of just using regular string types.
- Attributes to specify geometries are now of type `GeomName` or `GeomNameArray` rather than just "string".
- We have clarified that the UTF-8 character encoding is the default for MTLX files, and that this encoding is expected to be used for in-memory representations of strings.
- We have added a new `convolveops` "require" declaration for convolution opgraph nodes, and a new `shadernode` "require" declaration for using rendering shader nodes within nodegraphs. A few other "require" attributes have been renamed to match new terminology.
- The syntax for declaring nodegraph outputs has been simplified, and the concept of caching intermediate results within a nodegraph has been removed.
- Input connections for nodegraph nodes now use `<input>` elements rather than parameters of type "opgraphnode". This was done to unify the syntax for pattern and shader nodes, and to allow node inputs to be assigned a uniform input value rather than requiring a connection to a `constant` node.
- Input connections can now be "swizzled on input", allowing channels to be extracted or rearranged at input, e.g. one can now connect the green channel of a `color3` output to a float input. Because of this, the standalone `convert` node has been removed.
- The `image` operator now takes an optional `vector2` "texcoord" input rather than fixed `uvscale/rotate/offset` parameters. The `triplanarprojection` operator already had a `vector3` "position" input, but the `scale/rotation/offset` parameters have been removed. In both cases, manipulation of uv or xyz coordinates can now be done with a user-defined chain of `vector2/vector3` operators.
- The various `rampX` and `splitX` operators also now have optional "texcoord" inputs.
- The "scale" parameter for the `noise2d` and `cellnoise2d` operators has been removed, again in favor of driving a "texcoord" input with a user-defined chain of `vector2` operators.
- Added a multi-octave `fractal3d` noise procedural node.
- The `add`, `subtract`, `multiply`, `divide`, `modulo` and `exponent` math operators now take "in1" and "in2" inputs, rather than an "in" input and an "amount" parameter. This allows them to be more general purpose, and subsume the roles of the `plus`, `difference`, `minus` and `mult` blending operators (which have been removed). And the existing `min`, `max`, `dotproduct` and `crossproduct` blending operators have been recategorized as Math nodes.
- Added `vector2/vector3` `scale` and `vector2` `rotate2d` nodes, which scale or rotate texcoord or position vectors in pattern space.
- Added a `smoothstep` adjustment node.
- The "amount" and "premult" parameters for the `mix` node have been removed.

- Added Conditional operators `compare` (select one of two incoming streams based on comparing a separate float stream to a fixed cutoff value) and `switch` (select one of five incoming streams based on a float selector parameter).
- Float, `colorN` and `vectorN` parameters may now optionally specify `uimin` and/or `uimax` values to specify an allowed range for values, and string parameters may now optionally specify a `uienum` value to list allowable string values.
- UI-related attributes `xpos`, `ypos` and `uicolor` are now allowed on `<shaderref>`, `<material>`, `<light>` and `<look>` elements as well as any pattern or shader nodes.
- Attributes describing a "comma-separated list" of things that were previously just type string are now appropriate array types.
- ShaderRef's now refer to the "node" attribute value of the shader `nodedef`, rather than the `nodedef`'s "name", matching what would be used if the shader was invoked as a node within a `nodegraph`.
- The separator between hierarchical names for `publicnames` and `properties` has been changed from ":" to "/".
- There are now two reserved `GeomAttr` names, "udimset" and "uvtileset", used to explicitly list the `udim` or `uvtile` values allowed for a geometry.
- `LightIllum` and `LightShadow` assignments have been merged into a single `lightassign` element; the `global` attribute for `lightIllum` has been removed.
- It is now possible to define and assign a single property within a `look` without needing to create a named `propertyset` to hold it.
- General reorganization of the specification document to improve the ordering in which various concepts are introduced and used.
- Updates to `Definitions`, `Examples`, and `Diagrams` to match current concepts and syntax.
- Added a `Table of Contents`
- Various other minor clarifications and corrections.