

RenderMan For Katana 21.0

- [Welcome to RenderMan 21.0 for Katana](#)
- [What's New](#)
- [Additional Changes](#)
 - [Miscellaneous Changes](#)
- [Known Limitations](#)
 - [Live Rendering](#)
 - [Other limitations](#)

Welcome to RenderMan 21.0 for Katana

RenderMan for Katana (RfK) capitalizes on the changes made for newer versions of Katana and continues full support of the latest RenderMan ProServer 21.0.

We're especially pleased to have improved stability in Live Rendering and simplified workflows using macros for repetitive tasks. Light and material edits during Live Rendering should be more robust with complex networks and perform better overall. Adding analytic lights is a now breeze with a single click or shortcut. You can even add lights during Live Rendering!

We also introduce GafferThree light packages to handle the above improvements as well as direct control of light visibility.



Please note that RenderMan for Katana no longer includes the Reyes renderer. RenderMan for Katana is now based on modern and improved pathtracing technology and C++ plug-ins. Scenes using old RSL lights, displacements, and imager shaders will fail to render or render black because of this change.

This current release offers support for:

- Katana2.0v5
- Katana2.1v2
- RenderMan ProServer 21.0

Note that we no longer support Katana 1.x versions.

Please see the release notes below for all the new capabilities and known issues!

What's New

Better Defaults

We've removed much of the guesswork by setting appropriate defaults. These defaults are designed for common scenarios and performance benefits. Rendering should give you what you expect--a physically correct and intuitive result!

New Material!

PxrSurface is introduced to users from actual Pixar Animation production! The material is a physically based but flexible material with options to layer. PxrLayerSurface handles all of your layering needs in combination with PxrLayer nodes. Artistic control is maintained for all your art direction needs!

All New Lights!

All new lights and lighting tools are included in this release! You can now use the same lights and filters as Pixar Animation Studios. Useful features and performance enhancements are all included. We've also included an improved Portal Light for those difficult interior situations.

Texture Baking

It's now possible to bake patterns into 2D textures or point clouds! This allows artists to simplify complex networks to a single result for rendering. This can improve performance by reducing the cost in evaluating these patterns. Point clouds may be converted to ptex as a post process.

Denoise Improvements

Continued improvements in performance of the Denoise feature improves speed and quality. Also, using Denoise with CUDA capable GPUs can reduce Denoise processing times up to 60 times compared to CPU. Using the GPU requires CUDA 7.0 (compute capability 2.0 or later) and a [capable graphics card](#) with support. GPU denoising is enabled by default with an automatic fallback to CPU denoising if no compatible device is available.

Occlusion Integrator

PxrOcclusion is a new integrator that can be used to create non-physical renders for later compositing. Artists will find this very useful in adding details, knocking back unwanted elements, and creating occlusion passes for integration into background plates. No material or light changes are necessary, just change your integrator and render!

PxrDirt

PxrDirt is a new pattern made for artists wanting to create useful effects like procedural dirt maps or weathering. With useful controls and the new PxrBakeTexture pattern, artists can quickly generate and use results instead of hand painting maps. Combine this with the layering abilities of PxrSurface to bring your shading to another level!

Improved Holdout Workflows

Production feedback has helped us further refine our powerful holdout features. These now operate using all available features to the renderer including correct and complete light transport participation.

Physical Camera

You now have the option to define an arbitrary plane of focus using the PxrCamera. This can be used to create special effects and fields of focus to drive your story and focus your viewer in ways previously impossible.

OSL Patterns are Native

OSL Patterns no longer require added work to render in RenderMan and are handled natively. Using your own patterns has never been this easy or fast.

Additional Changes

- txmake no longer manipulates input images to conform to the OpenEXR latlong specification. An input image is converted directly to an OpenEXR format latlong texture with no image manipulation. The -extraargs exrlatlong flag is no longer needed or supported.
- Originally introduced in 20.6 as an option, PxrPathTracer now tracks the diffusedepth and speculardepth separately from each other based on which lobe of the BSDF is sampled. Previously rays that have both diffuse and specular contributions increment both the diffuse and specular depths. The new behavior is more intuitive and produce expected results when adjusting the trace:maxdiffusedepth and trace:maxspeculardepth attributes. This comes at the cost of, in some cases, effectively doubling the number of indirect ray bounces which can cause look differences and/or increase render times.
- There are now PrmanGlobalStatements and PrmanObjectStatements nodes to replace PrmanGlobalSettings and PrmanObjectSettings. Updates to these nodes are handled internally without requiring an update script.
- Camera visibility is now a parameter in the gaffer for Pxr lights.
- You can now add a mesh light by simply selecting an object and creating a "PxrMeshLight"
- You can now add lights during live rendering (note that in Katana 2.1 a new light is not automatically activated, you'll need to turn it on manually). See Known Limitations below.
- Material edits will update when only the material is marked as live. I.e. if you have a large number of objects with the same material you no longer need to mark all the objects live in order to update that material interactively. This will only work for "non-rebinding" edits. E.g. you can change a color but not change out a shader or enable a layer.
- Removed the ability to apply any scale PxrDome & PxrDistant lights
- Bxdf "PxrBlack" is assigned to mesh lights by default. This can be overridden by assigning a Bxdf to the light or the geometry source. PxrBlack is assigned to prevent secondary scattering from the light itself.
- Hierarchical material edits no longer incur an over-abundance of updates to Prman.
- Light muting is now handled natively using Ri overrides; RfK no longer uses 'intensity=0' as a mute edit workaround.
- Katana can now dynamically load custom integrators.

Miscellaneous Changes

- Added the darkfalloff hider parameter to PrmanGlobalStatements in the raytrace hider section.
- PrmanLightFilterCreate now supports light filter sharing! You can now either create light filters parented under a light or light filters that are referenced by any number of lights by setting the value of the "Shared Light Filter" parameter.
- Portal light no longer inherits dome light transforms
- The RfK package no longer contains deprecated files pertaining to RMS and Slim.
- No longer support light detail tag.
- Plugin directories have moved to a new location:

```
Previously
$RFTREE/lib/katana/Resources/PRMan20
$RFTREE/lib/katana/Src/PRMan20
Now it is
$RFTREE/plugins/Resources/PRMan21
$RFTREE/plugins/Src/PRMan21
```

Known Limitations

Live Rendering

- Light linking edits are not currently supported.
- Lights can be added during live rendering however, due to a bug in Katana 2.1, a newly-added light is not automatically 'live' and does not inherit its live state from the parent. When adding a light with Katana 2.1 you will need to manually activate that light in the Scene Graph. Newly added lights work correctly in Katana 2.0.
- Adding mesh lights during live rendering creates duplicate geometry, upon re-render this is solved.
- Changes to light filters are not processed unless the associated lights are also marked 'live'.
- Alembic geometry mesh light transform edits (i.e. Transform3D changes) are not updated correctly during live rendering.
- ROI can now be disabled during live rendering however it will not take into account the crop window settings
- Muting a PxrDomeLight with PxrPortalLight children does not successfully mute the portal lights.
- "Disable Local Assignment" does not work during live rendering with parameters of OSL shaders (e.g. PxrLayer and PxrLayerMixer).
- Disabling and re-enabling layers in PxrLayerMixer can cause instability in Live Rendering.
- Texture changes are not currently supported during live rendering.

Other limitations

- User attributes must have explicit "type" and "value" attributes set.
- If an object is assigned a material that has been disabled or does not exist the geometry will not render at all.
- If you are rendering to 'it' the Monitor tab must be open for the Render Log tab to receive output from Katana and prman.
- Although direct non-identity scaling has been removed from the PxrDomeLight and PxrDistantLight it is still possible to get an indirect scale via a constrained or inherited location. If the indirect scale is negative the Pxr light will flip to a reverse orientation.
- The "relative" transform mode for PxrBarnLightFilter, PxrCookieLightFilter, and PxrGoboLightFilter is not yet supported.
- RenderMan light filters are fully supported by RfK however due to restrictions in the current GafferThree API they must be created using the PrmanLightFilterCreate macro.
- PxrPortalLights are fully supported however they must be created in a separate GafferThree node with a reference created back to its dome light.
- Bake renders only work as Disk Renders. The render will fail for Live and Preview Renders.
- Those who were on the beta will find that any scenes using the 'Live Render' quality preset on PrmanGlobalStatements will get an error that says "Live Render". This is because the preset name has changed to "Draft". Change the Quality Preset value to "Draft" and the error will go away. The holdout_teapot.katana example scene also has this problem.