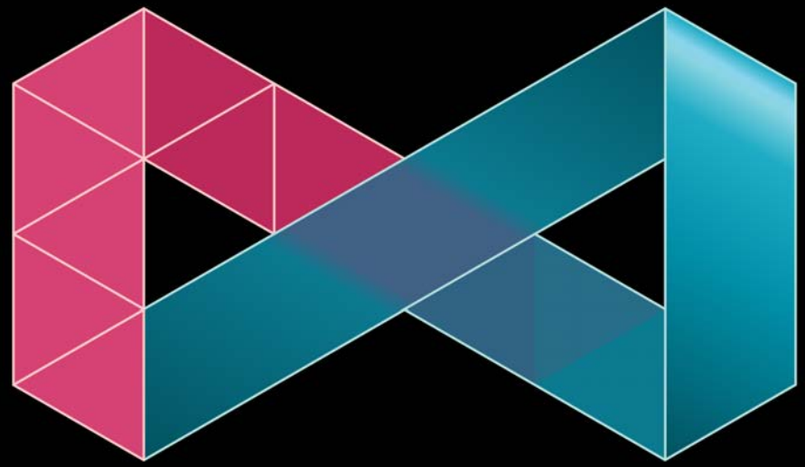



# **MaterialX**: An Open Standard for Network-Based CG Object Looks

# Presentations

| | |
|---|---|
| Doug Smythe ILM | New and Improved in MaterialX 1.37 |
| Jonathan Stone Lucasfilm ADG | MaterialX Technology Updates |
| Davide Pesare Adobe | Portable Shaders with Substance |
| Doug Smythe ILM | Material Interchange at ILM |
| Henrik Edström Autodesk | Autodesk Update on MaterialX in Architecture and Design |

Discussion and Q & A



# More Presentations

Autodesk Exhibitor Session:
Open Source Support at Autodesk - MaterialX

Wednesday July 31, 2019
3:15-4:00pm

L.A. Convention Center  Room 404A



# MaterialX Overview

- Schema and File Format used to describe "complete CG object looks":
  - Shading network topology
  - Complex materials with inheritance and multiple rendering targets
  - Texture, Material and Visibility assignments
  - Illumination and shadowing assignments for asset lights
  - Look variants, geometric primitive properties, and much more

- Specific defined behavior for "Standard Nodes"
  - Texture reading
  - Procedural pattern generation
  - Image processing operations



# Features of MaterialX

- Strong data typing
- Fully color managed
- Compatible with (but doesn't require) other open standards
  - e.g. OpenColorIO, Alembic, USD, OpenEXR, OSL
- "Live, Not Baked": Setups remain editable after re-import
- Extensible:
  - User-defined nodes and shaders
  - Application-specific node parameters and attributes
  - User-defined data types



# What's Happened in the Past Year

- Worked with Autodesk and Adobe to integrate prototype MaterialX support into their products
- Significant updates to the MaterialX Specification and library, now at v1.37
- Link to download at materialx.org : 

- Expanded and updated materialx.org site
- New sample Python code: mxvalidate.py, writenodegraphs.y, writelooks.py



# New and Improved in MaterialX 1.37

## ShaderGen !

# Other New Nodes and Node Changes

- New Procedural Nodes: worleynoise2d, worleynoise3d
- New Math Nodes: place2d, normalmap

- "Convert" operator now supports vector2<->vector3 and vector3<->vector4 by adding/removing extra "1.0" channel

- Transformpoint/normal/vector support transforming vector3 by matrix44

- Added vector*N* math (sqrt, exp, ln) and trig functions

- Many "parameter"s changed to connectable "input"s:
  - "amount" in <contrast>, <hsvadjust>, <rotate2d>
  - in and out low and high and gamma values in <remap> and <range>
  - "min" and "max" in <smoothstep>



# Look Groups

- Mechanism to group together a number of related Looks for an asset
- Can also specify one of the looks to be the "active" look

```
<lookgroup name="CarLooks" looks="normal,blue,red,wet"
           active="blue"/>
```



# Geometric Properties

- "GeomAttrs" were problematic when used in filenames —> "Tokens" added in 1.36
- Remaining functionality now described using GeomProps
  - Semantically equivalent to USD Primvars

- Declare using new <geompropdef> element:

```
<geompropdef name="MyProp" type="vector3"/>
<geompropdef name="UV1" type="vector2"
             geomprop="texcoord" index="1"/>
```

- Retrieve values using <geompropvalue>

```
<geompropvalue name="gp1" geomprop="MyProp" type="vector3"
               default="0,0,0"/>
```



# Node Output Definition

- Multi-output nodes already supported in MaterialX, but felt like an "afterthought"
- API access to actual node outputs and types differed for single vs multi output nodes

```
<nodedef name="ND_contrastF" node="contrast" type="float">
  ..
</nodedef>

<nodedef name="ND_dblclr" node="doublecolor" type="multioutput">
  ..
  <output name="c1" type="color3" default="1.0, 1.0, 1.0"/>
  <output name="c2" type="color3" default="1.0, 1.0, 1.0"/>
</nodedef>
```



# Node Output Definition (cont)

- Now definitions are unified:
  - Output names and types always declared with child <output> elements
  - Nodedef no longer itself declares a type
  - Multioutput nodes can now declare a default value or input passthrough per output

```
<nodedef name="ND_contrastF" node="contrast">
  ..
  <output name="out" type="float" defaultinput="in"/>
</nodedef>
<nodedef name="ND_dblclr" node="doublecolor">
  ..
  <output name="c1" type="color3" default="1.0, 1.0, 1.0"/>
  <output name="c2" type="color3" defaultinput="in1"/>
</nodedef>
```



# Other Changes

- "Swizzle on input" **`channel`** attribute for node elements reinstated
  - Most useful for on-the-fly type conversion and channel extraction

- New **`uivisible`** and **`uiadvanced`** attributes for parameters and inputs

- New **`mirror`** mode for **`uaddressmode`**/**`vaddressmode`** and **`frameendaction`**
- "**`black`**" mode now called **`constant`**, and is defined to return the default value for an <image> rather than zero

- Lots more: see the "Changes since v1.36" doc on materialx.org



# What's Implemented in 1.36.3

- New things that are **implemented in 1.36.3**:
  - Most PBR nodes
  - <place2d>, <normalmap>, new <convert> and <transformXX> variants
  - <geompropdef>
  - uivisible/uiadvanced, "mirror" and "constant" modes
  - parameter->input

- **Not yet implemented** but coming soon:
  - <worleynoise2d>, <worleynoise3d>
  - <geomprop>, <geompropvalue>
  - Look Groups
  - Nodedefs with type declaration moved to child <output> elements



# Future Directions

- More standard and PBR nodes
- User-contributed custom node libraries
- Some other things that Jonathan will mention
- Join the Discussion Forum through materialx.org and tell us what you'd like!

# Thanks!

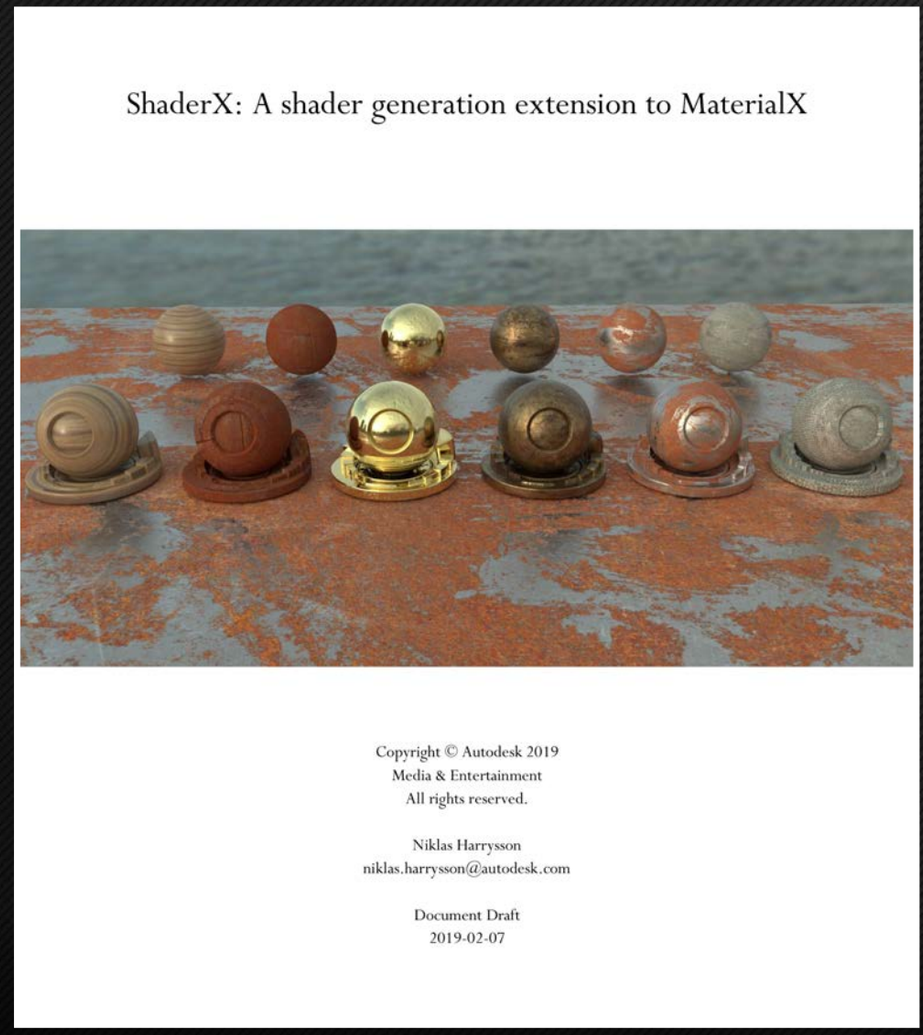

www.aswf.io

JONATHAN STONE - LUCASFILM ADG

# MATERIALX TECHNOLOGY UPDATES

# SHADERX COLLABORATION

- A partnership between Lucasfilm and Autodesk beginning in June of 2016
- Inspired by an internal Autodesk project, Abstract Material Graphs
- Autodesk begins developing two key extensions to MaterialX


ShaderX: A shader generation extension to MaterialX

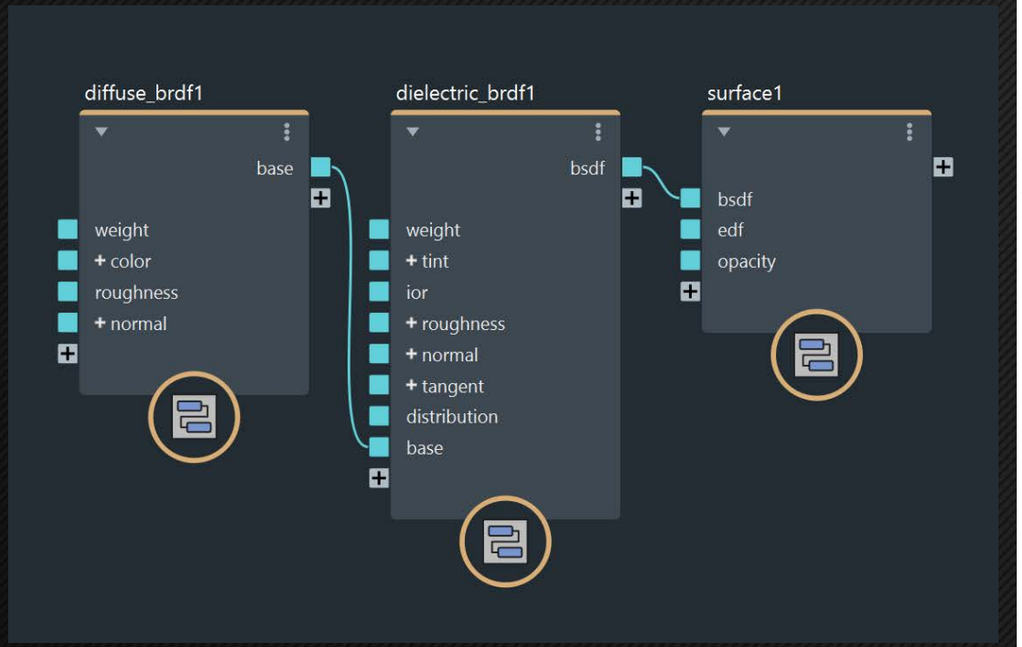


Niklas Harrysson
niklas.harrysson@autodesk.com

Document Draft
2019-02-07

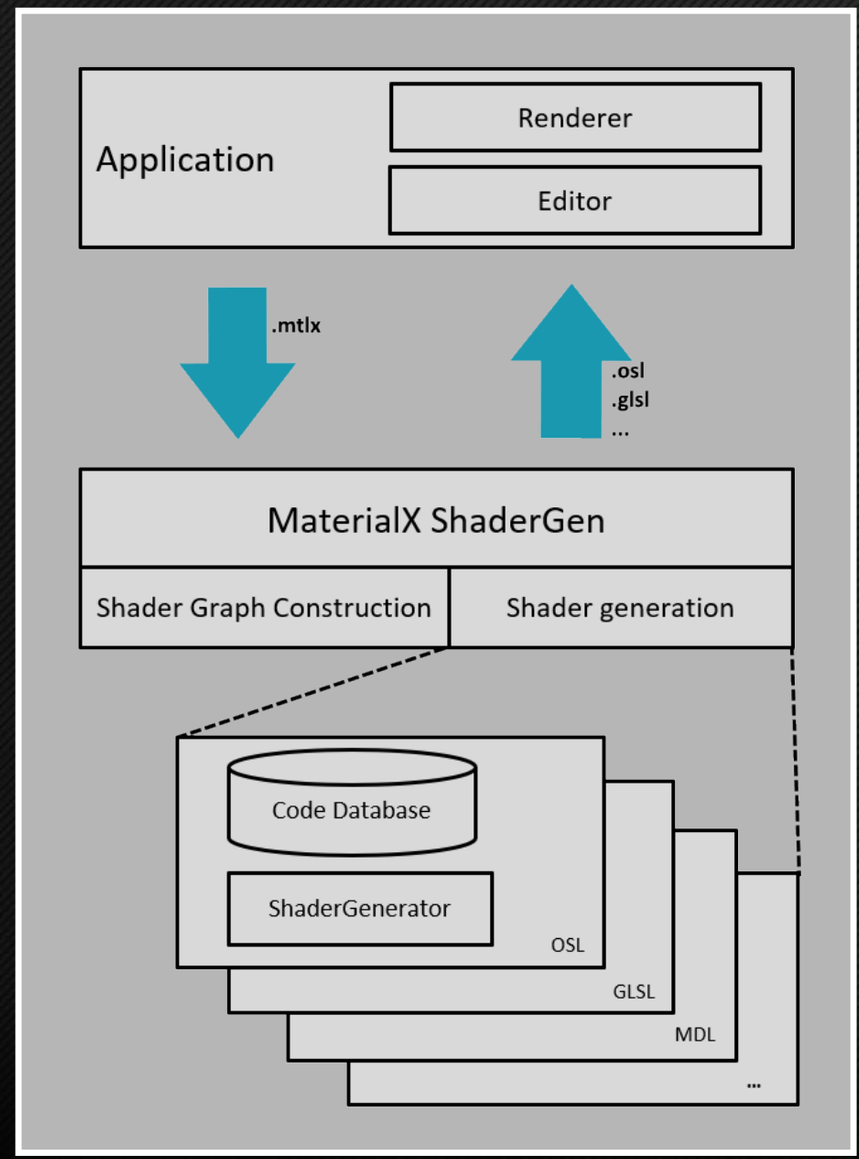
# PHYSICALLY-BASED SHADING NODES

- The first new feature is a standard set of physically-based shading nodes
- In addition to patterns, the underlying physically-based shaders can now be portably captured
- MaterialX ships with shading graphs for Standard Surface and USD Preview Surface



# SHADER CODE GENERATION

- The second new feature is a framework for shader code generation
- Automatic conversion of a MaterialX document to domain-specific shading code for rendering
- MaterialX ships with support for OSL and GLSL, with additional languages planned



# MATERIALX VIEWER

- Leverages MaterialX shader generation in combination with the open NanoGUI framework
- Provides a ground truth reference for renders of MaterialX content
- Provides a reference for integration of MaterialX shader code generation into other applications

MATERIALX UPCOMING WORK

# SHADER TRANSLATION GRAPHS

LUCASFILM UNIFIEDSRF

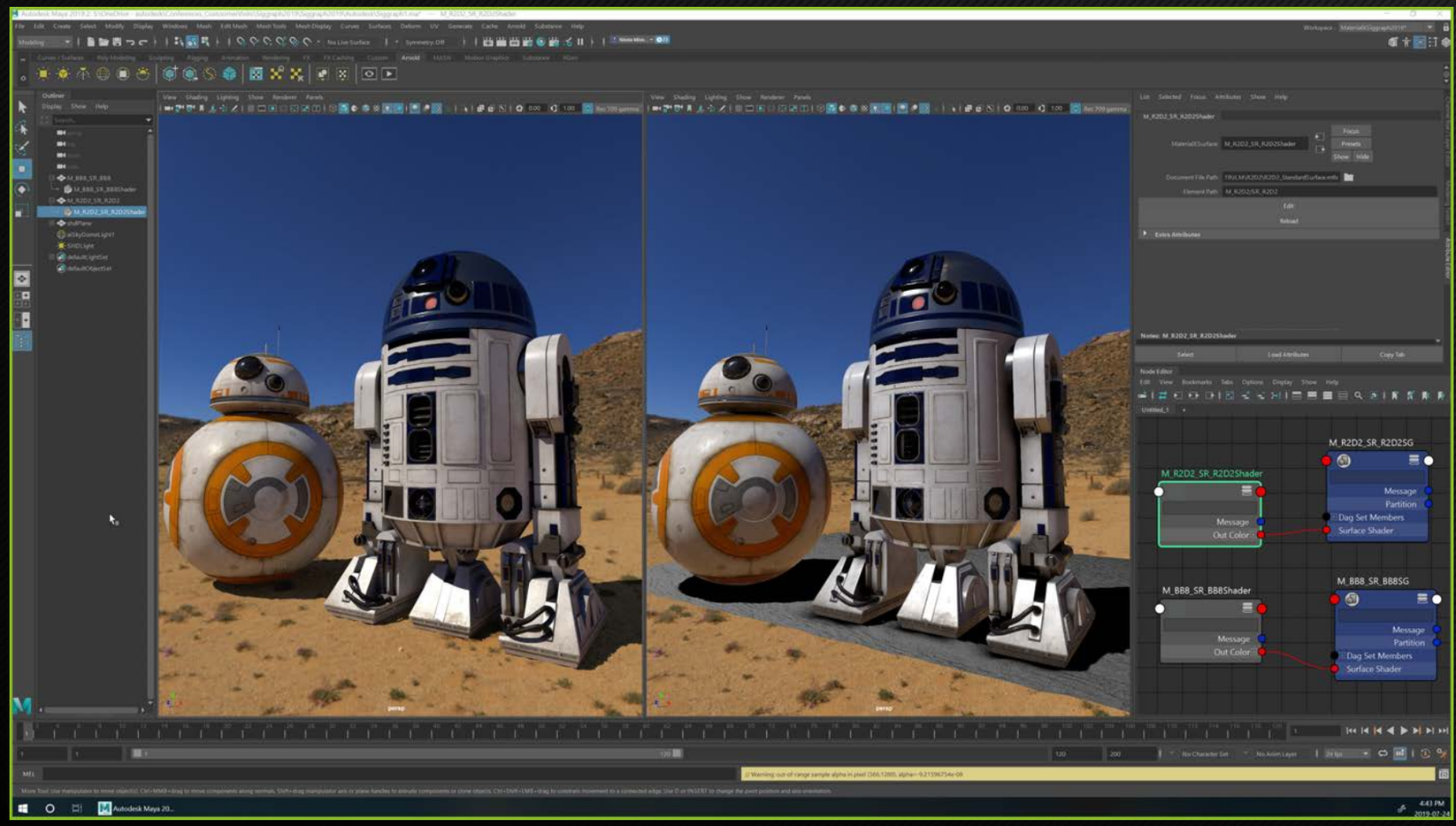
AUTODESK STANDARD SURFACE

# MATERIALX TEXTURE BAKING

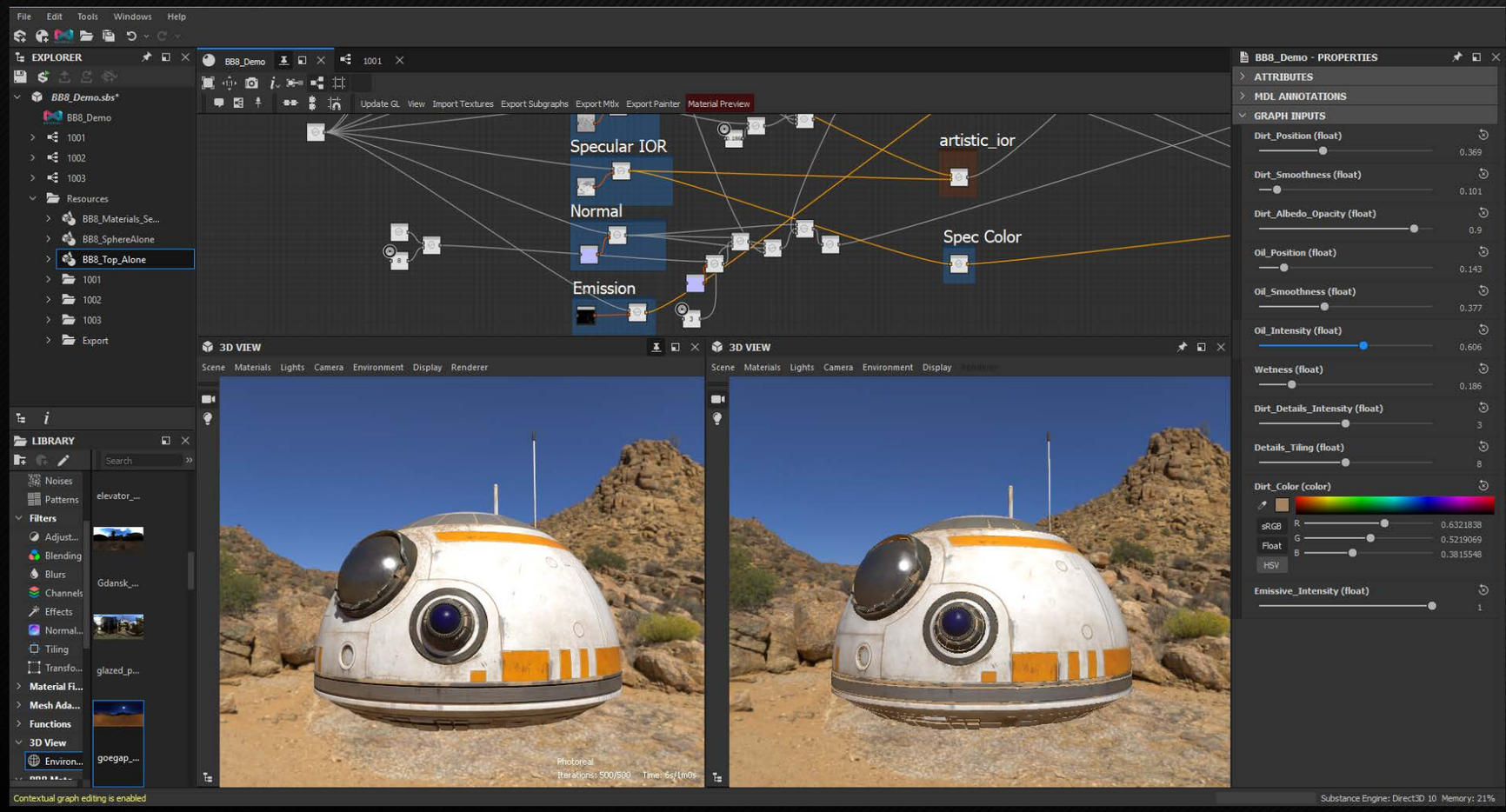
- Building upon the new shader generation framework to create an automated texture baker
- Would flatten an arbitrarily complex MaterialX graph down to a single texture (or UDIM set) for each shader input
- Could be combined with shader translation graphs to efficiently prepare materials for different rendering targets
- Reach out to our team if you're interested in contributing to development

MATERIALX UPCOMING WORK

# AUTODESK DEVELOPMENT



# ADOBE DEVELOPMENT



# THANKS!

- To Doug Smythe, Francois Chardavoine, Roger Cordes, and Rob Bredow
- With huge gratitude to the following contributors to MaterialX this year:
  - Niklas Harrysson, Bernard Kwok, Jonathan Feldstein, Ashwin Bhat, Eric Bourque, Davide Pesare, David Larsson, Sebastien Deguy, Eoghan Cunneen, Julien Cohen Bengio, Will Muto, Madeleine Yip, Sebastian Grassia, and Pol Jeremias-Vila


Birds of a Feather

# Portable Shaders with Substance

Davide Pesare - Head of Labs

# Making and sharing Materials

# Maximize Portability

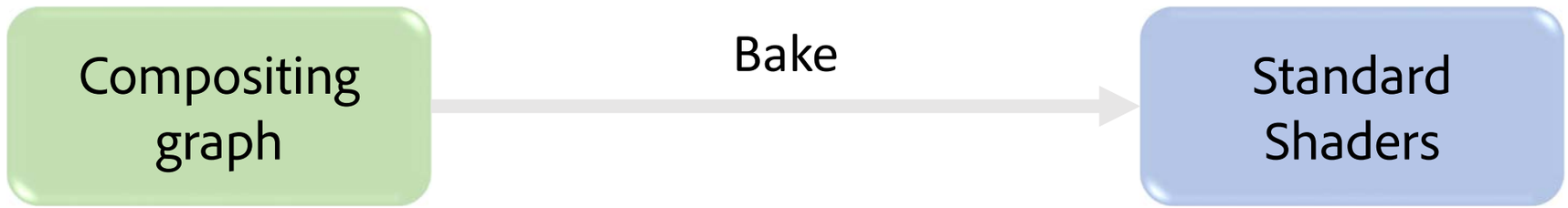
- Build procedural textures


Bake

SUBSTANCE by Adobe

# Maximize Portability

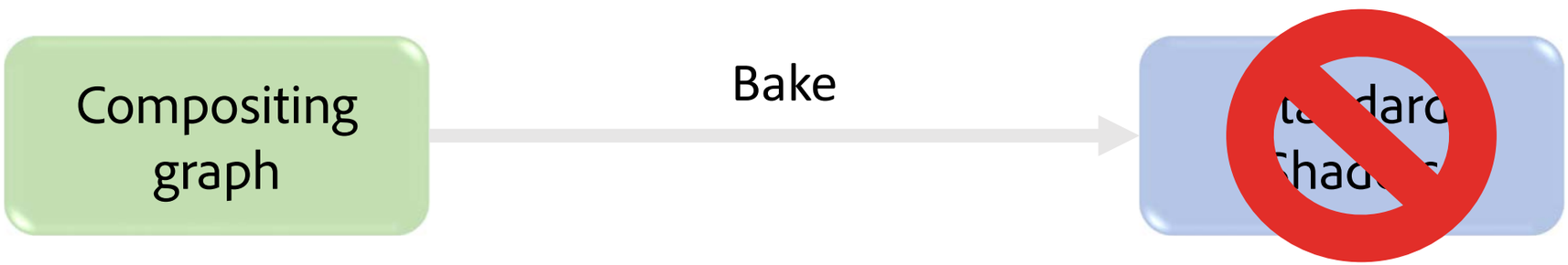
- Rely on standard uber-shaders
  - Our standard model based on Disney PBR model 2012




SUBSTANCE by Adobe

# Maximize Portability

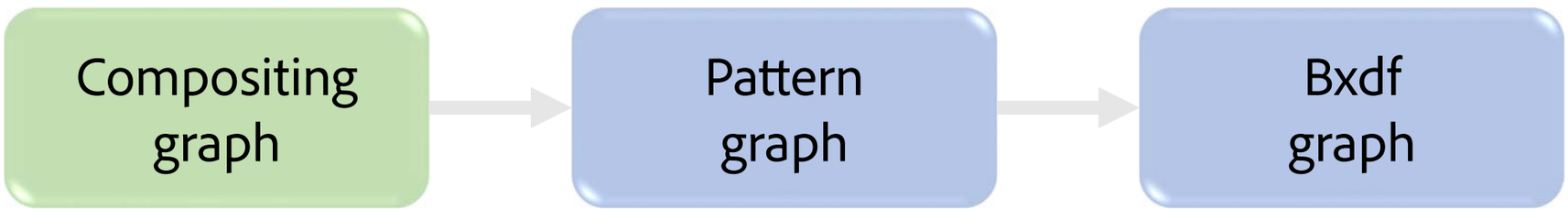
- Standard Uber Shaders don’t always suffice




SUBSTANCE by Adobe

# Extend to other use cases

- Need custom pattern and BXDF graphs
- Need editors for Shaders




SUBSTANCE by Adobe

# Editing Shaders

- We have a Shader Graph
- Successful for arch/design

# Editing Shaders

- Built on NVIDIA MDL
- Great for custom BXDF
- Enables rendering in Iray and other MDL-compliant renderers

# Editing Shaders

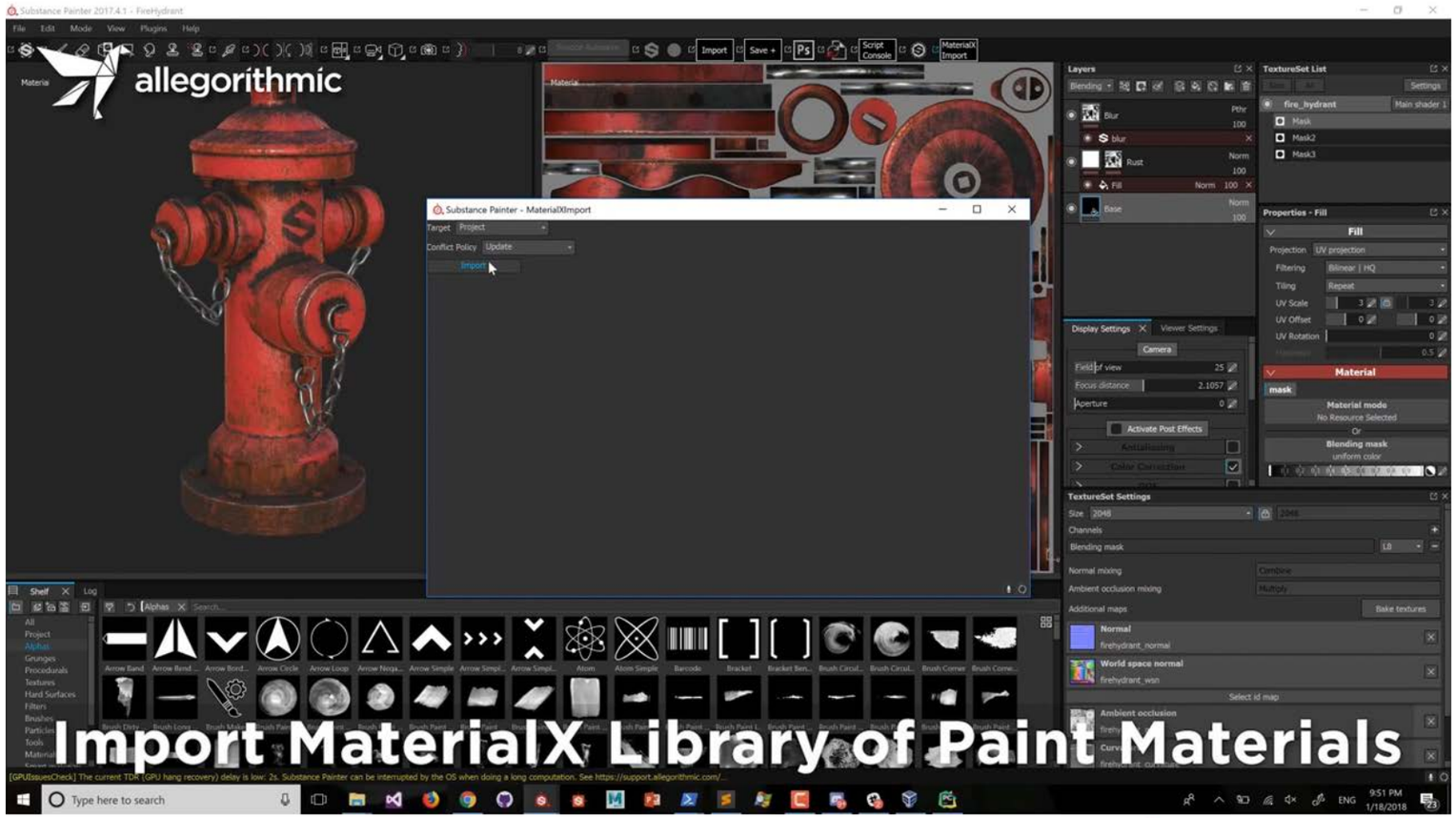
- Makes great shaders
- Not largely taken advantage of by the VFX industry


# We look at MaterialX

Not a new topic at Substance

# Last year at this BoF

- Import/sync MaterialX Libraries in Painter
- Export presets and bindings from Painter to MaterialX

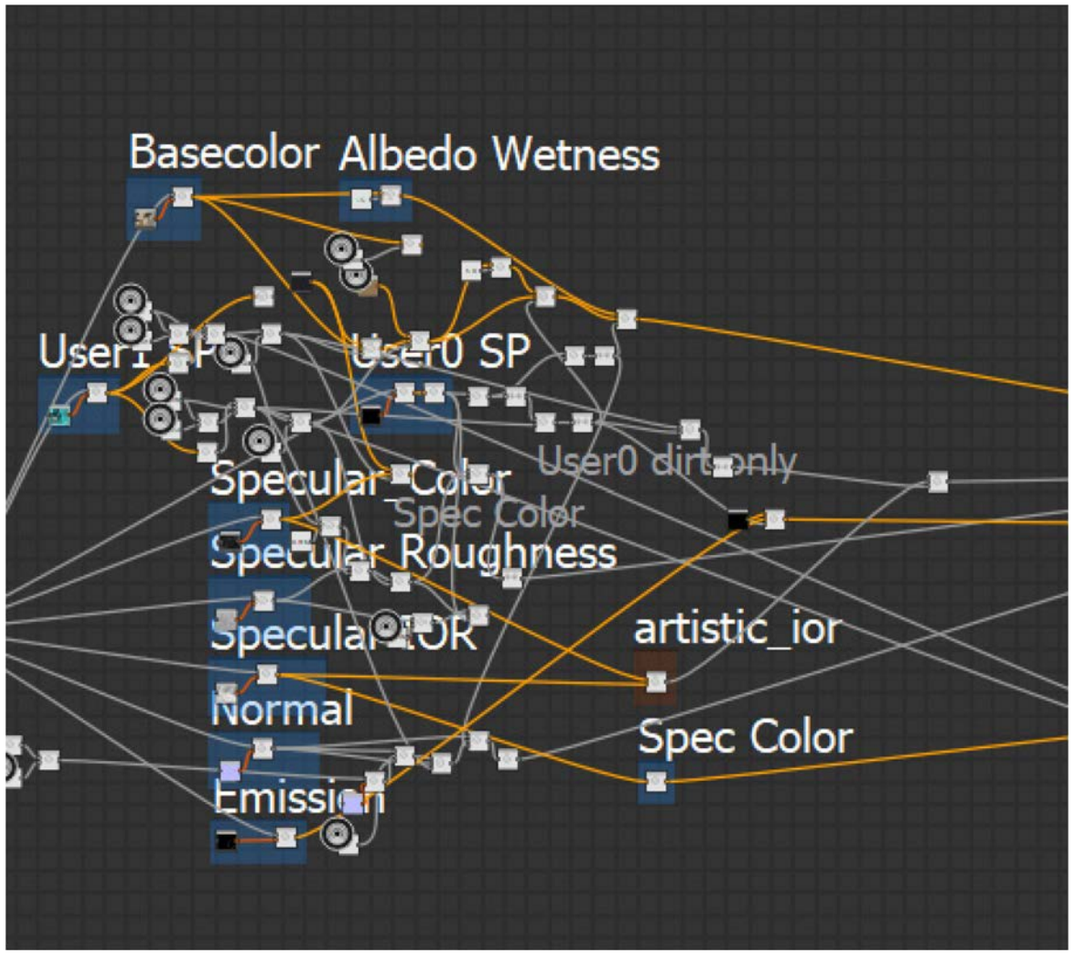



SUBSTANCE by Adobe


# Build on this idea

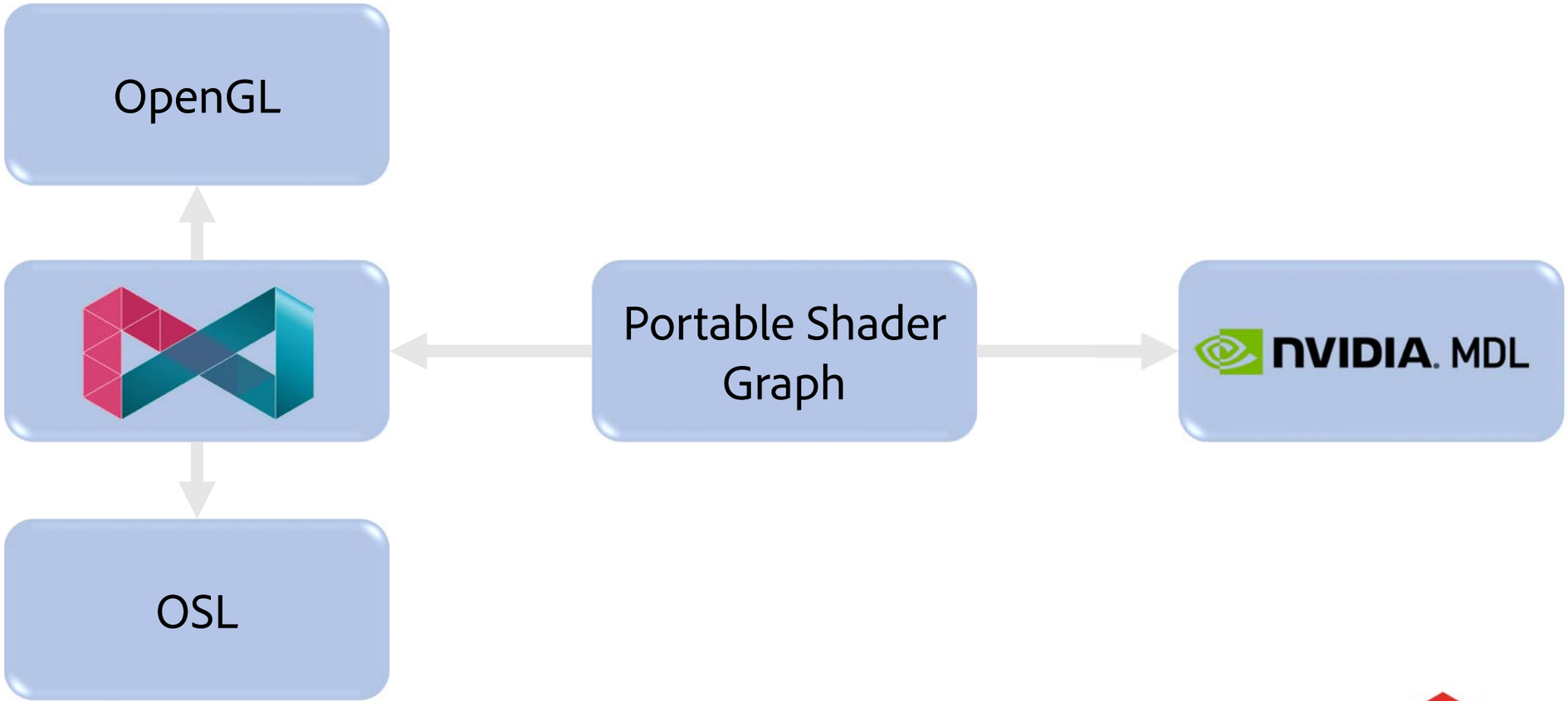
- Unlock new **Material Workflows**
  - But limit the scope to pattern modulation
  - Rely on StandardSurface as UberShader

# Extend the reach: Material Workflows




SUBSTANCE by Adobe

# Extend the reach: Material Workflows




SUBSTANCE by Adobe

# Dual Rendering


SUBSTANCE by Adobe

# Open in MaterialXView

- Verify correctness
- Play with parameters

# Extend the reach: Asset Workflows

- Import materials into Painter


SUBSTANCE by Adobe


Substance Painter 2019.1.3 - BB8_Demo
File Edit Mode Window Viewport Plugins Help
Size 0.34 Flow 100 Stroke opacity 100 Distance 3
Material
©&TM Lucasfilm Ltd. Used with Permission.
TEXTURE SET LIST
Settings
1001 Main shader
1002 Main shader
1003 Main shader
LAYERS
TEXTURE SET SETTINGS
Base Color
Cleaning_Mask Norm 100
BaseMaterial_1001 Norm 100
PROPERTIES - PHYSICAL PAINT
BRUSH
Size 0.34
Flow 100
Stroke opacity 100
Size Jitter 0
Flow Jitter 0
Alignment UV
SHELF
Tools
Search...
Particles
Tools
Materials
Smart materials
Smart masks
Environments
Autumn Le... BB8_Brush_01 BB8_HardW... BB8_PS_Pre... BB8_Wash BB8_Wash_01 Blur Brush_Flow... Brush_Spray Bullet Impact Cracks DEMO_Col... Dyn2_Blue Dyn2_Colori... Dyn2_Pink Dyn2_Red Dyn2_Yellow Dyn_2Col_F...
[Plugin - dcc-live-link] Command server error: The bound address is already in use
Cache Disk Usage: 22%

# Key Takeaways

- Worth pursuing portable shaders
- MaterialX is an effective carrier
  - We are interested in embedding in USD
- Improve API on Designer for custom node graphs
  - Our prototypes almost entirely coded in a python plugin


SUBSTANCE
by Adobe

# More in depth

Come to the **Autodesk Vision Series** tomorrow!

Come visit us at the **Substance booth**!

SUBSTANCE
by Adobe

A MATERIALX CASE STUDY:

# MATERIAL INTERCHANGE AT ILM

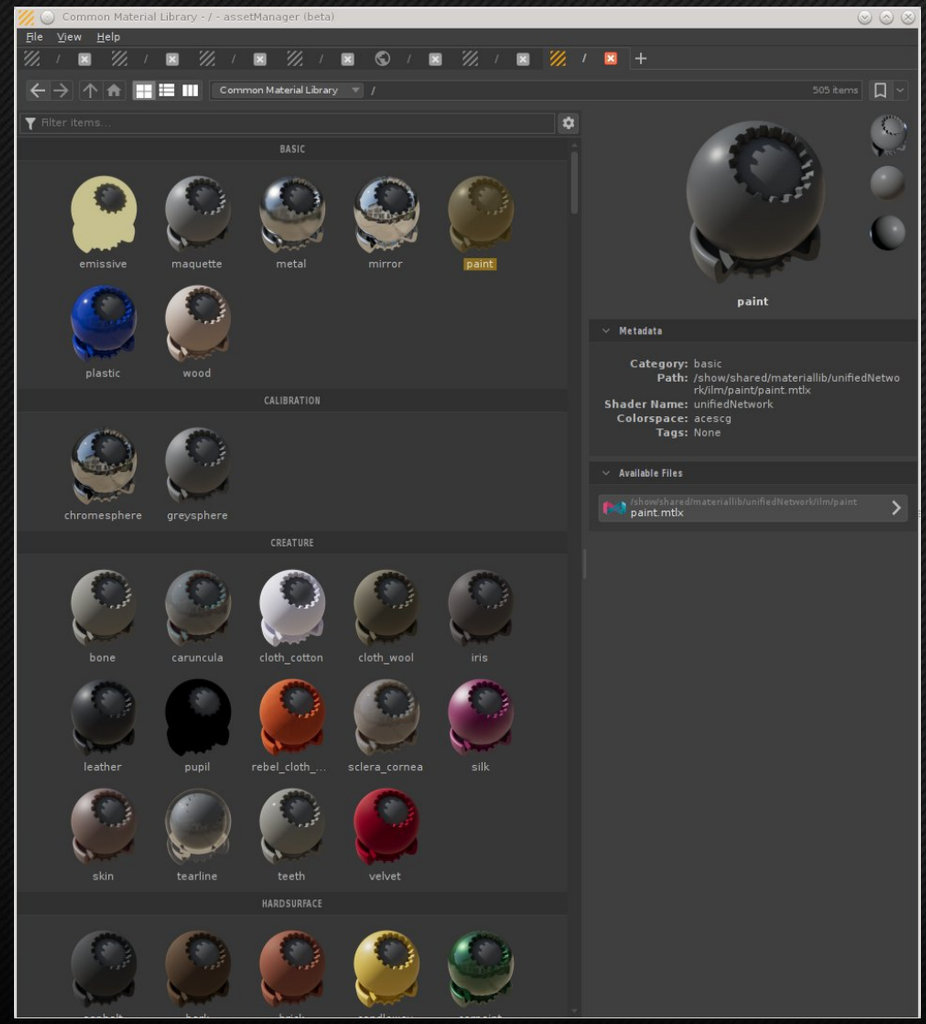
INDUSTRIAL LIGHT & MAGIC
SAN FRANCISCO SINGAPORE VANCOUVER LONDON SYDNEY

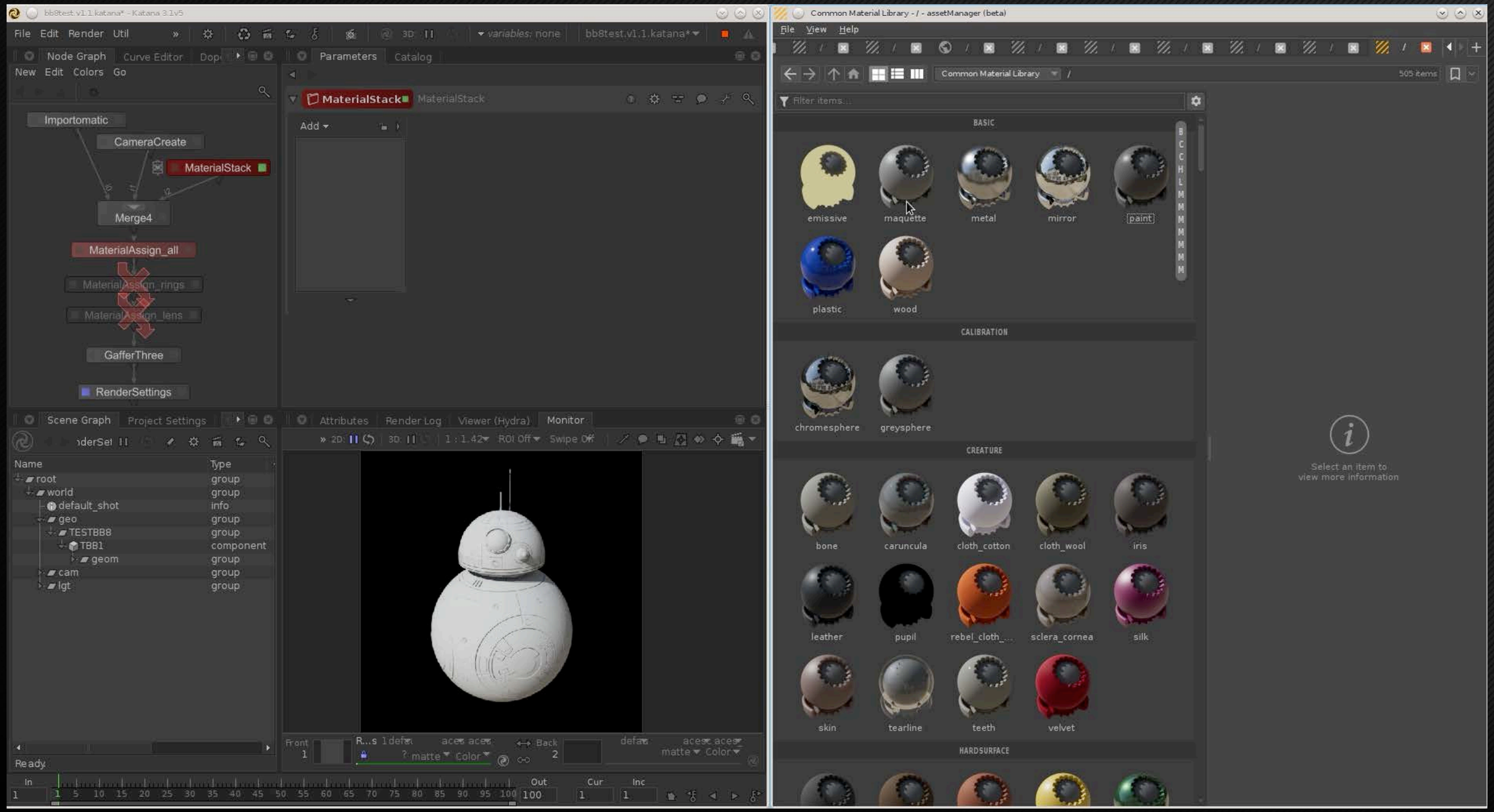
# THE PROBLEM

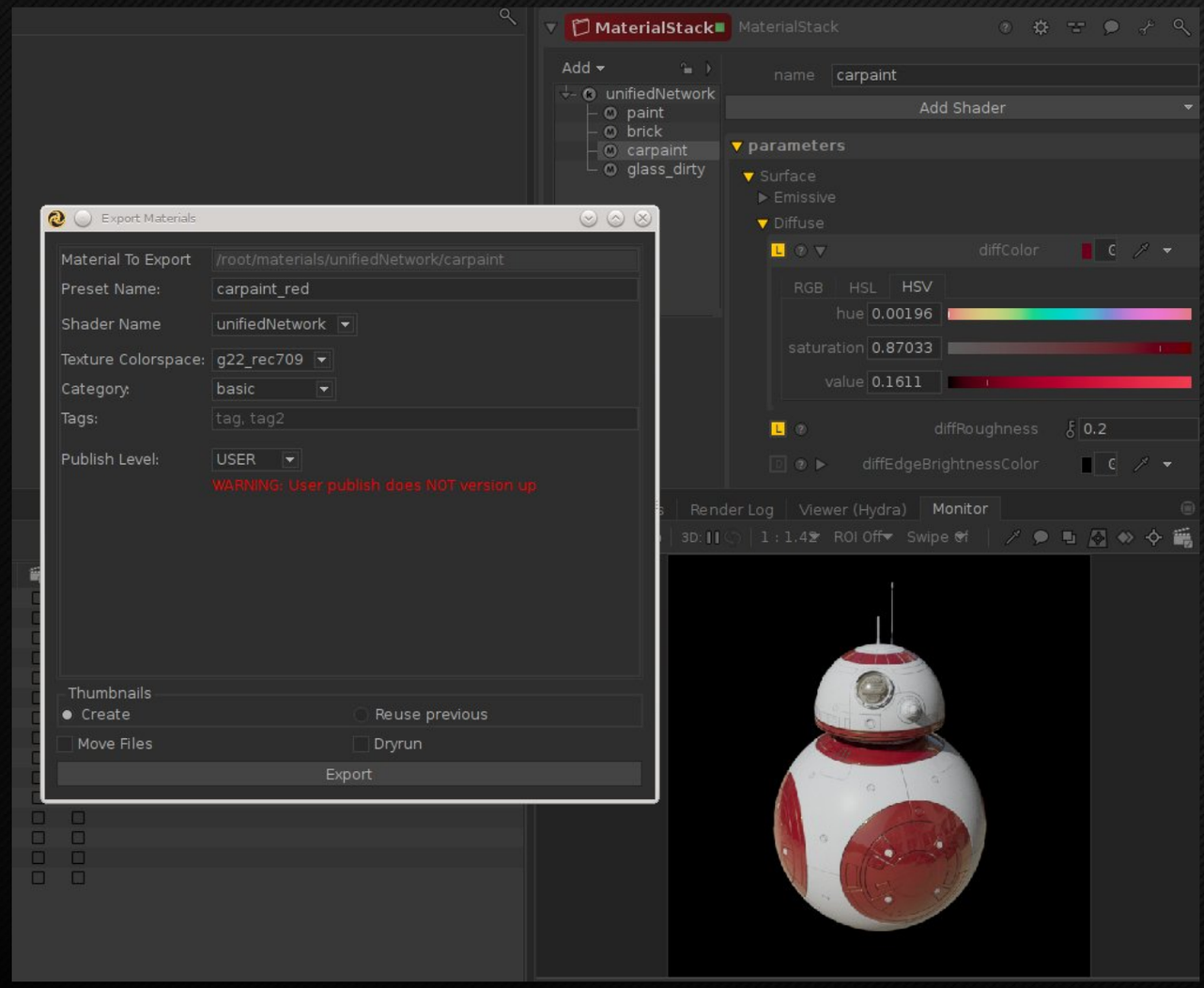
- Artists creating surface shading materials working in different content creation packages
- Quickly create and apply high-quality materials from a central library in any package
- Author final-quality materials in any one package, export back to library for use in any other package with minimal artist effort

# COMMON MATERIAL LIBRARY

- API wrapper for base MaterialX document access, publishing, version control, etc.
- Material browser panel:
  - Hub for shared materials across all packages
  - Standalone or plugin
  - Data stored in .mtlx format, with texture files




bb8test.v1.1.katana* - Katana 3.1v5
File Edit Render Util
Node Graph Curve Editor
New Edit Colors Go
Importomatic
CameraCreate
MaterialStack
Merge4
MaterialAssign_all
MaterialAssign_rings
MaterialAssign_lens
GafferThree
RenderSettings
Parameters Catalog
MaterialStack
Add
Scene Graph Project Settings
Name Type
root group
world group
default_shot info
geo group
TESTBB8 group
TBB1 component
geom group
cam group
lgt group
Ready
Attributes Render Log Viewer (Hydra) Monitor
Common Material Library - / - assetManager (beta)
File View Help
Common Material Library
505 items
Filter items...
BASIC
emissive
maquette
metal
mirror
paint
plastic
wood
CALIBRATION
chromesphere
greysphere
CREATURE
bone
caruncula
cloth_cotton
cloth_wool
iris
leather
pupil
rebel_cloth_...
sclera_cornea
silk
skin
tearline
teeth
velvet
HARDSURFACE
Select an item to view more information


MaterialStack
MaterialStack
Add
unifiedNetwork
paint
brick
carpaint
glass_dirty
name
carpaint
Add Shader
parameters
Surface
Emissive
Diffuse
diffColor
RGB
HSL
HSV
hue 0.00196
saturation 0.87033
value 0.1611
diffRoughness 0.2
diffEdgeBrightnessColor
Render Log
Viewer (Hydra)
Monitor
3D
1 : 1.4
ROI Off
Swipe
Export Materials
Material To Export
/root/materials/unifiedNetwork/carpaint
Preset Name:
carpaint_red
Shader Name
unifiedNetwork
Texture Colorspace:
g22_rec709
Category:
basic
Tags:
tag, tag2
Publish Level:
USER
WARNING: User publish does NOT version up
Thumbnails
Create
Reuse previous
Move Files
Dryrun
Export


Autodesk Maya 2018: untitled*
File Edit Create Select Modify Display Windows Key Playback Visualize Deform Constrain MASH Cache Arnold ToolCatalog Help
Workspace : Maya Classic*
Animation
No Live Surface
Symmetry: Off
Animation Curves / Surfaces Custom FX FX Caching Rendering Rigging Sculpting XGen anim_thirdparty blockParty2 cfl_crdev crdev_def crdev_sim DEV_crowdRigLib ebt_crdev faceUITools ILM ILM_anim ILM_anim_MOCKUP ILM_anim_preflight ILM_anim_retired
View Shading Lighting Show Renderer 2D Blast Panels
ilm_neutral (rec709g24)
persp
List Selected Focus Attributes Show Help
Make a selection to view attributes
Channel Box / Layer Editor
Modeling Toolkit
Attribute Editor
Select
Load Attributes
Copy Tab
No Character Set
No Anim Layer
24 fps
MEL
Select Tool: select an object

# UNDER THE HOOD

- Each Material references one of a fixed set of shader interface definitions
- Implementation of each "nodedef" for every application
  - Mari, Maya Viewport, MaterialXView: use native MaterialX + ShaderGen
  - RenderMan, Clarisse: custom network built from native nodes

# ACKNOWLEDGMENTS


- Vick Schutz: project lead
- Julien Cohen-Bengio: engineering lead
- Steve Muniz, Will Muto, Sarah Trop: project engineers
- Caitlin Hackett: project manager

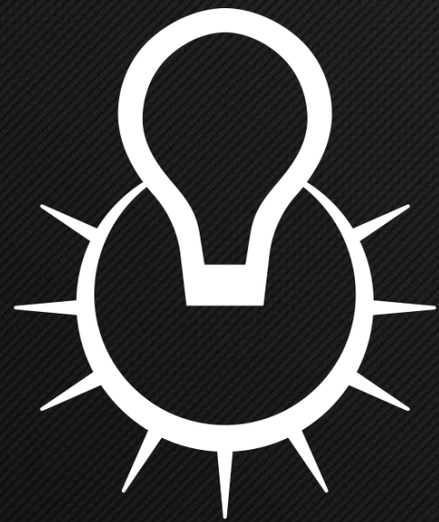
# We’re Hiring!

*Check our website for available positions at:*

**ILM.com/Careers**

Visit our ILM Recruiting team throughout SIGGRAPH at the
Disney Studios Suite  / Room 410

**San Francisco • Singapore • Vancouver • London • Sydney**



INDUSTRIAL LIGHT & MAGIC

SAN FRANCISCO SINGAPORE VANCOUVER LONDON SYDNEY

# THANKS!


MATERIALX

# Autodesk: Update on MaterialX in Architecture & Design

Henrik Edström

Sr. Software Architect, Graphics Technology

AUTODESK.

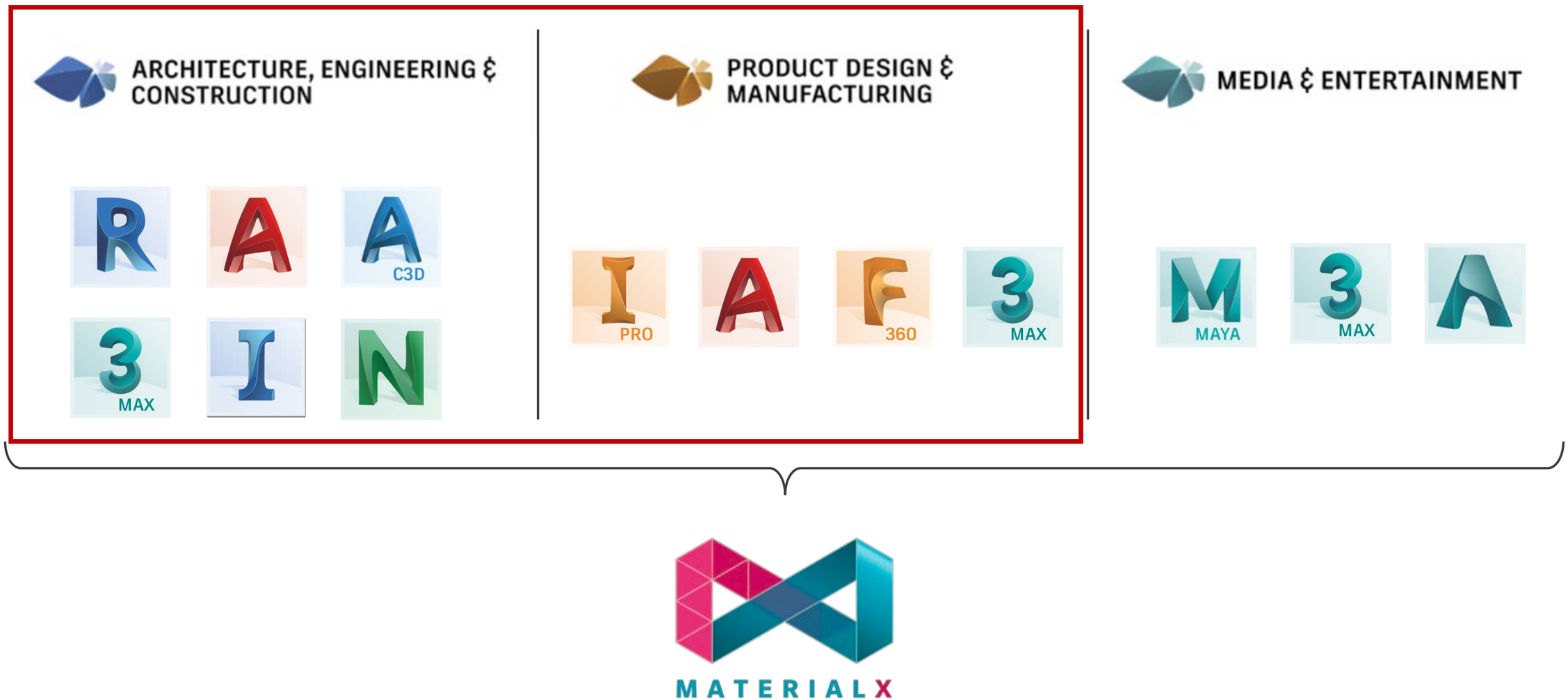
# Over 100 products and millions of users…


AUTODESK®

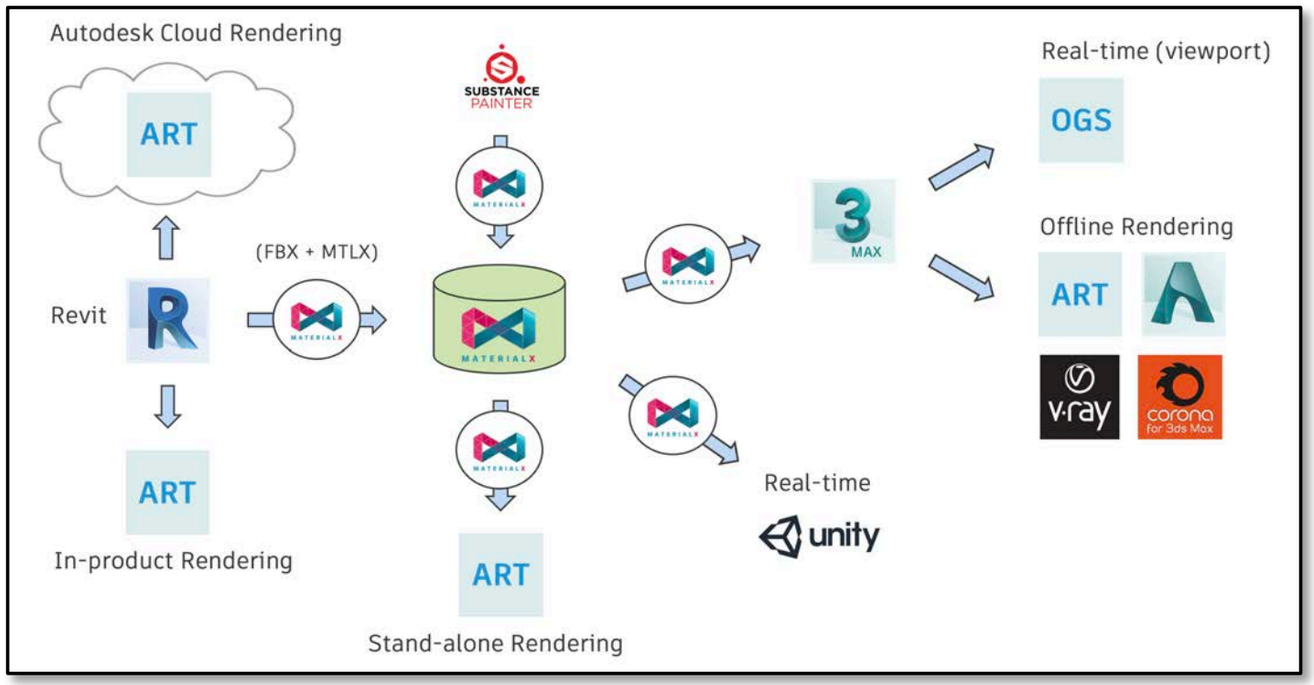
Make anything™


RCCf02

# One material standard across all industries

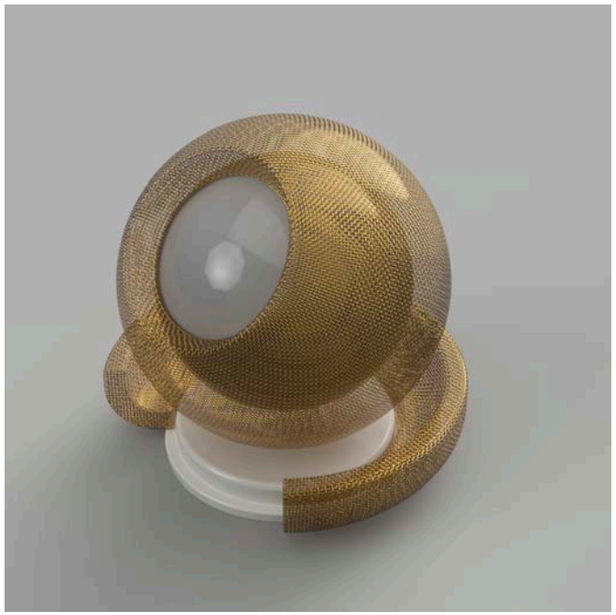


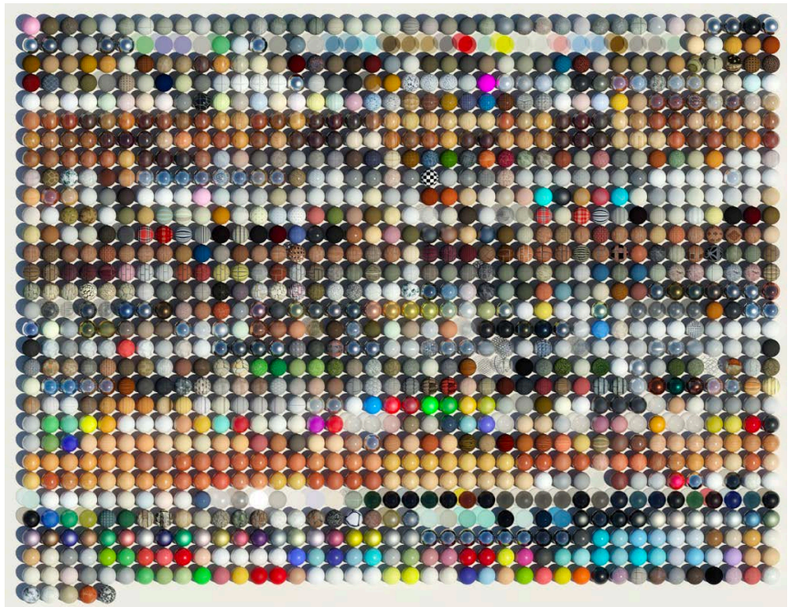
# MaterialX BOF 2018 – ArchViz Proof of Concept Demo



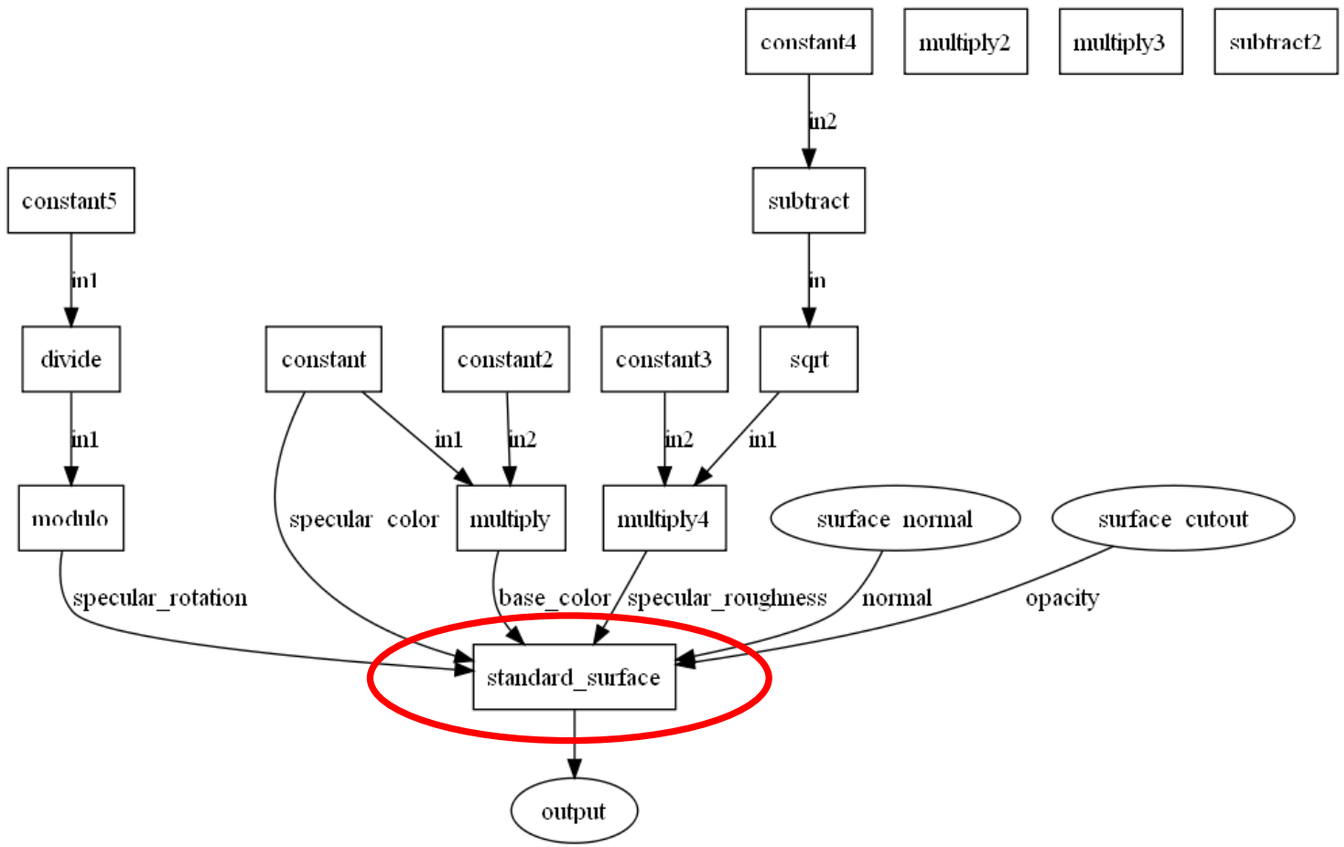

https://www.materialx.org/assets/MaterialX_Sig2018_BOF_slides.pdf

# Converting Autodesk Materials shared library to use MaterialX & Standard Surface

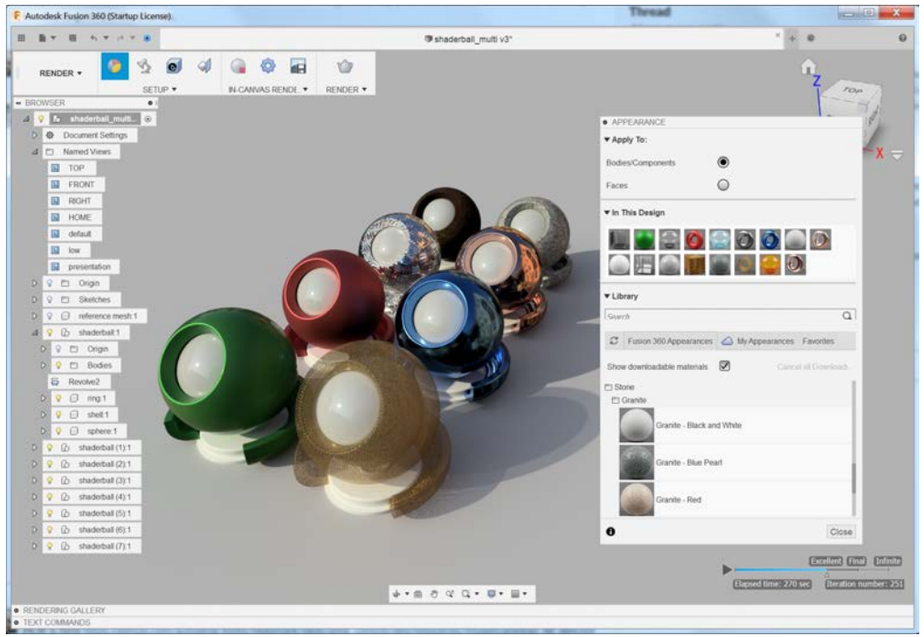


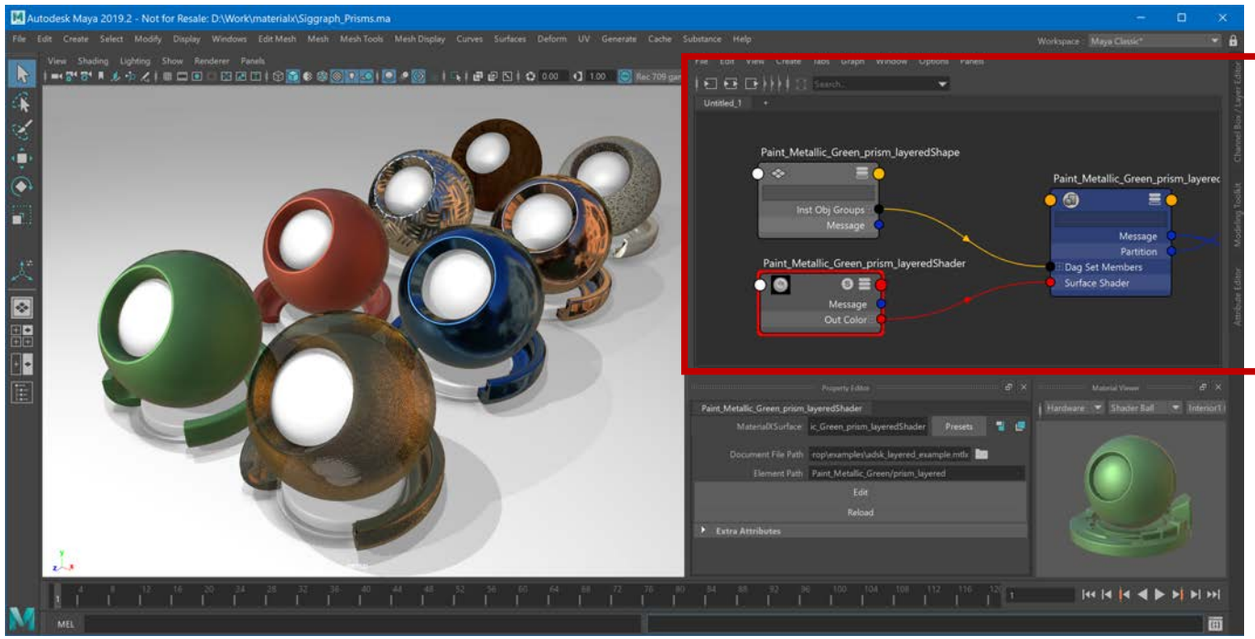


# Mapping existing PBR material classes to MaterialX & Standard Surface node graphs




# Rendering materials from Revit and Fusion 360 in Maya using MaterialX & Standard Surface

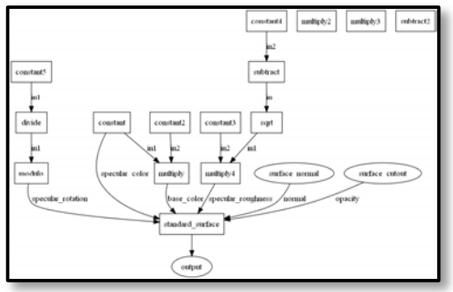


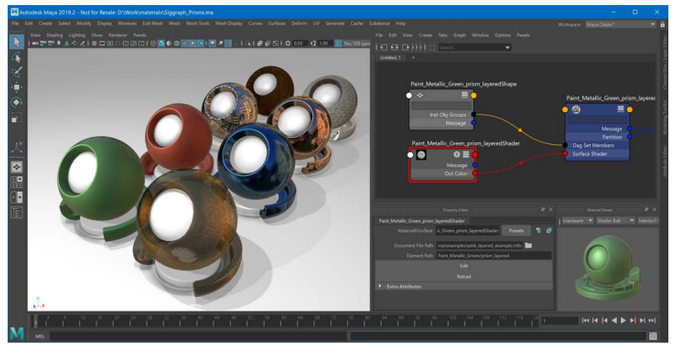




360

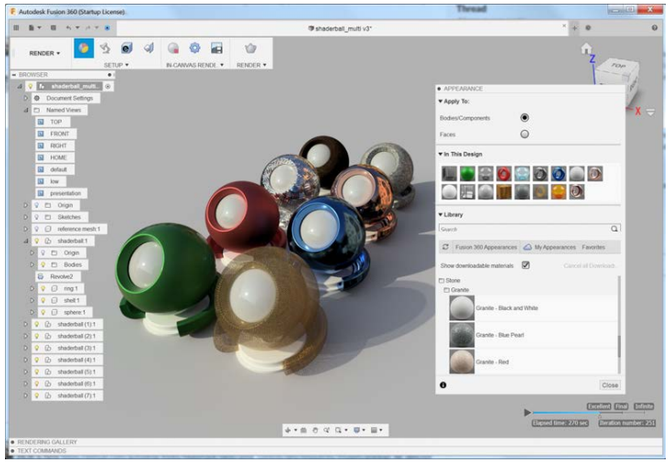
MAYA

*2019.2 Dev*

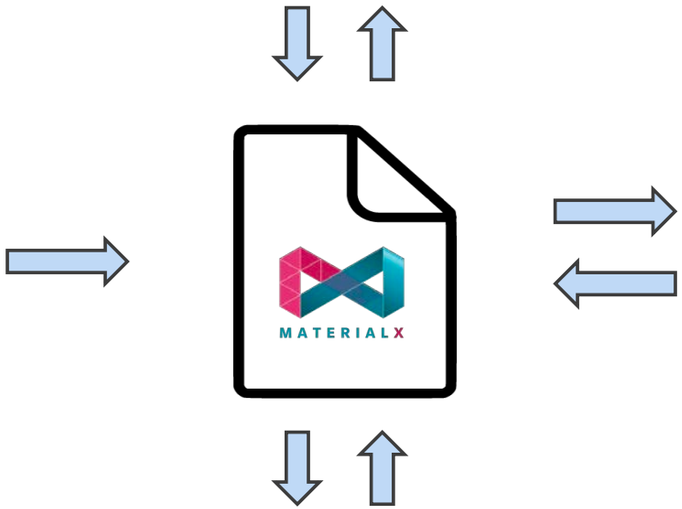


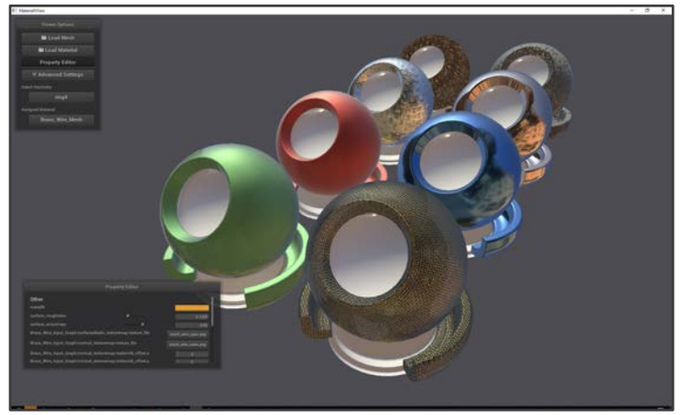



Maya

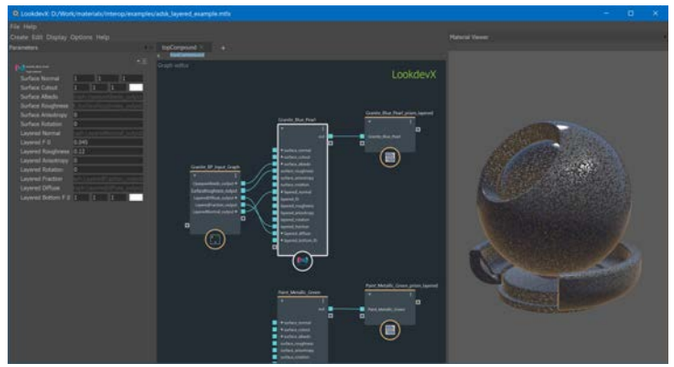


Fusion 360



MaterialXView



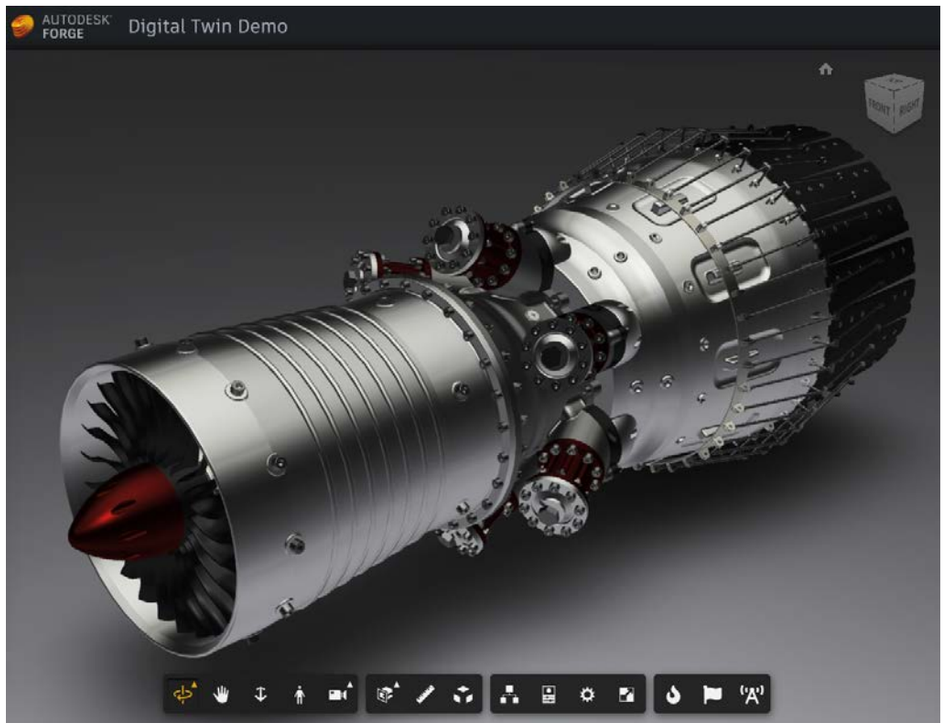

LookdevX

# Work in progress – adding *unit* attribute

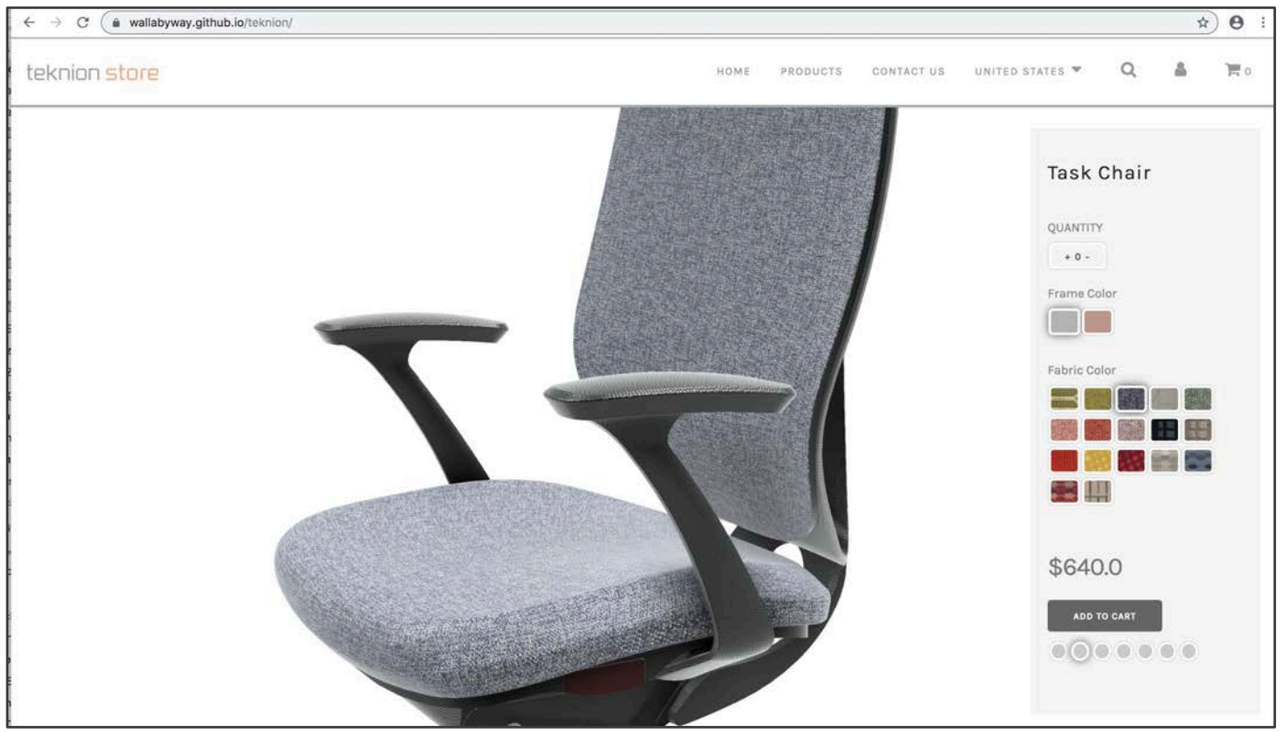
**Example MaterialX document using unit attribute**

```
<?xml version="1.0"?>
<!--
    Our working space distance unit is cm, angle unit is radians, temperature unit is Kelvin.
    Any value or texture in this document that are specified in a different unit will be converted to this working space unit.
 -->
<materialx version="1.37" unit="cm, rad, K">
    <!-- This constant will be multiplied by 100 -->
    <constant name="constant1" type="float">
      <parameter name="value" type="float" value="42" unit="m" />
    </constant>
    <!-- No unit specified - no conversion done -->
    <constant name="constant2" type="float">
      <parameter name="value" type="float" value="12" />
    </constant>
    <!-- Values read from this texture will be divided by 10 -->
    <image name="my_displacement" type="float">
      <parameter name="file" type="filename" value="some_displacement.jpg" unit="mm"/>
    </image>
</materialx>
```

# Work in Progress – GLSL ES ShaderGen backend for WebGL




Autodesk Forge Viewer




Third-party configurator on Forge Viewer

# Special thanks to

Ashwin Bhat

Bernard Kwok

Roberto Ziche

Lucasfilm & ILM

# Autodesk Standard Surface in MaterialX


https://github.com/Autodesk/standard-surface

Branch: master ▾ standard-surface / reference / standard_surface.mtlx Find file Copy path

niklasharrysson Added materialx definition and implementation (#1) 1324fd6 on May 28

1 contributor

253 lines (239 sloc) 14.8 KB Raw Blame History

```
<?xml version="1.0" encoding="UTF-8"?>
<materialx version="1.36">
  <!--
    Autodesk Standard Surface node definition.
  -->
  <nodedef name="ND_standard_surface_surfaceshader" node="standard_surface" type="surfaceshader" nodegroup="pbr"
           doc="A surface uber shader based on the Arnold standard surface shader">
      <input name="base" type="float" value="0.8" uiname="Base" uifolder="Base"/>
      <input name="base_color" type="color3" value="1, 1, 1" uiname="Base Color"  uifolder="Base"/>
      <input name="diffuse_roughness" type="float" value="0" uiname="Diffuse Roughness"  uifolder="Diffuse"/>
      <input name="specular" type="float" value="1" uiname="Specular" uifolder="Specular"/>
      <input name="specular_color" type="color3" value="1, 1, 1" uiname="Specular Color" uifolder="Specular"/>
      <input name="specular_roughness" type="float" value="0.1" uiname="Specular Roughness" uifolder="Specular"/>
      <input name="specular_IOR" type="float" value="1.52" uiname="Index of Refraction" uifolder="Specular"/>
      <input name="specular_anisotropy" type="float" value="0" uiname="Specular Anisotropy" uifolder="Specular"/>
      <input name="specular_rotation" type="float" value="0" uiname="Specular Rotation" uifolder="Specular"/>
      <input name="metalness" type="float" value="0" uiname="Metalness" uifolder="Specular"/>
      <input name="transmission" type="float" value="0" uiname="Transmission" uifolder="Transmission"/>
      <input name="transmission_color" type="color3" value="1, 1, 1" uiname="Transmission Color" uifolder="Transmission"/>
      <input name="transmission_depth" type="float" value="0" uiname="Transmission Depth" uifolder="Transmission"/>
      <input name="transmission_scatter" type="color3" value="0, 0, 0" uiname="Transmission Scatter" uifolder="Transmission"/>
      <input name="transmission_scatter_anisotropy" type="float" value="0" uiname="Transmission Anisotropy" uifolder="Transmiss
      <input name="transmission_dispersion" type="float" value="0" uiname="Transmission Dispersion" uifolder="Transmission"/>
      <input name="transmission_extra_roughness" type="float" value="0" uiname="Transmission Roughness" uifolder="Transmission"
      <input name="subsurface" type="float" value="0" uiname="Subsurface" uifolder="Subsurface"/>
      <input name="subsurface_color" type="color3" value="1, 1, 1" uiname="Subsurface Color" uifolder="Subsurface"/>
```

**standard-surface**/reference/**standard_surface.mtlx**

WED, JULY 31 3:15PM
ROOM 404A & B

REGISTRATION REQUIRE

VISION | SERIES

# Open Source at Autodesk - MaterialX

**ERIC BOURQUE**
Director of Engineering at Autodesk

**JONATHAN STONE**
Senior Software Developer at Lucasfilm

**DAVID LARSSON**
Senior Software Engineer at Adobe (Allegorithmic)

*Image courtesy of Vipir*

AUTODESK®

Make anything™