Features and Limitations

Below is a list of features supported and unsupported in this phase one release of XPU

General

- XPU currently requires a CUDA-compatible GPU to be installed, even if the intention is to run only in CPU mode.
- 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.
- Releasing a license: It is possible to get XPU into a situation where it doesn't release the license back to the RenderMan license server. If this happens and you end up using all of your licenses, just restart your application (Maya, Houdini, Katana, or Blender) and you will be able to continue your work.

DCCS

- XPU does not currently support redirection of errors away from the console, which means DCCs will not be able to report XPU errors in their UI.
- 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, Displacements, & Patterns	RIS	XPU	Notes
PxrSurface			Some Subsurface modes are not available. Certain single scattering scenarios are not supported. Second refraction roughness is not yet implemented.
PxrLayerSurface			
PxrDisneyBsdf			
PxrMarschnerHair			
PxrConstant			
PxrDiffuse			
PxrDisplace			
MaterialX Lama			
PxrVolume, Volumes			
OSL Patterns			PxrDirt & PxrCurvature not supported
C++ Patterns			
PxrSeExpr			
Baking			
Point Clouds			

- Custom Bxdf or Displacement plugins are unsupported.
- MaterialX Lama support will be implemented in a future release.
- In XPU, we support the following SSS models:
  - Burley Normalized
  - Exponential Path Traced
  - Non-Exponential Path Traced
- subsurfaceSubset is not yet supported.
- Single scattering support in PxrSurface is not complete because volumetric light transport is not fully supported in XPU.

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.

- XPU samples textures at one MIP level coarser than RIS so that it may store more textures in limited GPU memory. This results in slightly blurrier renders from XPU than RIS.
- We are currently in the process of updating the texture cache for the GPU to give better performance if it isn't large enough to hold the textures required for your render.
- Textures with a non square pixel aspect ratio are not supported.
- If you update a texture while the renderer is running, XPU may not update the texture. If this happens, restart your render.
- If you stop interactive rendering while textures are still being made, sometimes you will get a crash.

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.

Builtin	Supported in XPU	Alternative (if possible)
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)
dPdtime		use global dPdtime, but currently hardcoded to (0, 0, 0)
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)).
- 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 (this feature will be available in 24.1)

Lighting

Lights	RIS	XPU	Notes
Analytic Lights			PxrEnvDaylight, PxrCylinderLight not available; some light parameters not available

Dome Lights			Dome lights are always visible in refractions
Light Portals			
Mesh Lights			
Light Filters			
Light Linking			
Scalability to many lights			

- 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
- Thin shadows are currently not implemented.
- 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.
- VIRP (visible in refraction path) is not implemented (among other things, it requires LPE support).

> ⓘ **Lighting Limitations**
>
> As the light selection scheme in this phase one release of XPU is yet to be implemented, we recommend keeping the number of lights in your scene to a minimum.

Geometry

Geometry	RIS	XPU	Notes
Subdivision Surfaces			Catmull-Clark only. No support for hierarchical subdivision surfaces.
Polygonal Surfaces			
Nurbs			
Curves			
Points			Falloff is not supported
Quadrics and Blobbies			
Procedurals			
Nested Instancing			Varying attributes to drive differences in materials per-instance is not yet supported. Only 4 levels of nested instancing are supported.
Watertight dicing of subdivision surfaces			XPU cannot opt out of watertight dicing.
Watertight dicing of displaced polygonal surfaces			XPU cannot opt out of watertight dicing.
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.
- Object attributes such as maxdiffusedepth and maxspeculardepth have not yet been implemented.
- Deformation motion blur 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 and matrix geometry primvars are not supported.

Integrators

Integration & Ray Tracing	RIS	XPU	Notes
PxrPathTracer			Some features not available
PxrUnified			
PxrVCM			
PxrOcclusion			
PxrVisualizer			Some features not available
PxrDefault			

- numBxdfSamples, numLightSamples, numIndirectSamples (and their manually set counterparts) are not supported.
- clampDepth and clampLuminance (used to suppress fireflies) have not been implemented.
- Russian Roulette has not been implemented.
- allowCaustics is always 1.
- Non-stochastic presence is not currently supported - the maxNonStochasticOpacityEvents parameter is ignored

Rendering

Post Processing	RIS	XPU	Notes
Sample and Display Filters			

Pipeline	RIS	XPU	Notes
AOVs and LPEs			
EXR, TIFF			
Deep Output			
Matte, Holdouts			
PxrCamera			
Crop windows			XPU support will be available in 24.2. crop windows in 'it' will be available in 24.4.
Advanced DOF controls (bokeh, etc)			

- XPU currently only outputs EXR or TIFF frames.
- Adaptive sampling is yet to be implemented.
- Motion blur has not been fully implemented. Transformation motion blur is fully functional, but deformation motion blur of geometry is not supported.
- Trace Subsets have not been implemented. This impacts the subsurfaceSubset feature of PxrSurface.
- 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 built-in curvature calculation is different to RIS, which results in differences in specular mollification.
- The "weighted" option for the pixelfiltermode parameter to the Hider is not yet 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).