

RenderMan 22.6

Welcome to RenderMan 22.6!

Welcome to RenderMan 22.6. 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!

 Some recent changes to the API will require a recompile for custom plugins.

What's New

- **PxrCurvature** – A new [pattern](#) for rendering the resulting curvature of geometry is included. Useful for masking and other effects to show wear on edges or outlines
- **PxrSurface for Mari** – A new [material plugin](#) is provided to preview PxrSurface shading while painting in The Foundry's Mari
- **RenderMan integration with Hydra** – [hdPrman](#) is now available from the USD Github. Note that USD information is *specific to hdPrman* and available from the Github site alone. For help, take a look [here](#).

Miscellaneous Changes

- Improved performance when large numbers of trace membership groups are used
- Silence an error message from PxrCamera if the "matte" parameter is set to an empty string
- The simplified PxrPerspective projection now supports the same fovEnd parameter (for motion blurred zoom) as the full-featured PxrCamera
- We now support displacement material assignment/swapping edits in interactive rendering
- We now support changes to min/max samples and variance during interactive rendering
- Replacing inputAov with utilityPattern on PxrMarschnerHair. utilityPattern supports an arbitrary amount of connections (like on PxrSurface)
- Improved automatic selection of reference instances to favor instances inside the viewing frustum. If a dicing camera is defined it is used in the selection of the reference instance in place of the rendering camera. Improved off-screen dicing to not under-tessellate portions of instanced geometry when the instance happens to be partially outside of the screen

Fixes

- Fixed bug causing bxdf parameters with arraylength > 1 to potentially cause a crash or not execute properly
- Fixed a bug in PxrUnified where volume scattering would require larger maxPathLength values than expected, and would appear in the incorrect LPE. PxrUnified should now yield renders that are more consistent with PxrPathTracer
- An issue that may cause NaNs to appear or produce different results when using the aaOceanPrmanShader pattern has been fixed
- Fixed a bug where recovering from a checkpoint when display filters are active and asrgba is set could reload some channels incorrectly
- Fixed a crash with batched OSL when trying to compile a texture shadeop with uniform T-coordinate inputs
- Fixed a bug when recovering Cryptomatte images when manual cropping or autocropping is in effect
- Fixed a bug when validating the data window of recovered images against the expected data window due to cropping
- A bug that prevented Procedurals with displacement shaders to displace has been fixed
- A bug that caused blocky artifacts when using PxrMayaFluidShape's edgeDropoff parameter has been fixed. In addition, density values are now more accurately retained when used as the opacity input
- Fix a hang in PxrCryptomatte when trying to recover a checkpoint that does not exist
- Addressed a crash when manifolds were connected incorrectly in a pattern graph
- Fixed a crash that could occur when using overscan
- A bug that caused points with velocity in an alembic file to not motion blur in some cases has been fixed

API Changes

- Added projection DOF query from RixRenderState::GetOption as Ri:DepthOfField
- Add a new function to lighting services: UpdateWithResults, which enables an integrator to pass lighting results, transmission terms and bxdf evaluations back to lighting services to help with learning
- Added a parameter to the volume aggregate and lighting services to provide max distances when selecting lights for ray segments. Lights and parent clusters beyond this distance from the segment will not be considered. This allows for a potential speed up of equiangular sampling
- Added a parameter to lighting services so that it's now an integrator's responsibility to provide space for the storing of light selection probabilities

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
- 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
- PxrBakePointCloud 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 a new mesh light on existing geometry during IPR results in double geometry
- Motion blurred geometry does not motion blur normals when deformed
- 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.