

How to use Liquid

## **- Development environment**

see ./readme.txt

## **- Build**

see ./readme.txt

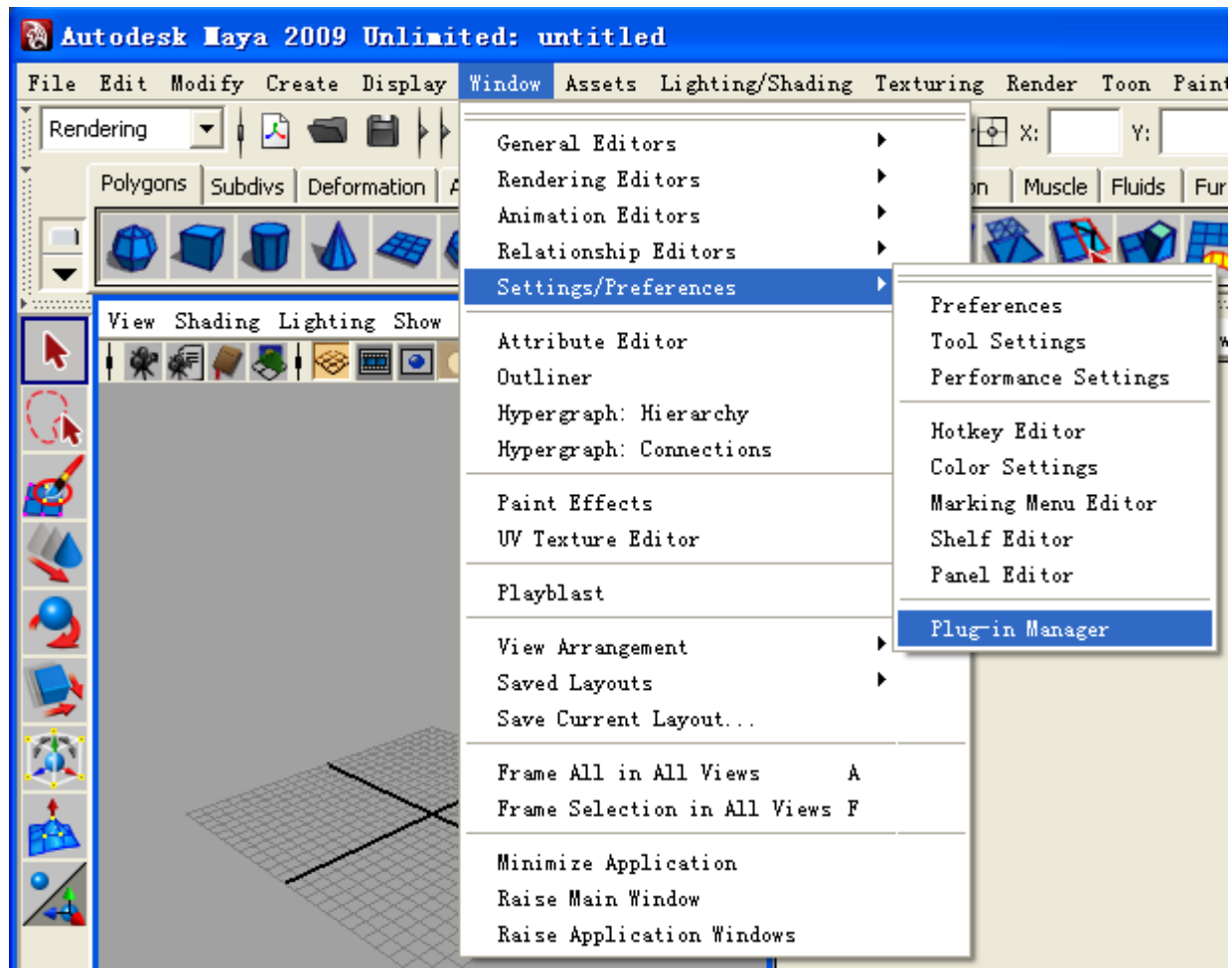
## **- Install**

see ./readme.txt

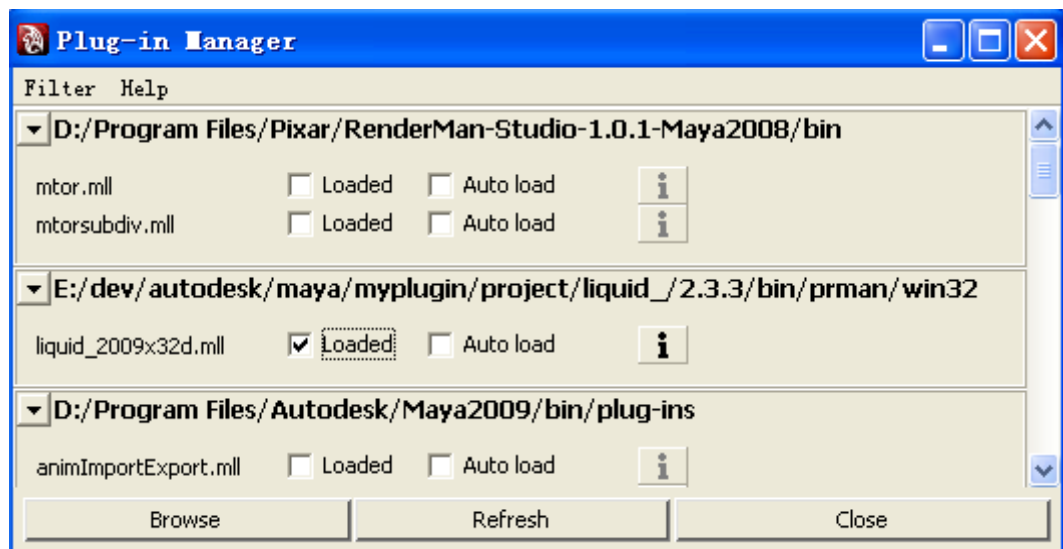
## **-load liquid plugin**

Demo scene is (\$liquid\_root)/2.3.3/doc/HowToUseLiquidmaya.ma

1)

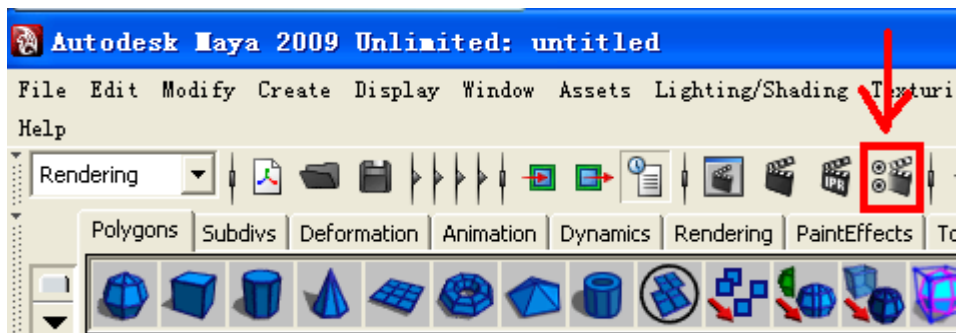


2) select liquid\_2009x32d.mll

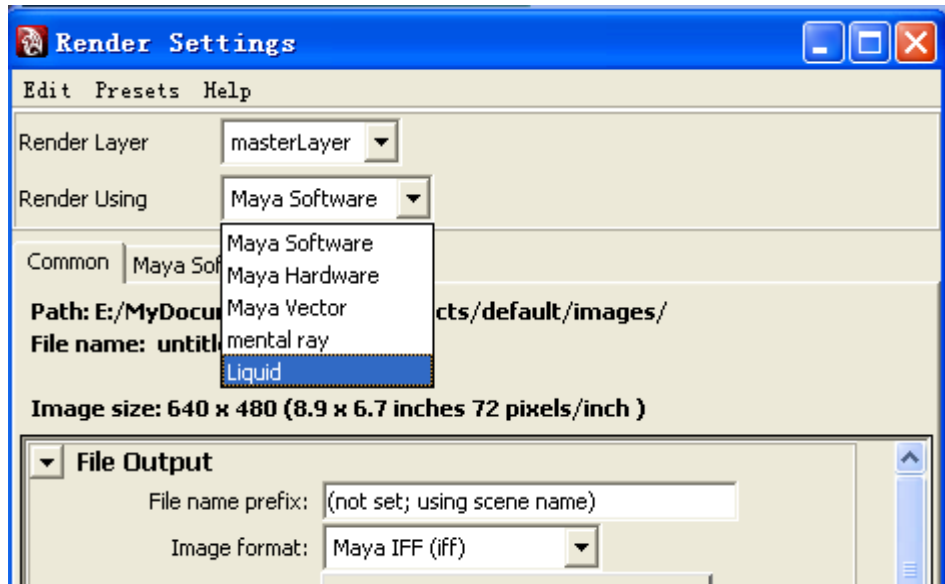


-select liquid renderer

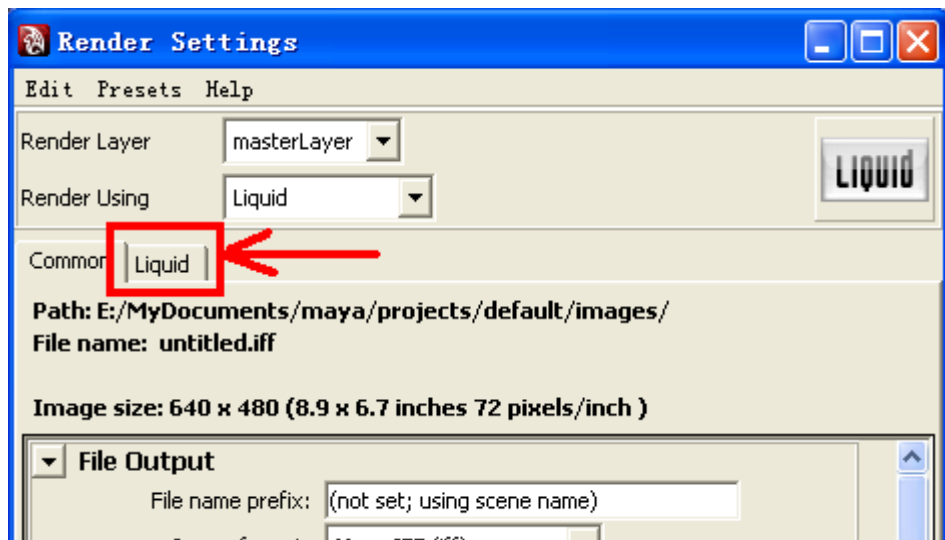
1)



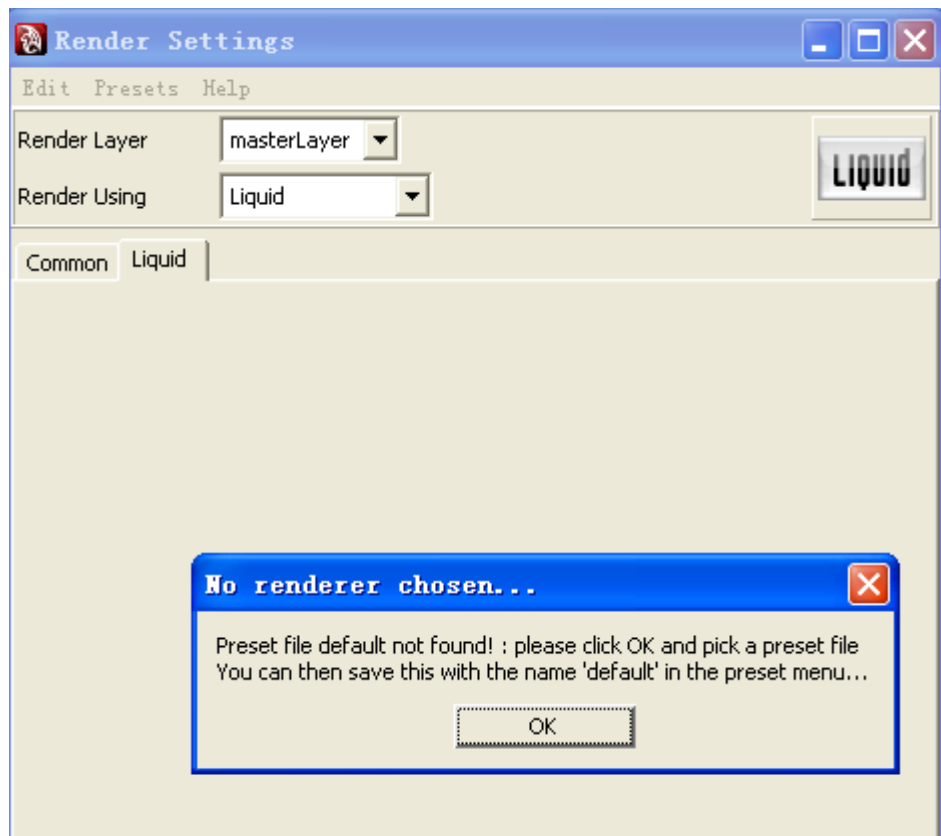
2)



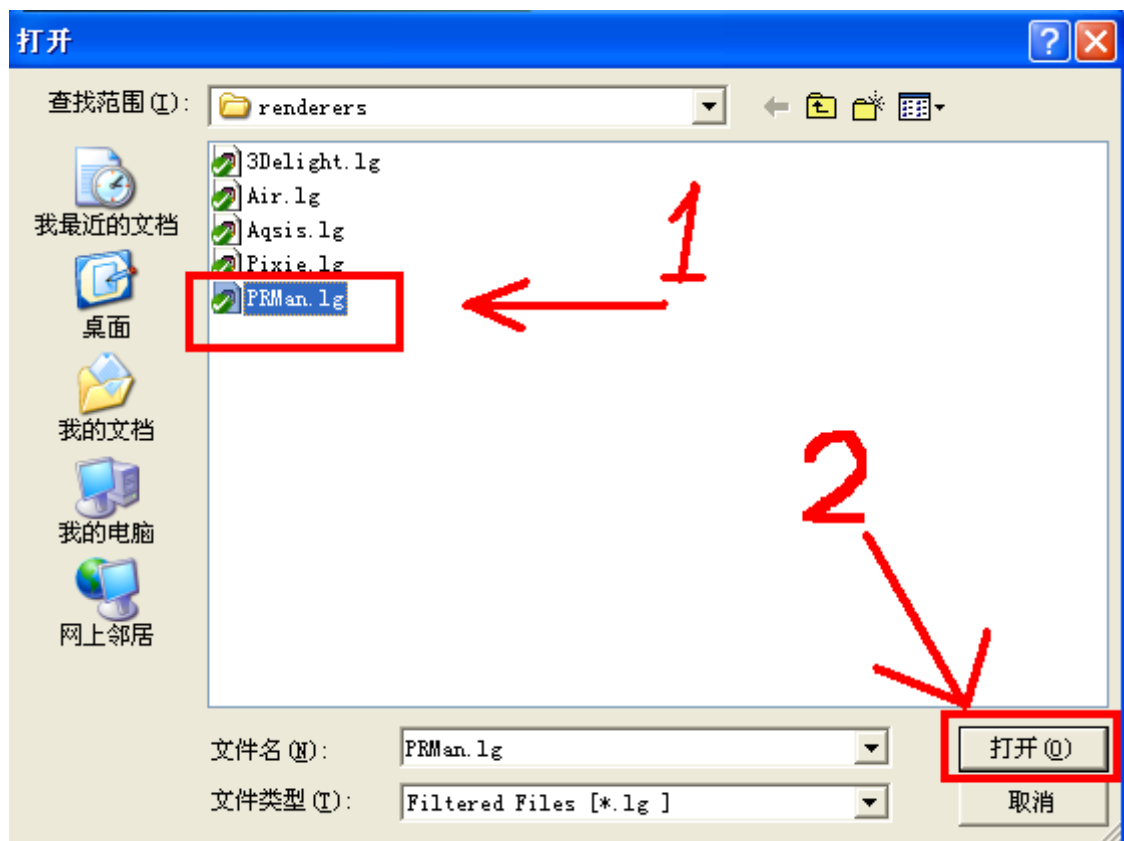
3)



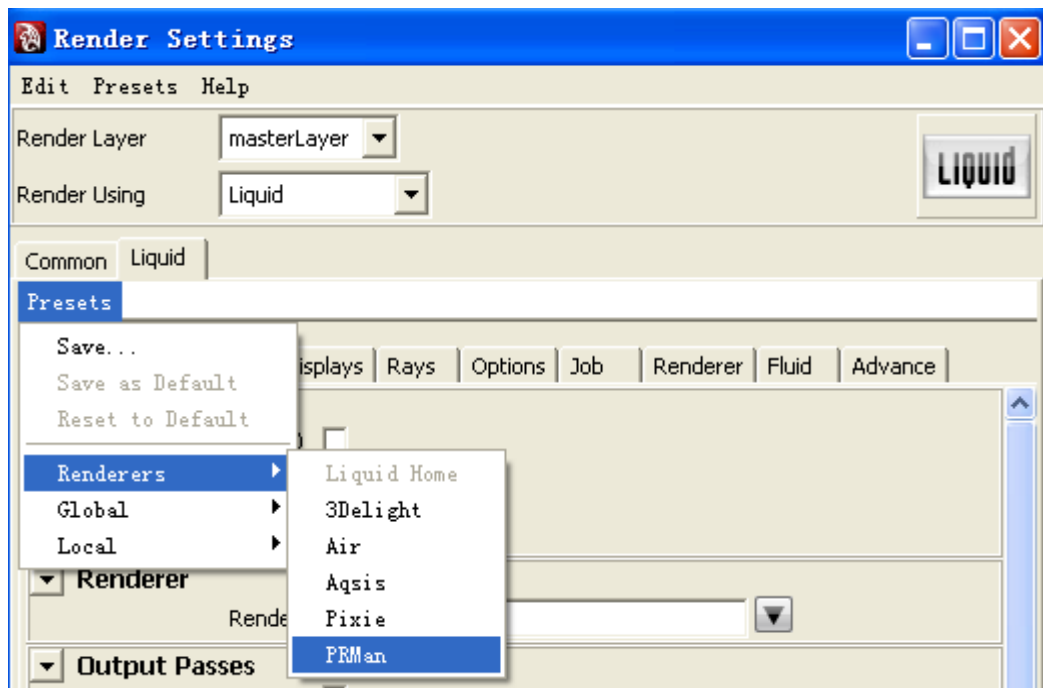
4)



5)



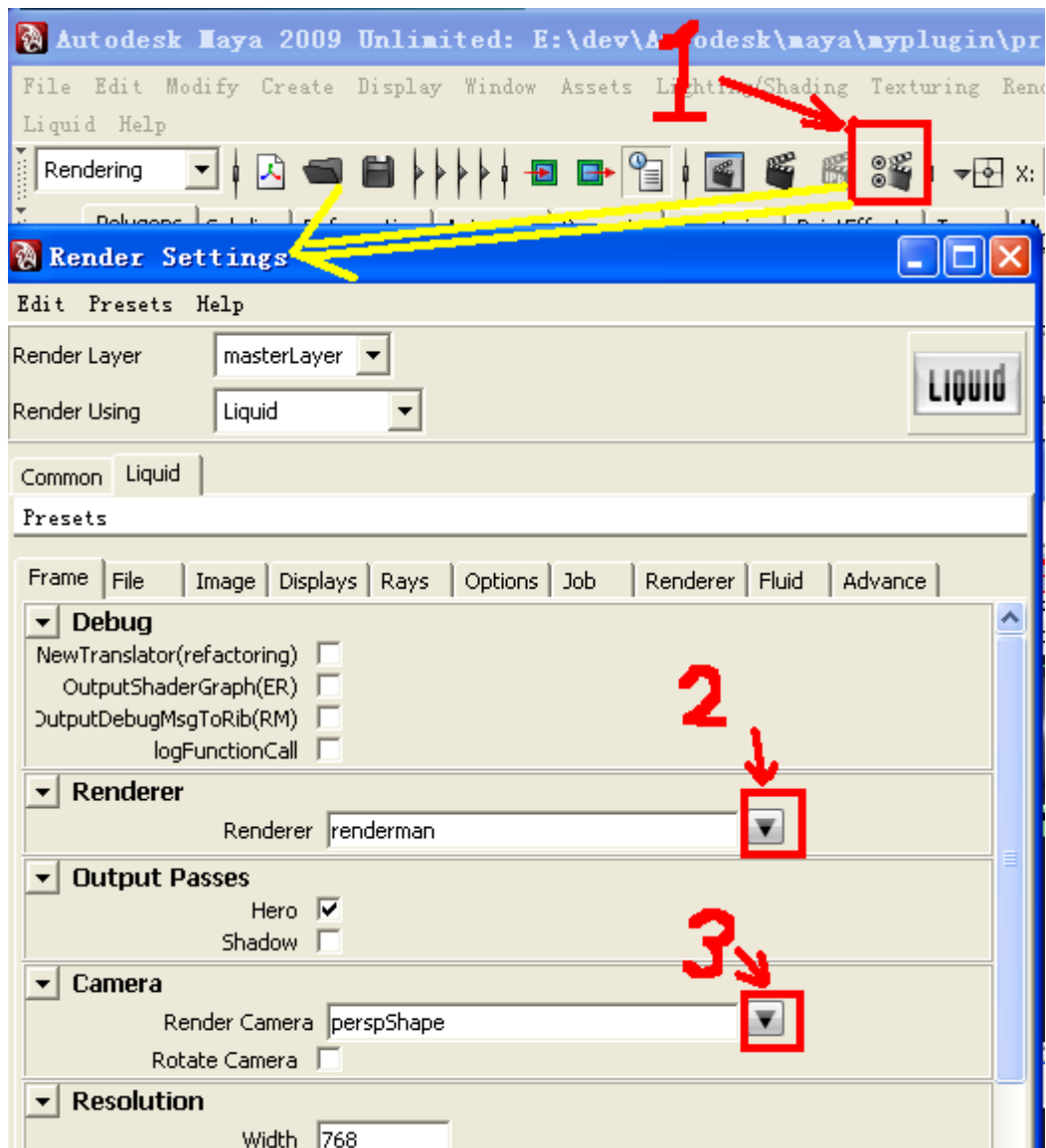
(you can select \*.lg in this way:



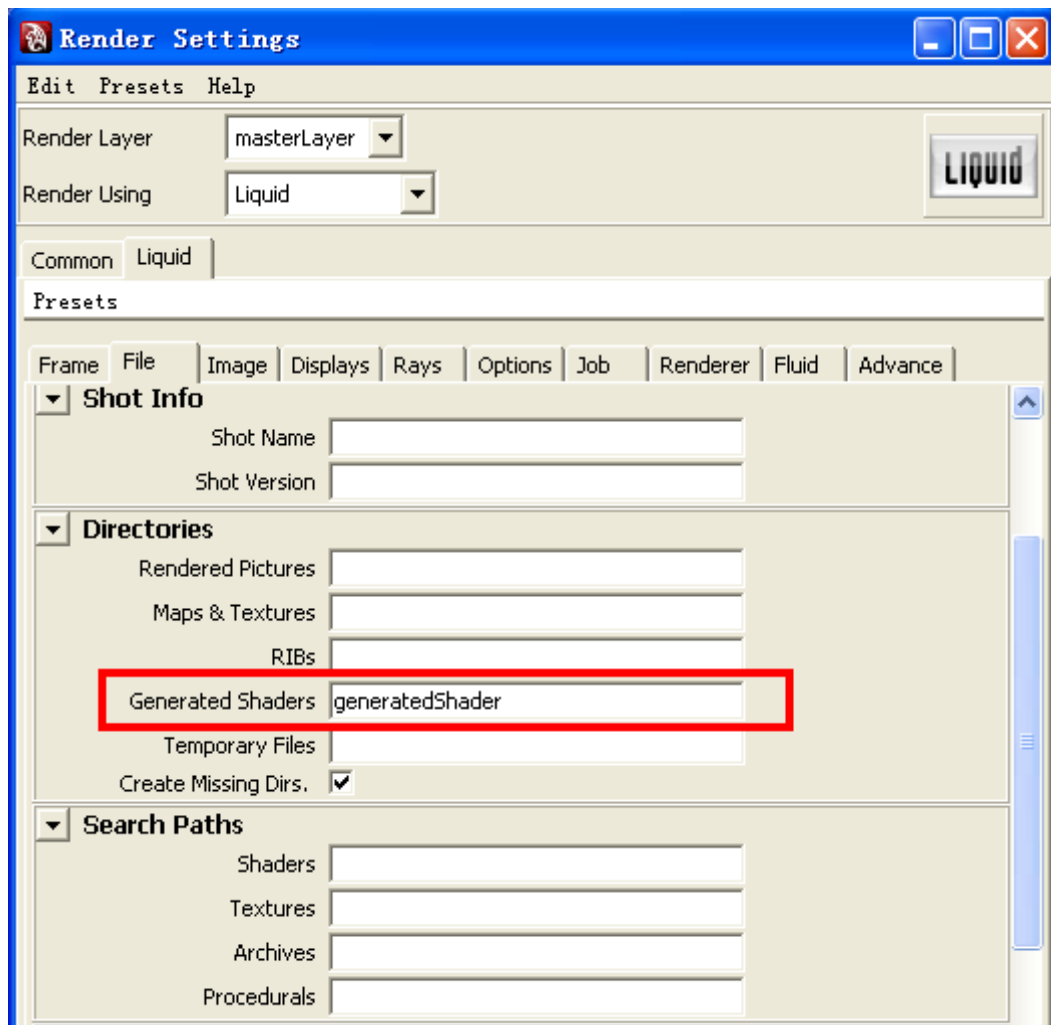
)

6) select "Renderer" and "Camera" in "Render Settings"

"Renderer" MUST be setted to "renderman" in liquidMaya,(in maya2renderer, you can set the "Renderer" to another renderer, e.g. elvishray)

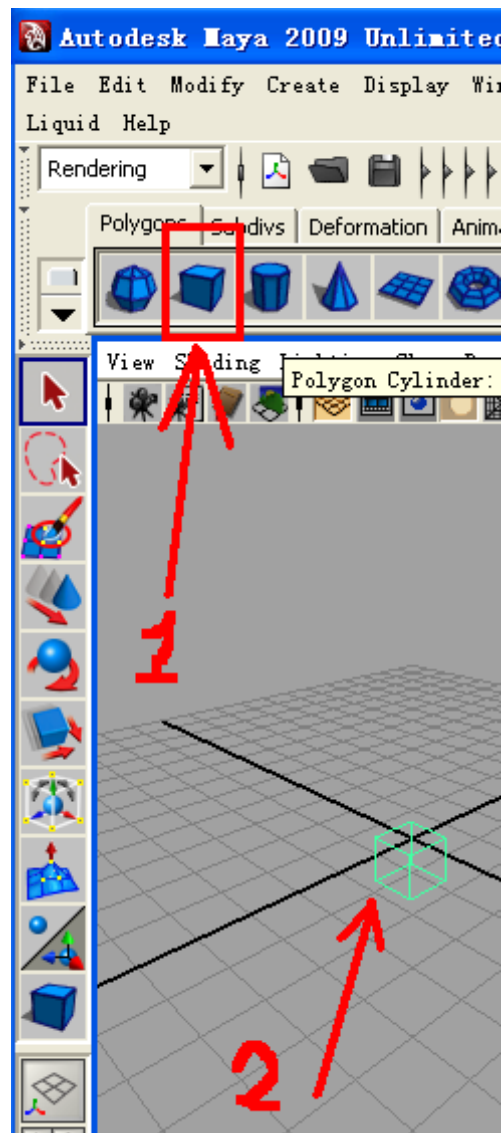


7)set GeneratedShaders directory



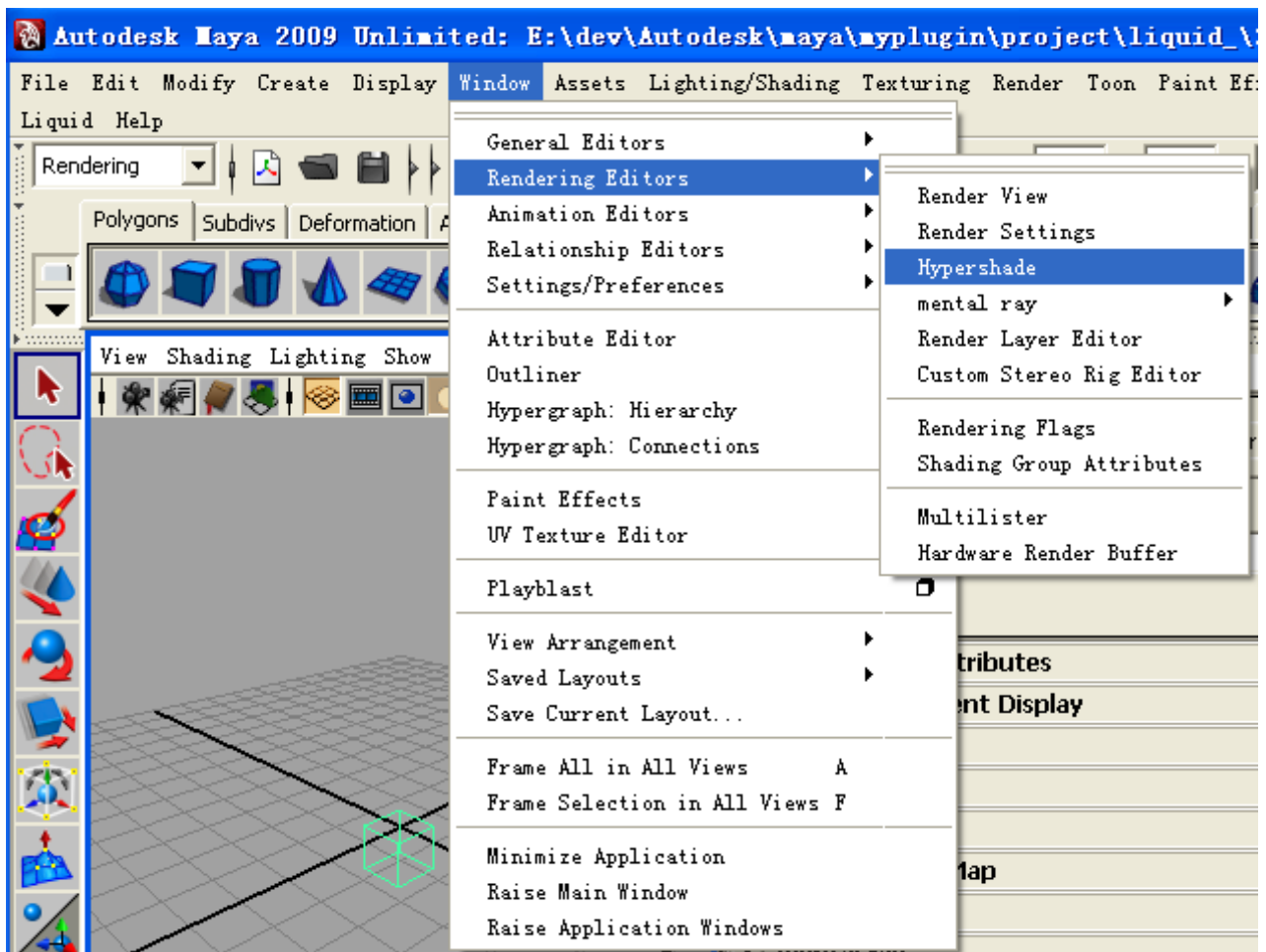
## Create the scene

1.create a box

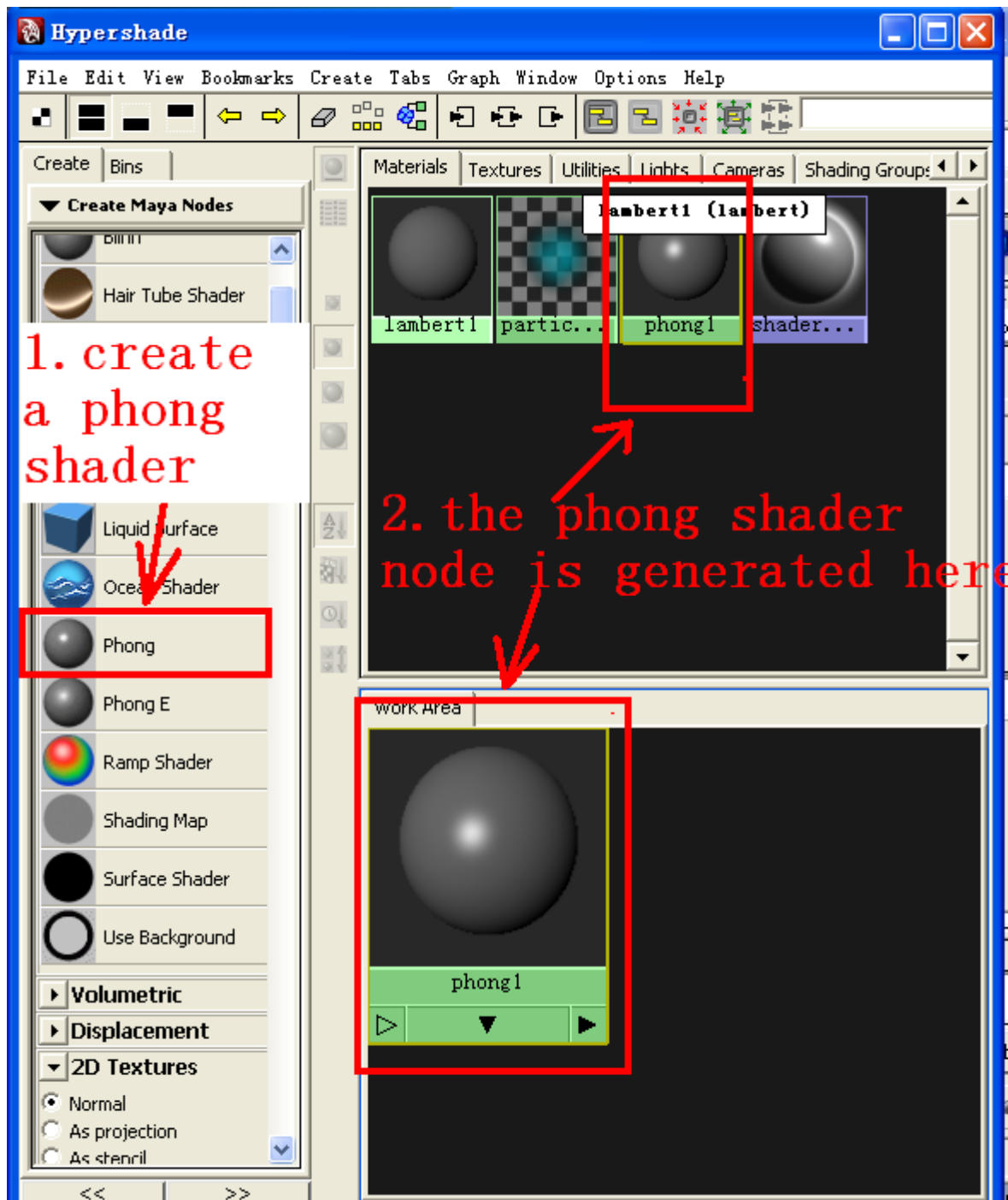


3. assigned a material to this box



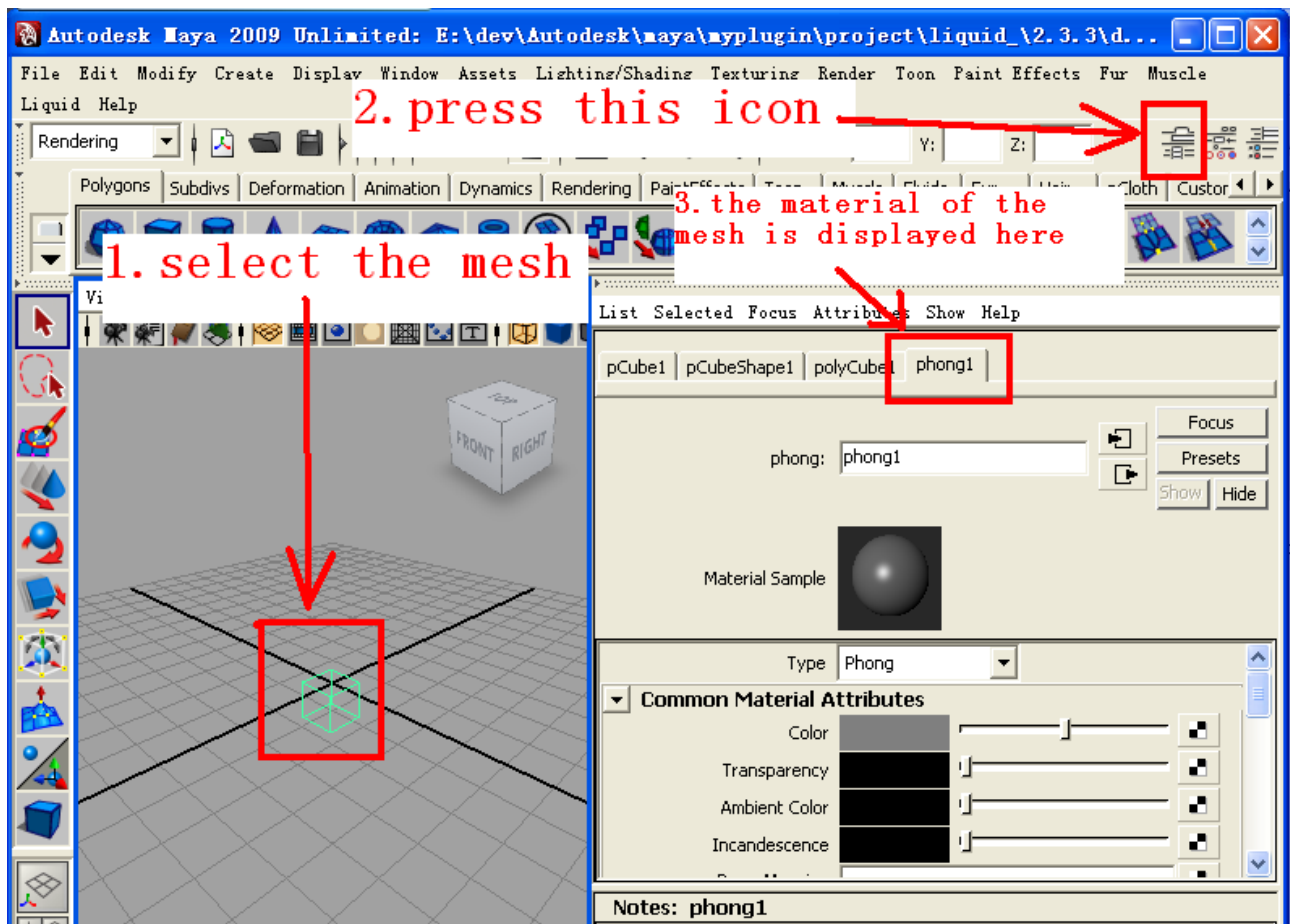


4. let's create a phong shader



5. select the mesh, then press(not click) the right mouse button on the phong1 node, and select "Assign Material To Selectedion."

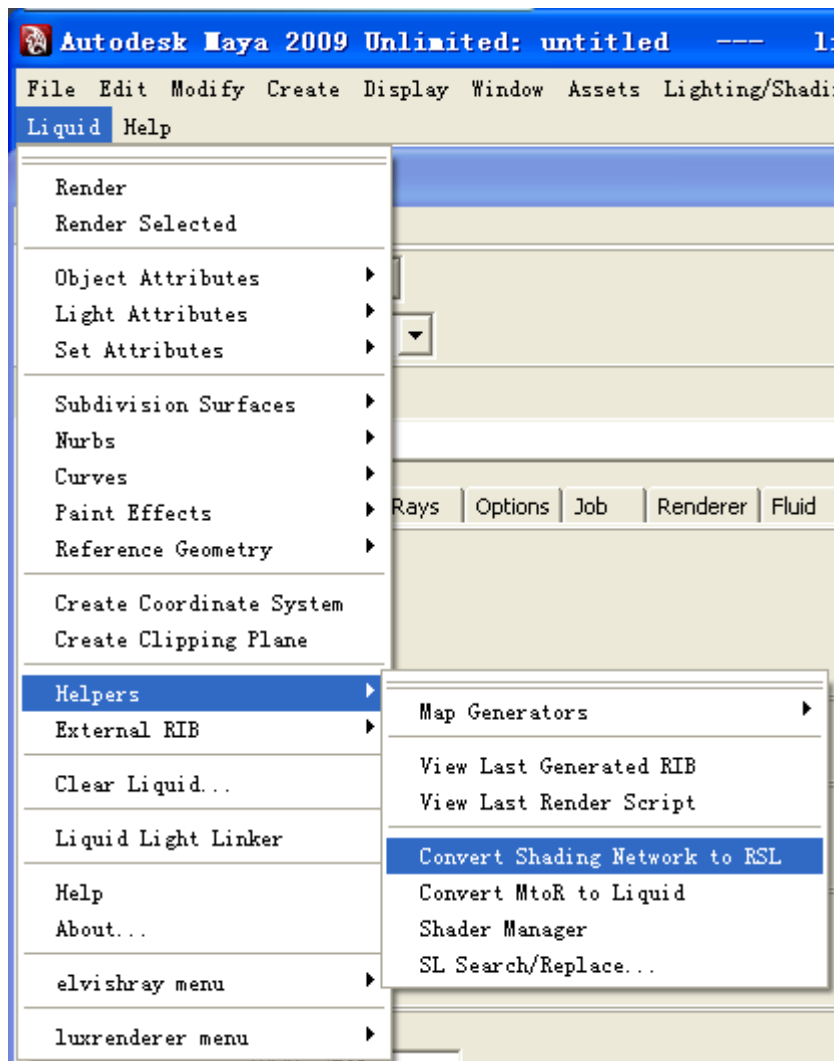
6. you can see the phong1 is assigned to the mesh.



## export the material of the mesh

In liquidMaya you MUST export the material of the mesh before you render that mesh. This is the steps:

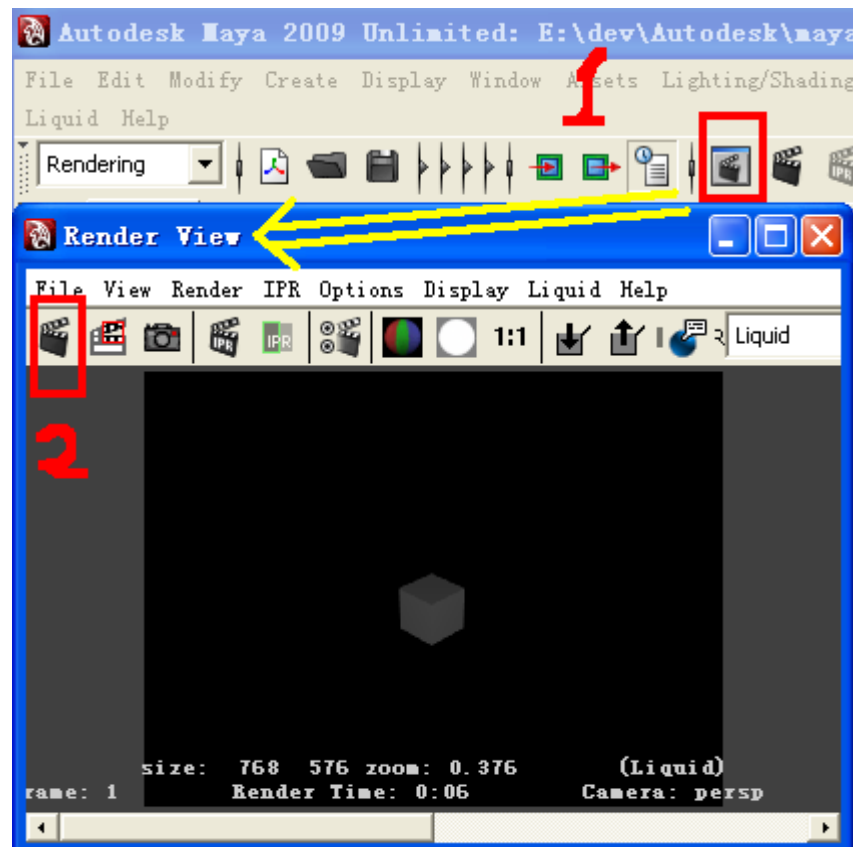
- make sure "GeneratedShaders" directory is set.
- select the mesh which you want to render
- Menu --> liquid --> Helpers --> Convert shading network to RSL:



phong1.sl and phong1.slo will be generated at  
 E:\MyDocuments\maya\projects\default\generatedShader\

## Render

Press the "RenderView" button, and press "Render" button:

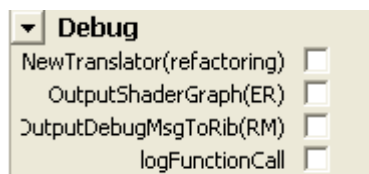


The rib file will be generated at E:\MyDocuments\maya\projects\default\rib, (e.g. E:\MyDocuments\maya\projects\default\rib\\_perspShape.0001.rib)

## -RenderSettings of maya2renderer

Demo scene is (\$liquid\_root)/2.3.3/doc/HowToUseMaya2Renderer.ma

### -Debug



### - NewTranslator(refactoring)

This is a new feature of maya2renderer.

I'm refactoring liquidMaya, and put the new export process into liqRibTranslator::\_doItNew();

The original export process is put into liqRibTranslator::\_doIt();

If NewTranslator(refactoring) is checked, liqRibTranslator::\_doItNew() is executed, otherwise liqRibTranslator::\_doIt() is executed.

**If you set NewTranslator(refactoring) to true, you are using maya2renderer,**

**If you set NewTranslator(refactoring) to false, you are using liquidMaya,**

## **- OutputShaderGraph(ER)**

This is a new feature of maya2renderer.

In liquidMaya you MUST export the material of the mesh before you render that mesh.

In maya2Renderer, if you set OutputShaderGraph(ER) to true, you don't have to convert the material manually. The materials will be converted automatically.

## **- OutputDebugMsgToRib(RM)**

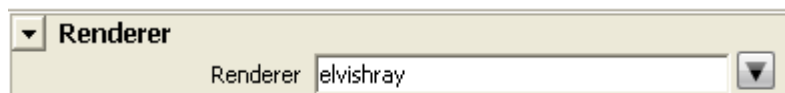
This is a new feature of maya2renderer. `_liqRIBMsg(const char* msg)` will output the msg to rib file.

## **- logFunctionCall**

This is a new feature of maya2renderer. `_logFunctionCall(const char* msg)` will output the msg. It is useful to trace the call stack in batchrender mode.

## ***-Renderer***

You can select other renderers in maya2renderer, e.g. elvisray.

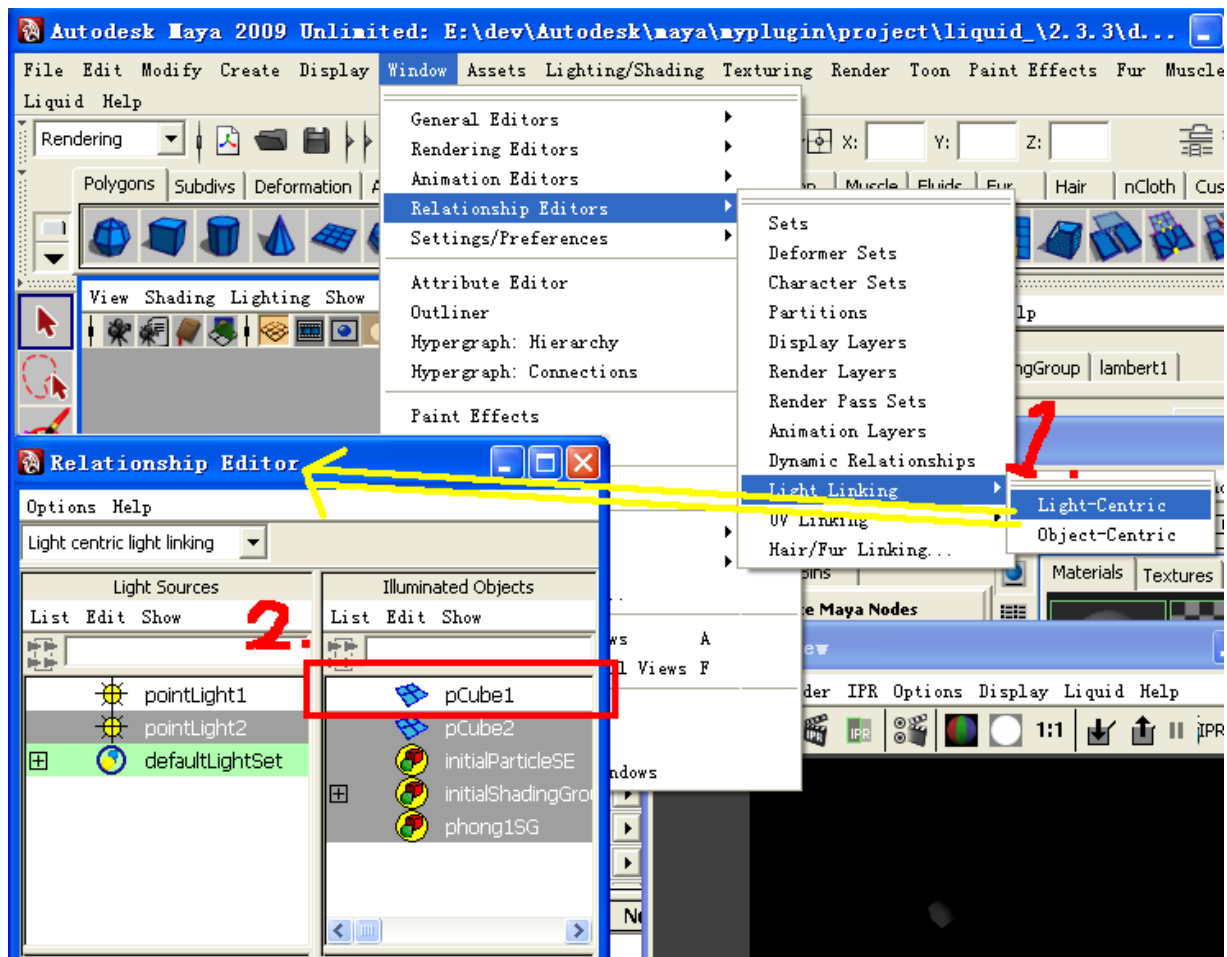


## ***Lightlink***

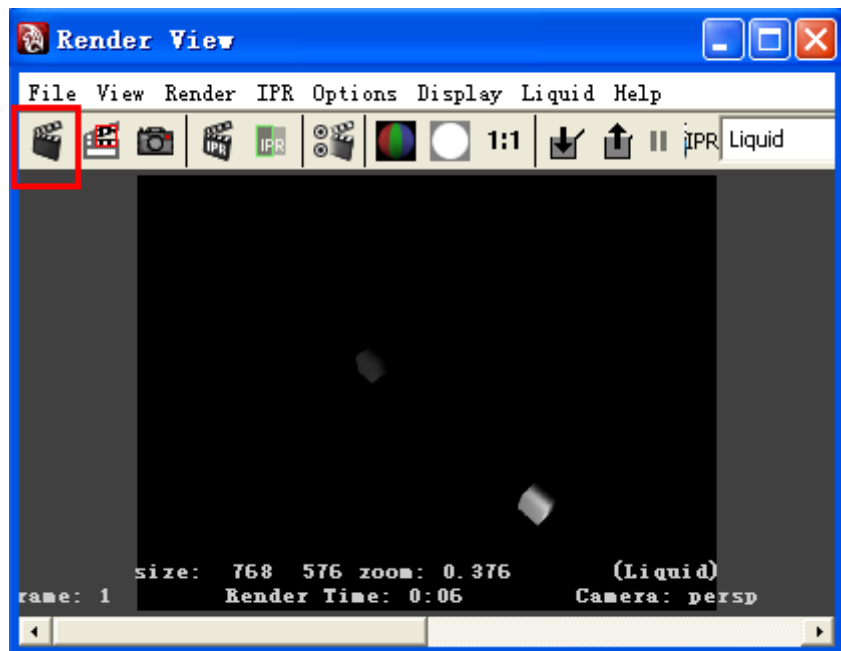
If all meshes are lighted by all lights, liquid will not output the light link for each mesh. In renderman, this is not a bug, because a mesh is lighted by all lights by default. But this is a bug in elvisray.

This bug will be fixed in the future. For now you have to do the following steps to avoid this bug.

- 1) add another mesh(e.g. Cube2) and another light(e.g. Light2)
- 2) open "Relationship Editor", and set light2 unlight Cube2:



## Render



It will generate \*.erapi at E:\MyDocuments\maya\projects\default\generatedShader\.

