

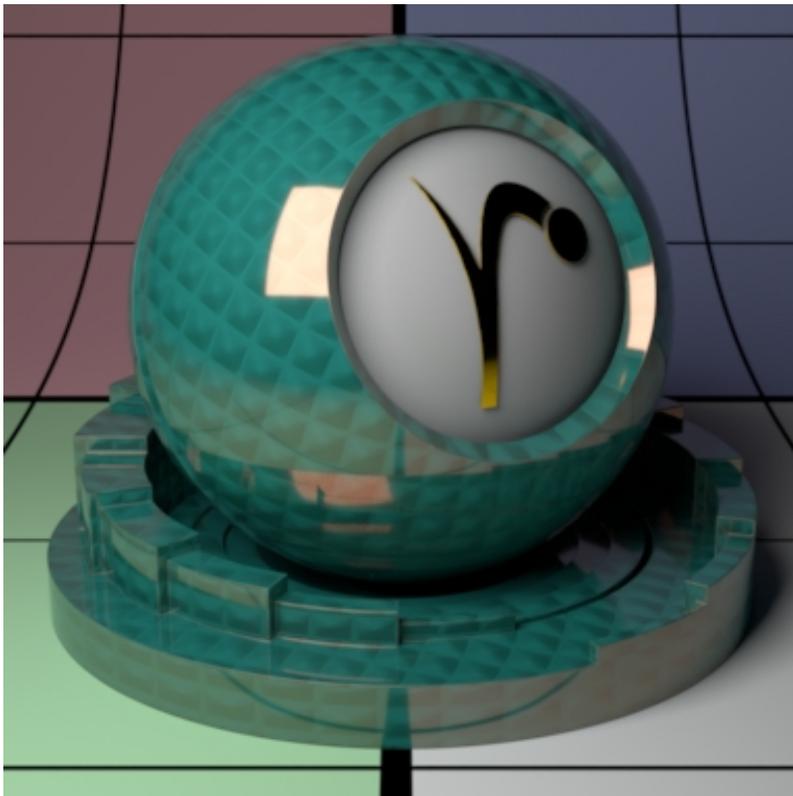
# RenderMan 21.6

## Welcome to RenderMan 21.6!

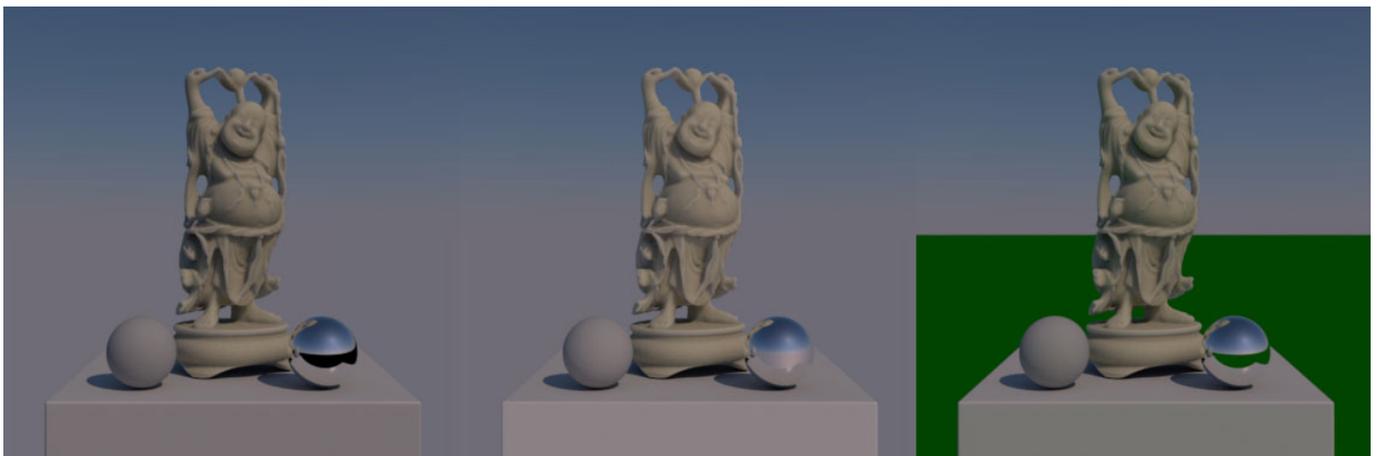
This release introduces the following improvements, fixes, and miscellaneous changes.

### New Features

- Added a control for "thickness" for Physical Clear Coats in PxrSurface.



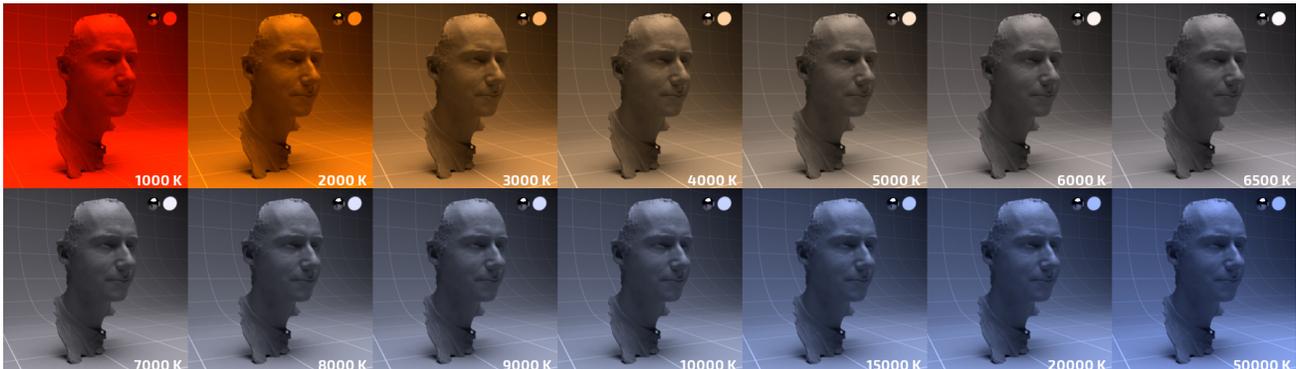
- Added new ground selections to the PxrEnvDaylight simulation:
  - Legacy: Old behavior
  - Clamped Hemisphere: "Smear" horizon values on the virtual ground plane
  - Diffuse Ground: A fake infinite ground plane with diffuse color



- The /prman/deepcomp/flagvolumes and Option "deep" "int flagvolumes" controls now interpret the value 1 as an "auto" mode. This is on by default, but off if generating any deep images in DTEX format. Use 2 to force it always on.
- Ray bias now affects transmission (shadow) rays as well as behavior of indirect rays related to it. Note that you must disable Auto Trace Bias in the global scene settings for this to affect your scene with your Trace Bias set. In scenes where dark areas may be caused by intersecting geometry this may reduce or remove the artifact.

## Miscellaneous Changes

- The Path Traced Subsurface Scattering control Zero Scatter Attenuation has been replaced by an IOR parameter causing total internal reflection. Note that your behavior may be different. Increasing the value from the default 1.0 may reduce fireflies in the SSS result, values that are too high may lose energy.
- There are significant performance increases for UDIM textures when many textures are in one directory
- The Alembic procedural now respects the visibility attribute.
- We now warn if a deep image has a pixel filter on alpha that is a poor match for the pixel filter on the other channels. This can lead to noise across the deep image.
- GetBuiltinVar queries of k\_lpeState or k\_wavelength no longer set the result pointer to null during displacement, cached presence /opacity, and baking.
- Added intensityNearDist parameter to [PxrMeshLight](#).
- Improved live-render edit speed in cases where several LPE-based AOV's were being used.
- For user-debugging purposes the RLFInjector can now be made to print out the exact Attribute Identifier path at each binding site by adding "-printaip 1" to the rif arguments. Rif arguments are set up in RfM via RenderMan\_for\_Maya.ini (see existing doc re -filtergeom). The path that is printed out is precisely the one that Xpath is matching against in a Dynamic Rule and can also be used (in the case of the new alembic-file-specific rif's) in the Static Binding table.
- RenderMan will now abort when attempting to recover non-incremental renders when bucket order is set to anything other than "vertical" or "horizontal".
- In OSL, getttribute() of the "context" object can be used to access the members of the RixShadingContext scTraits structure "eyePath", "lightPath", "primaryHit", "missContext", "reyesGrid" and "shadingMode".
- Extended widget types for OSL nodes loaded at startup: 'fileInput', 'assetIdInput', 'filename' (non-standard), 'text', 'popup' and 'boolean'.
- Fixed a crash when parsing OSL shaders with int parameters using 'min' or 'max' metadata.
- Improved quality of deep output compression when a deep id channel is present.
- Increased the upper range for RixKtoRGB, below we show new values above 10000



## Bug Fixes

- Fixed a bug where scalar displacement was backwards when dicing an instanced master and the orientation is either "lh" or "rh".
- Fixed bug causing potential crash when using aggregate volumes.
- Fixed a bug where using alembic archives (AlembicProcPrim) with motion blur would crash on Windows.
- Fixed a potential issue with infinite lights in very large scenes.
- Fixed bug that could cause a crash when using a gprim with very many vertices and/or primvars.
- Fixed bug causing reduced thread efficiency and potential hangs when rendering many things with different grouping membership, enabled lights, or enabled lightfilters state.
- Fixed a potential crash when a hierarchical subdivision mesh specified an invalid face index in a face edit.
- Fixed bug causing a very small chance of crashing when rendering a scene containing both infinite and local lights.
- Some fixes for subsurface scattering with the PxrVCM (and PxrUPBP) integrator(s) that may have cause them to render black or with artifacts.
- Fixed potential for crash or extra memory usage when rendering a large polygon mesh having non-quad faces and \_\_faceindex values for per-face textures.
- Fixed tile-shaped artifacts on materials with both subsurface and diffuse transmission.
- Added the correct attribute for importanceMultiplier to the PxrMeshLight
- Fix of incorrect normals of instanced NURBS patches with non-homogeneous control points. Speedup of ray tracing of some NURBS.
- Fixed a bug that could cause incorrect interpolation of primitive variables in some cases of coarsely tessellated displaced surfaces.
- Fixed a possible issue with alpha values on matte objects and volumes.
- Fixed a bilinear interpolation bug in the EvaluateAtLimitMultiple() call of the SubdivEval API

- Fixed a bug where OSL texture() shadeop would return random values on missing textures instead of the "missingcolor" and "missingalpha" or "fill" values.

## Known Limitations

### RenderMan Pro Server

- Full LPE support is only available with the PxrPathTracer and PxrVCM integrators.
- Light Filters in the PxrVCM integrator do not affect or shape photon emission.
- The Cone Angle parameter for lights is incorrect, it does not match the input angle, this will be fixed in a future version.
- The PxrAovLight does not work properly with PxrUPBP.
- Analytical lights placed inside volumes may yield artifacts when made visible to the camera. As a workaround, the light camera visibility should be turned off, and a geometry with a similar shape should be used (visible to the 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.
- Instances are not supported for baking.
- 3d baking: no direct bake-to-ptex support.
- PxrBakePointCloud cannot directly render ptex.
- No RixPTC/pointcloud API (so PxrBakePointCloud cannot read ptc files).
- 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.
- Camera visibility changes are not respected during Live Rendering.
- For PxrUPBP, If the light source is inside a volume, that volume needs to be defined as Volume "box"
- For PxrUPBP, To get a volume caustic, the object casting the caustic needs to have higher intersectpriority than the volume.
- For PxrUPBP, Overlapping heterogeneous volumes are not working yet. (However, overlapping homogeneous volumes do work.) This will be resolved in the future.
- PxrPortalLight may yield artifacts when generating photons for PxrVCM / PxrUPBP. This will be fixed in a future release.
- 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 have mesh lights attached to them.
- 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.