

XPU Features and Limitations



The first phase of XPU delivers a complete feature set for look development. However, some features are not supported yet but will be as XPU matures to replace RIS. Here is a high-level summary of those features supported currently in XPU

General

- The behavior of XPU when it runs out of memory on your GPU: XPU will crash if it runs out of memory on your GPU. If you are running XPU inside of Maya, Houdini, or Blender, this will bring down your application as well. Be sure to save often. We are going to work to mitigate the two biggest causes of this condition.
- If you have more than one GPU, we allow you to specify which GPU to use with XPU but do not yet support the ability to use more than one GPU at the same time.
- XPU is not yet supported on the Mac.

DCCS

- Warnings on the console when rendering to XPU from Katana. There are some warnings you will see on the console about a bad DSO being found for the socket display driver. You can ignore these.
- Picking from "it" does not relay the selection back to RenderMan for Maya
- RenderMan for Maya supports batch rendering to XPU (PxrPathTracer integrator)

Shading

BxDFs, Displacement & Patterns	RIS	XPU	Notes
PxrSurface	✓	✓	Some subsurface modes are not available
PxrLayerSurface	✓	✓	

PxrDisneyBsdf			
PxrMarschnerHair			
PxrConstant			
Lama			
Volumes			<i>Heterogeneous volumes are supported via aggregate volumes</i>
OSL Patterns			<i>PxrDirt and PxrCurvature are not supported</i>
C++ Patterns			
Baking			
Point Clouds			

- Custom Bxdf or Displacement plugins are unsupported.
- MaterialX Lama support will be implemented in a future release.

Patterns & Textures

- There is no support for custom C++ plugins. Patterns written in OSL are supported.
- PxrSeExpr is not supported.
- PxrDirt & PxrCurvature require the trace() OSL call, which is not yet supported.
- Point clouds are not supported.
- PxrBakeTexture and PxrBakePointCloud are not supported.
- Textures with a non square pixel aspect ratio are not supported.

Volumes

- In XPU, the expected workflow for heterogeneous volumes is via aggregate volumes. XPU has limited support for non-aggregate volumes: only homogeneous volumes are supported.
- XPU fully supports volume primitives of type "box" and "blobbydso" (ImplicitField plugins); however, VDB volumes are the optimized preferred choice, and non-VDB volumes will incur a cost for upfront internal conversion to VDB.
- Interior volume aggregates are not currently supported.
- Single scattering support in PxrSurface is not complete in XPU.
- Deformation motion blur via velocity is supported, but currently requires specification of the velocity grid directly to impl_openvdb.so as the third string parameter. We will also only support velocity that comes directly from a grid, and has not been computed or altered by arbitrary shading. We anticipate lifting some of these restrictions in a future point release.
- Joint sampling is not yet supported, so the convergence of volumes with complicated lighting that requires joint sampling will not be as performant as RIS on a per sample basis.
- Instancing of volumes is not fully supported.
- Certain editing operations on aggregate volumes are not currently supported; for example, editing the density of a volume material might not correctly update the rendered result in all cases. Restarting the IPR session should fix this.

OSL

- In XPU, the first argument to the OSL getbuiltin() shadeop is ignored. You may continue to specify a first argument in order for your shader to remain compatible with RIS, but the distinction between "primvar", "builtin", and "attribute" that exists in RIS does not exist in XPU.
- getattribute() calls asking for geometry primvars (in RIS, getattribute("primvar")) are fully supported.
- getattribute() calls targeting "builtins" (renderer computed quantities that are not directly geometry primvars) are partially supported. Some builtins known to RIS are not currently available in XPU. The most important changes are summarized in the table below. If an alternative is suggested, it means that OSL patterns should be rewritten to use the alternative to be as forward looking as possible, particularly if a OSL global exists.

P		global P may be preferred
PRadius		use filterwidth() (scaled to radius if needed)
Po		
Nn		use global N
Non		supported, but only if displacement took place
Ngn		use global Ng
Naon		point Po; getattribute("primvar", "P", Po); Naon = normalize(cross(Dx(Po), Dy(Po)));
Tn		
Vn, VLen		derive from global I: Vn = -normalize(I); VLen = length(I);
curvature		
dPdu, dPdv		use globals dPdu and dPdv
u, v		globals u and v may be preferred
st		
du, dv		use 0.5 * Dx(u), 0.5 * Dy(v)
dPdtype		
time		use global time
id, id2		

- The only getattribute("attribute", ...) lookups that are supported in XPU are "user" attributes (e.g. getattribute("attribute", "user:foo", foo)) and "user" options.
- getattribute("context") and getattribute("rendererInfo") queries are not supported in XPU.
- trace() is not currently supported. Support for a single level of recursion will be added in a future release.
- Dynamic string construction is not supported.
- It is possible in some cases that strings within OSL patterns are not properly constant folded, resulting in errors at runtime.
- Dynamic path construction in your own patterns is not supported, but please note that we **do** support the following path tokens <UDIM>, <u>, <v>, <U>, <V> and <primstr:varname> that will result in a "dynamic" path to your textures. <primstr:varname> can be used to reference the value in a constant primvar or user attribute to build a dynamic path

Geometry

	RIS	XPU	Notes
Subdivision Surfaces			Catmull-Clark only
Polygonal Surfaces			
NURBS			
Curves			
Points			Falloff is not yet supported
Quadrics and Blobbies			
Procedurals			
Nested Instancing			Only 4 levels of instancing are supported
Nested Dielectrics			XPU does not support intersectpriority or nested IOR tracking; refractions may be wrong when leaving an inner nested dielectric material
Dicing cameras			

- Loop Subdivision is not supported. Catmull-Clark for both quads and triangles **is** supported.
- Point falloff is not yet supported.
- Shading of nested instancing has artifacts. In our tests we have seen shading issues with leaves that are rendered as nested instances.
- Inverted normals are possible when doing displacement, depending on the input file and DCC. We have seen this with Alembic being imported into Houdini.
- String geometry primvars are not supported.

Lighting

	RIS	XPU	Notes
Analytic Lights			PxrEnvDaylight, PxrCylinderLight not available; some light parameters are not available
Mesh Lights			
Light Filters			
Light Linking			

Scalability to many lights			Some features are not available
----------------------------	---	---	---------------------------------

- Light Filters & Light Linking are currently not supported.
- "shot" based lighting features will be available in future releases. Examples include:
 - Emission Focus
 - Light groups
 - Manifold Next Event Estimation
- XPU does not support color temperature on lights yet. This means the correct color space won't be applied when a temperature is converted to an RGB value.

Integration & Ray Tracing

	RIS	XPU	Notes
PxrPathTracer			(some features are not available)
PxrUnified			
PxrVisualizer			(some features are not available)
PxrDefault			
Trace Sets			

- numBxdfSamples, numLightSamples, numIndirectSamples (and their manually set counterparts) are not supported.
- clampDepth and clampLuminance (used to suppress fireflies) have not been implemented.
- Non-stochastic presence is not currently supported - the maxNonStochasticOpacityEvents parameter is ignored

Post Processing

	RIS	XPU	Notes
Denoising			
Sample and Display Filters			

Pipeline

	RIS	XPU	Notes
AOVs and LPEs			
EXR, TIFF			
Deep Output			
Holdouts			
Advanced DOF controls (bokeh, etc)			
Checkpointing			

- XPU currently only outputs EXR or TIFF frames.
- Adaptive sampling is yet to be implemented.
- Only some geometric built-in AOVs are available under the same name as they are available for RIS.
- Multi-camera, multi-frame is not yet supported.
- The "weighted" option for the pixelfiltermode parameter to the Hider will not be supported.
- Baking - either at the pattern level via PxrBakeTexture or at the renderer level to bake global illumination is not yet supported.
- Cached presence or opacity is not supported in XPU. XPU renders of transparent surfaces may be sharper as a result (they will not be biased by the micropolygon size).

Projection Plugins

	RIS	XPU	Notes
PxrPerspective			
PxrOrthographic			
PxrCamera			Matte box and Circle of confusion not yet supported in XPU
PxrLightProbe			
PxrPanini			
Omnidirectional Stereo			

- XPU does not yet support bokeh and aperture control.