

RenderMan For Katana 22.0

- [Welcome to RenderMan 22.0 for Katana](#)
- [What's New](#)
- [Additional Changes](#)
 - [Miscellaneous Changes](#)
- [Known Limitations](#)
 - [Live Rendering](#)
 - [Katana Limitation](#)

Welcome to RenderMan 22.0 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 22.0.

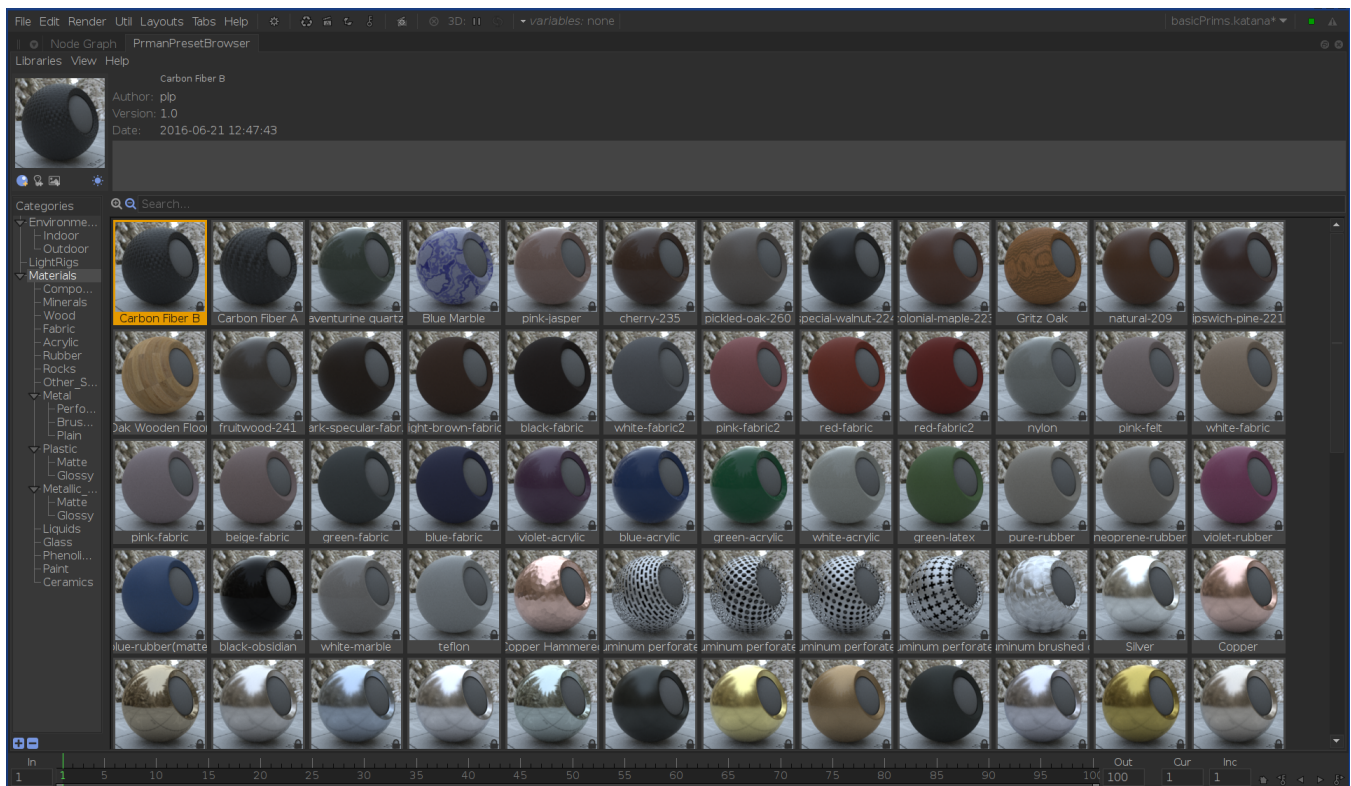
We're excited to have improved Live Rendering. All manner of changes and edits can be made during a Live Rendering session. Waits are minimal and results are stunning, the renderer will continue to refine your image continuously should you take a coffee break and pick up where you left off on your return. We've worked hard to avoid making the artist restart the render to see updates and stability is improved.

This current release offers support for:

- Katana 2.6
- Katana 3.0 (New!)
- RenderMan ProServer 22.0

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

What's New



The New Preset Browser

- Preset Browser for Materials and Environments
- Multi-threaded Katana Scenegraph traversal (faster time-to-first-pixel)
- Multi-threaded Alembic_In_Prman (improves speed when reading multiple alembics)
- Live add, delete and transform for geometry
- Live updates for light-linking
- Live light deleting
- Instanced lights
- Light array support
- Improved logic for PxrLayerSurface and PxrLayer. It's now more intuitive to enable and disable lobes.

- PxrMeshLight can be assigned to geometry with a MaterialAssign node
- Cone angle now visualized on Viewer Modifiers
- 64-bit object IDs, consistent across disk/live/preview renders
- Shadow Linking is supported through the GafferThree Linking tab.

Additional Changes

Miscellaneous Changes

- Materials set at instance source locations are ignored
- Dicing cameras are now specified as a camera's scenegraph location. The "worldcamera" and "framecamera" options are no longer supported. This also removes the need to add the dicing camera to a RenderOutputDefine.
- PxrSphereLights with cone angles are now facing the correct direction
- RfK now traverses past "geometry primitive" locations. So the children of a subdmesh/nurbspatch/etc location will now render.
- Instanced primvars are considered master attributes and cannot be altered per-instance. Use user attributes instead on instances, which allows for shading variation and flexibility.
- Leaf level instancing now requires that locations with the instance.ID attribute have no children. RfK will not traverse past a location with instance.ID set.
- RIB Include has been removed
- Integrator settings are now configured with a new node called PrmanIntegratorSettings. This node automatically picks up any integrators with Args files on RMAN_RIXPLUGINPATH.
- "insertWhen" directives for config.xml have changed.
- Parameter Layout has been removed from PrmanGlobalStatements and PrmanObjectStatements
- Attributes in PrmanObjectStatements are now split into either an "attributes", primAttributes or "primvars" group depending on whether the attribute can vary per instance.
- The grouping membership attribute no longer automatically prepends the value set with a "+". This is to allow fuller functionality to match the RenderMan attribute syntax. That syntax allows: "" - add to the membership specified at a higher level, "-" - remove from the membership specified at a higher level, "" - replace the membership.
- There is a new node called PrmanCameraSettings for setting camera settings
- There have been a few changes related to cameras and PrmanGlobalStatements: -
 - dofAspect was a hider setting, but is now found under prmanCameraStatements.camera.depthOfField.dofAspect.
 - Projection plugins can be created as materials assigned to /root. They are no longer available through PGS.
 - backgroundColor can be created as a "PxrBackgroundDisplayFilter" assigned to /root. It is no longer available in PGS.
 - The settings in the camera motionBlur section have changed and could not be automatically remapped. It is found in the [PxrCamera](#) projection node.
 - It is no longer necessary to create a camera with frameBeginCameraPath. All cameras in the scene will be created.
- No longer supported:
 - "prmanGlobalSettings.plugin.instanceSourceMode"
 - "prmanGlobalSettings.commandArguments.numThreadModeProcs"
 - "textures" attribute group
 - The PrmanLightFilterCreate macro has been removed in favor of creating light filters in GafferThree
 - camera fov motion blur
- Added new offscreen displacement bound primAttribute, please see: [displacementbound:offscreen](#)
- Arbitrary clipping planes are now supported
- The global traversal setting "flattenInstanceSources" indicates whether instance sources should be flattened internally in the renderer. By default this is on which is the most efficient and flexible for the common case. Disabling this option allows deeper nesting of instance sources with the caveat that RenderMan only supports 4 levels of nesting, and it prevents per-instance variation on instance source children. This can be overridden locally by the local PrmanObjectStatements parameter - "traversal.flattenInstanceSource" See the documentation on [Instancing](#) for an explanation of benefits and limitations
- OpenEXR outputs containing "Ci,a" as the first four channels will now write out an EXR with R,G,B,A as the first four channels (i.e. "asrgba" will be enabled for all beauty renders).
- Added extensibility to RfK's procedural argument parsing. Now RfK will look for an attribute "argsConversionFunc" which is the name of a attribute function for custom argument conversion
- Add support for prman command line options "woff", "logfile", and "loglevel"
- RfK will now use the basis from the alembic file unless overridden by the curve.basis setting in PrmanObjectStatements (NOTE: only supported with AlembicInPrman initially for 2.6)
- The default type in PrmanOutputChannelDefine is now "color"
- Live render edits to ROI / cropWindow now use the intersection of the two attributes
- Alembic cameras now use the camera settings from prmanGlobalStatements
- OSL parameters now have connection type checking for the inputs and outputs of PrmanShadingNode
- OSL parameters support a "tag" hint in the metadata to allow users to specify a connection type for the parameter. For multiple types on outputs, use a "|" to delimit the options (eg "struct|manifold")
- RfK now supports connections between any color, normal, point, and vector parameters

Known Limitations

Live Rendering

- Creating a mesh light from existing geometry will duplicate the geometry in-render. Restart the render to remove the duplicate.
- Adding a portal light to a dome light doesn't delete the dome light
- Cannot change a geometry primitive type during live rendering (e.g. from NURBS to polymesh)
- Edits to mesh light parameters require that the geometry location is enabled in the live render working set

- When assigning a material to a Scene Graph location, that location must be enabled in the live render working set
- Cannot create new volumes or enable existing-but-disabled volumes during live rendering.
- The Katana Viewer will crash when muting a light with a light filter reference with "Lock Size to Cone Angle" enabled

Katana Limitation

- When rendering to "it" from Katana, do not stop the render from "it", abort the render from Katana. Your Katana session may freeze for a time if you abort from "it". If you make this mistake you can restore Katana to operation by terminating the prman render process manually. This will be fixed in a future version. You can also avoid this entirely by rendering to the Katana Monitor.
- Light Linking volumes is not currently supported
- We do not receive live render edits from Katana for nodes added at the end of the node graph, right above the Render node. If a no-op node (e.g. Merge) is inserted above the Render node and the node is added above that then the edit is received.
- Any live updates will cause interactive motion blur to be disabled. The render must be restarted.
- Support for light arrays of mesh lights is coming in a future version of 2.2
- Render is not correctly terminated when errorHandler = 'abort', this only aborts traversal, not the render
- PRMan error handlers are not fully supported yet.
- Instanced lights with filters using the "Light Filter" coordsys have an incorrect transform. The workaround is to promote the light filter to a shared light filter using a light filter reference.
- There are a few live render limitations in Katana 2.6 that have been resolved in Katana 3.0 based on the improvements to 3.0, typically limitations with live working sets and adding/deleting locations in 2.6