Differences between MaterialX Specification v1.32 and v1.33:

- Added a `vdirection` root-level attribute, indicating whether V coordinate values in uv space should be interpreted as increasing in the "up" or "down" direction, to establish the coordinate direction for opgraph sources such as `<ramptb>` that use the concepts of "top" or "bottom" to define their output. The various `noise` sources also respect the setting of vdirection.
- The `texturerole` <materialx> attribute has been renamed `texturecolorspace`, and now refers to an actual color space name rather than an OCIO role.
- What were previously called "regex expressions" are now instead "geometry name wildcards", and we have clarified that these follow csh glob syntax rather than traditional regex syntax. The functionality of `regex` attributes in various elements has been folded into `geom` attributes, which now accept comma-separated lists of geometry names using wildcards.
- The `frameendaction` values "hold" and "bounce" for the various Texture nodes are now called "clamp" and "mirror", to match the values used for `u/vaddressmode` parameters.
- We have clarified that ramp, split and noise sources use only the fractional component of uv values for their coordinates, discarding any integer portion, which is necessary to maintain compatibility between systems that do or don't use tiled textures.
- The `<time>` node now accepts an optional float "fps" parameter to explicitly state the frames-per-second conversion factor.
- The `<reorder>` operator is now called `<swizzle>`, and can output a different number of channels from its input. `<swizzle>` can thus be used to extract a single channel from a vectorN or colorN stream. In keeping consistency with actual functionality of most DCC packages, we have removed the ability of this operator to set a channel value to 0 or 1; instead, a combination of `<swizzle>` and `<pack>` must be used, which is more portable.
- There is a new operator `<pack>`, which packs the channels from two separate streams into a single stream, e.g. to combine two float streams into a vector2 stream, or a color3 RGB stream and a float stream to form an RGBA color4 stream.
- The `includechildren` attribute for `<collectionadd>` and `<collectionremove>` was deemed not very transportable and has been removed.
- We have clarified that `<geomattr>` elements define sets of named attributes with constant values rather than arbitrary and possibly varying externally-defined geometry-specific data, and the `extern` attribute for `<geomattr>` has been removed. External varying data on geometry should instead be accessed using Geometric nodes within opgraphs.
- The `file` (formerly `sourcefile`) attribute for `<implementation>`s of shaders and custom opgraph nodes has been changed to refer to the file containing the source code for the shader's or function's entry point, rather than just a file or directory name. There is also now an optional companion `function` attribute, specifying the name of the function within that file which defines the shader/function entry point.
- `valueformat` strings for <typedef> elements and specified values for custom types now use a comma-separated list of types or values rather than space-separated.
- The `nodecolor` attribute has been renamed `uicolor`, and can now be defined on any opgraph node as well as <shader> and <material> elements, not just on <backdrop> nodes. We have also clarified that `uicolor` values are vector3 type and not color-managed, and are expected to be specified in a normalized-value display-referred color space, typically sRGB.
- Various other minor clarifications and corrections.