

XSIIndigo Exporter 1.0

Renders performed by

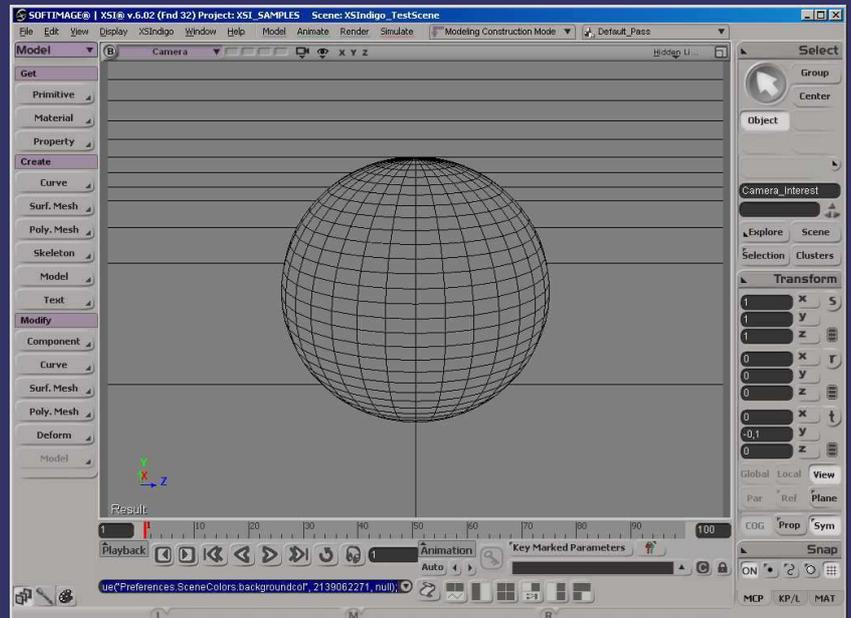


0. Installation

Being XSIIndigo provided as an XSI add-on, the installation is quite straightforward. Simply drag and drop the file in the main XSI window or go to “File -> Add-on -> Install”, select the XSIIndigo.xsiaddon file and press “Install”. The Add-on creates a structure in your User Add-on folder adding all the file needed to work with XSIIndigo.

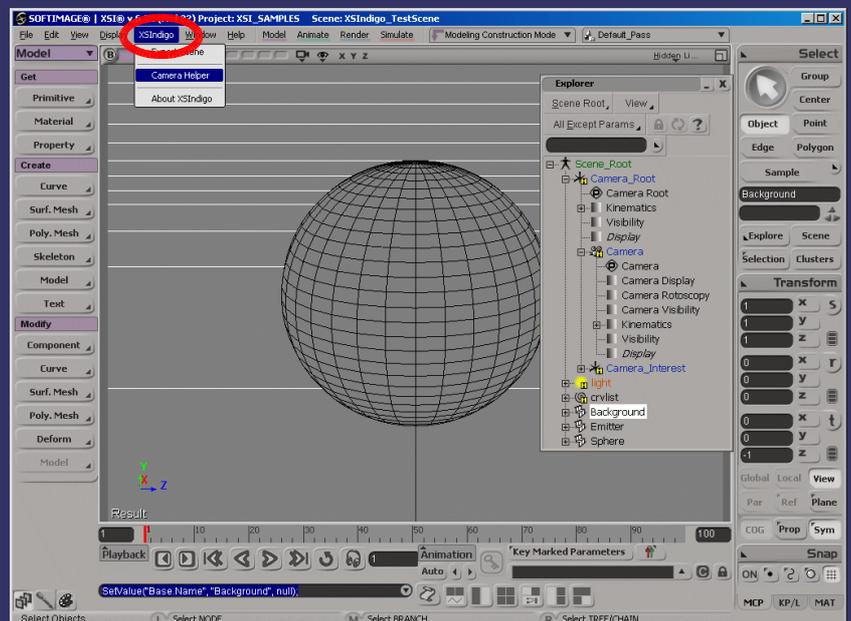
1. Basic Workflow

Start a new XSI session and load our test scene. In this specific case the scene is composed by all polymesh* elements: a background, a sphere and an emitter (hidden). (*Please be sure to convert all your geom items in polymesh before exporting



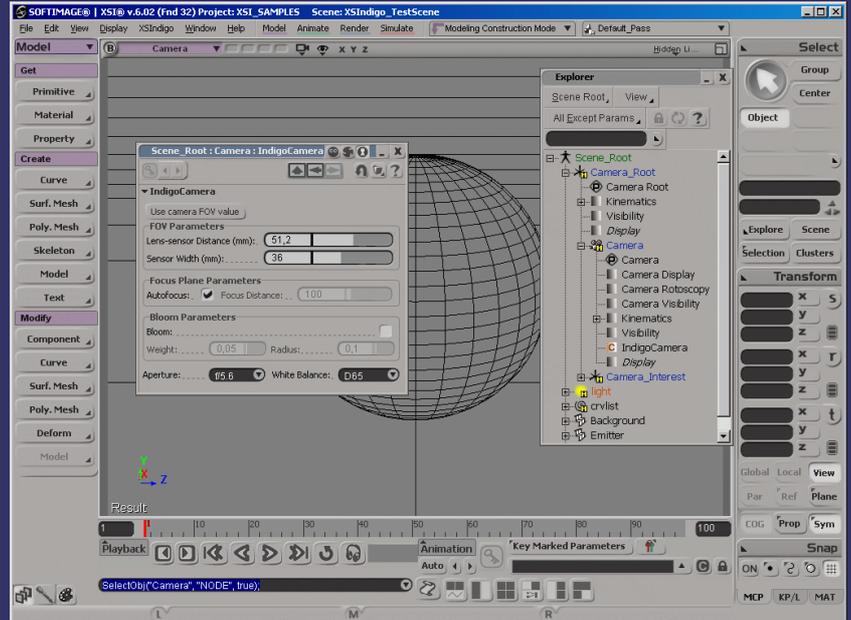
The first thing to notice is the lack in the standard XSI camera of some parameters needed by Indigo to correctly handle the camera object. For this reason from the top menu “XSIIndigo” invoke “Camera Helper”. This command, if nothing* selected prompt us to select a camera node.

(* In case a camera has been previously selected, the command directly creates the custom property node required.

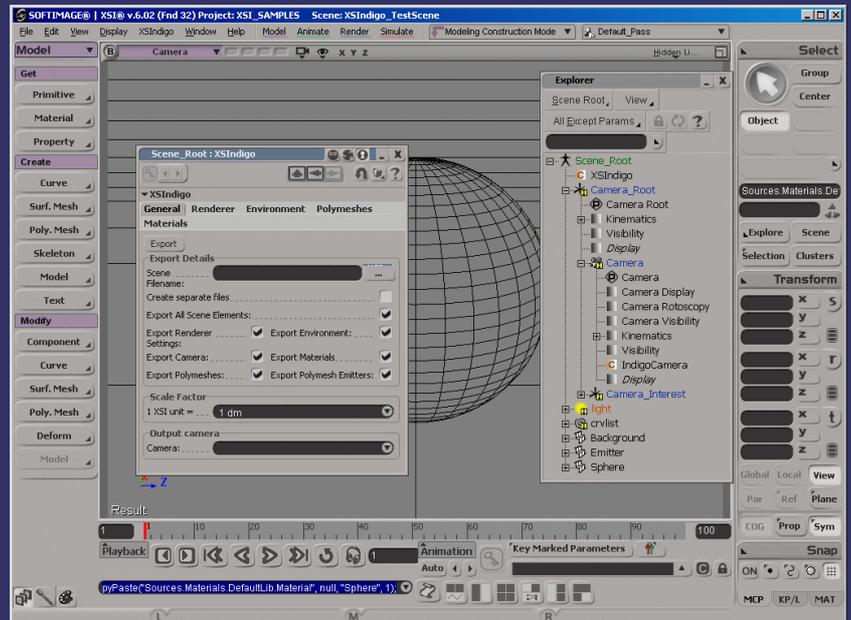


XSIIndigo Exporter 1.0

With the picking tool activated, select a camera (not the camera rig) and with the RMB end the picking session. At this point XSI will open the IndigoCamera property page to tune all the indigo's camera parameters.

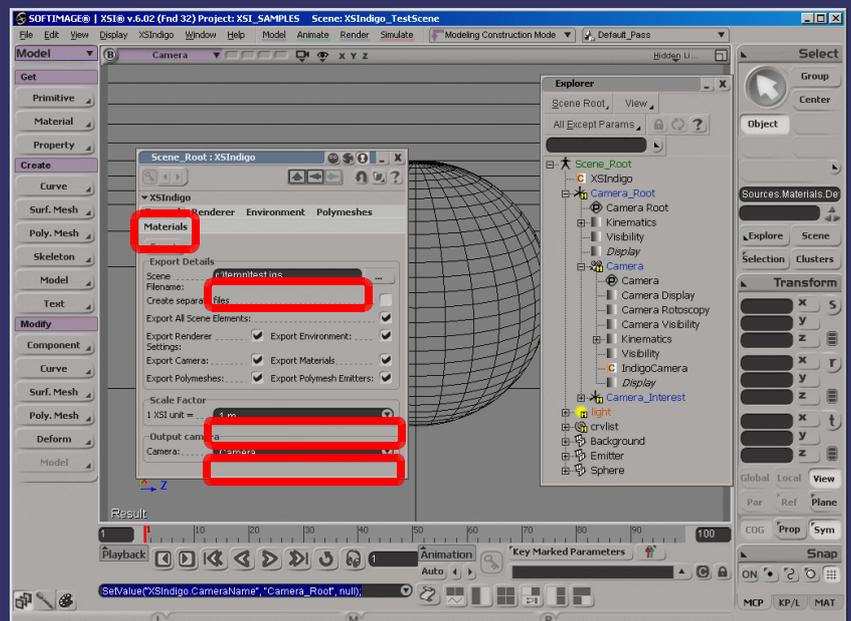


Once done, we are ready to invoke from the top XSIIndigo menu, "Export Scene" to create a basic .igs file to be rendered with Indigo. As the XSIIndigo property page is shown, we have direct access all the settings needed to define our Indigo scene



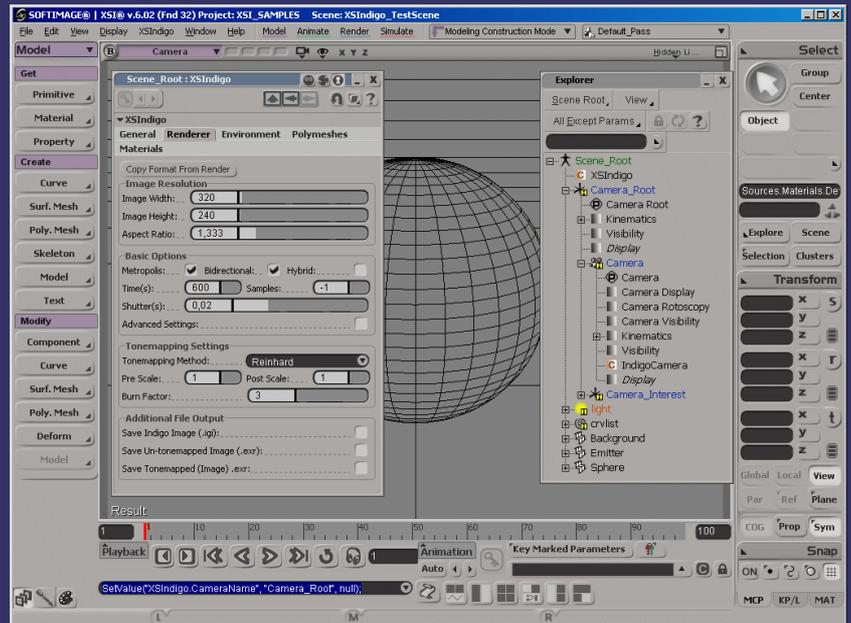
The first parameter to set is the "Scene filename" which is the full path to the scene file to be created. Then we select the scale factor to convert XSI units to Indigo ones*. Last parameter to set in the "General" tab is the camera to be exported. Remember that this camera should have an IndigoCamera property node attached otherwise the exporter fails.

(* Indigo units are standard meter so the scene is scaled in according to this parameter

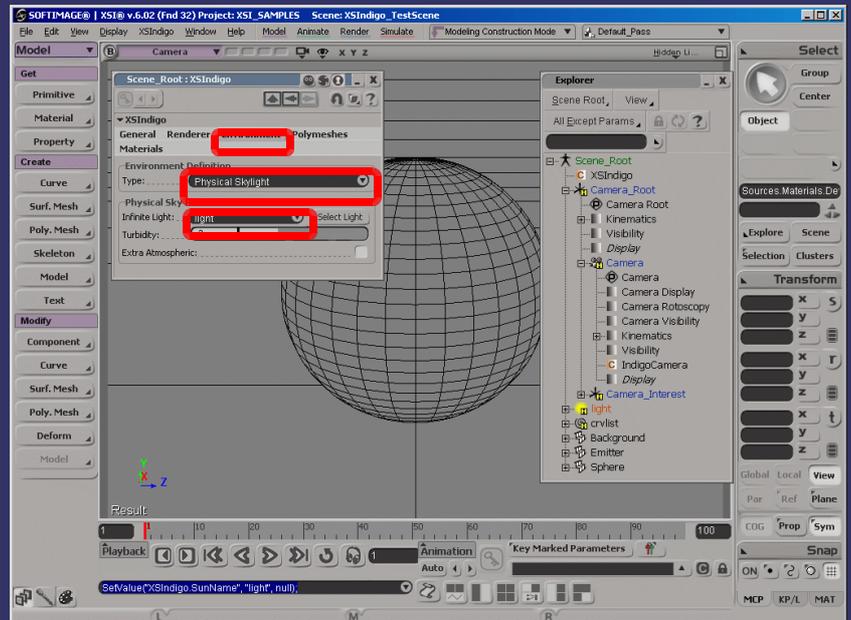


XSIIndigo Exporter 1.0

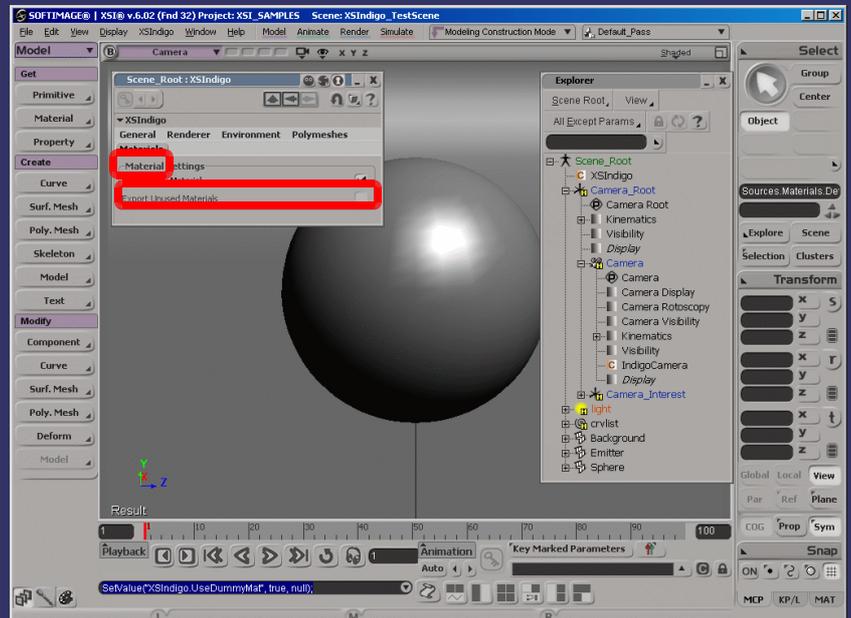
Passing to the “Renderer” tab we access to all those general parameters to properly set the image output and to roughly set the rendering engine parameters.



In the “Environment” tab we can define the environment surrounding the scene. In our case let's select a Sun/Skylight environment. Then select the default XSI infinite light as light source.

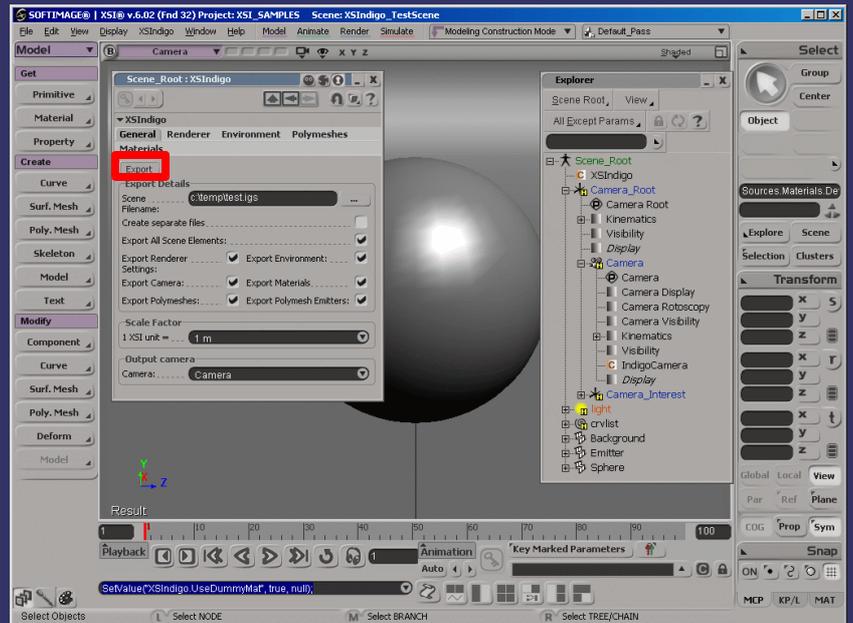


Last parameter to check for a simple scene export is “Use Dummy Material” which is located in the “Material” tab. The dummy material is a simple lambertian diffuse shader applied to all the scene elements enabling us to have a fast overview of the overall scene.

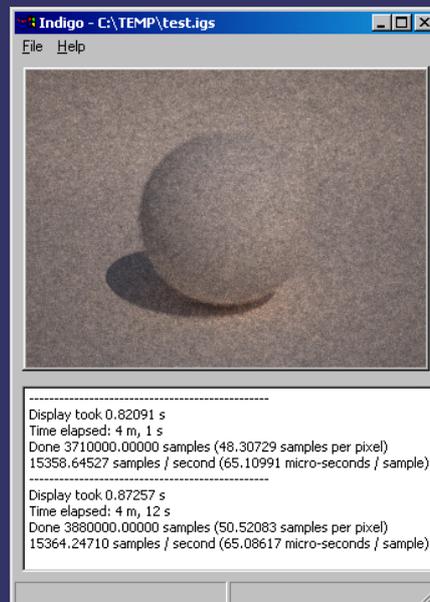


XSIIndigo Exporter 1.0

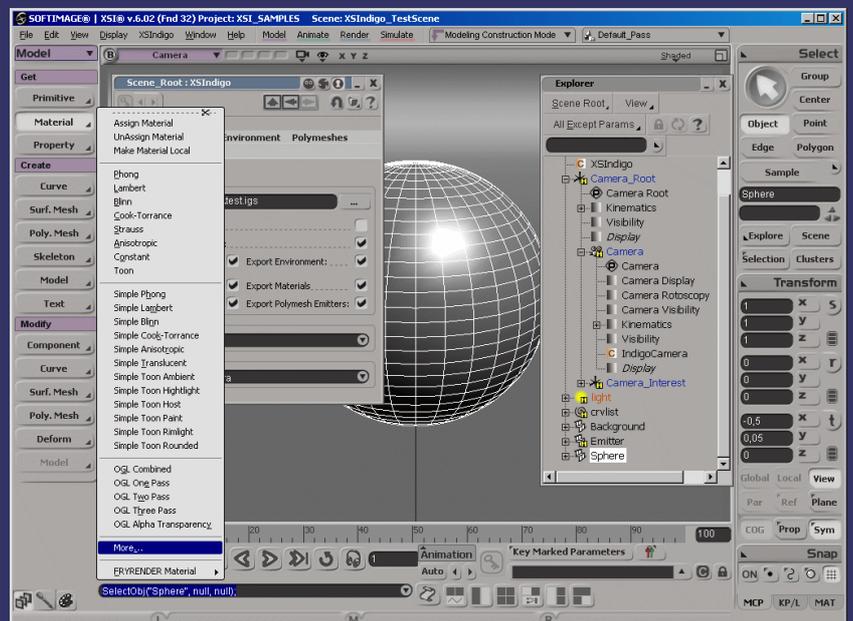
Pressing “Export” in the “General” tab, we create our Indigo scene in the specified path previously defined.



As we start Indigo through “indigo.exe” and open the file just created, the renderer starts rendering our image in the proper Indigo window

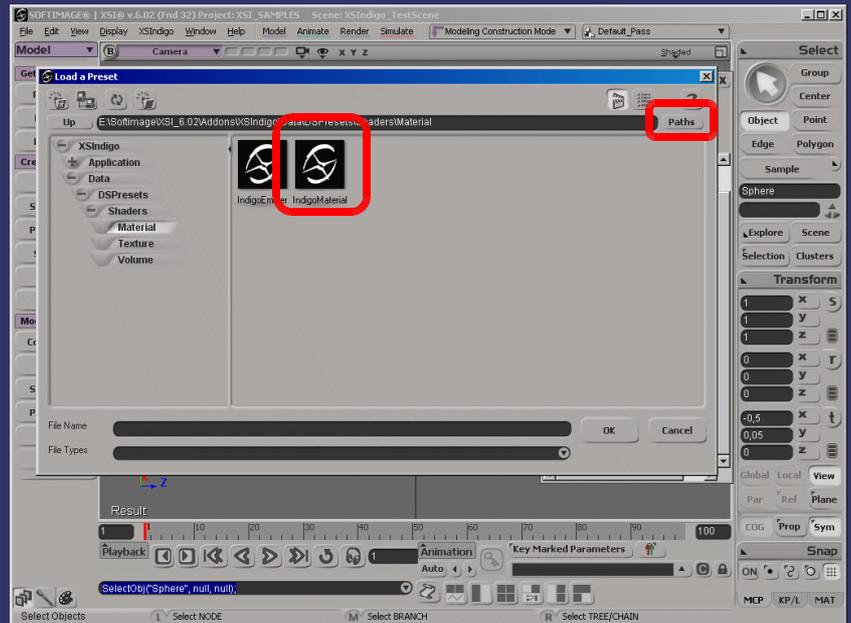


Since we're interested to apply custom materials (phong, specular, SSS and so on), these are defined through the use of IndigoMaterial shader nodes which are provided with the XSIaddon. A simple phong material is obtained composing an IndigoMaterial node and an IndigoSpectrum node. For obtain this result we select the sphere and click “Material->More”

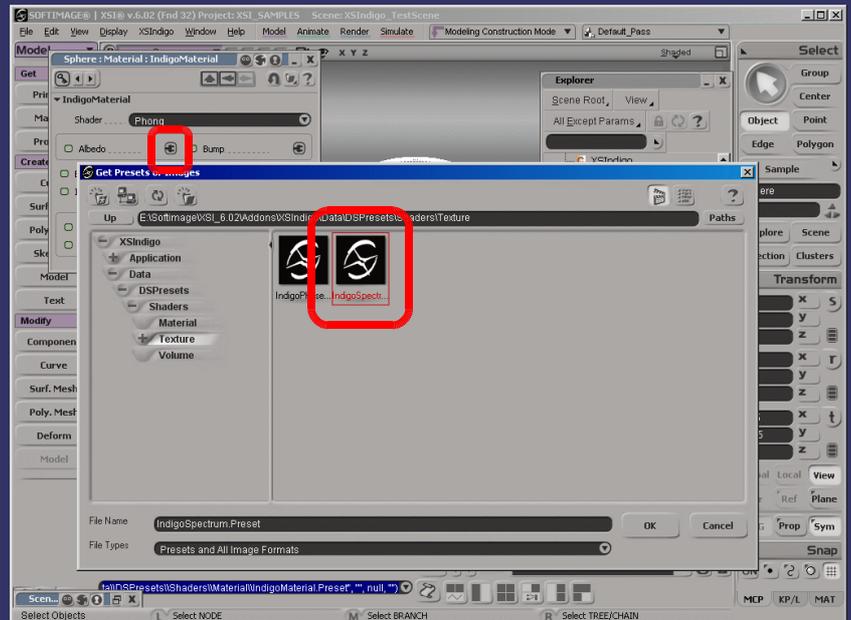


XSIIndigo Exporter 1.0

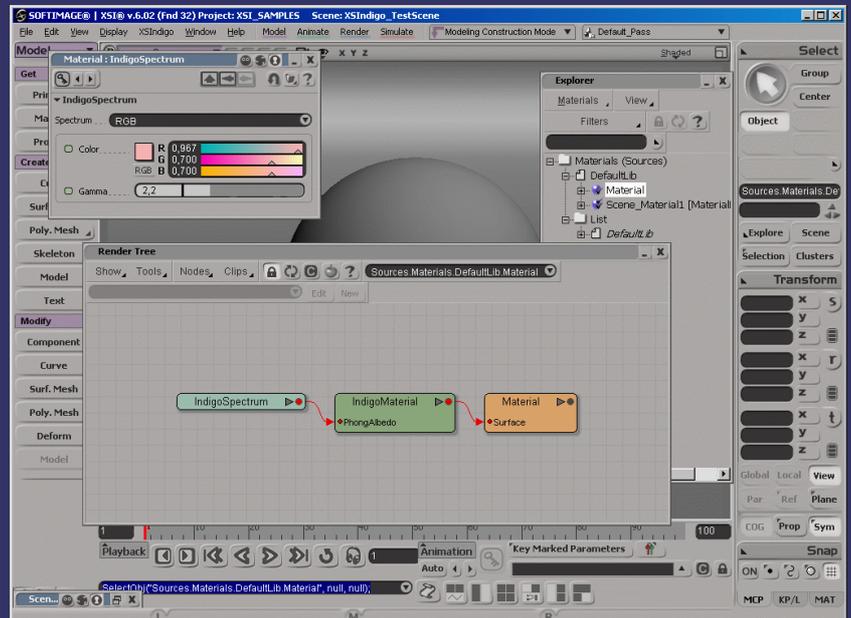
In the File browser we go to the XSIIndigo addon path with the "Paths" button go down to "Shaders -> Material" folder and select "IndigoMaterial"



The IndigoMaterial property opens enable us to access all the parameters related to Indigo's materials. The first thing to do is to connect an IndigoSpectrum to the Phong albedo port. This could be done via the "node" button or via the RenderTree window. Let's click "Albedo" node button then "More". As for the material we have to go to XSIIndigo Addon directory and select "IndigoSpectrum"

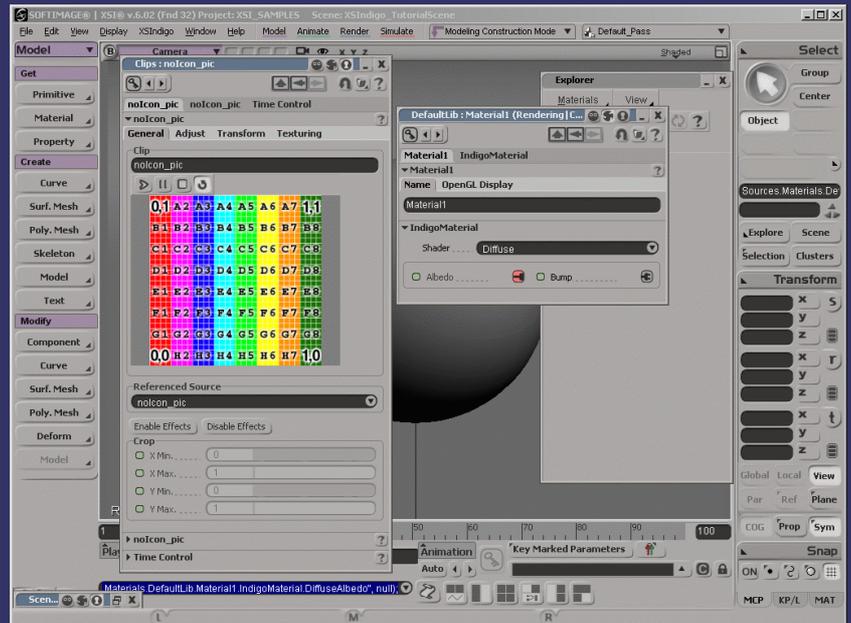


Tweak the RGB handle to select your albedo color. Opening the RenderTree, the selected material should appear composed by three nodes as shown.

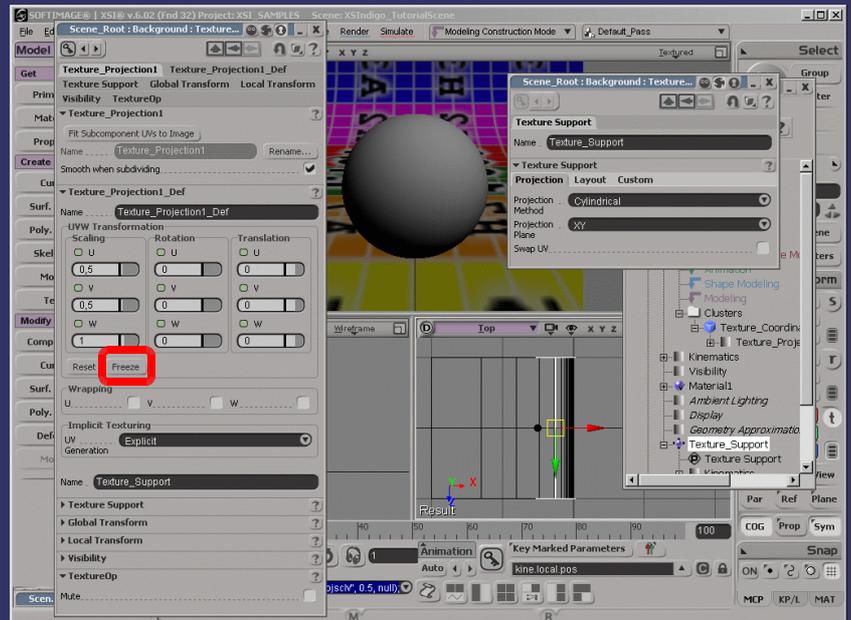


XSIIndigo Exporter 1.0

Now let's create a diffuse textured material for the Base element. As done for the sphere select your object and apply an IndigoMaterial. Change the shader to "Diffuse" then connect to the albedo port your preferred image.

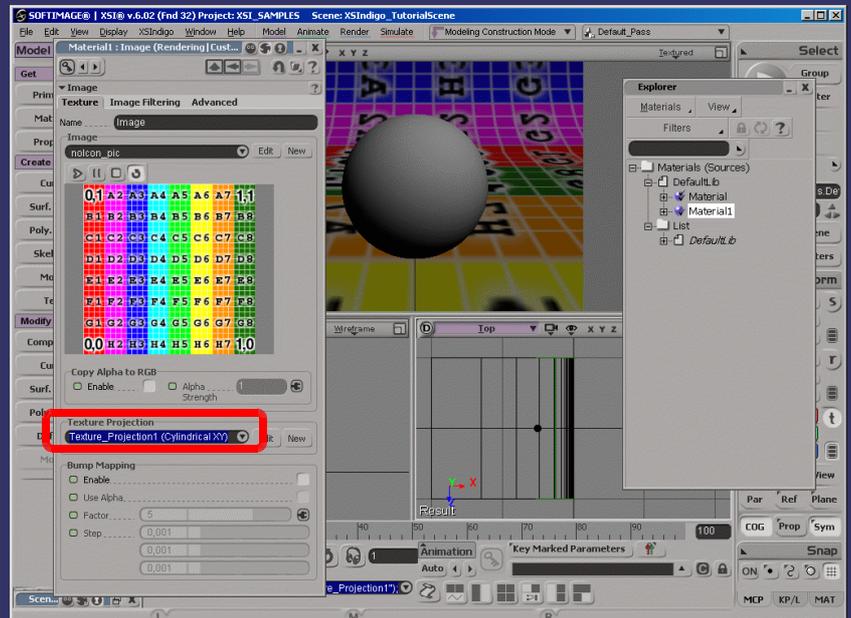


Being a textured material applied on a mesh missing UV info, we have to apply a texture support together with a texture projection to our object. Remember to freeze any changes you apply to UVW Transformation under your texture projection property page



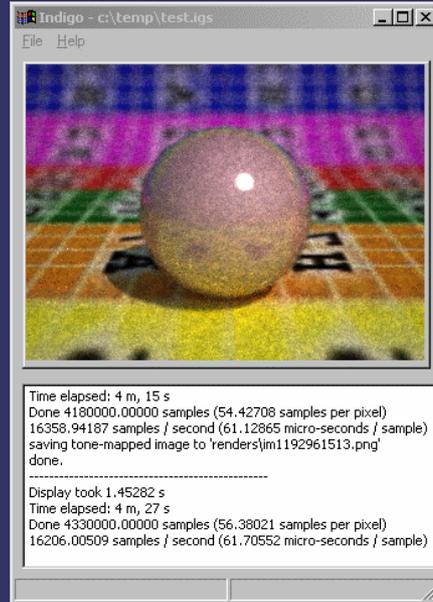
Last thing to do before proceeding in export the updated scene, is to set in the Image property page the uv-dataset to use for the selected object. Although XSI sets by default the current TextureProjection, XSIIndigo wants the user to specify it*. Open the diffuse material property page, click the albedo connector and set In the "Texture Projection" the proper texture previously created

(*)Specifying the TexProj let's XSIIndigo export the correct UV-dataset for each object having that image connected

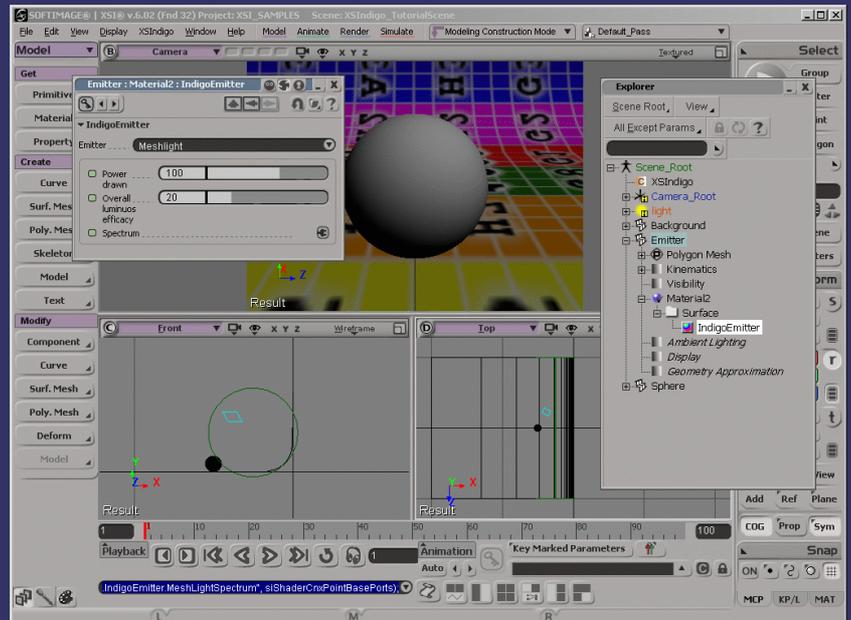


XSIndigo Exporter 1.0

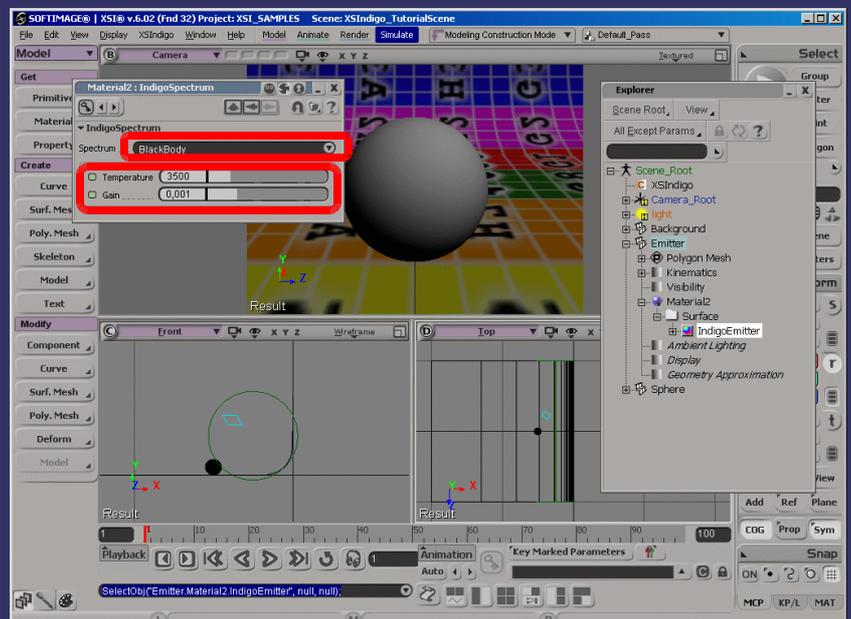
Going back to the XSIndigo Property (you can access it either clicking in the Explorer either clicking in the top menu) disabling “Use Dummy Material”, exporting the updated scene and launching a new render in Indigo should bring to the result beside. Durign the export process you could notice a Prompt shell opening. This let XSIndigo to directly convert all your imageclips in image “comfortable” to be used as maps for Indigo.



Last thing to do is to add an emitting material to our selected polymeshes. For this purpose has been provided the IndigoEmitter material that can be applied as the standard IndigoMaterial to any XSI item. Let's show our “Emitter” from the Explorer and apply from the IndigoEmitter.

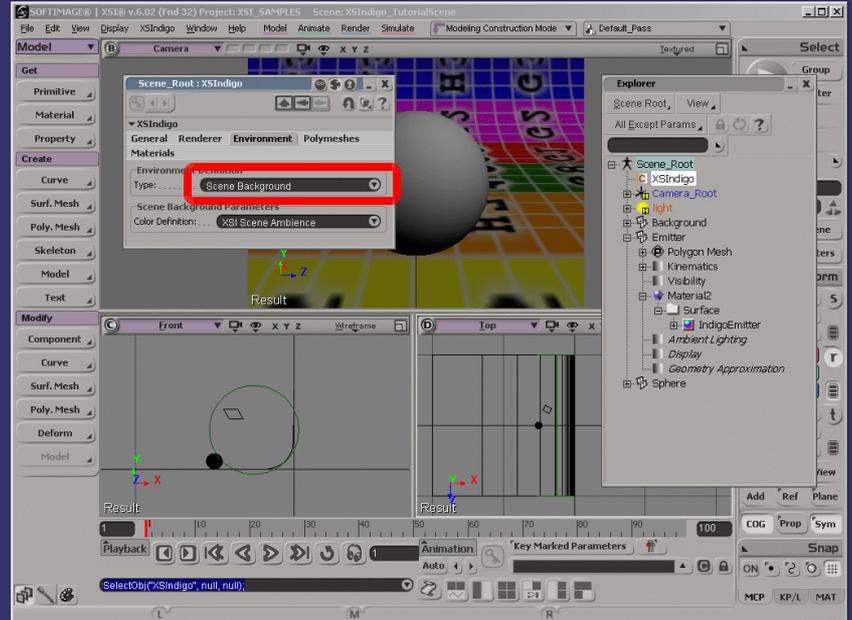


To enable emitting properties we have to connect an IndigoSpectrum or an image clip to the “Spectrum” connector. In case you choose an image-clip, remember to apply a texture projection to your meshlight (as shown for applying an albedo texture). Now we apply an IndigoSpectrum with a BlackBody spectrum definition.



XSIIndigo Exporter 1.0

Going back to the XSIIndigo Property remember to change the environment settings to "Scene Background" (otherwise you won't distinguish any light coming your meshlight emitter) and simply export again your scene.



Finally render with Indigo, tune your parameters, and enjoy results.

