

# RenderMan 24.3

Release date: January 6, 2022

## Welcome to RenderMan 24.3!

RenderMan 24.3 brings some new functionality to artists, as well as several bug fixes.

These release notes build on the the release notes for 24.0, 24.1, and 24.2 so please see those notes first.

For each of the bridge product integrations, please see the Release Notes within each of their respective sections.

Dive right in using the table of contents below!

- [Welcome to RenderMan 24.3!](#)
  - [What's New](#)
    - [RIS and XPU](#)
    - [RIS](#)
    - [XPU](#)
    - [Other](#)
  - [Changes in Stylized Looks](#)
  - [Other Bug Fixes](#)
    - [RIS and XPU](#)
    - [RIS](#)
    - [XPU](#)
    - ["it"](#)
    - [hdPrman - the RenderMan Hydra Renderer Delegate](#)
  - [Miscellaneous Changes](#)
    - [RIS](#)
    - [XPU](#)

## What's New

### RIS and XPU

- `txmake`: Added support to `txmake` for aborting on input NaN's by using `-checknan`

### RIS

- OSL: Custom LPE match and throughput values can now be accessed in OSL via `getattribute("lpe:match"/"lpe:throughput", "<customlpenname>"/"<customlpeexpression>",val)`; This wires the `RixCustomLPE` API into OSL via a custom `getattribute` call. This will allow the shading to vary based on the throughput or in paths specified by a custom LPE. For example, a shader can respond and return different values depending on whether it a ray has refracted through a particular object or not.
- LPEs: Added 'white' LPE flag, allowing the color of shading events to be excluded from light path expressions. This allows other properties of the shaded points to be written to AOVs without the color being multiplied in, extending the set of things which LPEs can be used to express.

### XPU

- Geometry: Added support for matrix primvars

### Other

- We now support Mari 4.5 up to 5.0

## Changes in Stylized Looks

`PxrStylizedLines`:

- Optimizations for faster edge detection
- UI updates: Dilate Sort Page

`PxrStylizedControl`:

- New features: Sections Add, Sections Quantize
- Updates to Section Spread Scale algorithm

## Other Bug Fixes

### RIS and XPU

- Stats: Stability improvements. When running Live Stats it is no longer necessary to configure a port number. An available port will be chosen automatically by the system

- Patterns: PxrFractal: Fixed a regression related to the application of the frequency parameter. Added a new parameter for backward compatibility if needed
- Patterns: PxrFractal: Improved frequency clamping to optimize render times

## RIS

- Integrators: Issues with unpredictable random number generation for photon generation in the PxrVCM and PbsPathTracer/PxrUnified integrators have been fixed
- Volumes: Issues with incorrect alpha channel results for volumes that are marked matte or holdout have been fixed, including cases where the matte volume holds out an overlapping non-matte volume. One remaining known issue in 24.3 is that matte volumes will not produce the correct alpha channel if accumOpacity is turned on in the PxrPathTracer integrator.
- Volumes: An issue with artifacts in the shadows of aggregate volumes rendered with numLightSamples > 1 in PxrPathTracer has been addressed
- Textures: Fixed incorrect derivatives when using mipBias or maxResolution in texture patterns
- PxrVCM: Fixed a bug in caustics when the light source is a scaled rect light with emission defined by a texture
- BxDFs: Parameters "iridescenceAnisotropy", "iridescenceBumpNormal", "glassAnisotropy" and "glassBumpNormal" are added to PxrLayerSurface
- BxDFs: Lama: LamaDielectric and LamaGeneralizedSchlick now support the binding of interior aggregate volumes via the new parameter "volumeAggregate".
- BxDFs + Integrators: Fixed an issue with PxrPathTracer when aggregate volumes were bound to the interior of an object via PxrSurface or LamaDielectric. This case now works.
- Lights: Fixed a flickering issue when lights overlapped with each other
- Lights: Fixed a case where mesh lights wouldn't properly cast shadows

## XPU

- BxDFs: Fixed XPU PxrSurface specular energy compensation for "specularFresnelMode" 1
- BxDFs: Fixed a crasher that resulted if subsurface scattering from PxrSurface was used within nested instances
- Geometry: Fixed an issue where some polygons could render black
- Memory: Fixed several memory leaks that would manifest during interactive renders
- Crop Windows: Fix XPU lockup when crop window is very small

"it"

- Stats: Greater stability when interacting with the live stats system

## hdPrman - the RenderMan Hydra Renderer Delegate

- RenderMan hydra delegates no longer include a version string in the label
- Improved support for Cryptomatte in hdPrman

## Miscellaneous Changes

### RIS

- Lighting: Faster startup if your scene has mesh lights

### XPU

- Stats: More memory statistics are being reported