

How to use Maya2Renderer

## **- Development environment**

see ./readme.txt

## **- Build**

see ./readme.txt

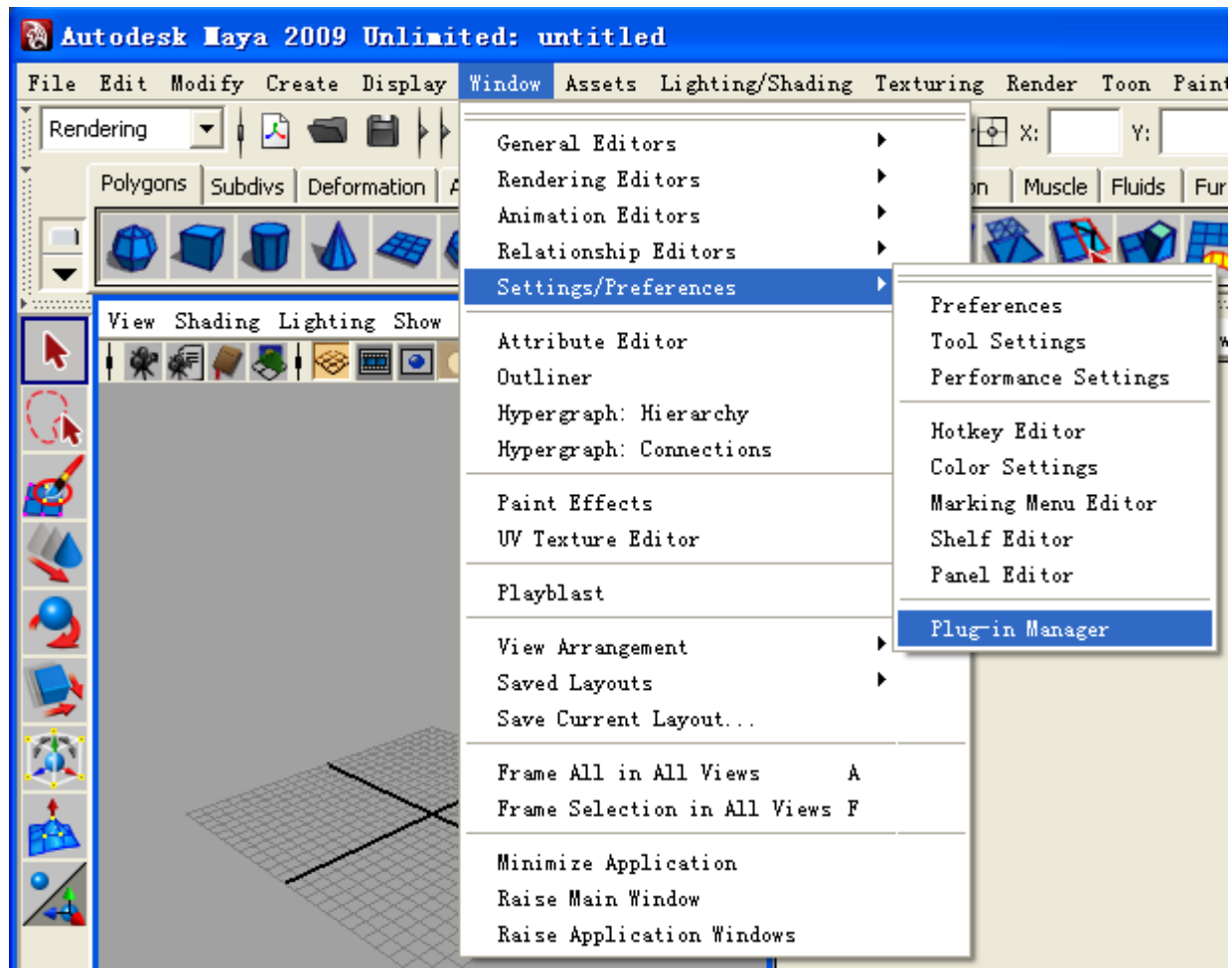
## **- Install**

see ./readme.txt

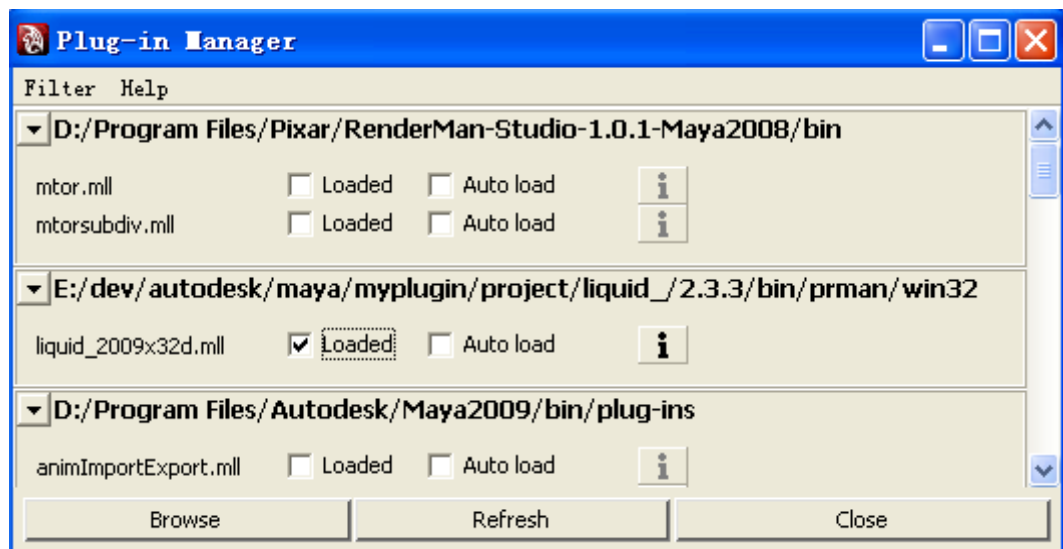
## **-load liquid plugin**

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

1)

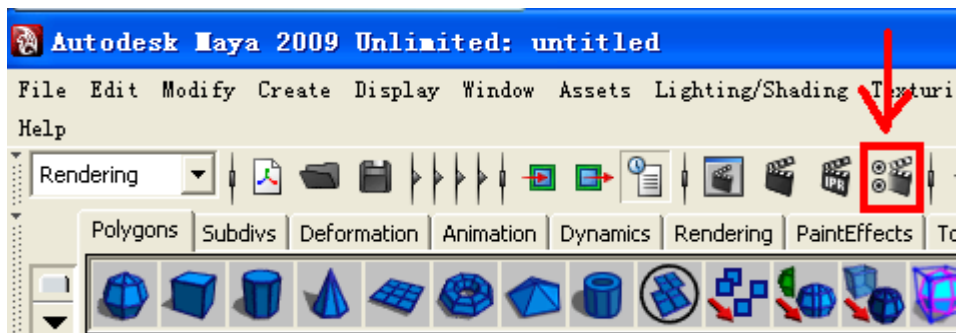


2) select *liquid\_2009x32d.mll*

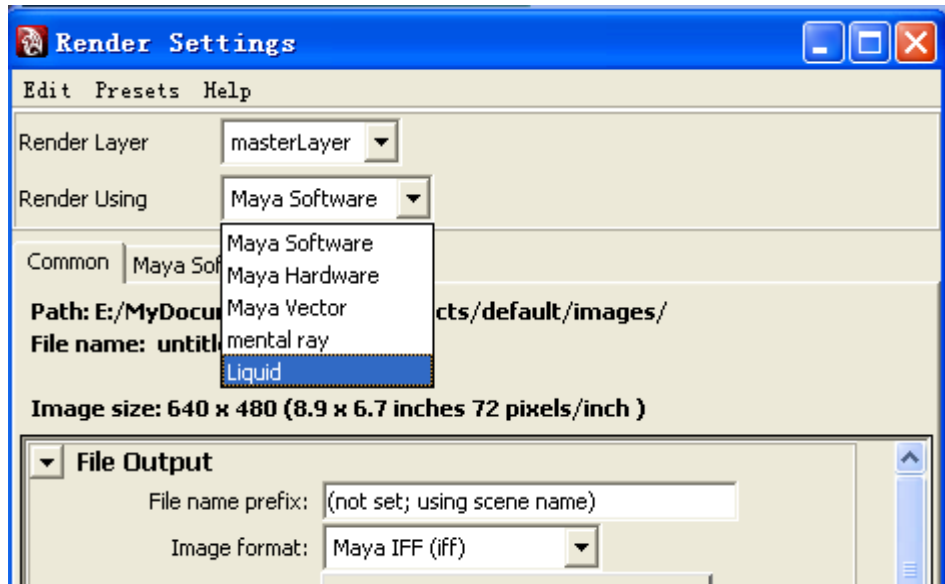


-select liquid renderer

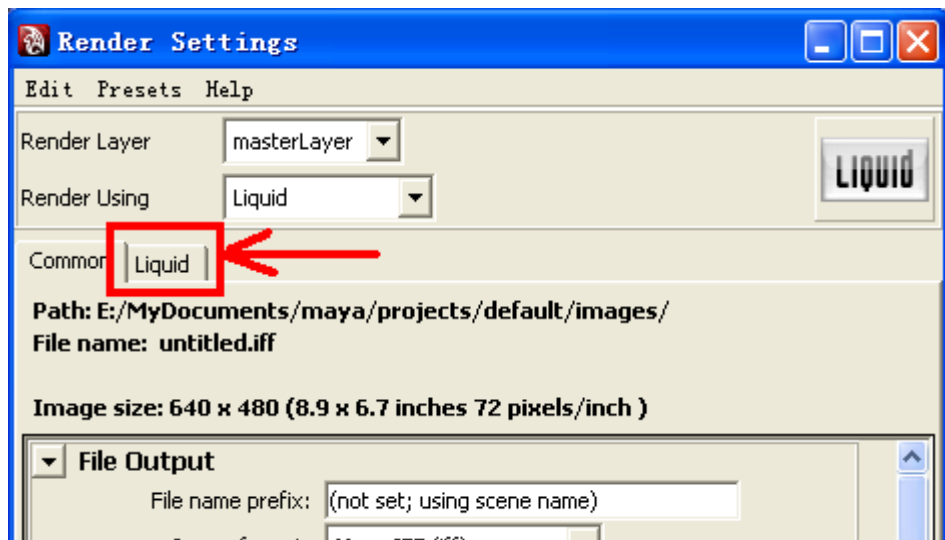
1)



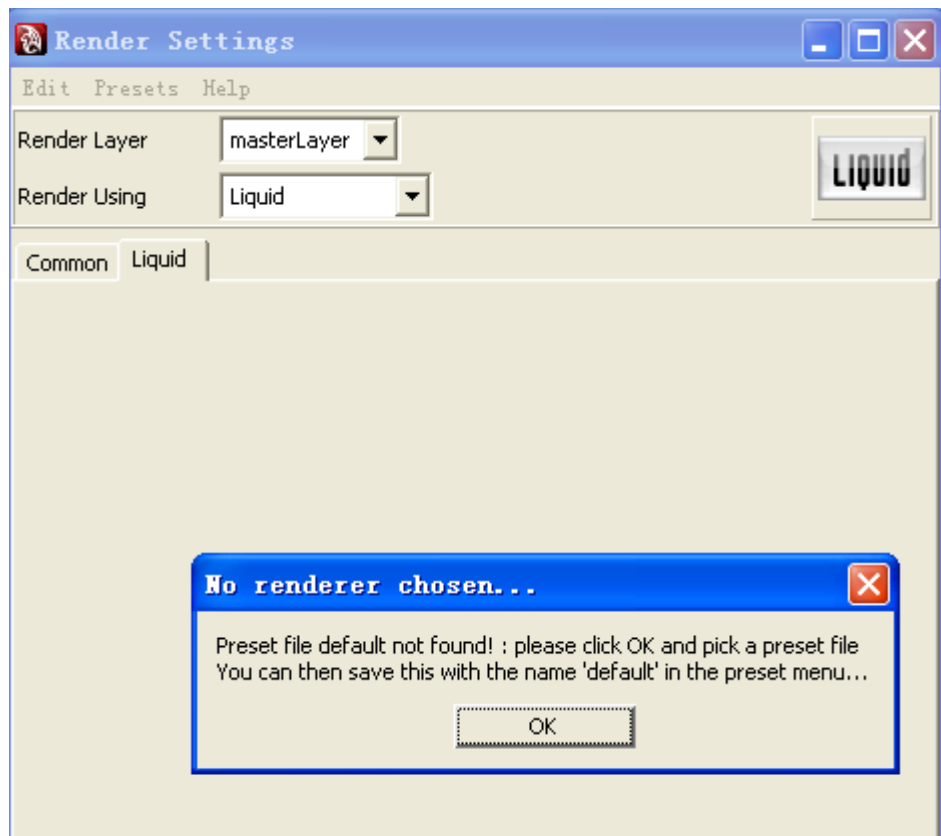
2)



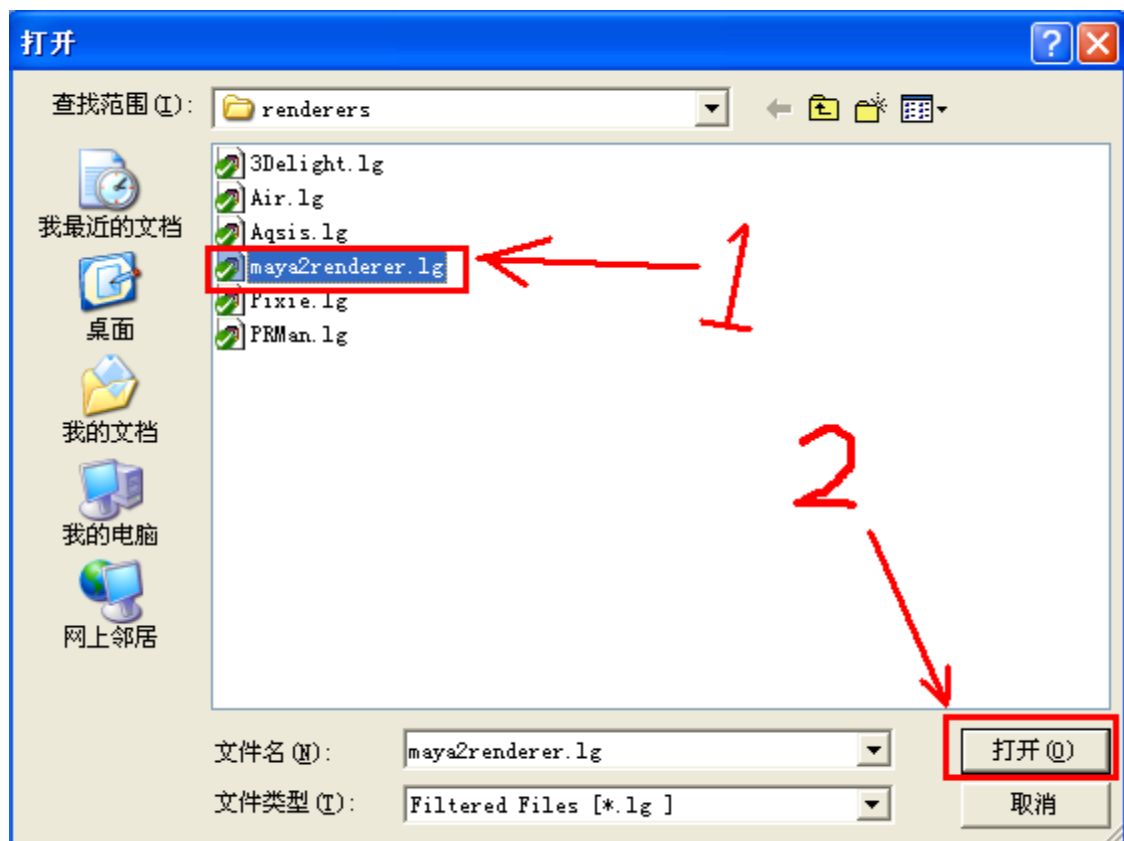
3)



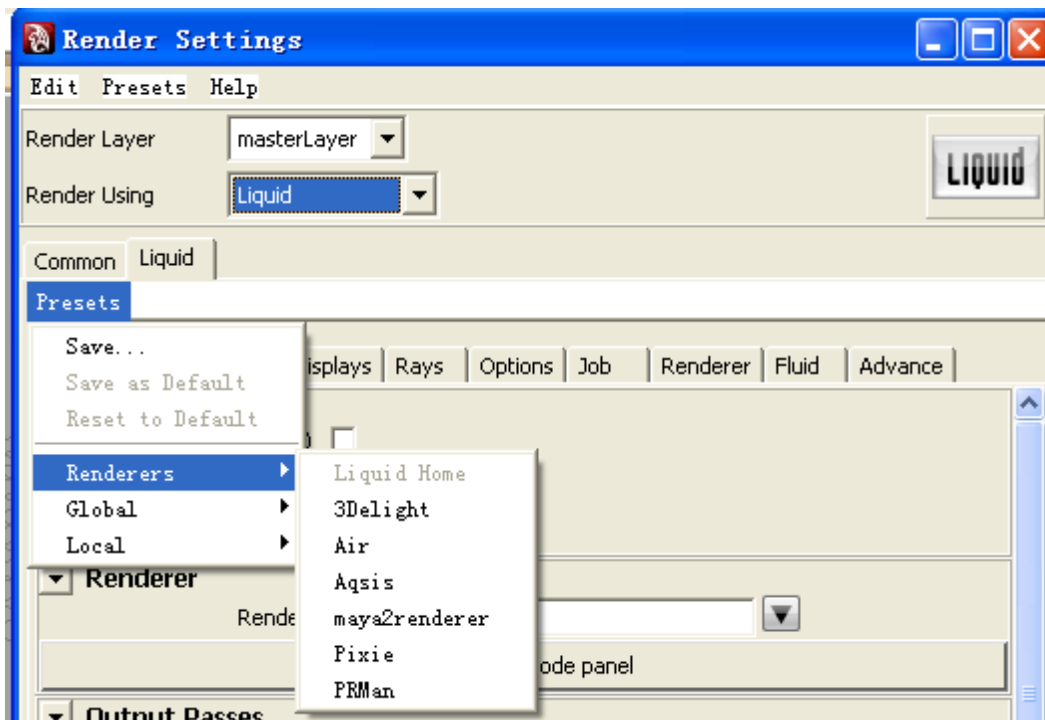
4)



5)



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



)

## -RenderSettings of maya2renderer

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

### -Debug



#### 1. 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,**

#### 2. 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.

#### 3. OutputDebugMsgToRib(RM)

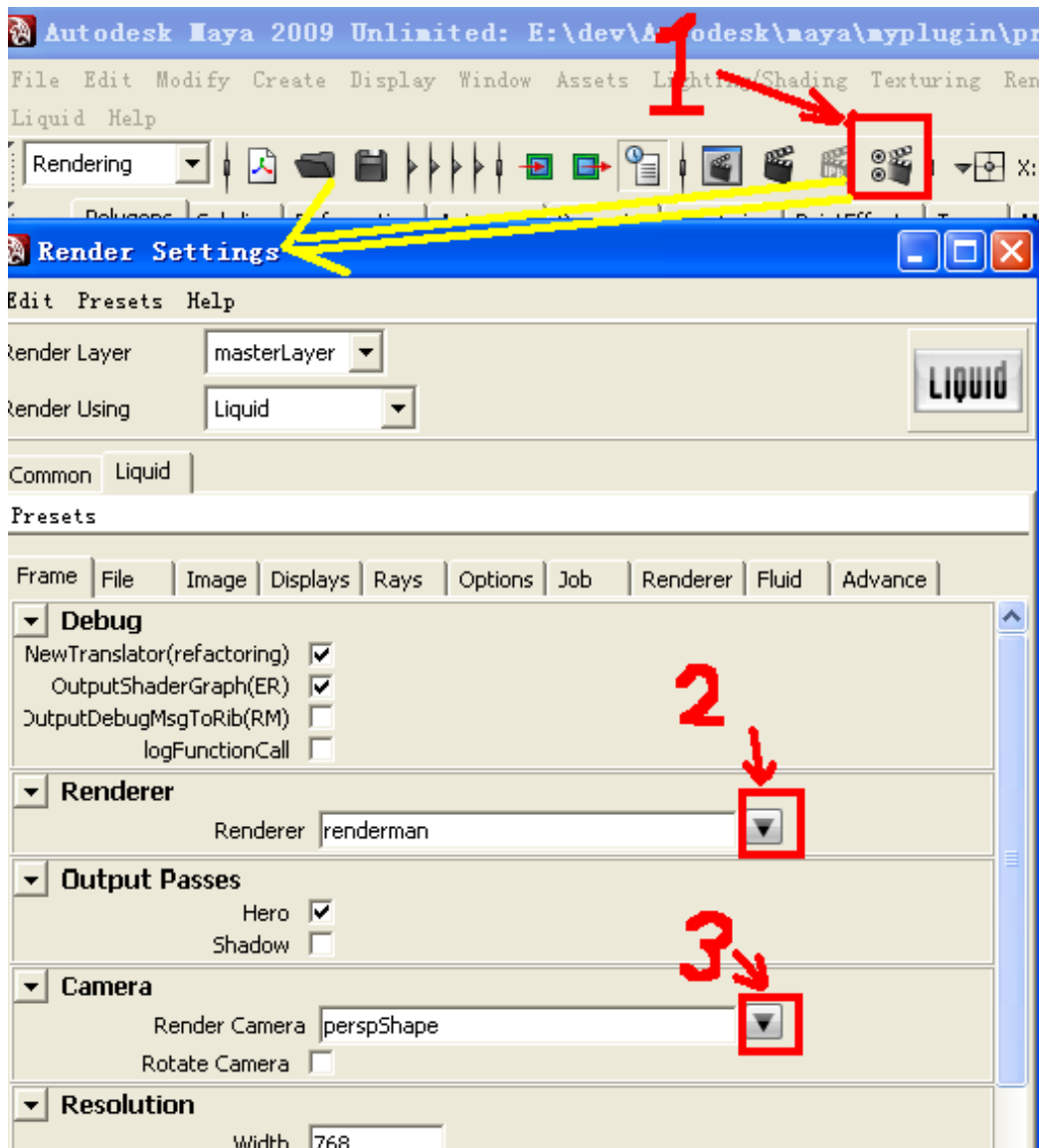
This is a new feature of maya2renderer. \_liqRIBMsg(const char\* msg) will output the msg to rib file.

#### 4. logFunctionCall

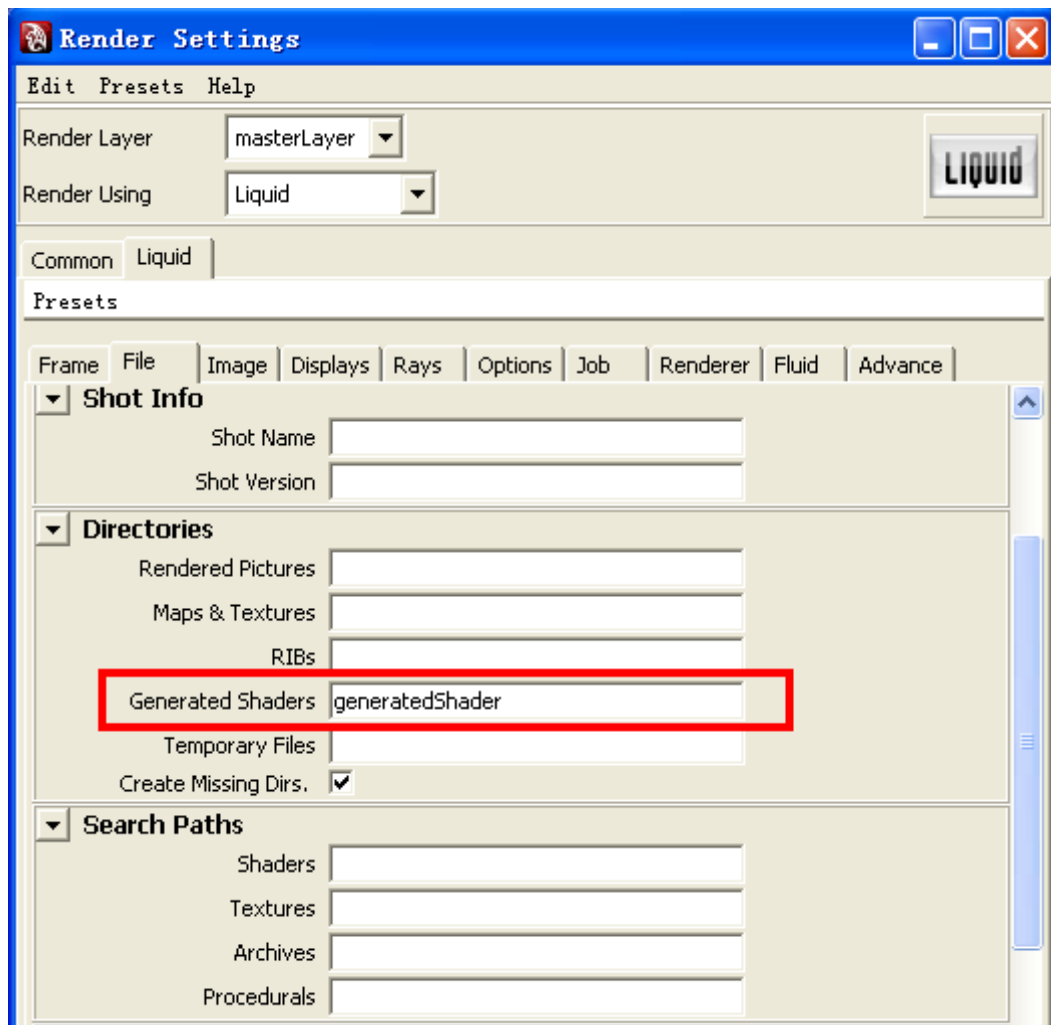
This is a new feature of maya2renderer. `_logFunctionCall(const char* msg)` will output the msg. It is useful to trace the call stack in batchrender mode.

#### 7) 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)

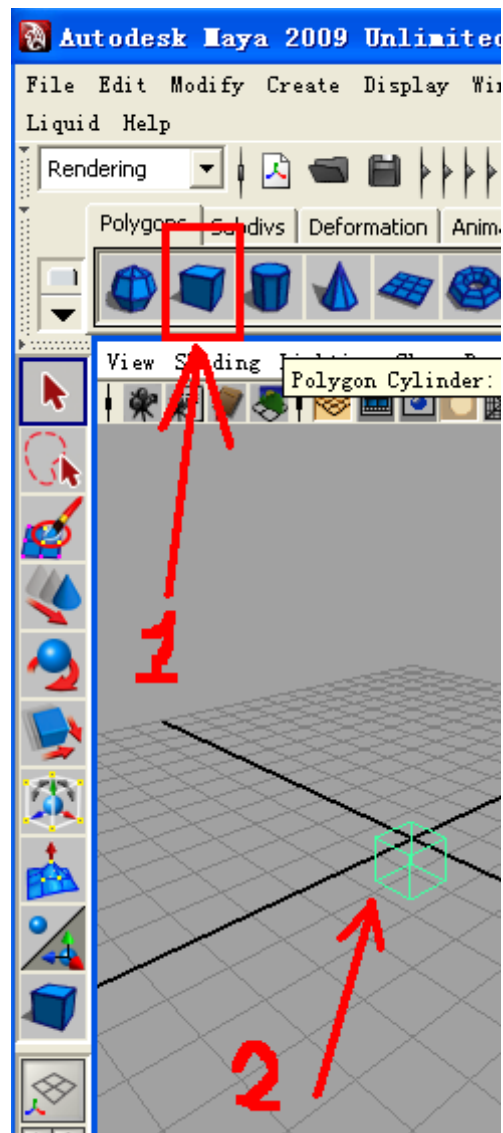


#### 8) set GeneratedShaders directory



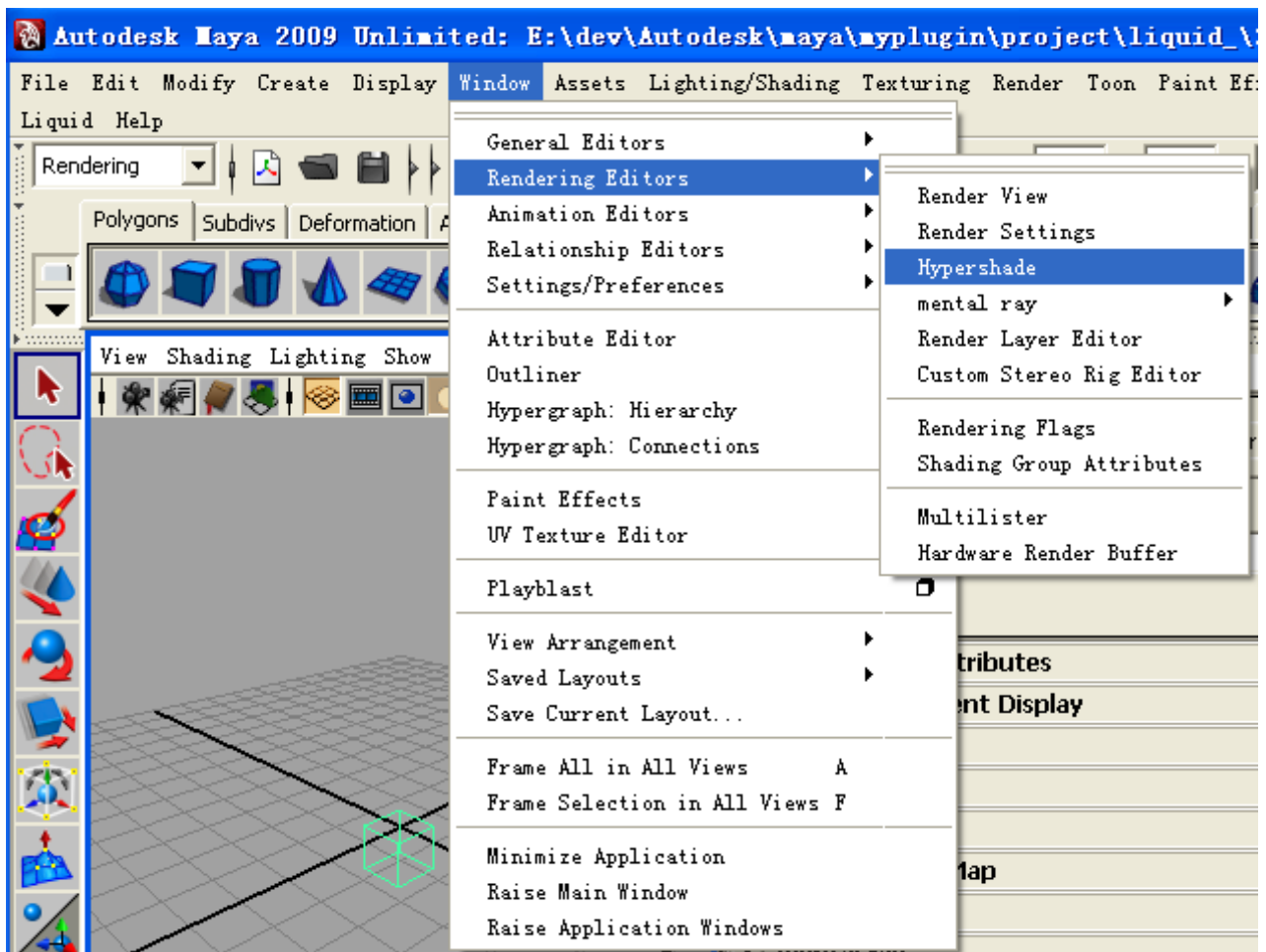
## -Create the scene

1.create a box

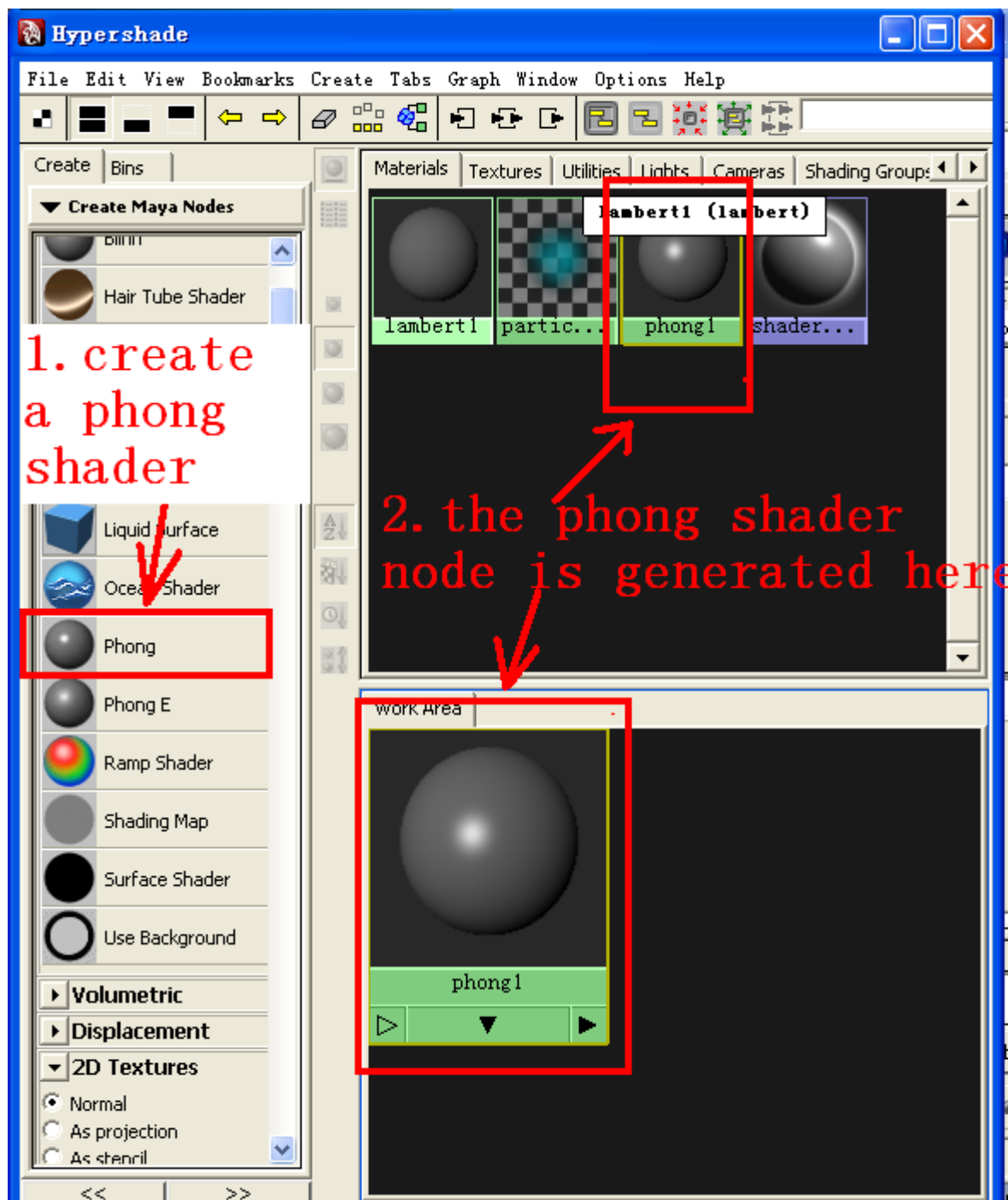


3. assigned a material to this box



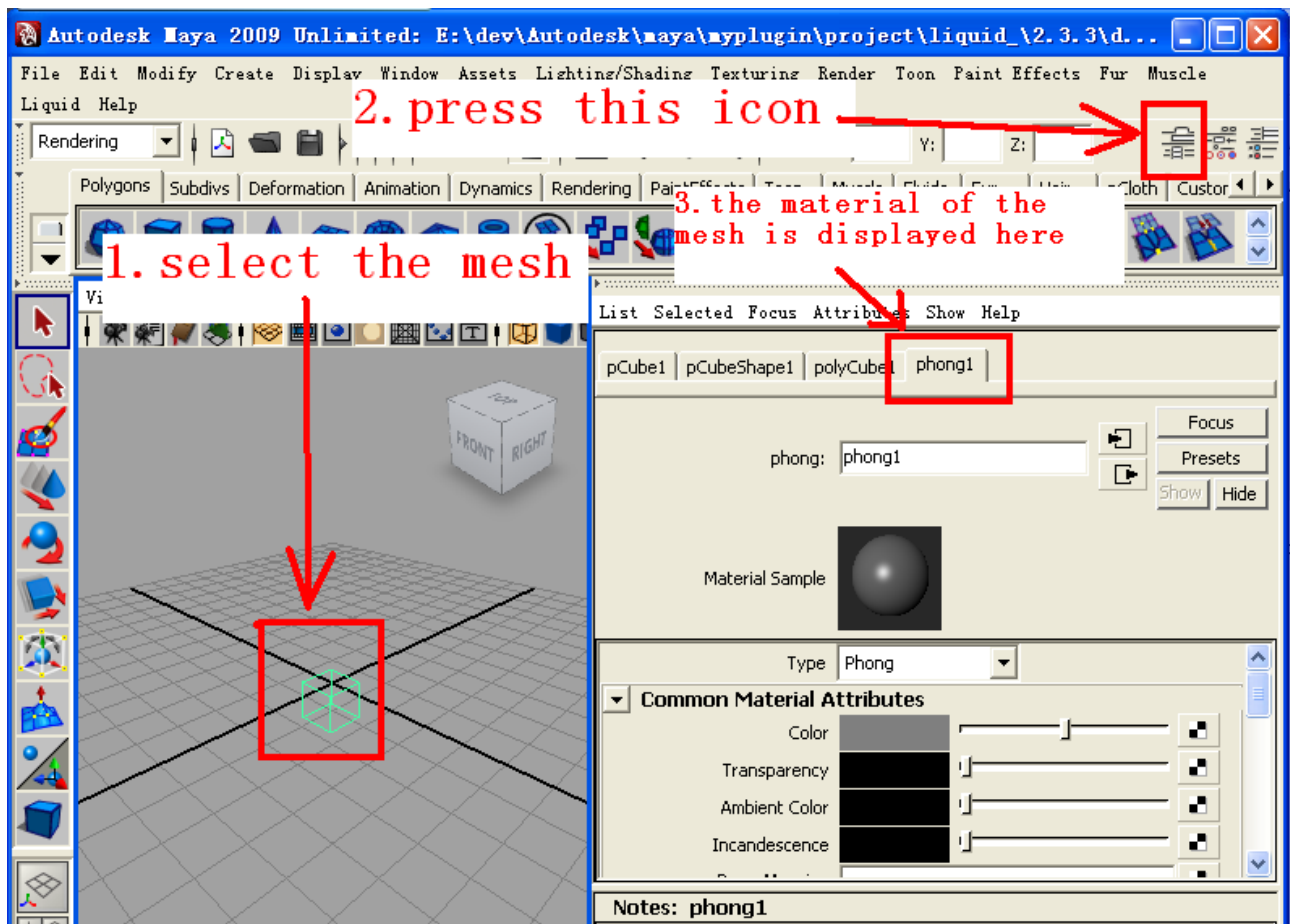


4. let's create a phong shader



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

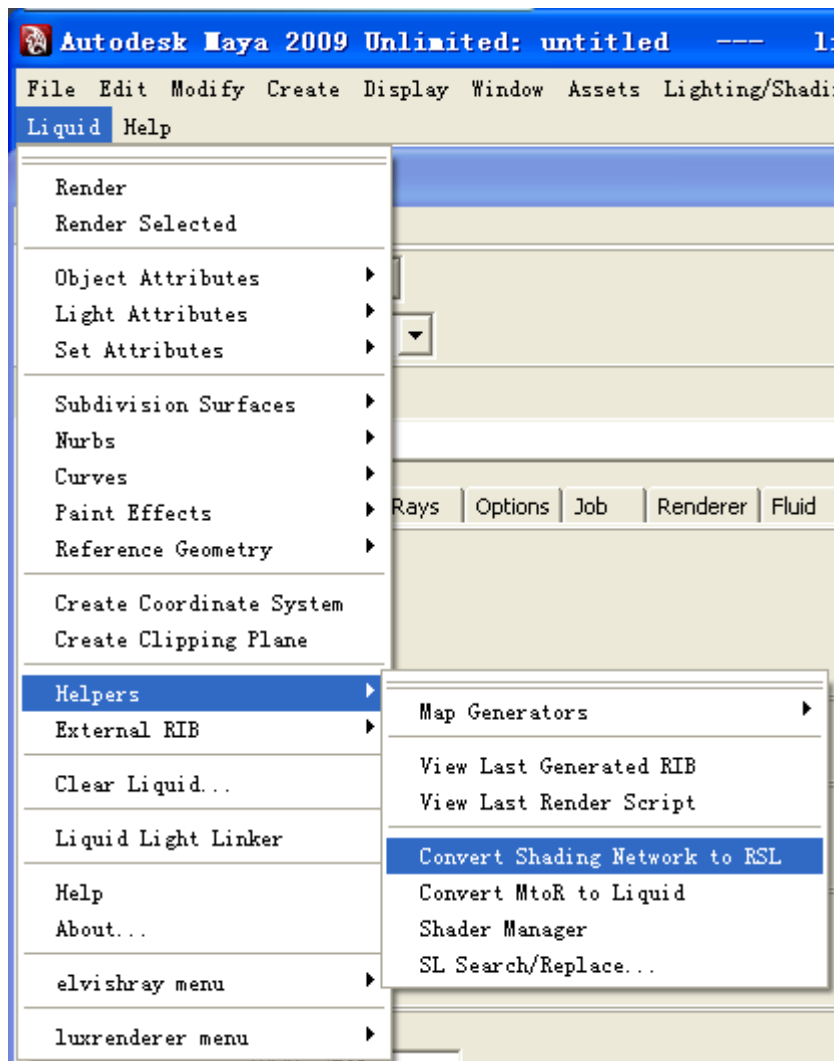
6. you can see the phong1 is assigned to the mesh.



## -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:

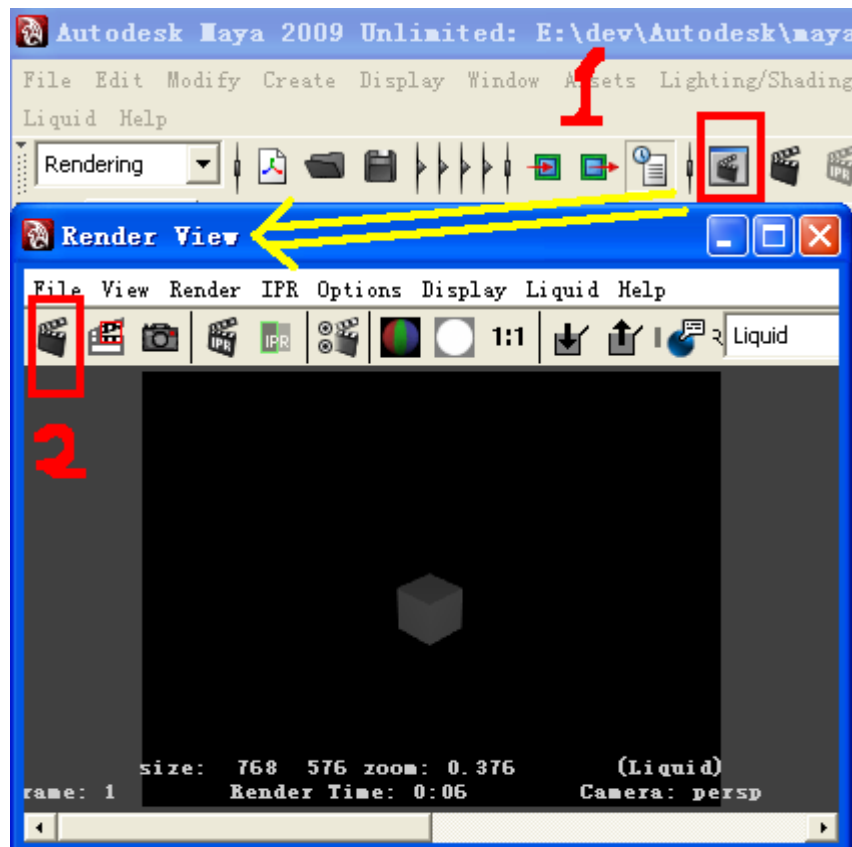
- make sure "GeneratedShaders" directory is set.
- select the mesh which you want to render
- Menu --> liquid -->Helpers --> Convert shading network to RSL:



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

## -Render

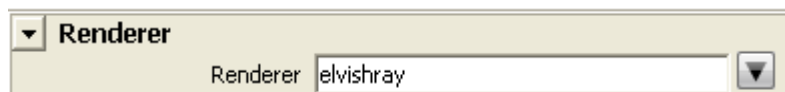
Press the "RenderView" button, and press "Render" button:



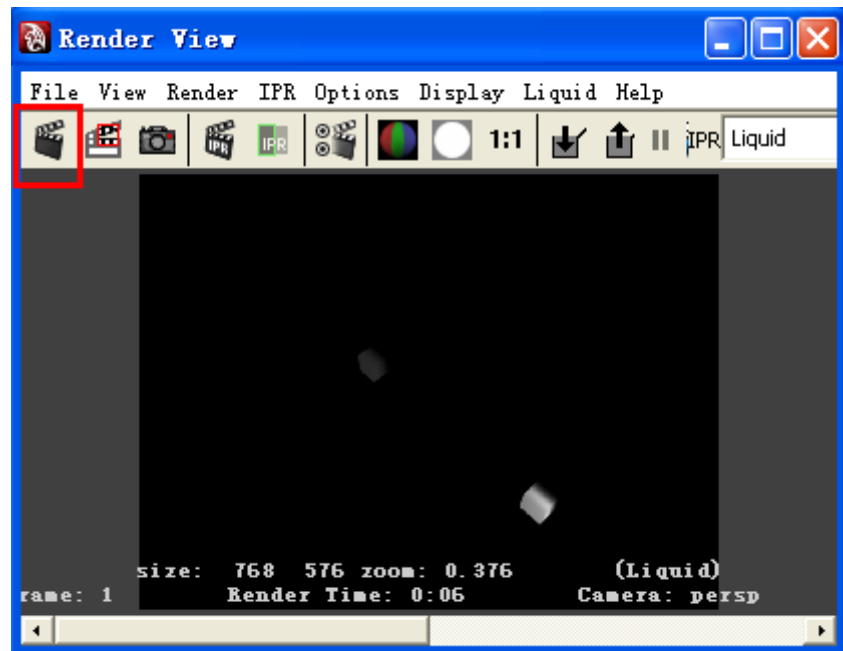
The rib file will be generated at E:\MyDocuments\maya\projects\default\rib, (e.g. E:\MyDocuments\maya\projects\default\rib\\_perspShape.0001.rib)

### **-Renderer**

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



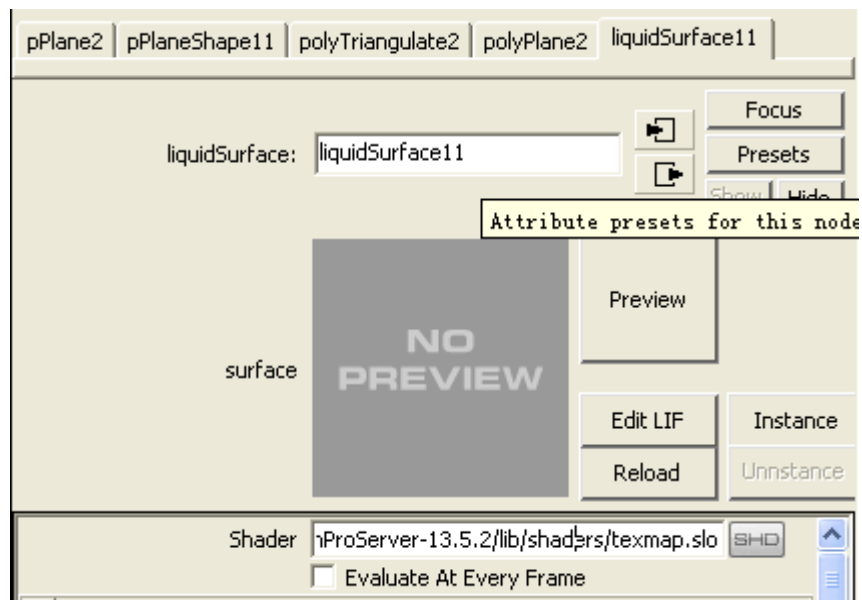
## Render



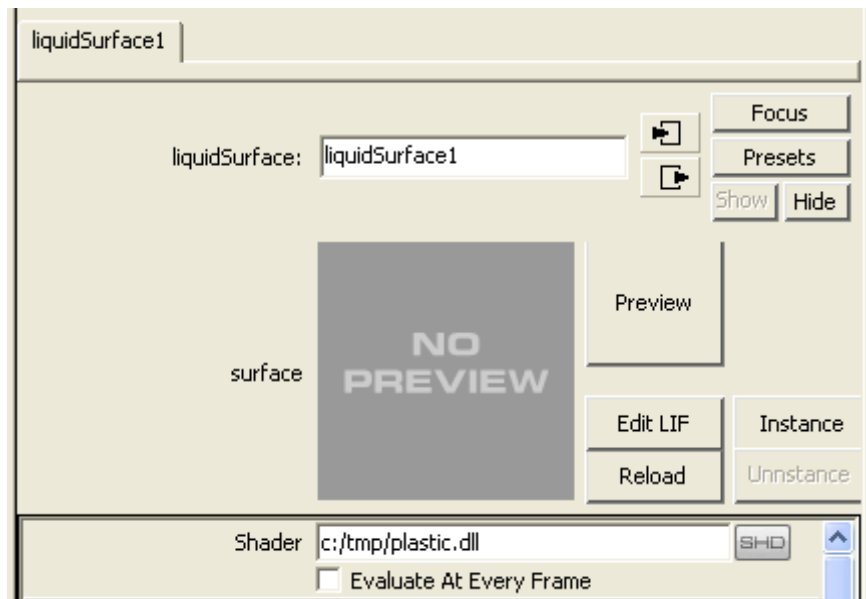
It will generate <file:///<MayaProjectDirectory>\rib\<SceneName>.er> which trace the invoking of ERAPI

### - *liquidSurfaceShader*

- In liquidMaya, liquidSurfaceShader is used for user-defined surface shader( it is \*.slo for prman)



- In maya2renderer, liquidSurfaceShader is extended to supported user-defined shader for other renderers.



If you use prman, you set the Shader to a slo file, and don't have to do anything else.

If you use elvishray, e.g. In liquidSurface1, if you set Shader to <c:/tmp/plastic.dll>, you **have to create** <c:/tmp/plastic.pl>. Because maya2renderer will search <c:/tmp/plastic.pl>, and read this pl file to generate shader parameters automatically.

## - PL file

pl file contains shader type, shader parameter list.

pl file must locates at the same directory as the shader file. For elvishray, the shader file is a dll file.

pl file is used to generate the UI of parameters of user-defined shader automatically in Maya. The following image show this process in details

[%liquidRoot%\2.3.3\doc\HowToGenerateShaderParameterUI\HowToGenerateShaderParameterUI.png](#)

or

<https://github.com/maya2renderer/maya2renderer/blob/master/2.3.3/doc/HowToGenerateShaderParameterUI/HowToGenerateShaderParameterUI.png>

Here is an example of pl file:

```
#shaderType // shaderType: surface, volume, displacement, light. shaderType must be set before shadename
```

```
surface
```

```
#shaderName //shadename must be set after shaderType
```

```
liquidchecker
```

```
//shader parameter list. One line for each parameter,
```

```
#Name:Details>Type:IsOutput:Defaults:ArraySizes//Attributes of each parameter, and the attributes are seperated by colon.
```

```
frequency:uniform:float:0:8.5:-1
```

```
mode:uniform:float:0:0.0:-1
```

```
mode2:uniform:float:0:0.5:-1
```

