Migrating from 21 to 22

This page will help you with the major changes made to operating RenderMan in Maya. RfM2, as we call it, has been extensively re-written to avoid redundant menus, reduce the number of clicks you need to make, and present RenderMan options as native Maya menu items. This means users can jump right into RenderMan with what they already know.

Many options have been removed from RenderMan, these decisions were to improve performance of the renderer and the workflow for artists, as such, the options provided are to help guide the user to the correct workflows and reduce headaches. As always, avoid using optimization controls (like those found in integrators and some materials like PxrVolume) to achieve a specific look. These techniques may be improved or modernized in later versions and you will not have access to them.

Please note that the original RMSTREE variable is no longer used and instead we use RFMTREE for the Maya plugin environment variable. You can find details here.

## Main Changes

Shading

- RenderMan 22 maintains the same shading options supported in RenderMan 21. However, some materials have been removed since their deprecation in RenderMan 21, of note are the removal of:
  - LM Materials
  - PxrGlass
  - PxrSkin

Please use the available BxDFs in the Hypershade menu.

- You can assign, manipulate, and import entire shading networks while rendering in IPR mode through the viewport or "it"
- When applying a material, we create a Maya common material for OpenGL viewport viewing. The RenderMan materials plug into a RenderMan option on the shading group node. You can disable this in the preferences.
- In the initial release, we do not render a material swatch or Hypershade preview, you will get an icon. Instead we encourage you to use the VP 2.0 rendering by selecting RenderMan from the "Renderer" viewport panel menu. You may playblast to this window if scale is set to 1.0 in the Maya settings. (added in 22.6)
- Material overrides on instances is now straightforward as everything is treated as an instance in RenderMan now.
- OSL materials with appropriate metadata can auto-populate a UI, this removes the need for a separate .args file. Examples are the PxrLayer nodes.
- We don't support the color management options in Maya in the initial release
- Texture paths using environment variables in older scenes (such as $HOME) should be replaced with the new token system to correctly render.
- We don't support the material viewer in the initial release (added in 22.6)
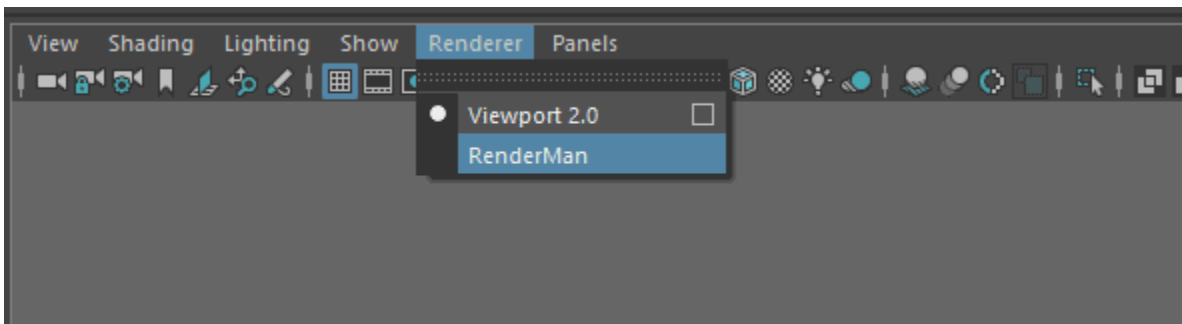
Render to the Viewport

Geometry

- You can model interactively while rendering in IPR mode through the viewport or "it"
- You may import or update assets while rendering in IPR mode
- We still support all the common geometry types, however, support for hierarchical subdivision surfaces has been removed. If you have a scene with one of these, it's an old scene isn't it?
- We no longer require users to add RenderMan attributes, instead the transform (attributes) and shape (primvar) nodes contain valid parameters.
  - You may set a widget type on the nodes through the Attribute Editor now.
 - You can find examples in the rmanGlobals.json file
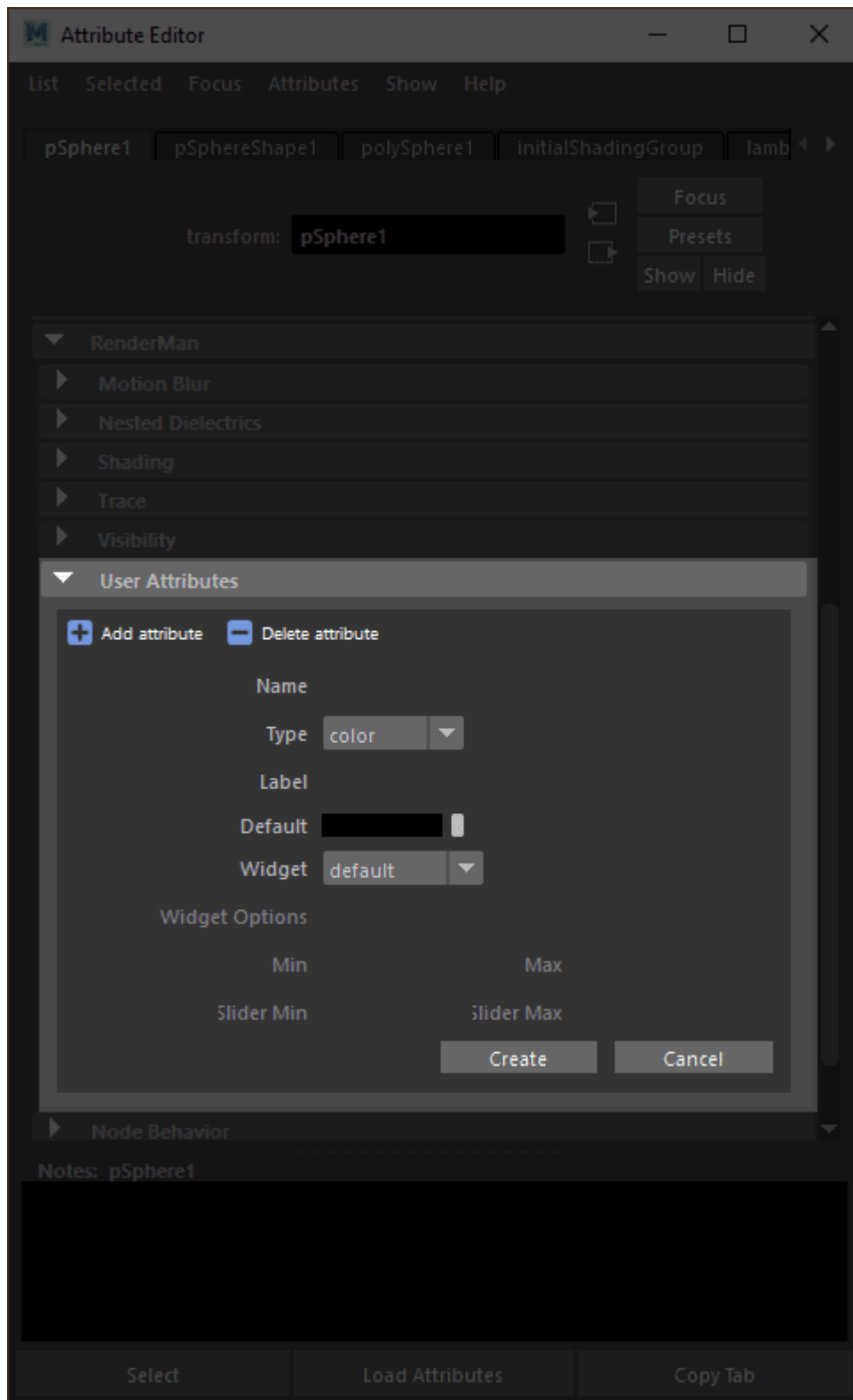
- We no longer render camera-facing (billboard) curves, instead they default to rounded curves. If you provide normals you will get a ribbon. Rounded curves perform well and are quite realistic, hence the default.
- Workflow for setting __Pref (texture reference objects) has been simplified, you'll find the controls on the shape node.
- We do not yet support Bifrost

Lighting

- Lights remain the same from RenderMan 21, however, you will find in your imported old scenes, the HDRI in dome lights is reversed, remove the negative scale we automatically applied in the previous version.
- You can light interactively when rendering to the Viewport or "it" in an IPR session

Menus and Scripting

- The projection editor is gone, please use the Maya camera projections instead.
- Xpath has been removed.
- The Dynamic Rules menu has been rewritten
- The Light Lister is removed, use the Maya attribute spreadsheet now.
- Tcl support is removed
- The Preferences and Render Settings menus have been reorganized and simplified.
- Most .ini files are now .json files, these can be customized for your pipeline, please make a backup of the original .json files.
- Many menu fields allow use of pre-built tokens, just right-click in the field, the Workspace tab makes use of this.
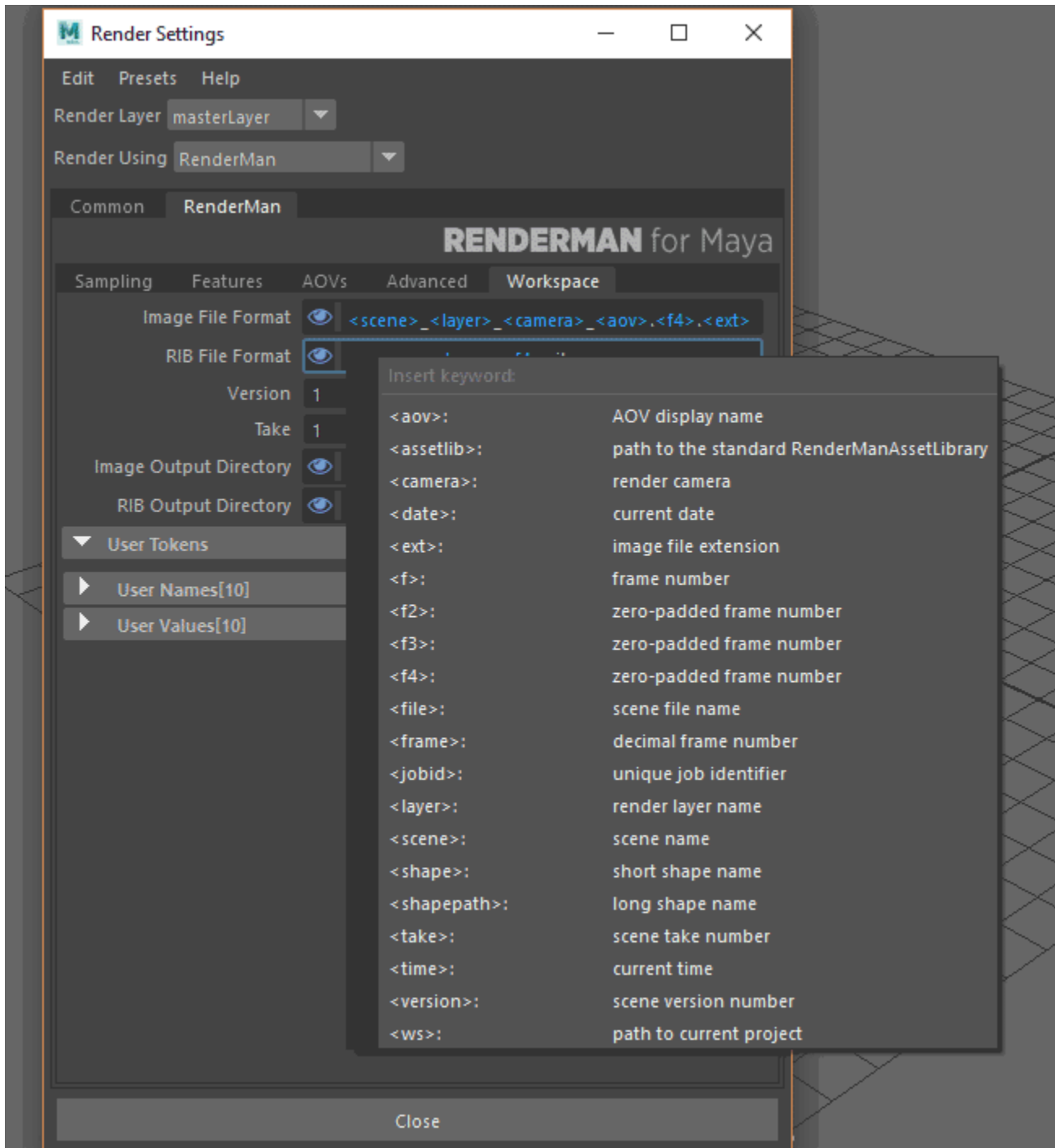
Output

- The AOV menu has been rewritten
- Denoise has been moved to the AOVs/LPE tab of the Render Settings. It is activated per-LPE now.
- Choose your file type from the Display Drivers in the AOV tab, we default to EXR and recommend this setting.
- The number of available Display Drivers has been reduced to the most commonly required file types.
- Note we added a dPcameradtime AOV that contains camera blur separately from the dPdtime AOV (which no longer includes camera blur)
- We no longer write to RIB for interactive rendering sessions, since we render to memory it's much faster and easier to manipulate. However, save often when making risky changes as crashes may corrupt a Maya session. Our Version and Take system in the Workspace Tab will help you keep track of your refinements.
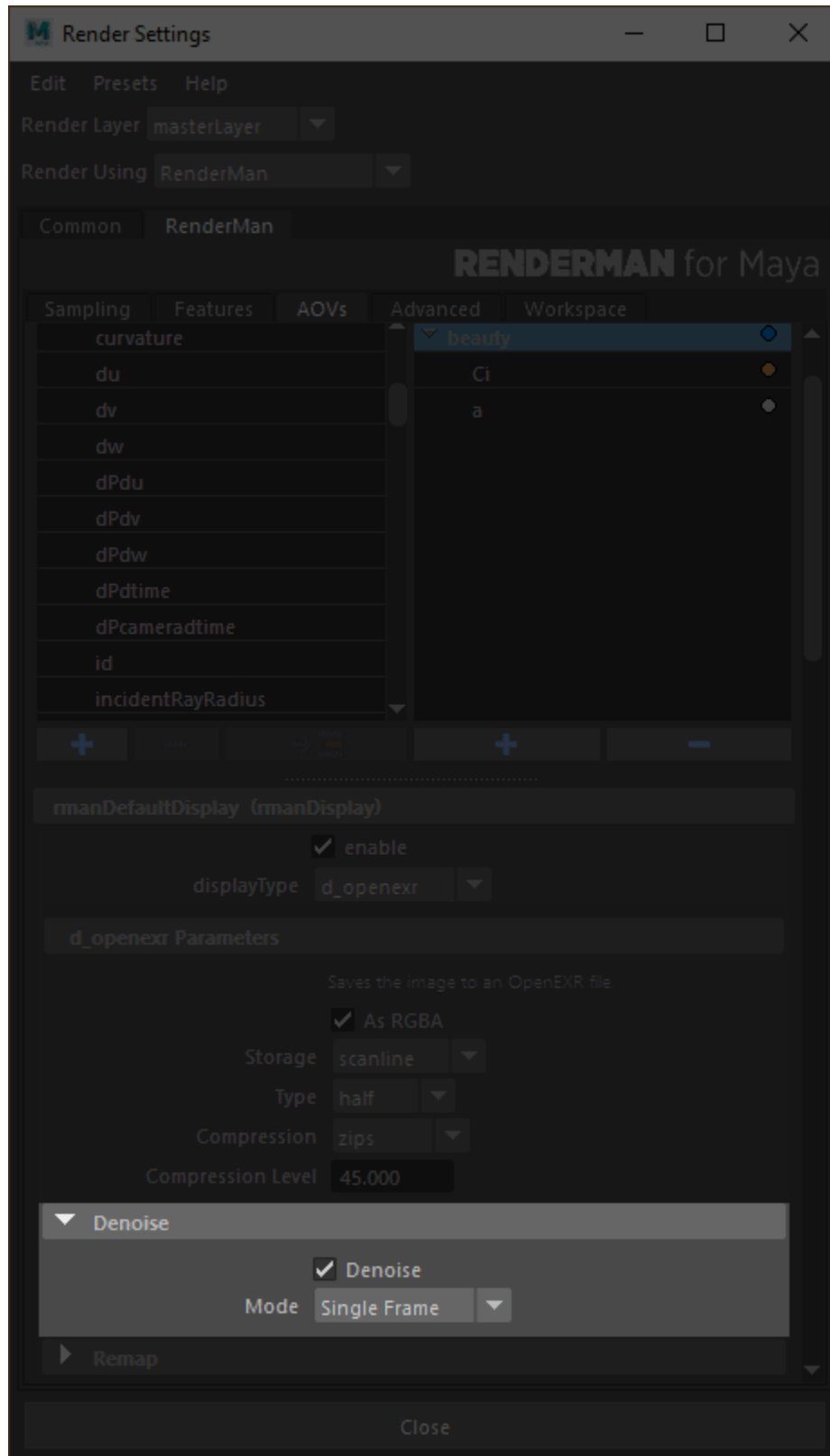
Individual Scripts

Older scenes may have issues with migration, areas of note are the PxrRamp, subdivision settings, and AOVs. The following python script can correct these or they can be run individually depending on your scene's needs or included features from RenderMan 21. We do not run this script on import to avoid performance problems loading large or complex scenes. Instead these are provided to the user to use as needed.

```
import rfm2.utils.scene_updater
rfm2.utils.scene_updater.update_scene()
```

Older Maya scenes that had PxrRamp nodes in it may break. To fix, you can run this python script in the script editor –

```
import rfm2.utils.scene_updater
rfm2.utils.scene_updater.update_pxrramp_nodes()
```

Older scenes may have incorrect geometry settings and can be updated specifically using the below python script –

```
import rfm2.utils.scene_updater
rfm2.utils.scene_updater.convert_subdivs()
```

Older scenes with AOVs may fail to show in the new UI, the following script can solve this issue –

```
import rfm2.utils.scene_updater
rfm2.utils.scene_updater.convert_displays()
```

Tractor

> ⚠ For Tractor 2.2 and prior

Due to changes in RenderMan, we are providing some helpful files here for rendering with Tractor on a farm.

The shared .*. envkeys should be copied to the "config" subdirectory in the Tractor install. The TrEnvHandler.py file should be copied to lib/python2.7/site-packages/tractor/apps/blade/ subdirectory (if you're copying to Windows the path should be lib/python2.7/Lib/site-packages/tractor/apps/blade) Restart the blades and render. **Be sure to back up the original files.**

Exporting RIB

If you need to export a RIB file using a script, you can use the below python script. Note that this export will follow whatever settings you have provided in the Render Settings.

```
import rfm2.render
rfm2.render.frame('"-ribFile /var/tmp/foo.rib"')
```