

RenderMan 22.3

Welcome to RenderMan 22.3!

Welcome to RenderMan 22.3. This release introduces improvements to the previous RenderMan.

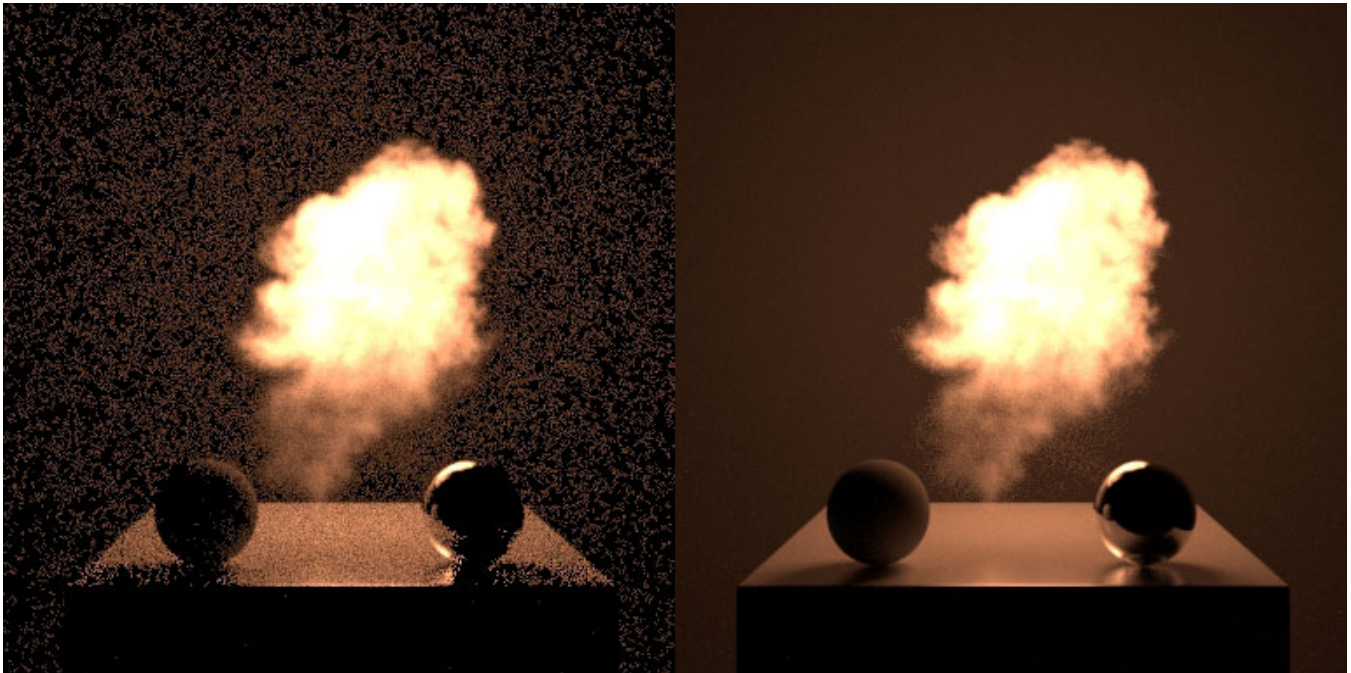
Please dive right into the release notes below for more detailed information on the latest version of your favorite renderer!



Note that there are API changes in RenderMan 22.3 that require you recompile your plugins. You can view and download the Doxygen files [here](#).

What's New

- **Use Emissive Volumes as a Light Source** — A new option on [PxrVolume](#) allows the renderer to sample emissive volumes as light sources. This improves performance for scenes where the volume contributes to lighting, like fire and explosions illuminating their surroundings. The below image shows off versus on with the same settings.



- **The Light Learning scheme**— for improved performance is now default. This should improve convergence for scenes with complex setups involving many lights seeing the most improvement. This version is not deterministic and small differences can be seen if not well converged
- **Checkpointing with Deep Data** — is supported as well as additional performance options for output. See the documentation for more details
- **Further improvements to OSL performance** — The SIMD version of [OSL](#) is optimized to execute 16 samples at a time with Intel® Advanced Vector Extensions 512 (Intel® AVX-512). It can also execute using Intel® Advanced Vector Extensions 2 (Intel® AVX2), Intel® Advanced Vector Extensions (Intel® AVX), or Intel® Streaming SIMD Extensions 4.2 (Intel® SSE4.2) at reduced performance levels.

Miscellaneous Changes

- Added more mitigation against corruption in denoised images when there are NaNs in the albedo variance channel
- Add basic scene graph translation stats to "Expert View > Plugins"
- Add primvar buffer (scene translation) stats to "Expert View > Memory"
- Improved threading behavior of light selection setup
- Remove unused master attribute "trace:samplemotion", we suspect you want true raytraced motion blur everywhere by now
- The texture3d() OSL function has been implemented. It can read 3D texture data from point cloud files, organized point cloud files, and brick map files. The return type can be float or color. (The OSL function resembles the old RenderMan RSL texture3d() function.)
- Improved performance of light selection setup in scenes with many thousands of lights and light linking

- On Windows (only) the default number of textures that can be opened is changed, which can result in dramatic speed ups for scenes with thousands of textures. The rendermn.ini setting /prman/textures/maxfiles is no longer ignored on Windows. On Windows the default is now 2048 (was 512) which is the same as the maximum allowed on Windows
- Significant memory reduction of curves geometry, the improvement is more pronounced for cubic b-spline and catmull-rom basis. This may cause a very small look difference if examining close individual strands
- Reduced OSL compile times
- Images rendered with more than 4K samples per pixel will have less noise than before
- The time to first pixel for dense subdivision meshes with watertight dicing has been improved
- Max resident memory reporting on Linux has been improved, often showing a higher memory footprint
- Peak memory consumption for dense subdivision surfaces partitioned into many facesets has been greatly reduced
- Checkpoints are now written to .part files during checkpointing and only replace the prior checkpoint once all of them have been successfully written
- Memory statistics tracking for scenes with huge numbers of instanced objects has been significantly improved
- OSL shading system optimization levels have been adjusted for better performance out-of-the-box
- Added new "objectdistance" dicing strategy that is guaranteed to be stable across all geometry instances. This dicing strategy dices to a object space distance length set by dice:worlddistancelength
- A new attribute, "displacement" "ignorereferenceinstance" has been added. It takes the value 0|1 and is off by default. If set to 1, displacement shading will ignore the referenceinstance; this may be useful for eliminating any instance-related variability in displacement shading
- Extended RixSCAccess to allow labeling BXDF inputs with the execution modes where they will be needed to enable future renderer-side optimizations
- The OSL source for the blue marble preset is now included in the preset assets
- Improved memory statistics for lighting
- "checkpointElapsed" is now a float attribute that can be added to EXR files

Fixes

- Fixed a bug where editing a geometry master instanced inside a group would yield a crash
- The normal N on brickmap gprims has been fixed
- Addressed a bug with checkpointing that could corrupt tiled openexr file output
- Fix incorrect transformations for geometric lights
- Bug fix in "it" that would lead to a crash if you opened a image file from disk while the A.I. denoiser was enabled
- Fixed bug in PxrRemap pattern where uninitialized values are returned when the value of inputMin and inputMax are equal
- A problem with facevarying interpolation near semi-sharp geometric creases on subdivision meshes which could lead to swimming textures has been fixed
- Fixed a bug that could lead to incorrect transformations when using named coordinate systems during cached presence and opacity calculations
- The peak memory consumed by the ray accelerator is now correctly counted
- A small memory leak incurred by the ObjectInstance RI call has been fixed. Besides plugging this leak, the memory consumption per instance has also been reduced slightly
- Fixed a bug that could cause checkpoint recovery to fail when the "origin" Display parameter is used
- Fixed an issue where a Specular Gint setting of 0 caused a crash in PxrMarschnerHair
- A stack overflow when rendering volumes on OSX has been addressed
- Improved memory statistics for lighting
- Fixed calls to RixTransform/RiTransformPoints when called from Ri procedurals

API Changes

- RixLightFactory::Create() and RixLight::Edit() have a new parameter: "RtUString const name". The name of the light is the attribute "identifier: name" of the instance of the light. This is used to properly track lights in lpes and aovs.
- There is a new output parameter for RixLight's GenerateSamples(), EvaluateSamples() and GenerateEmission(), the light's normal.

Interactive/Live Rendering Limitations

- Crop window edits are restricted to fall inside the original crop window

RenderMan Pro Server Limitations

- PxrUnified integrator does not yet support all the standard rendering features
- Meshlights cannot be instanced
- We do not read point data from OpenVDB files
- PxrSurface back diffuse color is not output to the albedo color AOV
- Analytical lights placed inside volumes may yield artifacts when made visible to the camera. As a work around, the light camera visibility should be turned off, and a geometry with a similar shape should be used (visible to camera, invisible to transmission and indirect rays), with the proper emissive bxdf
- Using the '.' character in the handle for an OSL shader could cause unpredictable results during re-rendering
- Per-Instance baking is not supported, only the reference instance
- 3d baking: no direct bake-to-ptex support
- PxBakePointCloud cannot directly render ptex
- Sample/Display filter plug-ins do not have access to lighting services for light dependent effects, e.g. lens flare
- Adding new mesh light on existing geometry during IPR results in double geometry
- Motion blurred polygons do not motion blur normals when deformed. Use Subdivision meshes instead
- When attempting to access an array primvar, you must first check the size of the array primvar and allocate the appropriate space. Not doing so may lead to a crash
- Points and curves cannot be used as geometric (mesh) lights
- Deformation motion blurred volumes don't currently work with densityFloatPrimVar or densityColorPrimVar. You will need to use a PxrPrimVar node connected to densityFloat and densityColor instead



The Centos KDE style "Oxygen" installs a version of Qt and sets the user's environment variable `QT_PLUGIN_PATH` forcing "it" and LocalQueue to attempt to load an incompatible Qt library. Either avoid installing the Oxygen theme or unset `QT_PLUGIN_PATH` before running "it" or LocalQueue. Other KDE styles may also install this theme.