

REDSHIFT TO RADEON PRORENDER CONVERSION REPORT

Version 2.9.4, 17 December 2018

This report summarizes results of tests made to convert Redshift nodes to Radeon ProRender nodes.

SUMMARY

For the report, **54** Redshift nodes within **7** node groups has been tested in total. The results of conversion are as follows:

- **CONVERTIBLE**: **10** nodes (**18%**), [see details](#)
- **PARTIALLY CONVERTIBLE**: **7** nodes (**13%**), [see details](#)
- **NOT CONVERTIBLE**: **22** nodes (**41%**), [see details](#)
- **RESEARCH NEEDED**: **15** nodes (**28%**), [see details](#)

Currently, most of the nodes are converted by the script. In some cases, we preserve original information as is, in some we are trying to accommodate the input values from Redshift to available RPR features. Architectural, rsMaterial, Incandescent, VolumeScatter – the majority of their parameters are converted as close to original, as possible. This is a continuous process of refining the formulas, to better match render results, as more testing provides more sample data. Conversion for some of the nodes is yet to be implemented, such as lens and tonemapping effects.

REPORT DETAILS

In this report:

- [What's New](#)
- [Script Link](#)
- [Known Issues](#)
- [Complex Scenes](#)
- [Conversion Status by Node Group](#)

WHAT'S NEW

What's new between versions 2.9 > 2.9.4:

1. Fixed IES orientation.
2. Support for the whole range of units for light sources (refer to comparison screenshot for value proportions).
3. Refining reflection color formula for metallic surfaces (for more complex cases with Color + edge tint Fresnel mode).
4. Subsurface usage for multiscatter layers.
5. Transparency for incandescent materials (inverted alpha value).
6. Refining volumeScattering conversion.
7. Conversion for metallness from architectural materials with extinction coefficient.
8. Refining Subsurface materials.
9. Save more bright colors.

SCRIPT LINK

The latest version of the conversion script: [download script](#).

KNOWN ISSUES

The following JIRA issues affecting the conversion process were identified:

- Shadow catcher render with a lot of noise and sometimes gets bad results ([RPRMAYA-801](#))
- Textures are being downscaled by supported standard Maya nodes ([RPRMAYA-848](#))
- [Core] Artifacts when using Displacement ([RPRMAYA-329](#))
- [Core]Sky System issue ([RPRMAYA-147](#))

Currently, we have several issues with how some of the values get converted. Mostly, this concerns SSS and Metallic effects in rsMaterial and SubsurfaceScatter materials. RPR handles color brightness and saturation differently, so to have a better match to Redshift output, we are researching the way to bring it as close as possible. For metallic effects, we covered the Fresnel effects for rsMaterial and rsArchitectural, but more formula refining and error fixing is needed. In some cases, metal materials acquire hue shifted or darkened colors in reflection. This issue is being investigated (error in Reflectivity and reflection color mixing process is suspected).

VolumeScatter needs further refinement, to compensate for difference between RPR and RS scatter algorithms.

Area light shapes, PhysicalSky conversion, Displacement node value conversions still needs to be implemented.

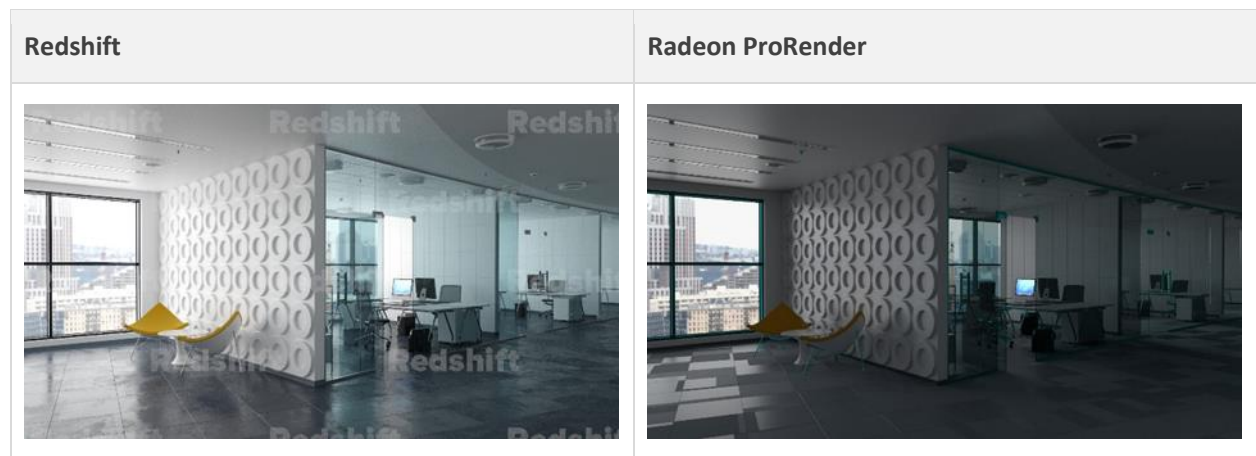
COMPLEX SCENES

Scene 1

Office interior. Small scene with Sun and Sky, refractive materials, and procedural maps for bump and roughness.

Known issues:

1. Error in Metal reflection color conversion, first appeared in this version. Grey metal gets color shift into blue.
2. Noise procedural for roughness and bump got lost.
3. Area light is dimmer, but most of the brightness is lost due to Tonemapper, and dimmer reflection from materials. Needs more research, is that how RPR handles energy conservation, or something else.

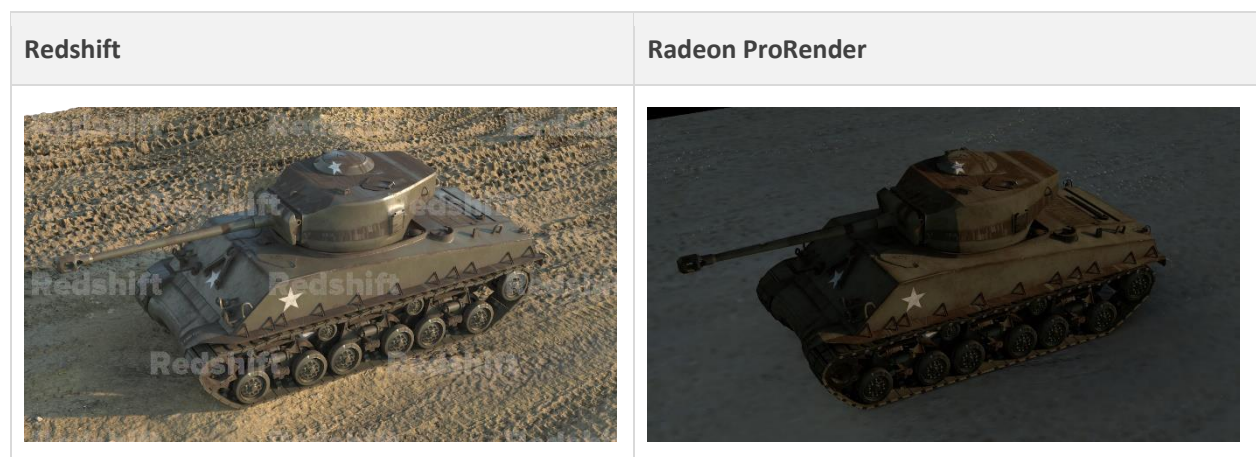


Scene 2

Tank. Object render with displacement ground.

Known issues:

1. Ground Material. Currently, the color layer node is being converted in suboptimal way. To better implement it via series of arithmetic nodes, more complex algorithm is needed.
2. Displacement got lost in conversion. It was supposed to be added into Ubermaterial slot, instead of the shading group slot due to one of the RPR bugs ([RPRMAYA-675](#))
3. Brightness and tonemapping - same as a previous scene, but more severe.

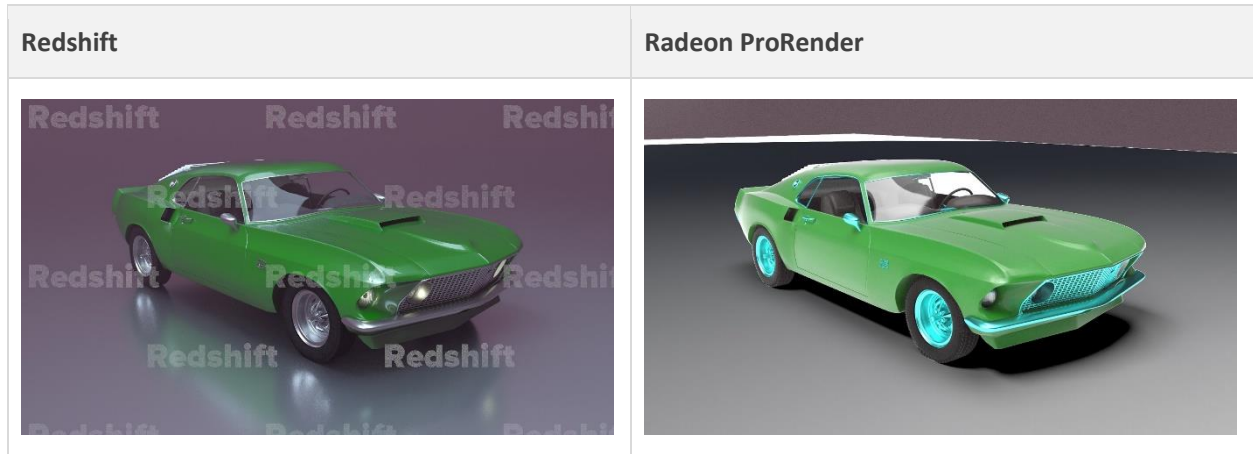


Scene 3

Mustang. Object scene with MatteShadow, VolumeScatter and Carpaint.

Known issues:

1. Car paint doesn't store the edge falloff color for now. Color mixing through the Fresnel node will be needed, with additional testing for map mixing.
2. Same as the first scene, metal is converted with color shift.
3. RPRVolume is not dense enough for this scene, and it isn't visible against the shadow catcher, with very hard border between what's covered with RPRVolume, and what's outside with ShadowCatcher. Possible RPR bug needs investigation.

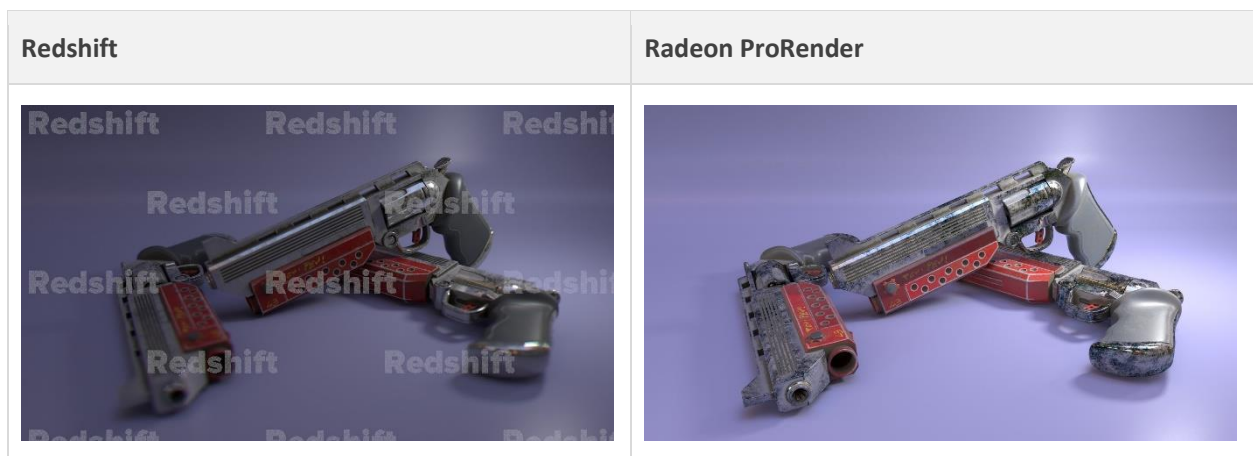


Scene 4

Complex baked maps.

Known issues:

1. DoF needs to be implemented with the default camera parameters. Possible tonemapper needs to compensate for bokeh.
2. Metal color, same as above.
3. Mixed maps looks too high-contrast.

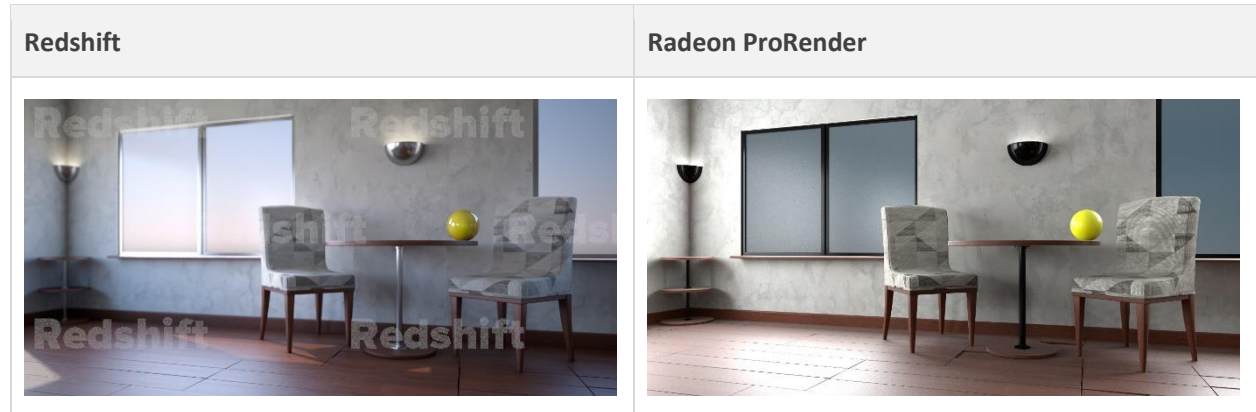


Scene 5

Simple interior, part 1

Known issues:

1. Physical sky needs to have more brightness with conversion.
2. Portal lights are converted to area light with white color.
3. DoF, as above.

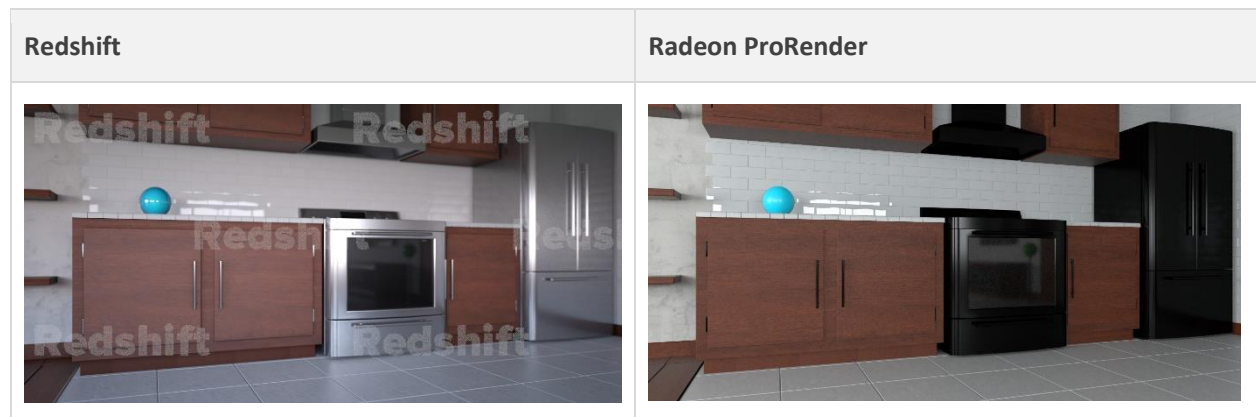


Scene 6

Simple interior, part 2

Known issues:

1. Metal for Architectural material gets converted into too dark reflection color.
2. Reflections for Architectural non-metals needs compensation for more effect.
3. DoF and Portal lights as above.

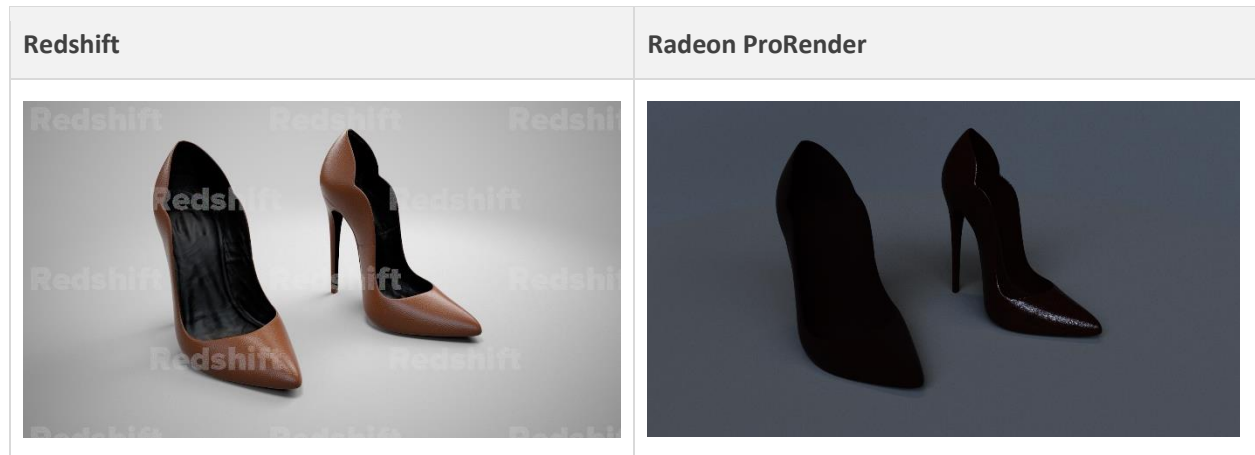


Scene 7

Shoes

Known issues:

1. Tonemapper as above.
2. Issue with brightness and saturation of the SSS material. Needs more research to determine how to transfer values from RS to RPR with compensation for brighter colors in RS.

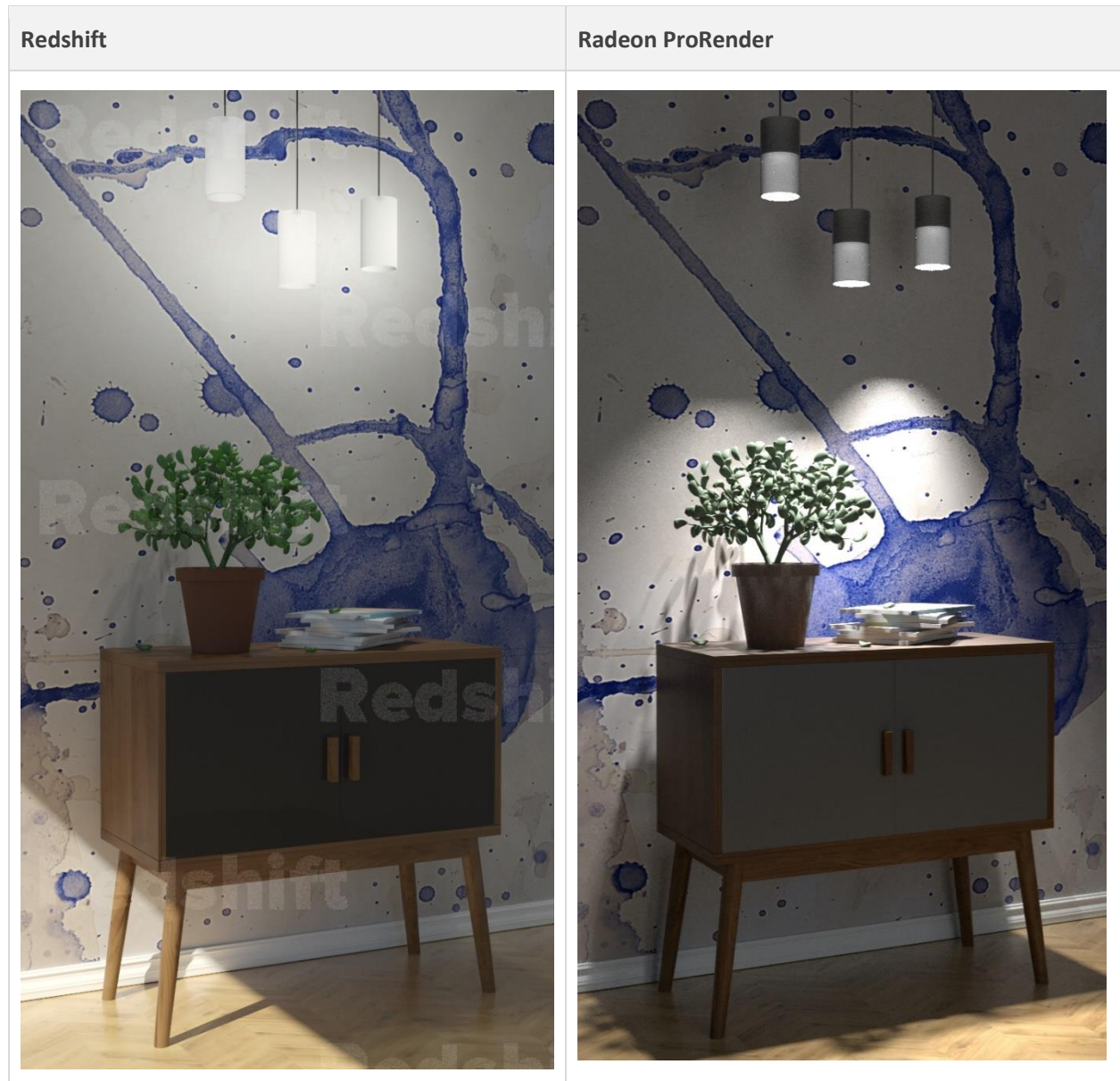


Scene 8

Small object render, lit by Physical Sun and arealights.

Known issues:

1. Translucency->Backscatter conversion issue, need to find more accurate formulas to convert
2. Grey color gets too light. Needs investigation.
3. Sun currently doesn't convert to the Directional light, TBD.



CONVERSION STATUS BY NODE GROUP

Node Group	Total Nodes	Convertible	Partially Convertible	Not Convertible	Research Needed	Details
Environment	1	1	0	0	0	Link
Lens	3	0	0	1	2	Link
Lights	7	2	2	1	2	Link
Materials	11	2	2	2	5	Link
Physical Sky	1	0	0	0	1	Link
Utility Nodes	29	5	3	17	4	Link
Volume Scattering	2	0	0	1	1	Link
Total	54	10	7	22	15	Link