

Migrating from 23.X to 25

Many user facing changes are listed in the [Release Notes](#).

This document lists the main issues for developers who are migrating from 23 to 24/25.

For migration from 22 to 23, please refer to [Migrating from 22.x to 23](#).

Parameter Lists

- `pxrcore::ParamList` has been split into `ParamList` and `PrimVarList`; the distinction is that `RtPrimVarList` is preferred for geometry calls.
- `RtParamList.h` has been removed; the typedefs for `RtParamList` and `RtPrimVarList` have been moved to `RtTypesHelper.h`

Riley

There have been many changes in Riley in order to streamline the creation of displays and cameras, as well as to allow for the editing of options. Please note that as of PRMan 24.0, Riley **should still be considered an unstable interface**.

- The Riley header file is now namespaced by version (currently 0.3). The Riley class is now accessible as `'riley::vX_Y::Riley'`. In order for this to be forward-declarable, a new header `'Riley_version.h'` has been added which only contains the versioned namespace information. By default, the versioned namespace is also imported into the `'riley'` namespace, so that the various Riley API components can also be accessed by only using `'riley::Riley'`.
- The `Begin()` and `End()` routines have been removed.
- The use of the term "Master" has been completely replaced with "Prototype" throughout.
- `RenderViews` are now first class citizens:
 - `RenderViewId` has been added.
 - `CreateRenderView()`, `ModifyRenderView()`, and `DeleteRenderView()` has been added.
 - The `RenderSettings` struct has been removed.
 - `Render()` now takes a `RenderView`, which contains the information about the camera, integrator, and render target to render.
- `SetRenderOptions()` has been renamed to `SetRenderSettings()`.
- `SetActiveCamera()` has been renamed to `SetDefaultDicingCamera()`.
- `CreateGeometryMasterDeferred()` has been removed.
- Several new structs have been added in order to simplify parameter lists for certain routines:
 - `RenderOutputList` (affects `RenderTarget` and `Display` routines)
 - `FilterSize` and `Extent` (affects `RenderOutput` and `RenderTarget` routines)
- A new `riley::UserId` class has been added to facilitate statistics gathering.
- `CreateMaterial()` and `ModifyMaterial()` now accept parameter lists.
- The `ScopedCoordinateSystem` struct has been renamed to `CoordinateSystemList`.
- `CreateDisplayFilterChain()` has been replaced with `CreateDisplayFilter()`, `ModifyDisplayFilter()`, and `DeleteDisplayFilter()`.
- `CreateSampleFilterChain()` has been replaced with `CreateSampleFilter()`, `ModifySampleFilter()`, and `DeleteSampleFilter()`.
- `SetClippingPlanes()` has been replaced with `CreateClippingPlane()`, `ModifyClippingPlane()`, and `DeleteClippingPlane()`.
- `CreateRiley()` now accepts a `RtParamList`, which allows the behavior of the Riley variant to be changed based on supplied parameters.
- A new `InvalidateTexture()` method has been added, allowing textures to be invalidated directly via Riley.
- Two new methods `SetRenderOptions()` and `SetRenderMetadata()` have been added to support general options edits. These calls can be made while `Render()` is running.
- The `Id` types have been changed from enums to classes.
- Inconsistencies in the order of array parameters and the size of those arrays have been addressed.
- Inconsistencies between the `Create` and `Modify` routines with respect to modifiable parameters that have array types have been addressed.

Rix Shading Plugins

- The use of the `RtPointer` and `RtConstPointer` types has been eliminated in favor of `void *` and `const void *`. Some usage of `RtInt` and `RtBoolean` has also been changed to `int` and `bool`.
- Usage of `#include "ri.h"` has been deprecated for Rix headers in favor of `#include "RiTypesHelper.h"`.
- The `RixSCShadingMode` enum has been extended with a new `k_RixSCBakeIntegratorQuery`, which allows plugins to distinguish between baking of patterns and baking of illumination.
- `RixShadingContext::pointWeight` has been changed from a float to a `RtColorRGB`.
- `RixBxdfFactory::BeginScatter()`, `RixBxdfFactory::BeginOpacity`, `RixBxdfFactory::BeginInterior()`, `RixBxdfFactory::BeginScatter()` have been extended to accept an opaque data pointer. When using nested Bxdfs in a shading network, this opaque pointer can be used by a parent Bxdf to pass data to children Bxdfs. When these methods are called by the renderer on the parent Bxdf this pointer will always be `nullptr`.
- The number of available Bxdf user lobes as defined in `k_RixBXMaxNumUserLobes` has been set back to 8 (internally, the limit has always been 8)
- Light filters now have new entry points:
 - `ShadowFilter()`: this is called after `Filter()` has run and after the default shadow rays have been traced. The new method can be used to calculate new transmittance values.
 - `CreateMutableContext()`: this method can be used to re-evaluate connected inputs to the lightfilter, which would be typically done after changing builtins of the shading context with `RixShadingContext::SetBuiltinVar()`.
- The `RixRNG` API remains mostly unchanged with respect to support for the new Blue noise pmj02 samples. There is an exception: for photon emission, the new methods `ScrambledSample2D()`, `MultiScrambledSample2D()`, etc should be used instead of the `Sample2D()` variants in order to ensure sufficient entropy in photon emission directions.

Deprecated APIs

The following interfaces should be considered deprecated, and will be removed in a future release:

- `dtex`
- `RixDeepTexture`
- `RixSymbolResolver`

- RixResourceResolver
- RixStorage
- RixMutex
- RixThreadUtils