MaterialX Shader Generation

Bernard Kwok and Ashwin Bhat

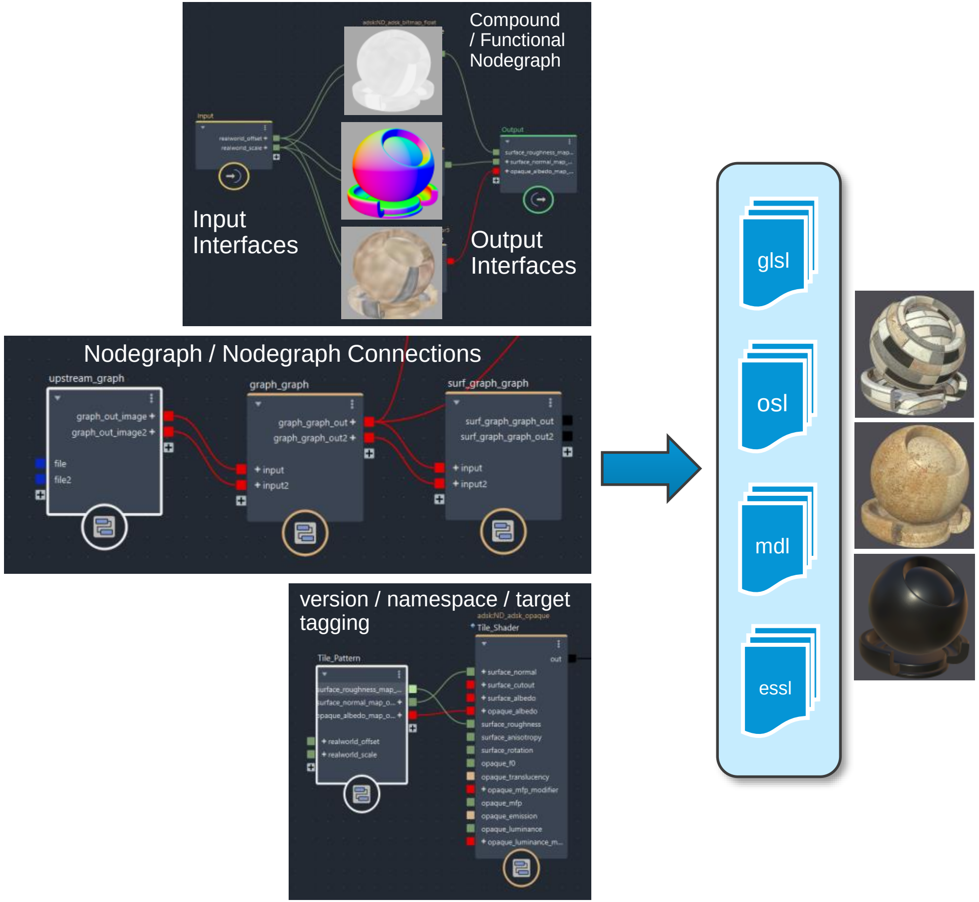
bernard.kwok@autodesk.com
ashwin.bhat@autodesk.com

AUTODESK

1.38.x Updates

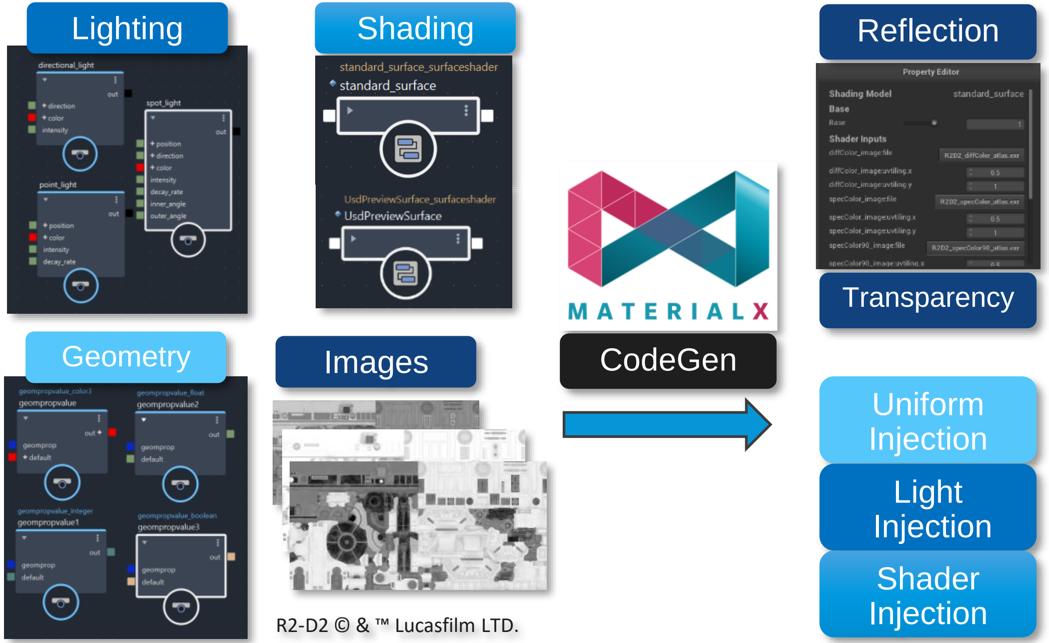
Shading Graph Configurability

- Consistent and robust compound and functional graph support
- Improved traversal logic for node and graph interface connections
- New: Nodegraph-to-nodegraph connections, Translation graph support.
- Improved namespace, version, target support
- Improved input value resolution to handle: inheritance, interface connections, geometry and filenames (incl. tokens)
- Improved ability to code generate for individual nodes, and sub-graphs.



Code Generation Configurability

- Improved light injection and geometry stream bindings
- Improved uniform injection including layout support
- Improved reflection for resource binding and transparency heuristics
- Improved image format and texturing support



Property Editor

Shading Model standard_surface

Base

Base 1

Shader Inputs

diffColor_image:file R2D2_diffColor_atlas.exr

diffColor_image:uvtiling.x 0.5

diffColor_image:uvtiling.y 1

specColor_image:file R2D2_specColor_atlas.exr

specColor_image:uvtiling.x 0.5

specColor_image:uvtiling.y 1

specColor90_image:file R2D2_specColor90_atlas.exr

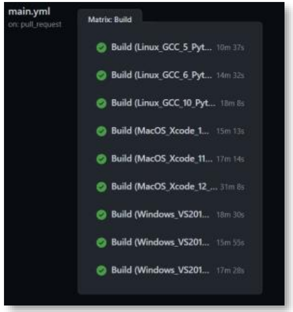
specColor90_image:uvtiling.x 0.5

R2-D2 © & ™ Lucasfilm LTD.
Used with permission

R2-D2 © & ™
Lucasfilm LTD. Used
with permission

Infrastructure

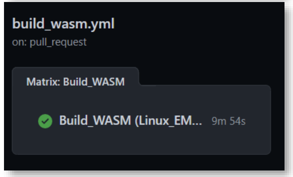
Github Actions Migration



main.yml
on: pull_request

Matrix: Build

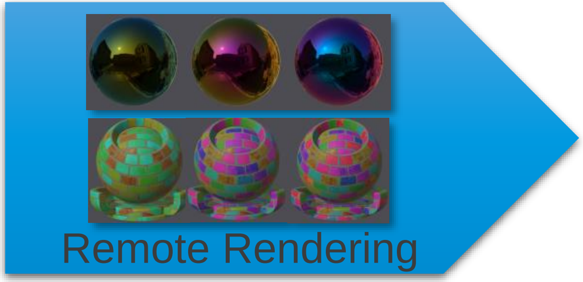
- ✓ Build (Linux_GCC_5_Pyt... 10m 37s
- ✓ Build (Linux_GCC_6_Pyt... 14m 32s
- ✓ Build (Linux_GCC_10_Pyt... 18m 8s
- ✓ Build (MacOS_Xcode_1... 15m 13s
- ✓ Build (MacOS_Xcode_11... 17m 14s
- ✓ Build (MacOS_Xcode_12_... 31m 8s
- ✓ Build (Windows_VS201... 18m 30s
- ✓ Build (Windows_VS201... 15m 55s
- ✓ Build (Windows_VS201... 17m 28s

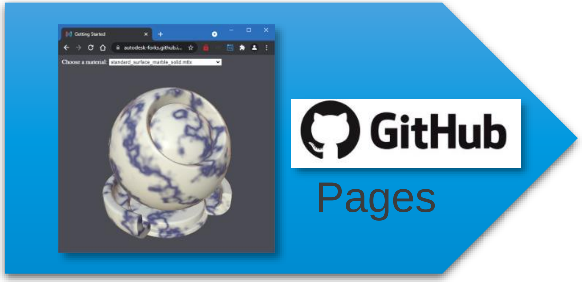


build_wasm.yml
on: pull_request

Matrix: Build_WASM

- ✓ Build_WASM (Linux_EM... 9m 54s





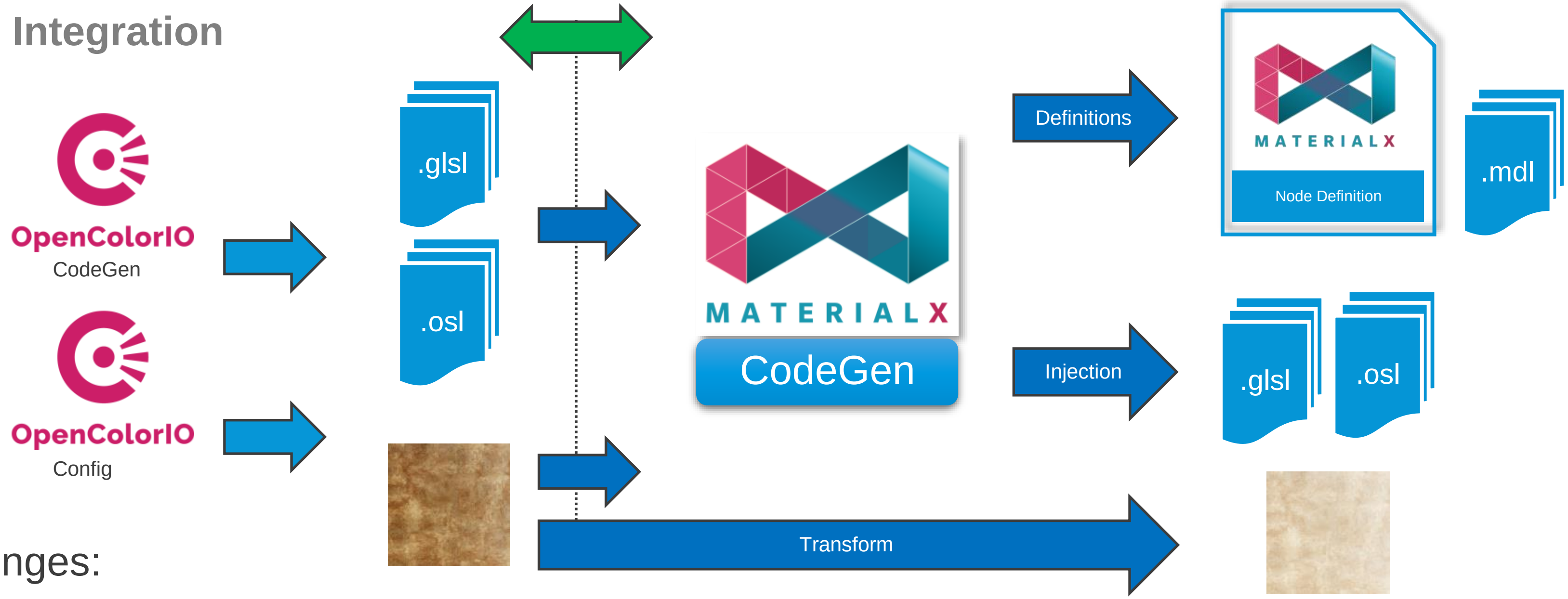
Choose a material: standard_surface_marble_solid.mtlx

- Rendering validation (Intel OpenSWR)
- MaterialX Web: WASM generation and Github pages hosting.
- Goal: support fully automated code generation / rendering validation

Initiatives In Progress

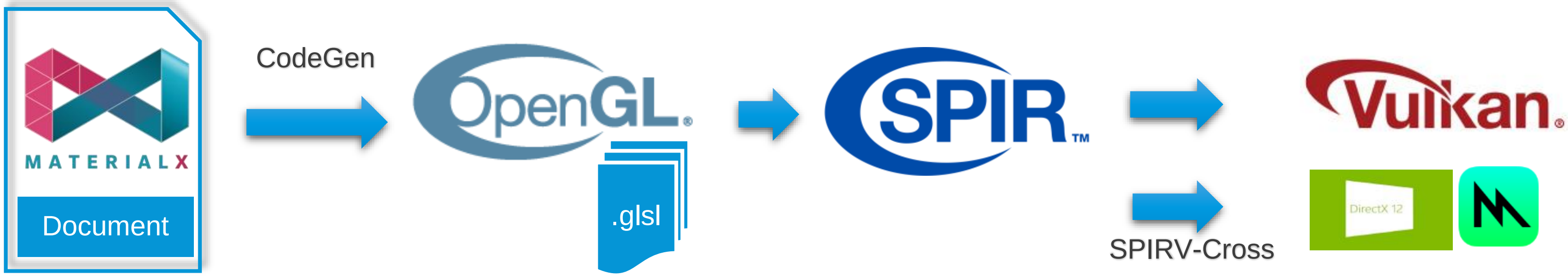
Color Management

OCIO v2 Integration



- Challenges:
  - ACEScg color space naming consistency
  - Code generation targets: GLSL, OSL, MDL, ESSL
  - Deployment flexibility: pre-compute, function generation, full shader, reference definition
  - OCIO enhancements for uniform injection / format control

SPIRV Code Generation Overview



- Use `mx::GlslResourceBindingContext`
- Generate SPIRV compatible GLSL.

  E.g., use `#extension GL_ARB_shading_language_420pack`
- Demonstrated feasibility of Cross Compilation during SIGGRAPH 2020 Autodesk Vision Series demo.
- Explore and improve **KhronosGroup/SPIRV-Tools** to provide per target Shader Reflection.

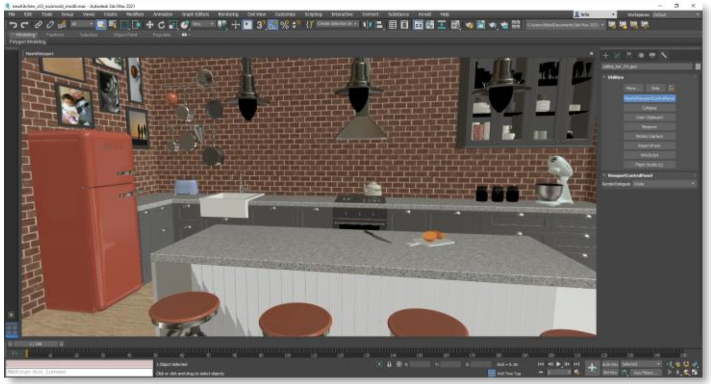
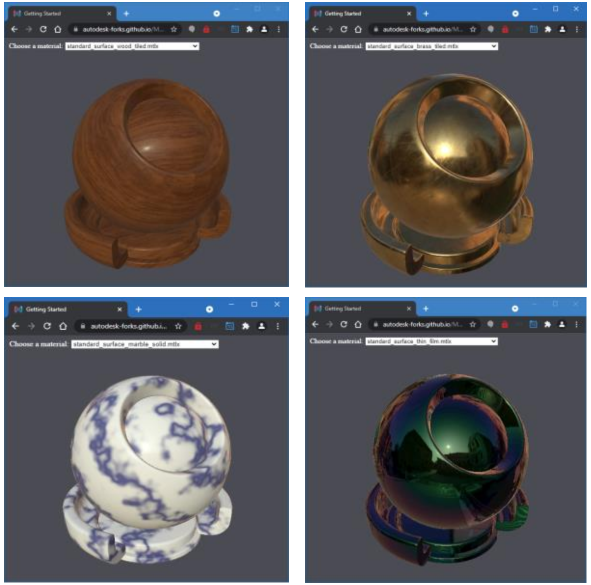


Image credit:
3ds Max: Open Standards & Next Generation Viewport Framework (SIGGRAPH 2020 Autodesk Vision Series)

MaterialX for Web

MaterialX JavaScript library

- In progress project for upcoming release.
- Components:
  - JavaScript Bindings + Web Assembly.
  - CodeGen for OpenGL ES 3.0.
  - Web Viewer Sample Application https://autodesk-forks.github.io/MaterialX/
- Fully compatible with current GLSL implementation.
- Supported Browsers Chrome, Firefox, Edge, Safari*
- Supports material shading graphs and pattern graphs (textures, procedurals)
- Framework agnostic.


Getting Started
autodesk-forks.github.io/M...
Getting Started
autodesk-forks.github.io/M...
Getting Started
autodesk-forks.github.i...
Getting Started
autodesk-forks.github.io/M...

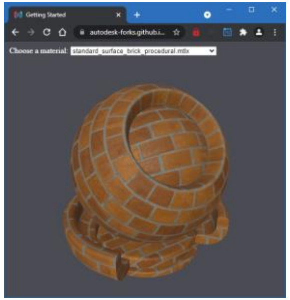
Choose a material: standard_surface_wood_tiled.mtlx

Choose a material: standard_surface_brass_tiled.mtlx

Choose a material: standard_surface_marble_solid.mtlx

Choose a material: standard_surface_thin_film.mtlx

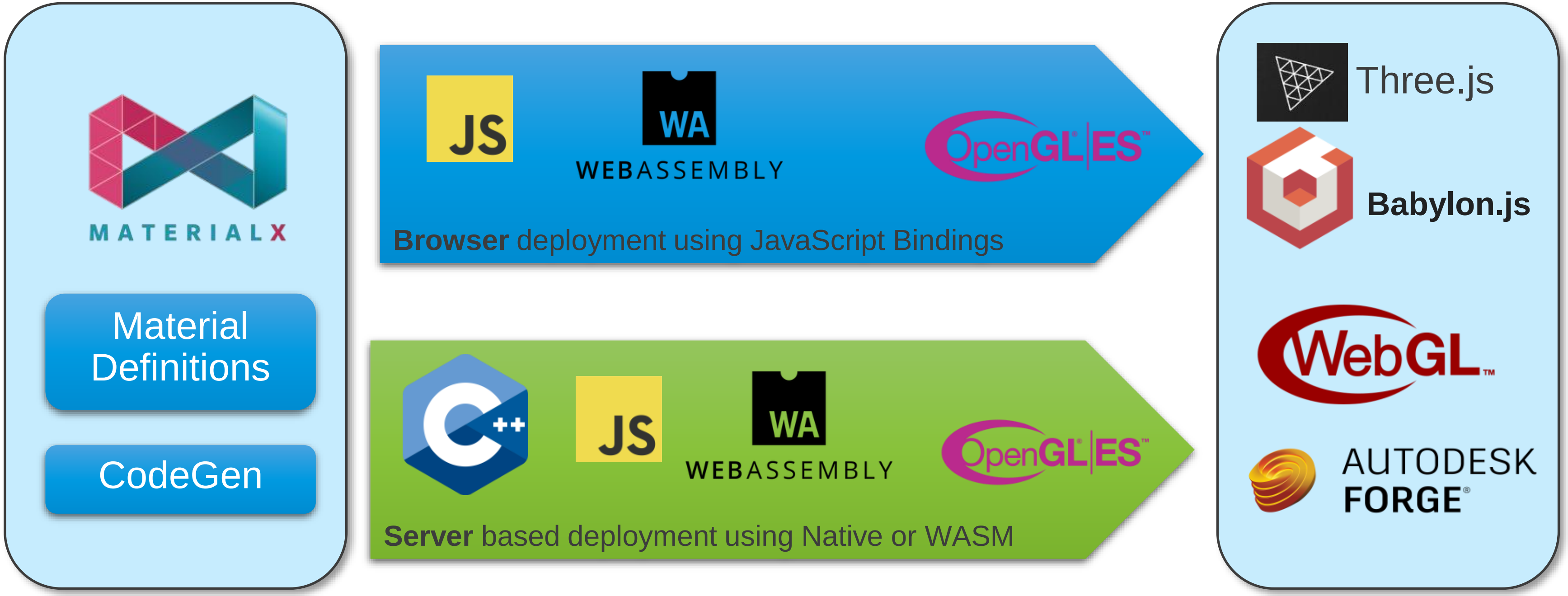
Above: Examples from MaterialX distribution using Standard Surface in Google Chrome.
Below: Example procedural material from **Adobe Substance** as MaterialX in Google Chrome.



Choose a material: standard_surface_brick_procedural.mtlx

MaterialX for Web

Deployment options (framework agnostic)



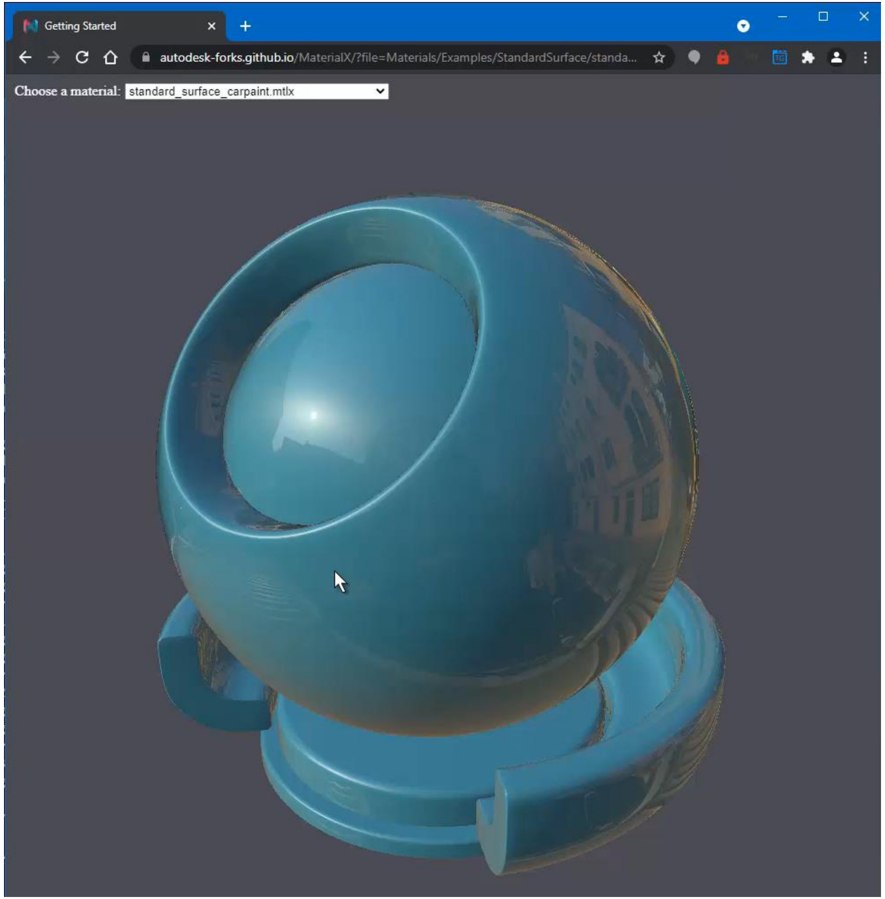
Material Definitions

CodeGen

Browser deployment using JavaScript Bindings

Server based deployment using Native or WASM

MaterialX WebGL (in Google Chrome)

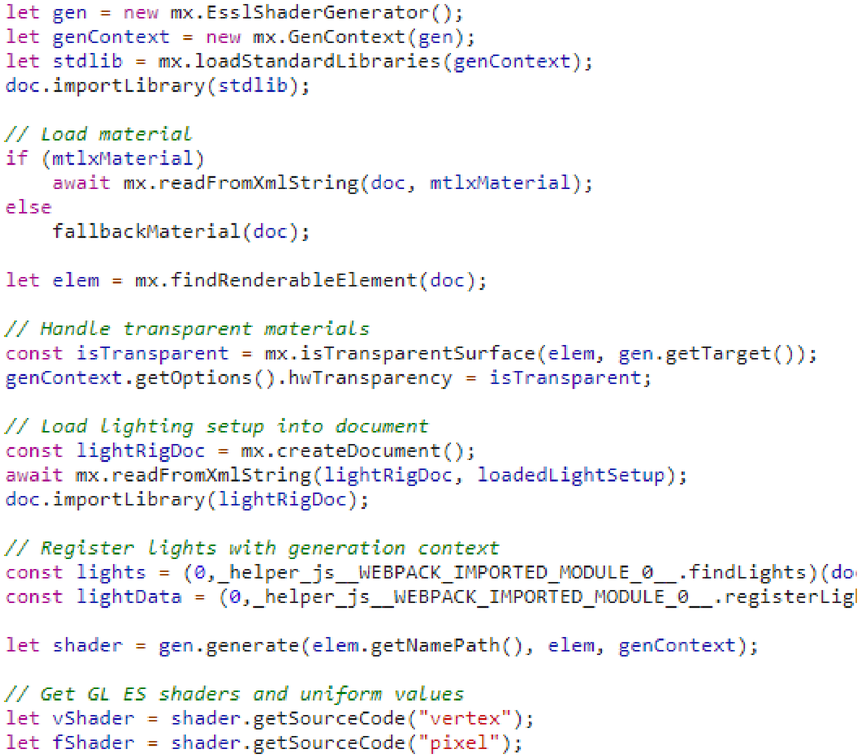


Getting Started

autodesk-forks.github.io/MaterialX/?file=Materials/Examples/StandardSurface/standa...

Choose a material: standard_surface_carpaint.mtlx

MaterialX API in JavaScript, using GL ES Shader Generator

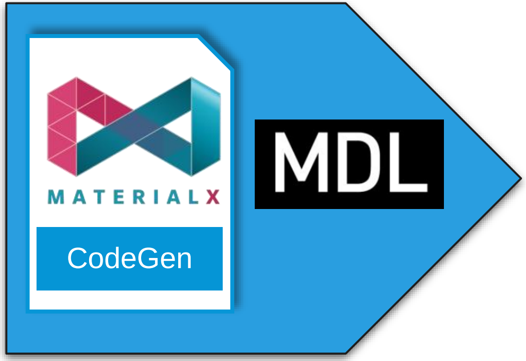


```
let gen = new mx.EsslShaderGenerator();
let genContext = new mx.GenContext(gen);
let stdlib = mx.loadStandardLibraries(genContext);
doc.importLibrary(stdlib);

// Load material
if (mtlxMaterial)
    await mx.readFromXmlString(doc, mtlxMaterial);
else
    fallbackMaterial(doc);

let elem = mx.findRenderableElement(doc);

// Handle transparent materials
const isTransparent = mx.isTransparentSurface(elem, gen.getTarget());
genContext.getOptions().hwTransparency = isTransparent;

// Load lighting setup into document
const lightRigDoc = mx.createDocument();
await mx.readFromXmlString(lightRigDoc, loadedLightSetup);
doc.importLibrary(lightRigDoc);

// Register lights with generation context
const lights = (0,_helper_js__WEBPACK_IMPORTED_MODULE_0__.findLights)(do
const lightData = (0,_helper_js__WEBPACK_IMPORTED_MODULE_0__.registerLig

let shader = gen.generate(elem.getNamePath(), elem, genContext);

// Get GL ES shaders and uniform values
let vShader = shader.getSourceCode("vertex");
let fShader = shader.getSourceCode("pixel");
```

Future Work

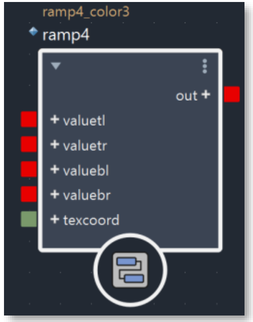
NVIDIA MDL Updates

- Forthcoming MDL 1.7 release will have better alignment with MaterialX (e.g., sheen layer, unbound mixer nodes)
- End of year target to have MaterialX import for Omniverse
  - Background improvements in MDL generation and consumption (E.g., resource path handling)
- See SIGGRAPH 2021 updates from NVIDIA.

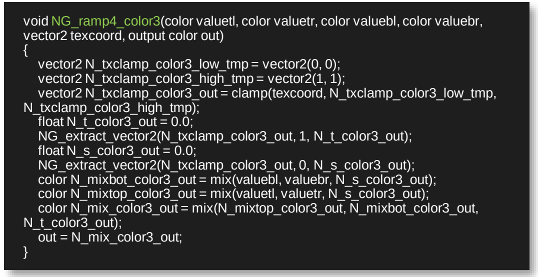


Generation Configurability

- Fragment / Function Export vs new generator derivation
- Uniform format control / reflection

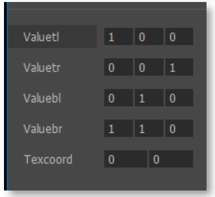


Fragment



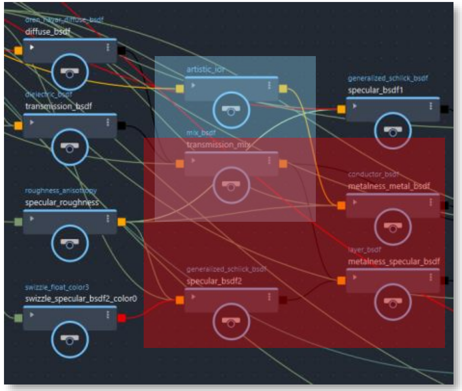
```
void NG_ramp4_color3(color valuetl, color valuetr, color valuebl, color valuebr,
vector2 texcoord, output color out)
{
    vector2 N_txclamp_color3_low_tmp = vector2(0, 0);
    vector2 N_txclamp_color3_high_tmp = vector2(1, 1);
    vector2 N_txclamp_color3_out = clamp(texcoord, N_txclamp_color3_low_tmp,
N_txclamp_color3_high_tmp);
    float N_t_color3_out = 0.0;
    NG_extract_vector2(N_txclamp_color3_out, 1, N_t_color3_out);
    float N_s_color3_out = 0.0;
    NG_extract_vector2(N_txclamp_color3_out, 0, N_s_color3_out);
    color N_mixbot_color3_out = mix(valuebl, valuebr, N_s_color3_out);
    color N_mixtop_color3_out = mix(valuetl, valuetr, N_s_color3_out);
    color N_mix_color3_out = mix(N_mixtop_color3_out, N_mixbot_color3_out,
N_t_color3_out);
    out = N_mix_color3_out;
}
```

Uniform Format / Reflection

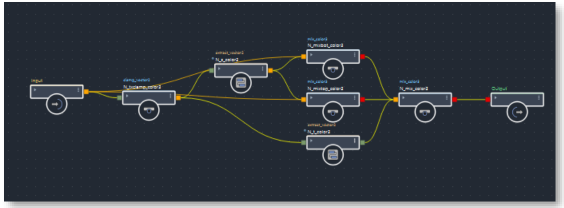


Valuetl	1	0	0
Valuetr	0	0	1
Valuebl	0	1	0
Valuebr	1	1	0
Texcoord	0	0	

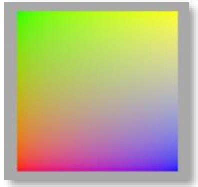
- Sub-graph / node export as graphs or images



Export

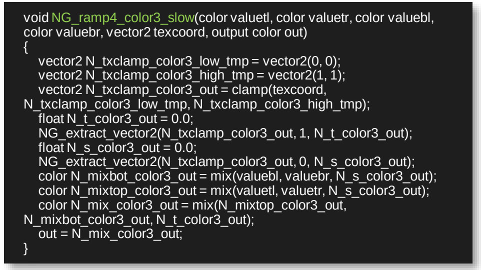


Bake



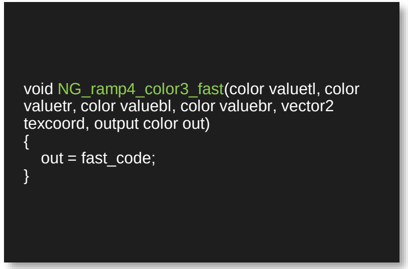
Generation Optimization

- Performance optimizations for language / platform / workflow



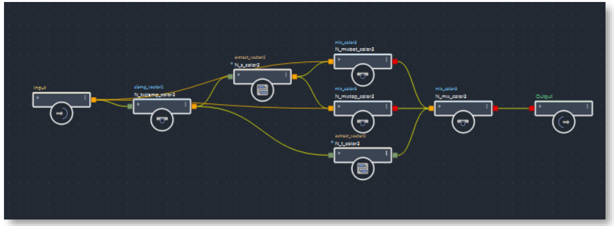
```
void NG_ramp4_color3_slow(color valuetl, color valuetr, color valuebl,
color valuebr, vector2 texcoord, output color out)
{
    vector2 N_txclamp_color3_low_tmp = vector2(0, 0);
    vector2 N_txclamp_color3_high_tmp = vector2(1, 1);
    vector2 N_txclamp_color3_out = clamp(texcoord,
N_txclamp_color3_low_tmp, N_txclamp_color3_high_tmp);
    float N_t_color3_out = 0.0;
    NG_extract_vector2(N_txclamp_color3_out, 1, N_t_color3_out);
    float N_s_color3_out = 0.0;
    NG_extract_vector2(N_txclamp_color3_out, 0, N_s_color3_out);
    color N_mixbot_color3_out = mix(valuebl, valuebr, N_s_color3_out);
    color N_mixtop_color3_out = mix(valuetl, valuetr, N_s_color3_out);
    color N_mix_color3_out = mix(N_mixtop_color3_out,
N_mixbot_color3_out, N_t_color3_out);
    out = N_mix_color3_out;
}
```

Optimize



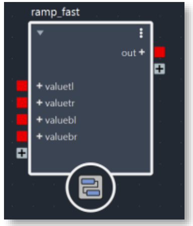
```
void NG_ramp4_color3_fast(color valuetl, color
valuetr, color valuebl, color valuebr, vector2
texcoord, output color out)
{
    out = fast_code;
}
```

- Optimize at code, node, and/or definition level

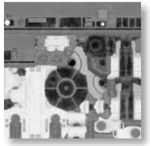


Optimize

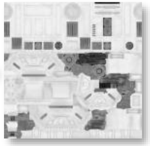
Reparametrize



- Repackaging of resources: baking, packing, access atlas / arrays (e.g. UDIMs), alternate formats (e.g. IBL cubemaps)

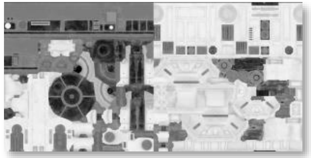


+



R2-D2 © & ™ Lucasfilm LTD. Used with permission

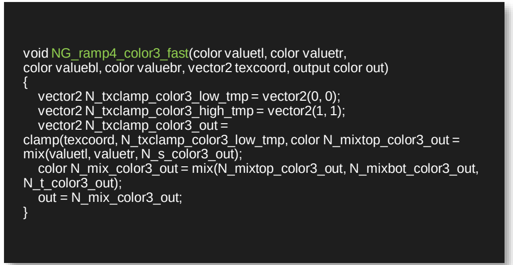
Format / Pack



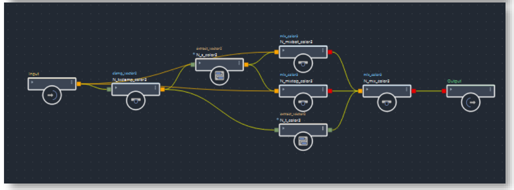
Generation Deployment

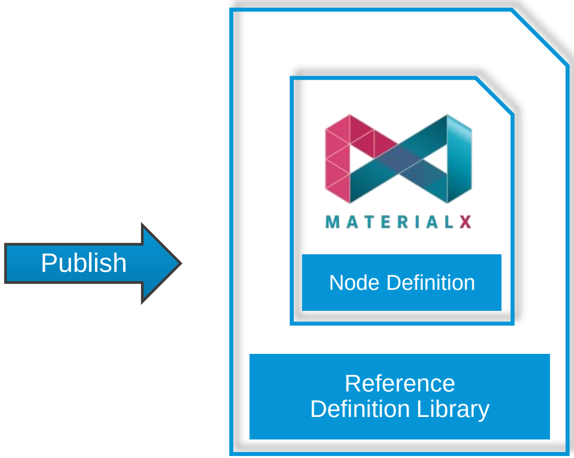
- Publishing for reuse, produce reference libraries (e.g. OSL reference library)

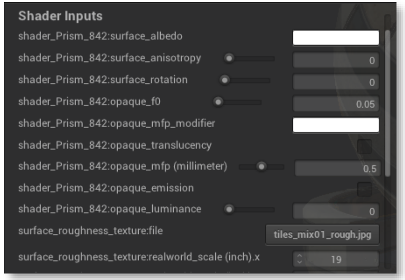
- Realtime Updates:
    - Observability
    - Change management
    - Diagnostics / Feedback



```
void NG_ramp4_color3_fast(color valuetl, color valuetr,
color valuebl, color valuebr, vector2 texcoord, output color out)
{
    vector2 N_txclamp_color3_low_tmp = vector2(0, 0);
    vector2 N_txclamp_color3_high_tmp = vector2(1, 1);
    vector2 N_txclamp_color3_out =
clamp(texcoord, N_txclamp_color3_low_tmp, color N_mixtop_color3_out =
mix(valuetl, valuetr, N_s_color3_out);
    color N_mix_color3_out = mix(N_mixtop_color3_out, N_mixbot_color3_out,
N_t_color3_out);
    out = N_mix_color3_out;
}
```

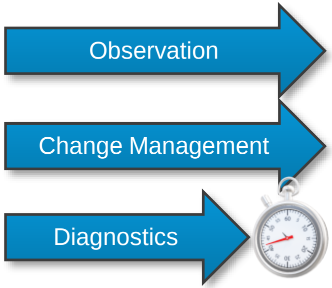


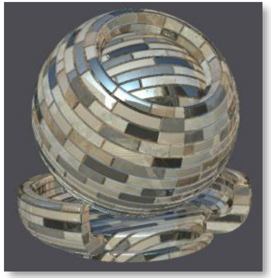




Shader Inputs

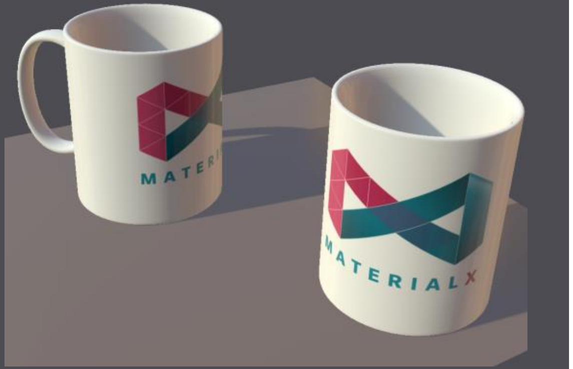
shader_Prism_842:surface_albedo	
shader_Prism_842:surface_anisotropy	0
shader_Prism_842:surface_rotation	0
shader_Prism_842:opaque_f0	0.05
shader_Prism_842:opaque_mfp_modifier	
shader_Prism_842:opaque_translucency	□
shader_Prism_842:opaque_mfp (millimeter)	0.5
shader_Prism_842:opaque_emission	□
shader_Prism_842:opaque_luminance	0
surface_roughness_texture:file	tiles_mix01_rough.jpg
surface_roughness_texture:realworld_scale (inch).x	19





Credits

- Adam Felt
- Aura Munoz
- Brent Scannell
- Cedrick Muenstermann
- David Larsson
- Doug Smythe
- Doug Walker
- Dusan Kovic
- Eric Bourque
- Fedor Nikolayev
- Gareth Morgan
- Guillaume Laforge
- Harv Saund
- Henrik Edstrom
- Jan Jordan
- Jerran Schmidt
- Jerry Gamache
- Jonathan Stone
- Kai Rohmer
- Kevin Zhang
- Krishna Kalvai
- Krishnan Chunangad Ramachandran
- Krystian Ligenza
- Lutz Kettner
- Mauricio Vives
- Nicolas Savva
- Niklas Harrysson
- Nikola Milosevic
- Patrick Hodoul
- Philippe Frericks
- Phenix Xu
- Rishabh Bisht
- Roberto Ziche
- Sankar Ganesh
- Sebastian Dunkel
- Toni Qin
- Wayne Catalfano
- Will Telford
- Zap Andersson



AUTODESK®

Make anything™