This document is a small primer for people new to VRay and to help get the best out of the renderer in short order. By no means is this the 100% mathematically accurate way of doing things but it's a method that's worked for me. The way I use VRay is with a full linear workflow. This means I linearize non linear textures and I also linearize the render view (VFB) so I can deal with things in the compositor properly. Suffice it to say that explaining the theory of linear workflow is something beyond this document.

VRay Basic Settings

So, let's start with the first part, and that's the common tab. Here you can save files automatically in various formats. You can save as a multichannel openEXR for use in your favorite compositing app. If you're on windows, you can turn on the VRay VFB. This will display the proper VFB and allow you to hit the sRGB option to linearize your render view. However, due to some windowing toolkit issues, linux and osx users need to suffer a little. There is an option of using the standalone renderer, which will use the VFB and that's effective. I prefer this method. It will require a small mel script to be written so that you call the standalone and render your scene. The other method is to use Maya's built in frame buffer. If you use this you can turn on the checkbox at the bottom labeled "Convert image to sRGB for RenderView". This way VRay will place the gamma curve on top of the render so you don't need to worry about any additional lens shaders or other hacks. Simple is good. However, if you want to do the standalone and want a quick start on the mel code, my hack looks like this :

```
$tmpfile = "/var/tmp/tmp.vrscene";
setAttr vraySettings.vrscene_render_on 0;
setAttr vraySettings.vrscene_on 1;
setAttr -type "string" vraySettings.vrscene_filename
$tmpfile;
renderIntoNewWindow render;
removeRenderWindowPanel renderView;
system ("/Applications/Autodesk/maya2008/Maya.app/
Contents/vray/bin/vray  -scenefile=\"" + $tmpfile + "\" &");
system ("rm " + $tmpfile);
```
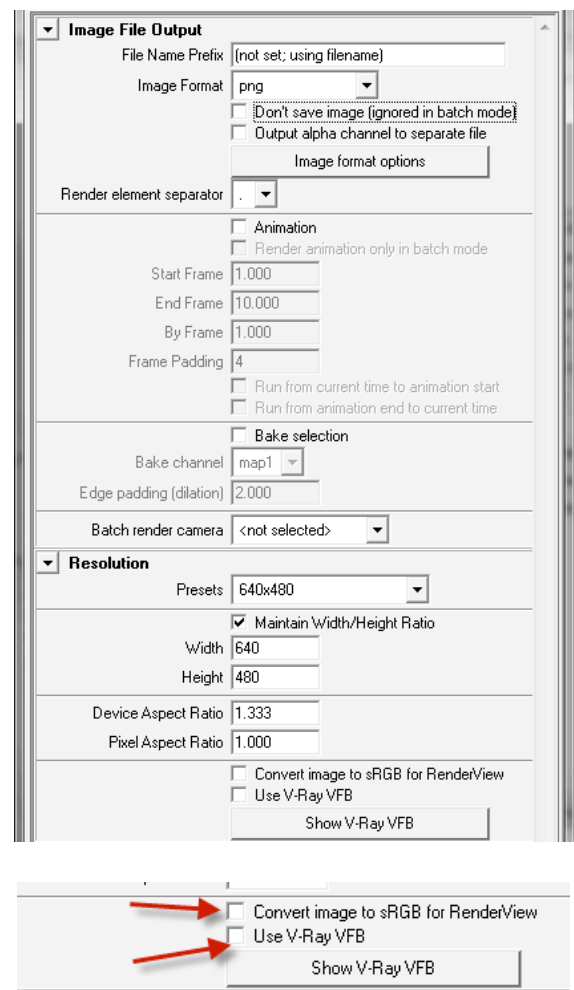
Yes, I realize thats genius mel code. I seriously have no idea what I am doing there. However, let's move on to the next part of our adventure.

Render settings

Figure 1 shows what the settings look like before we fiddle with them. This will get you a decent render but it's not optimized for speed and "ease of use" in my book. How I work is that I tell the renderer that everything is a noise problem. This way the only thing that we have to worry about is how much noise we're willing to tolerate vs render time. Sounds simple eh? Yep. It is. Let's get to Figure 2. These are the settings I like to start out with. First we set the sampler to Adaptive DMC. Adaptive subdivision basically renders higher resolution and then scales down to deal with AA, but Adaptive DMC allows you to sample per pixel and deal with a contrast value for finding out when to sample next. For filters, I like the Gaussian. Yep, a softening filter. I don't like Mitchell, etc because they are sharpening filters and can cause negative data. Gauss with a setting of 2.0 or 2.5 is pretty nice. The min and max values here are a good starting point as is the Threshold. Threshold is basically telling the renderer that if it sees a delta of 0.01 or greater on neighboring pixels then it will sample the current pixel again until it reduces the contrast to >=0.01 or until you hit the max subdivs. This is your noise "tolerance" area. I then set my gamma to 2.2 (approximately sRGB), turn on "linear workflow" and "adaptation only". Linear workflow takes the 8 bit image data, and applies the inverse gamma to them, in effect linearizing them. Yes, this is to avoid the gamma nodes at .454 in the hypershade. Neat. A checkbox! What does "adaptation only mean"? It tells the renderer to sample the noise with this gamma but

Fig 1

Fig 2
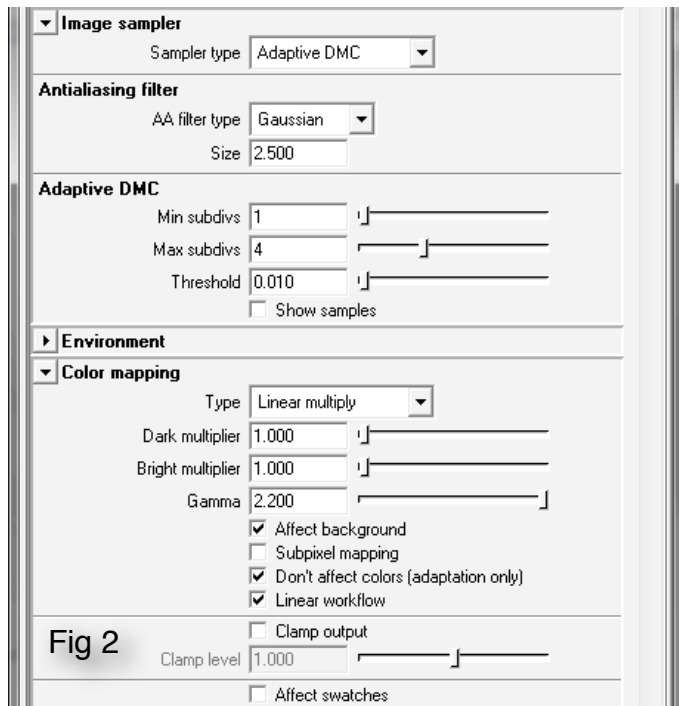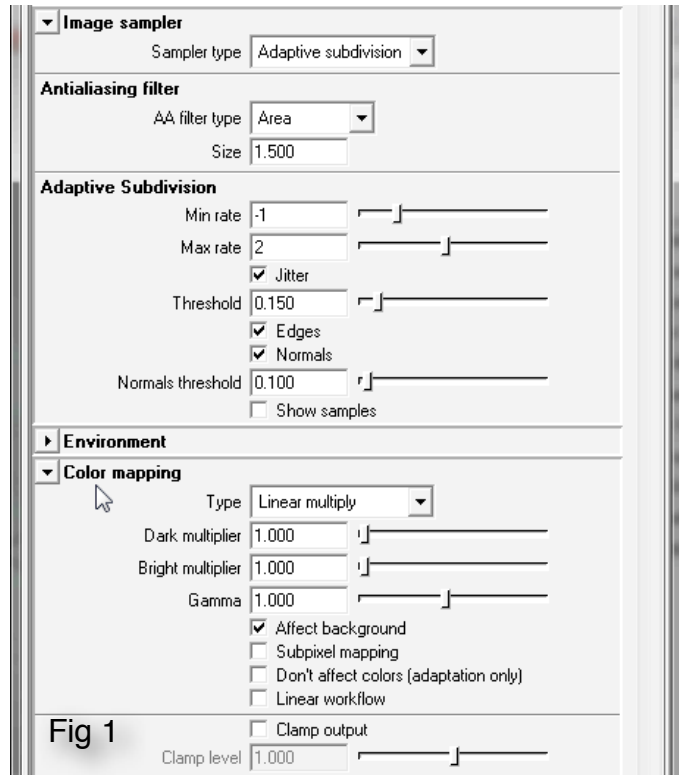
not to actually bake the gamma into the render itself. You don't want to bake the 2.2 into the render of course. However, if you don't set this checkbox on, then the renderer will sample as if you're working with gamma 1.0. So, what in essence will happen is that you view the image with a 2.2 gamma. So visually you're multiplying the values by the 2.2 gamma. However, because noise levels happen more in dark areas, those values need to be multiplied at 2.2 gamma as well in order to get sampled. Otherwise the threshold will not pick up the deltas between neighboring pixels and you will not get the additional samples you need to reduce noise. Like I said, it's a primer and I really don't want to get into the intricacies of linear workflow. With that said, just set the gamma to 2.2 and turn on the "adaptation only" checkbox. See, easy! So, now that we've done that we need to make sure we're working in a fully adaptive mode. This allows VRay to treat everything as a "glossy" or noise issue. Go to the settings tab and you'll see something like Figure 3. Change the adaptive amount to 1.0 so it looks like Figure 4. Yes there is an adaptive threshold here too, but I don't mess with this value because I control it all from the sampler in the previous panel. Now that we've done this we're ready to render and we'll get a reasonably clean render with reasonable render times. I know you're going to ask "what's the best setting"? Well, I'll give you two answers. The first one is that if you set your minimum DMC samples to 2 and max to 100, then you deal with the threshold as your noise issue and that's perfecto. Very simple. I like to

Fig 3

Fig 4

set the minimum to 2 just for fine edge details that I want to get an additional sampler hit. I generally find that for most things, I can get good renders with min 2, max 25 and .005 for threshold.
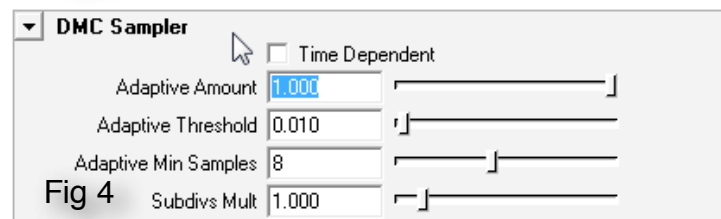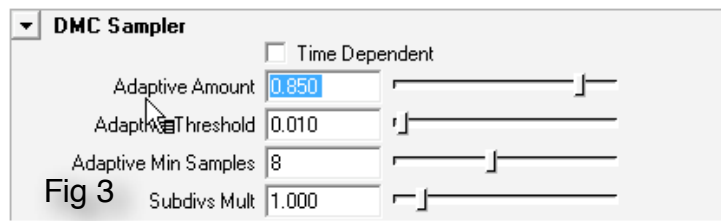
Quick GI Setup
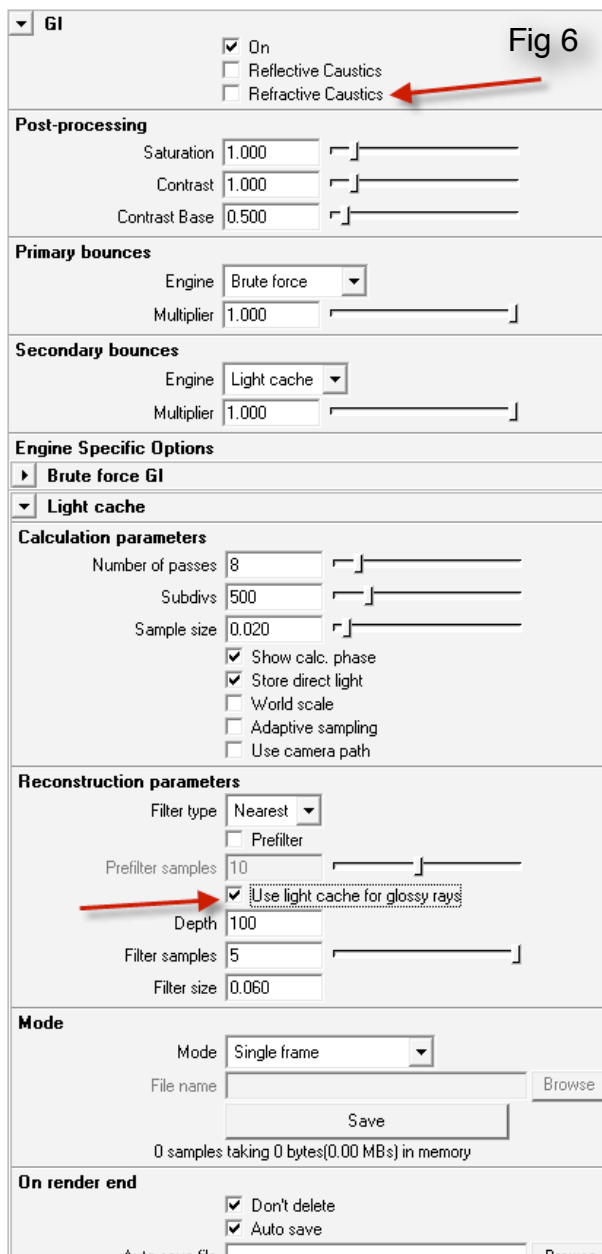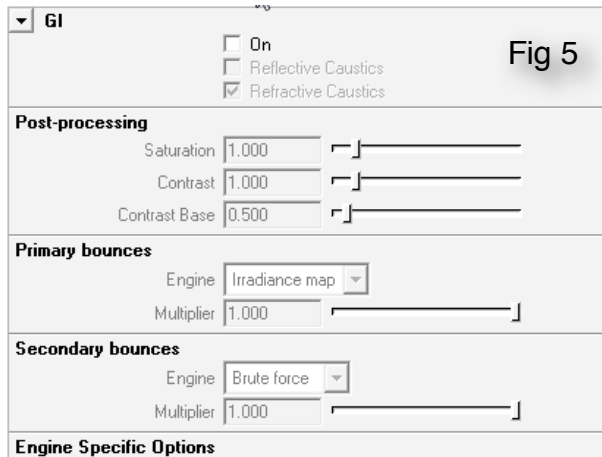
Fig 5

Fig 6

So now that we've gotten that out of the way, let's look at some basic GI settings. By default it's turned off (Fig 5). Let me switch a few fiddly bits and we can take a look at the difference (Fig 6). First, turn of the refractive caustics. That has caused me many issues with splotches, etc when I did not expect to see them. Next, I use Brute Force for Primary, and Light Cache for Secondary. I set the subdivs on the lightcache to 500 for starters but you're looking at around 2500-3000 for HD final frames. Next I turn on "Use light cache for glossy rays". What this does is take calculations that don't need tons of accuracy (aka blurred reflections) and pushes that into the GI caching system. This can save time because less noise is generated for those calculations. Don't bother to mess with the Brute Force rollout since the AA sampler will deal with the noise that the primary GI will create. See? Easier! To be honest, that's really the basics of the render settings. The rules are not set in stone hard and fast of course. There are other ways to deal with getting a great image from VRay but these are the settings I use for general purpose that will get me good results with minimal time invested by the artist or the machine.

Playing with Render Elements

Rendering passes with email is relatively easy. It's done with the Render Elements tab in the render settings panel. Let's start with a simple ambient occlusion element. First thing that needs to be done is to create a VRay Dirt shader (Fig 7). Yep, that's the AO shader. Nice thing is you can assign this in various slots of the main VRay shader, like reflection. This way you can assign burnishing maps on edges of geometry. Handy! So, get over to the render settings and into the
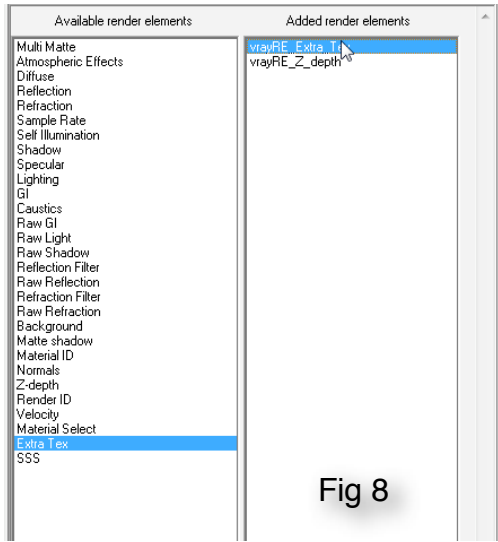
Render Elements tab. See the thing labeled ExtraTex? At the bottom there...(Fig 8) Double click to add that to the list of things to get rendered. Now that we've done that we need to add the created VRay Dirt shader into this render element. Click on the render element in the right hand side of the render settings pane. In the attribute editor you will see a UI very similar to Fig 9. Middle click drag the texture into the "Texture" slot of the "Extra VRay Attributes" rollout. Bam. Connection made. You can optionally not have AA sample this texture layer.
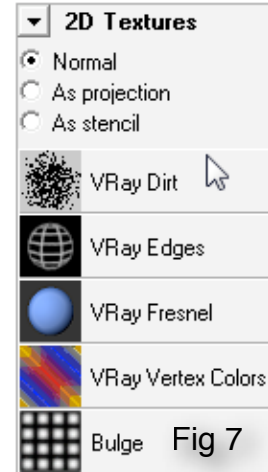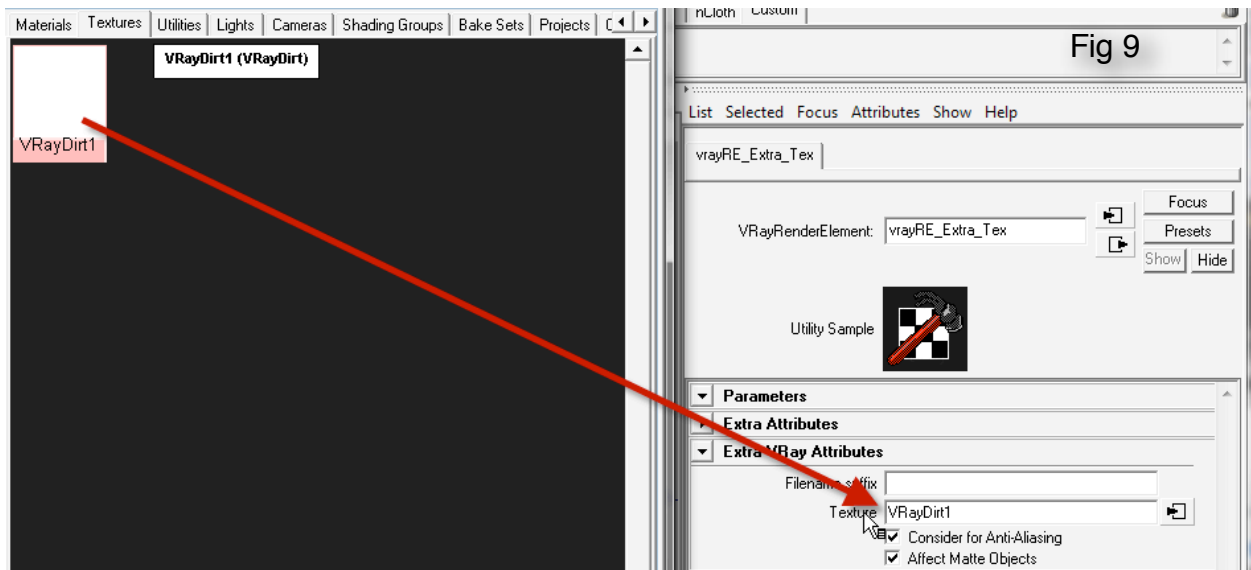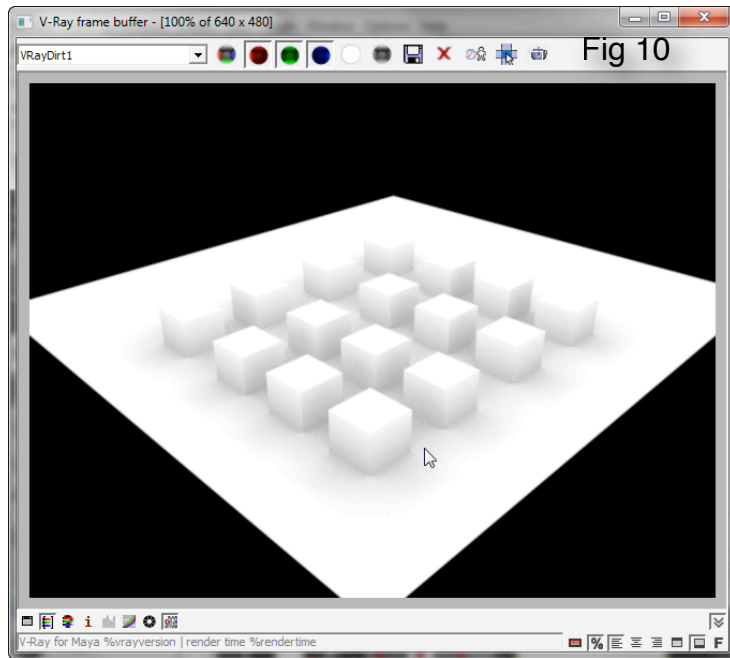
Fig 7

Fig 8

Fig 9

Some elements you may not want to have AA applied to them. Now, these initial settings will get you a traditional AO render (Fig 10)

Fig 10

The standard AO is nice, but the only real way to apply this is to multiply the RGB by the AO. Now being that I am a Nuke user, I want to create a control channel for color correction. Looking at Figures 11 and 12 you can see the before and after of the Dirt shader. Basically I swap the black and white. This will result in the render you can see in Figure 13. Now, whatever you call the Dirt shader, is what the EXR layer/ channel will be called. I just name mine AO and then I know what the heck it is.
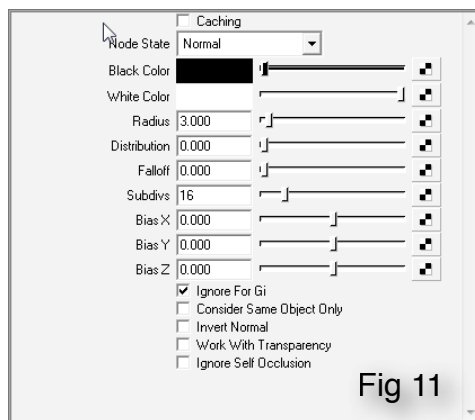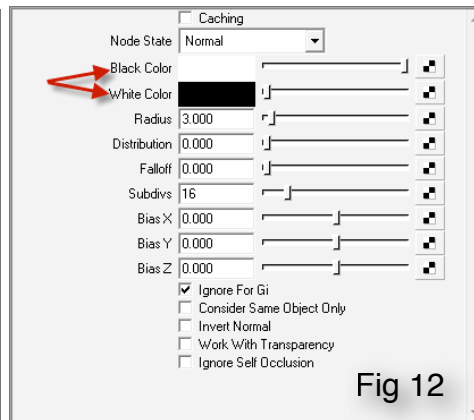

Fig 11

Fig 12

Fig 13