

Call ordering

Here is the order in which various RixShadingPlugin API methods are called.



When a method is called for all shading plugins, we will use the RixShadingPlugin base class method.

Order	Name	Comment	Granularity	Access to options	Access to RixIntegratorEnvironment
Once per rendering session.					
#1	<i>RixShadingPlugin::Init()</i>	When the first instance of a shading plugin is created.	For each plugin.	Deprecated.	No.
Before the first render and between renders.					
#2	<i>RixShadingPlugin::CreateInstanceData()</i>	When the first instance of a shading plugin is created, after the plugin's <i>Init()</i> has been called.	For each shading plugin instance.	Deprecated.	No.
	<i>RixIntegratorFactory::CreateIntegrator()</i>				
	<i>RixProjectionFactory::CreateProjection()</i>				
	<i>RixLightFactory::CreateLight()</i>		At least once for each light plugin instance (*).		
#3	<i>RixBxdf::GetInstanceHints()</i>	Shortly after <i>RixBxdf::CreateInstanceData()</i> is called.		Deprecated.	No.
For each render.					
#4	<i>RixProjection::RenderBegin()</i>		For each <i>RixProjection</i> .	Yes.	No.
	<i>RixIntegrator::RenderBegin()</i>	Responsible for filling the <i>RixIntegratorEnvironment</i> structure.	For each <i>RixIntegrator</i> .	Yes.	Writing.
#5	<i>RixShadingPlugin::Synchronize()</i>	With the synchronization message <i>k_RenderBegin</i> .	For each plugin.	Yes.	Yes.
#6	<i>RixShadingPlugin::SynchronizeInstanceData()</i>	Only if <i>CreateInstanceData()</i> returned a non-zero <i>InstanceData::synchronizeHints</i> .	For each plugin instance.	Yes.	Yes.
Once per rendering session.					
#7	<i>RixShadingPlugin::Finalize()</i>	Before unloading the shading plugins.	For each plugin.	Deprecated.	No.



(*) *RixLightFactory::CreateInstanceData()* will be called once for each light plugin instance. *RixLightFactory::CreateLight()* will be called once per light plugin instance, *per instance of the corresponding light*. This is a behavior specific to the *RixLight* API.