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