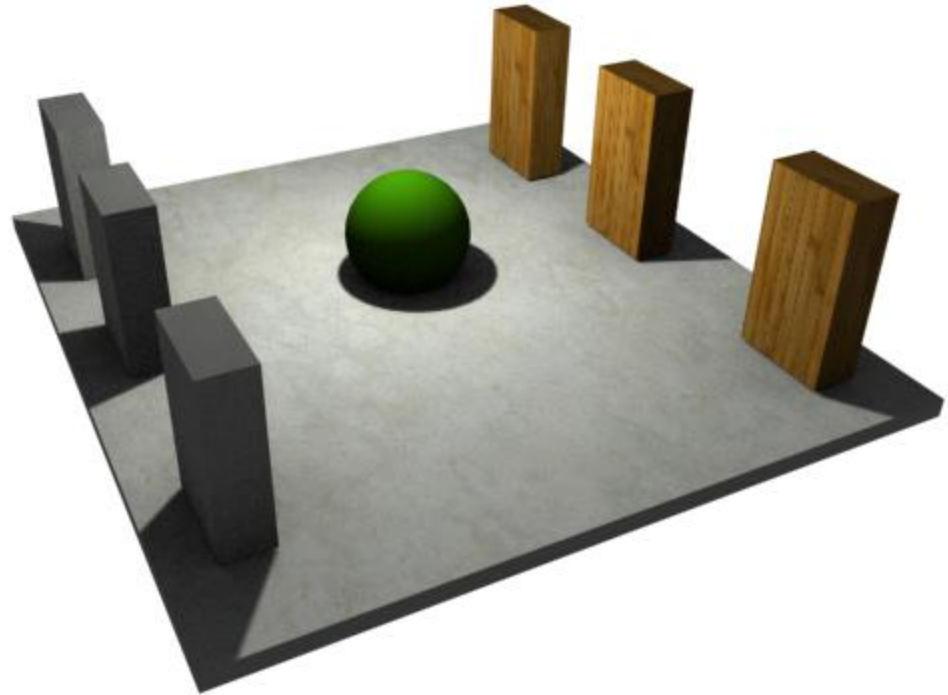


Vizard

3ds MAX Workflow

Topics covered:

- Render to texture
- Complete maps
- Light maps
- Export settings
- Vizard Inspector

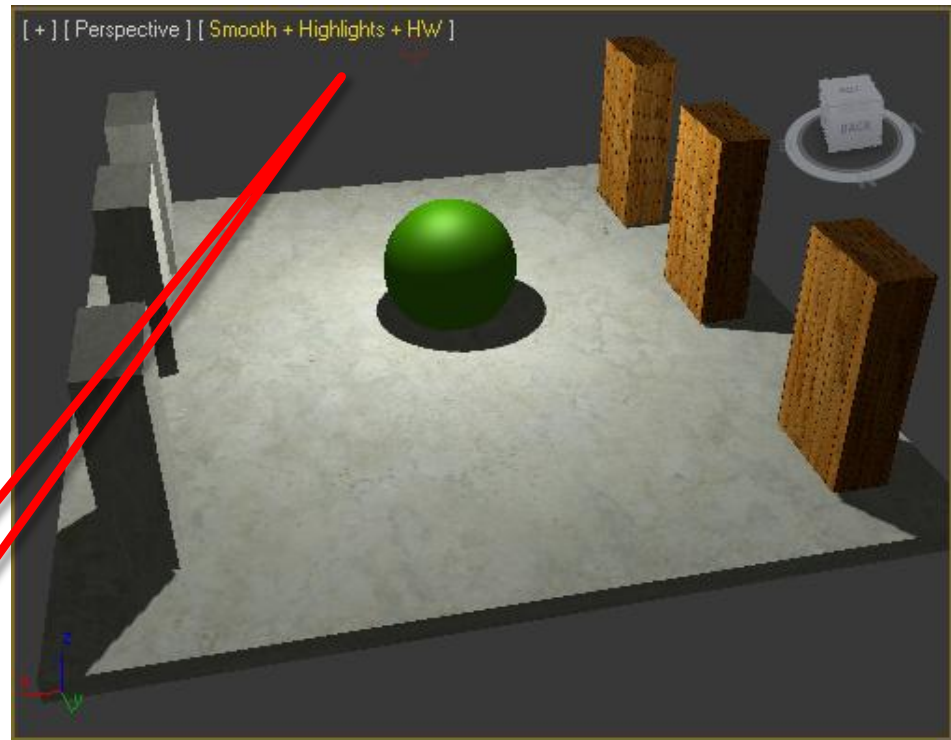


Set viewport render properties

Recent versions of max have added in a hardware renderer which gives you a realistic preview of the lighting in your scene.

The additional realism does come at the cost of performance, so for complex scenes, you'll need to back off these settings.

The hardware renderer is further improved in Max 2012.



Right-click for options

- Smooth + Highlights
- Hidden Line
- Wireframe
- Flat
- Edged Faces

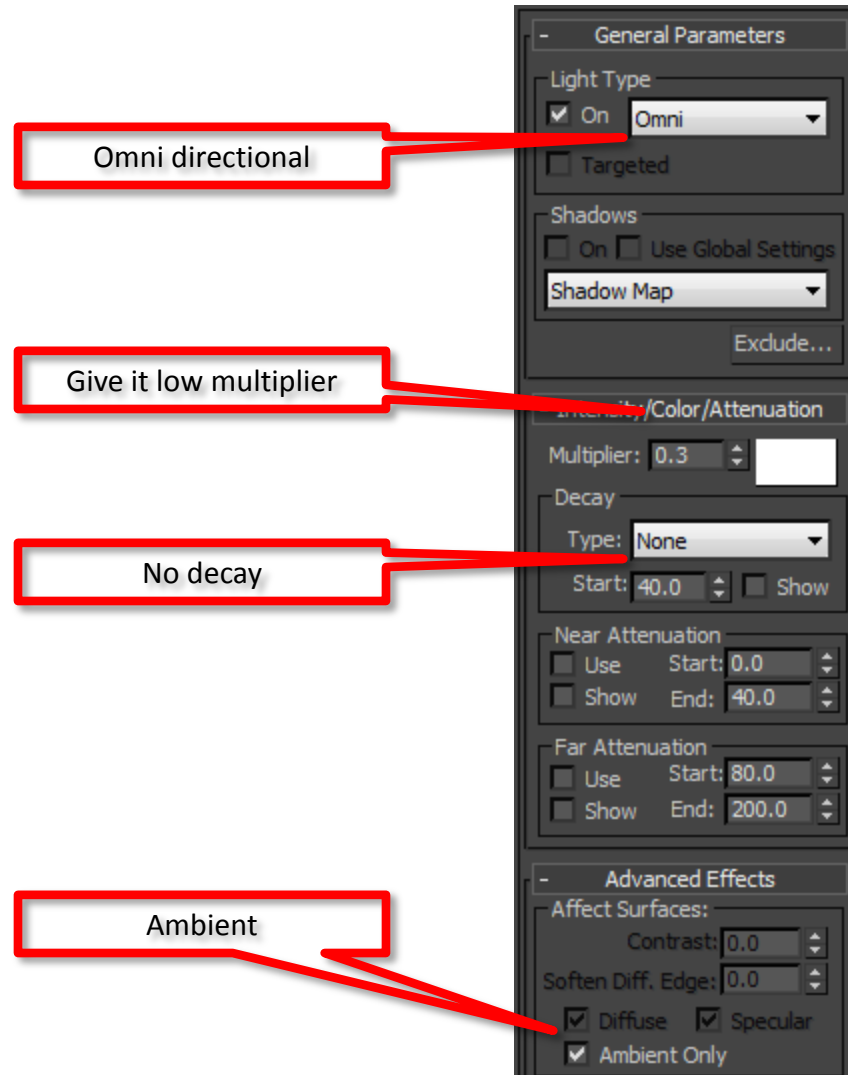
- Lighting and Shadows
 - Illuminate with Scene Lights
 - Illuminate with Default Lights
- Transparency
- xView
- Other Visual Styles
- Viewport Background
- Enable Hardware Shading Shift+F3
- Enable Exposure Control in Viewport
- Enable Shadows
- Enable Ambient Occlusion
- Configure...

Ambient Light

Without ambient light, it is difficult to see detail in shadowed areas. In the real world, direct light bounces off the environment, brightening up darker areas with indirect light (global illumination). Calculating this accurately requires a lot of time and processing power.

If you are baking a scene and care more about quick turnaround than photorealism, consider using a flat ambient light instead of physically accurate global illumination.

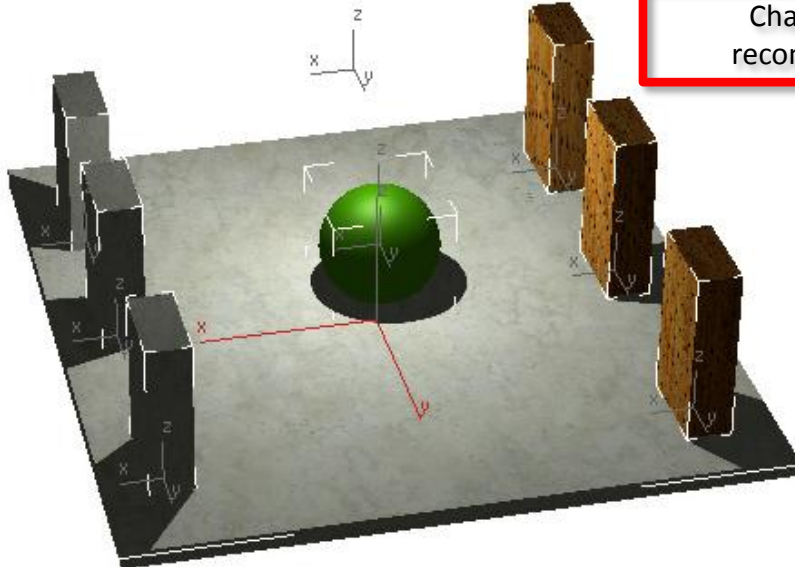
MAX's "Environment ... Global Lighting" ambient light does not export in Vizard IVE format. Instead, use an omni light in ambient mode, as in the graphic to the right.



Render to Texture (step 1)

From the viewport, select all objects you want to bake. These objects will show up in Render to Texture under “Objects to Bake”.

For consistency, bake Lightmaps and Completemaps to Mapping Coordinate channel # 3. (UVW Channel 3)

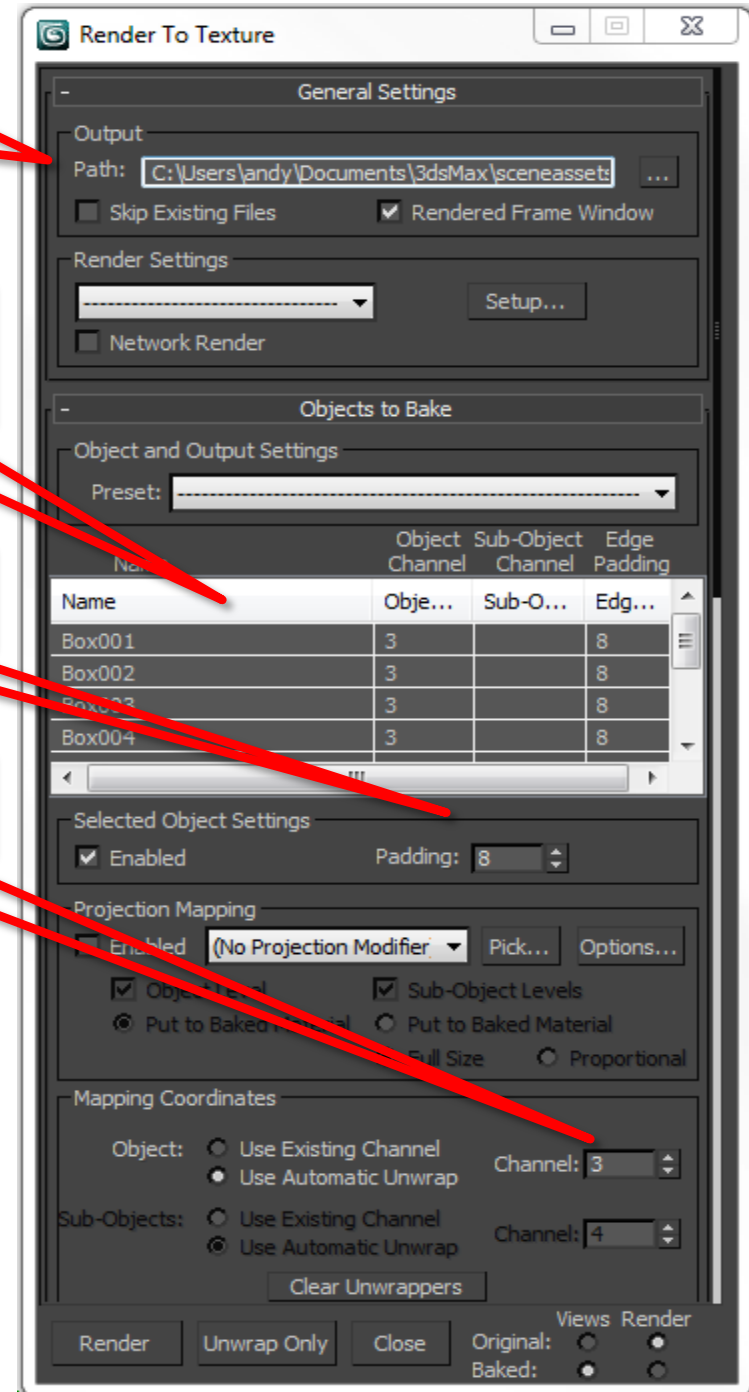


Pick the location where the baked images will be stored.

Selecting objects in viewport puts them in the “to bake” list

“8” or more pixels recommended

Channel “3” recommended



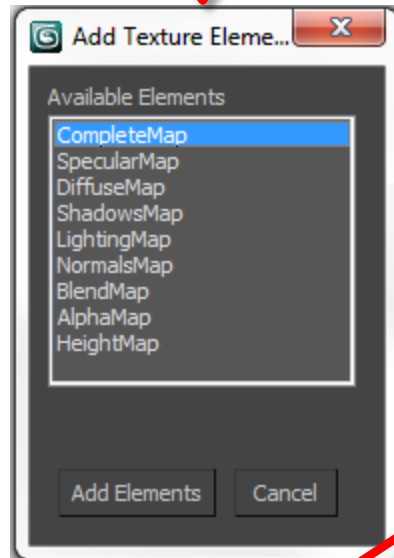
Render to Texture (step 2) CompleteMap method

Completemaps are a record of brightness and color stored inside an image file.

Completemaps are the most effective means of capturing lighting and materials into a static texture.

The main limitation is a lack of realtime reflection, though this can be achieved through other methods where necessary. Static textures cannot directly produce view dependent effects.

The resolution of close-up detail of your real-time scene will entirely depend on the size you choose for the rendered textures.

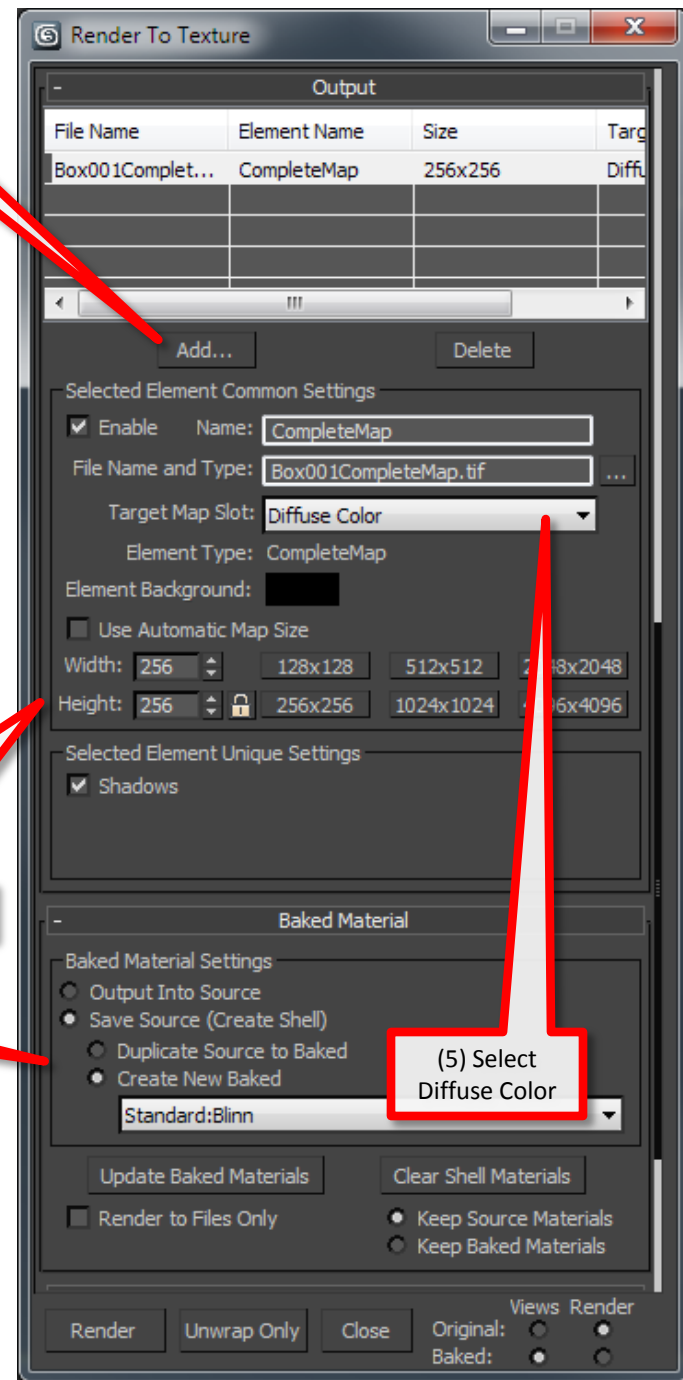


(1) Click Add ...

(2) Select "CompleteMap"

(3) Choose a map size for rendered textures

(4) Create new baked – Standard: Blinn



(5) Select Diffuse Color

Render to Texture (step 2) LightMap method

Lightmaps are a record of luminance values stored inside an image file.

Compared to Completemaps, lightmaps provide:

1. Versatility. Lightmaps keep lighting separate from surface color, allowing you to change diffuse maps without needing to rebake the scene.
2. Optimization. Map size only affects the quality of the shadows and light effects. Lighting is mostly composed of smooth gradients, so it does not need as much close-up detail as a diffuse texture.

(1) Click Add ...

(2) Select "LightMap"

(3) Choose a map size for rendered textures

(4a) If you use all Standard materials, use Duplicate Source to baked.
(4b) Otherwise, use Create New Baked: Standard Blinn.

(5) Select Self-Illumination

(6) If you had to do step 4b, you will need to copy each of your diffuse textures to the new baked materials.

Render to Texture

File Name	Element Name	Size	Target
Box001Lighting...	LightingMap	256x256	Self-Illumination

Add Texture Element

Available Elements

- CompleteMap
- SpecularMap
- DiffuseMap
- ShadowsMap
- LightingMap
- NormalsMap
- BlendMap
- AlphaMap
- HeightMap

Render to Texture Settings

Output

Selected Element Common Settings

- Enable Name: LightingMap
- File Name and Type: Box001LightingMap.tif
- Target Map Slot: Self-Illumination
- Element Type: LightingMap
- Element Background: [Color]
- Use Automatic Map Size
- Width: 256 (128x128, 512x512, 1024x1024, 2048x2048)
- Height: 256 (256x256, 1024x1024, 4096x4096)

Selected Element Unique Settings

- Shadows
- Direct Light On
- Indirect Light On

Baked Material

Baked Material Settings

- Output Into Source
- Save Source (Create Shell)
- Duplicate Source to Baked
- Create New Baked
- Standard:Blinn

Update Baked Materials Clear Shell Materials

Render to Files Only Keep Source Materials Keep Baked Materials

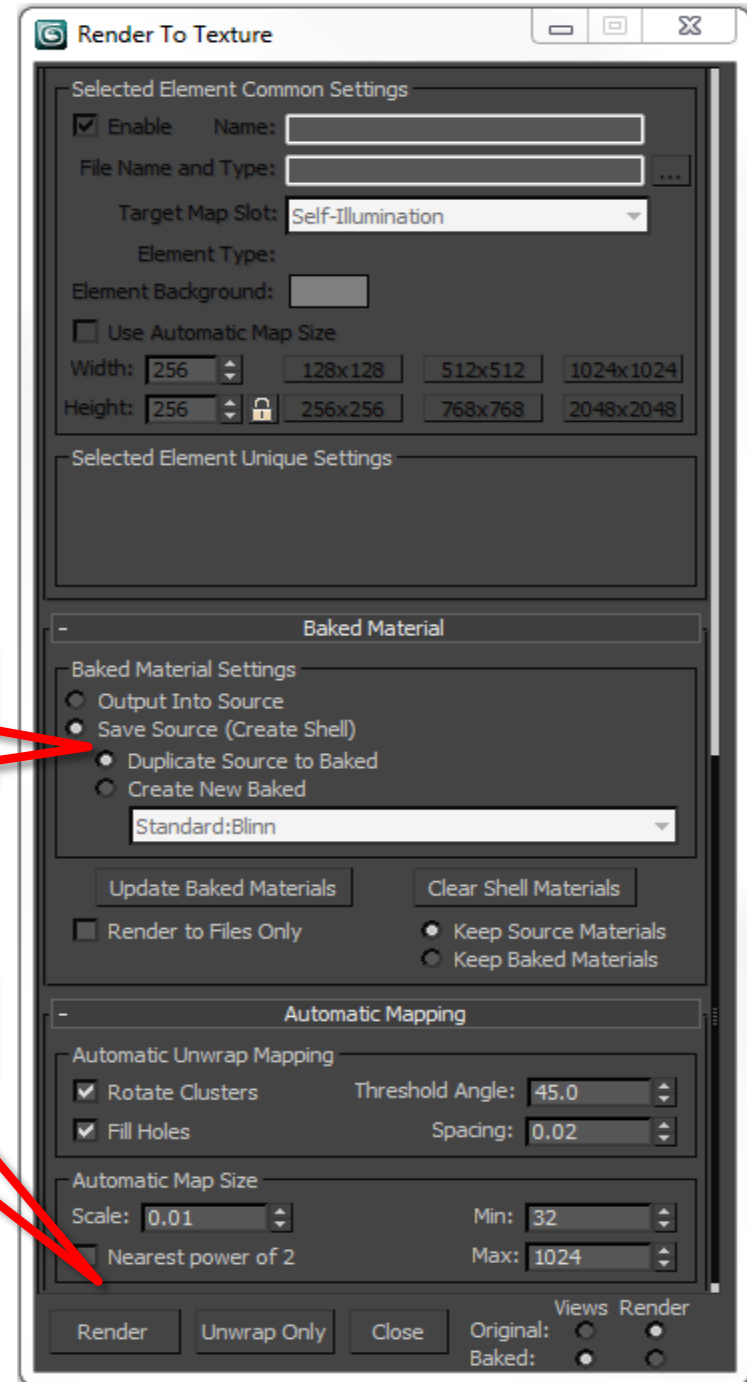
Render Unwrap Only Close Views Render Original: Baked:

Render to Texture (step 3) Baked Material Settings

When “baking” your rendered scene into textures, instruct MAX to save the result in a new shell material. When you export your scene to Vizard, the exporter will choose any baked shell materials over your original materials when available.

Preserve your original maps by creating a new shell material

Click “Render” to begin final process



Export to Vizard

Depending on the extension used a binary or text version of your file will be saved. The binary has the advantage of embedded compressed textures and faster load times.

Vizard 3 and earlier, and Architecture Interactive:

Binary Format: .IVE

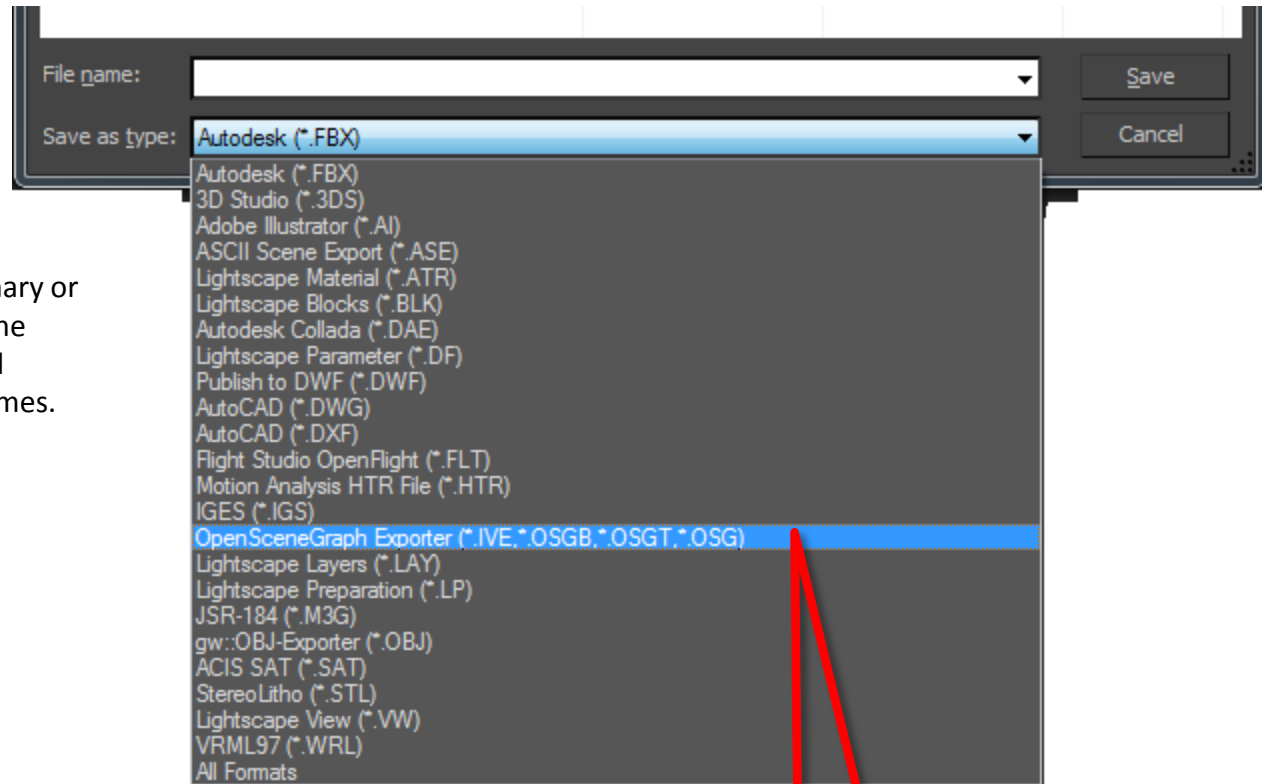
Plain Text Format: .OSG

Vizard 4.0+

Binary Format: OSGb or .ive

Plain Text Format: OSGt or .OSG

.IVE and .OSG are being phased out.



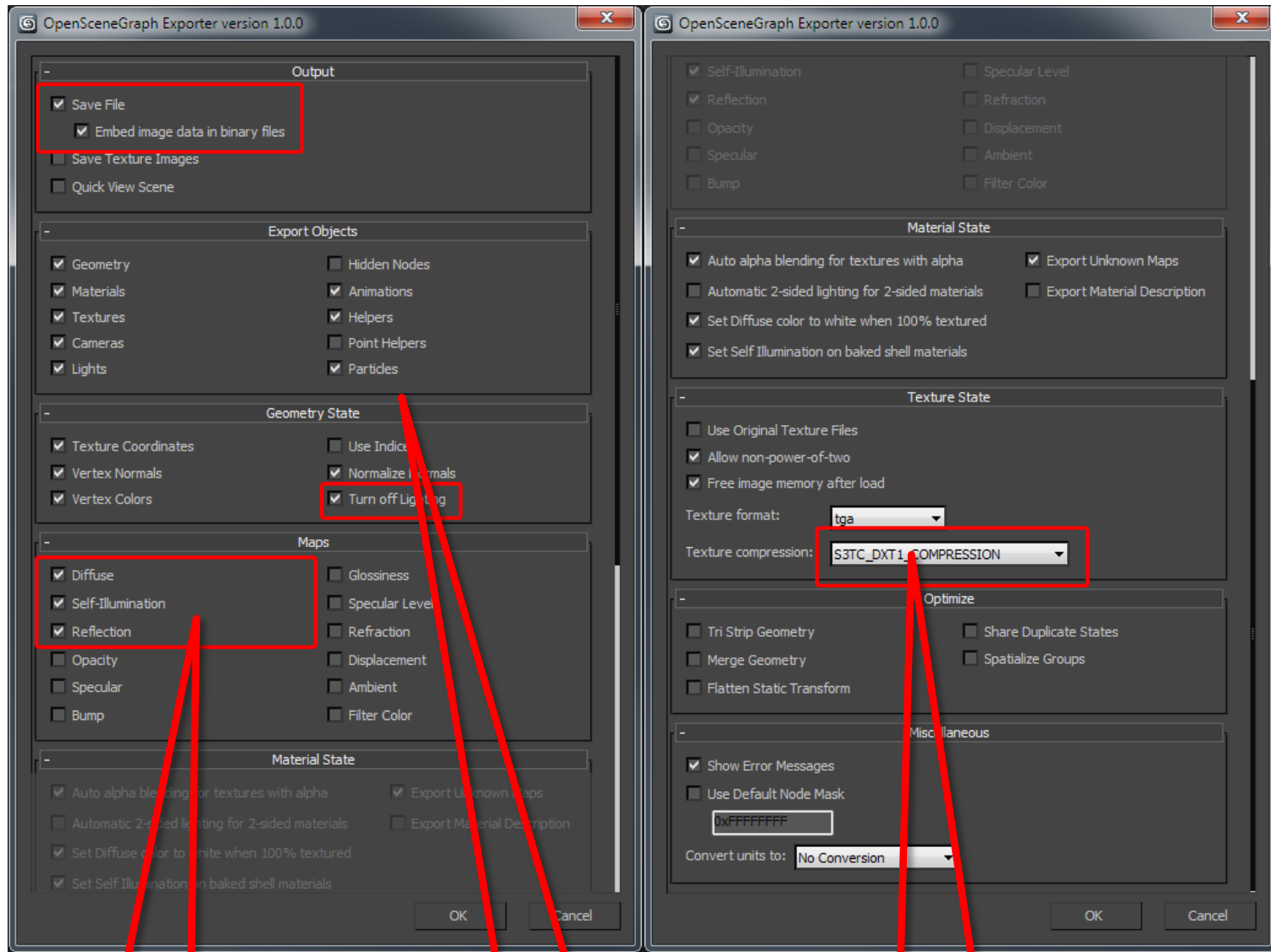
Export to OpenSceneGraph
format.

Exporter Settings

Show here are the typical settings recommended for baked environments.

Most of the settings should be fairly self-explanatory to an experienced 3d artist.

Read through them and note what options are available.



Always include these maps

For baked scenes, Check to disable real-time lighting effects

For resource intensive scenes, use DXT compression

Use Vizard Inspector to examine your bake

After exporting your baked scene, open your IVE or OSG file into Inspector. You will be able to see your scene as it will appear in Vizard.

