

How to correctly utilize a linear workflow in 3ds Max, Various Renderers and Various Compositing Applications

I have seen quite a lot of confusion surround Linear Workflow and 3ds Max. This is understandable, as there are many guides online, some with conflicting information. Another confusing stage is how to composite these renders correctly.

This is why I have written this guide. I hope it will explain everything you need to know to work in a full Linear Workflow.

For a fantastic technical breakdown on what a linear workflow actually is and why you should care, read this article by Max Attivo here: <u>http://maxattivo.blogspot.com.au/2011/07/what-is-gamma-correction-in-image.html</u>

Why use a linear workflow?

Using a linear workflow results in images that better match how the human eye perceives the world. Essentially, using a linear workflow makes it possible to have a more realistic output. As an added benefit in some cases, using a linear workflow can reduce calculations needed for rendering, therefore speeding up your render time. This is accomplished by more accurately displaying light falloff and travel distance, meaning you may not have to increase GI samples or depth to get correct illumination.

Sounds great! So, how do I do it?

PART 1: Setting up 3ds Max

Recommended Gamma and LUT Settings (found under Customize- Preferences- Gamma and LUT):

Preferenc	e Settings					HX
Inverse	e Kinematics	Gizmos MA	XScript mental	ray Containers	Help	RealFlow
Genera	al Files	Viewports	Gamma and LUT	Rendering	Radiosity	Animation
1	✔ Enable Gam	ma/LUT Correction	1			_
	Display			Materials and Col	lors	3
	Autodesk	View LUT		Affect Cold	or Selectors	
	Browse			Affect Mat	erial Editor	
2	Gamma	2.2 🛟		<u> </u>		
				Bitmap Files		
				Input Gamma:	2.2 🗘	4
				Output Gamma	a: 1.0 🛟	5
						_
					OK	Cancel

Figure 1

1. Enable Gamma/LUT Correction (enabled)

This tick box tells 3ds Max to use the Gamma Correction. To work with a correct linear workflow, we want this activated.

2. Gamma: 2.2

This is the display Gamma. It tells 3ds Max what gamma to display for your viewport and in the Render Frame Buffer. You could leave this at 1.0, but by making sure it is set to 2.2, you enable gamma correction <u>as a preview only</u>. In other words, setting this to 2.2 <u>doesn't affect render output</u>. When you save your files, they will still be Gamma 1.0, but you are more easily able to understand what the final color corrected output will look like. (see Figure 2a and Figure 2b)



<u>Figure 2a</u>-

Gamma Set to 1.0: This shows the gamma of the final output when everything else is setup properly, but can make choosing and adjusting colors and tones difficult.





Gamma Set to 2.2: This shows how your final output will look when corrected to a gamma of 2.2 when everything else is also set up correctly. Essentially, this means what you see is what you get.

3. Affect Color Selector and Affect Material Editor (both enabled)

When both are enabled, the Gamma setting affects the display of colors on the standard 3ds Max Color Selector, the Object Color dialog and in the Material Editor. This means when selecting colors and viewing materials they will match the final output when it has been corrected to a gamma of 2.2. *(see Figures 3a and 3aa and Figures 3b and 3bb)*





Affect Color Selector disabled: Colors are displayed with a gamma of 1.0. This means any color you choose will be brighter than expected when rendered and gamma corrected to 2.2.





Affect Color Selector enabled: Colors are displayed with a gamma of 2.2. This means the colors you select will match the colors in the final render when gamma corrected to 2.2.





Affect Material Editor disabled: Materials are displayed as Gamma 1.0, which makes correctly changing parameters difficult.



Figure 3bb-

Affect Material Editor enabled: Materials are displayed as Gamma 2.2, correctly showing how the material will look once your final output is corrected to 2.2 gamma.

4. Input Gamma: 2.2

Tells 3ds Max what gamma is already applied to images you are importing (such as texture maps). If you leave it on 1.0 you have to override the gamma for every diffuse image when loading the images through Bitmap, but it's much less work to set Input Gamma to 2.2 and only have to change normal, displacement, bump and HDR images to a gamma of 1.0 on import. Do keep in mind though that some formats such as TGA may have gamma settings that 3ds max will read on import. *(see Figure 4a, Figure 4b and Figure 4c)*



<u>Figure 4a</u>- Input Gamma: 1.0: 3ds Max assumes the gamma of the original imported image is 1.0, so applies a gamma of 2.2, unless you manually override the gamma to 2.2 on each texture you import. This results in an incorrect "double gamma" effect. The render on the right was done using Input Gamma: 1.0 and the resulting render (which has 1.0 gamma) was gamma corrected to 2.2 for final output. As you can see, this results in bright washed out textures.



<u>Figure 4b</u>- Input Gamma: 2.2: This tells 3ds Max that the imported images have a gamma of 2.2, so no correction is done for the preview (because of our other settings telling 3ds Max to display with a gamma of 2.2).Behind the scenes though, 3ds Max is changing the gamma to 1.0 to allow for correct rendering. The render on the right was done using Input Gamma: 2.2 and the resulting render (which has 1.0 gamma) was gamma corrected to 2.2 for final output. This results in a correct render.



<u>Figure 4c</u>- Here is a close up, split screen of the two renders above. Input Gamma: 1.0 on the left and Input Gamma: 2.2 on the right, both rendered with a 1.0 gamma and then gamma corrected to 2.2 for final output. You can see the brightness and contrast differences in the textures. This can be more evident depending on the texture.

5. Output Gamma: 1.0

This tells 3ds Max to save the rendered image with a gamma of 1.0. By doing this, we will have access to the full tonal range during compositing. (*see Figure 5a Figure 5b*)



<u>Figure 5a</u>- Output Gamma: 2.2: This is the render exactly as it is saved out, without any gamma correction. While this image is correct, gamma 2.2 has been baked in which will limit us with tonal range when it comes to compositing.



<u>Figure 5b</u>- Output Gamma: 1.0: This is the render exactly as it is saved out, without any gamma correction. Although it looks strange, this is the result we want. It will be gamma corrected to 2.2 for final output.

By completing the steps above, we have told 3ds Max to display everything with a gamma of 2.2, but save the rendered output to disk as Linear.

A note about Output Gamma: The steps I have outlined assume you are rendering out to a high bit format (explained later in this tutorial). For the best flexibility, you want the most tonal range you can which is why you save out with a gamma of 1.0. If you don't need to do any compositing or color correction, you could set Output Gamma to 2.2. As shown above, the render will look as expected. This could change per project depending on the required output, but I find it easiest to set to 1.0 for everything, and keep my workflow consistent.

IMPORTANT: Because we have told 3ds Max that all textures have a gamma of 2.2, when importing Normal Maps, Bump Maps or Displacement Maps or Vector Displacement Maps you must specify a gamma of 1.0. The reason for this is that these images are based purely on data. This could vary though, depending on how you are generating the maps and what formats they are saved into, so I do recommend trying both manually setting gamma to 1.0 on import and letting 3ds max assume a gamma of 2.2. Though you will only need to test this if you are getting unexpected results. In most cases, setting gamma to 1.0 for these types of images is the standard. You can do this directly from the Open window when importing a texture. Just check Override Gamma and make sure it is set to 1.0

Now 3ds Max knows exactly what to do with our images and renders. But what about our chosen render engine? Does it know how to interpret everything, and how we want to save it? Let's check it out!

PART 2: Setting up your Render Engine

Scanline and MentalRay: By default, there is nothing you will have to change for these render engine, unless you are using exposure controls.

Vray: To utilize a linear workflow in Vray, you will need to change some settings in the Vray Color Mapping rollout in the Render Settings.

[+ [+	V-Ray:: Adaptive DMC image sampler V-Ray:: Environment			
-	V-Ray	:: Co	olor mapping	
Type:	Linear multiply	- I	Sub-pixel mapping	
			Clamp output Clamp level: 1.0	÷
Dark multiplier: 1.0 Bright multiplier: 1.0		:	 Affect background Don't affect colors (adaptation only) 	
		:		
	Gamma: 2.2	¢	Linear workflow	

I recommend setting Type to Linear multiply as it is truly linear, but you can read up on the alternatives here: http://renderstuff.com/best-vray-settings-antialiasing-and-color-mapping-cg-tutorial/#VRay Color mapping rollout

You also want to make sure Clamp Output is disabled when saving to a 32bit file, as clamping the render means you will lose information. This can cause artifacts though in certain situations, so keep an eye out for this.

Set Gamma to 2.2: This tells Vray to calculate everything for an eventual output to a gamma of 2.2. If it was left at 1.0, you could end up with artifacts from things like GI and antialiasing.

Don't affect colors (adaptation only): Tells Vray that the Gamma: 2.2 we just set is only to be used for calculations, and to not apply it to the final render.

<u>DO NOT</u> tick the Linear Workflow box. Although it sounds like exactly what we want, it is actually an old brute force method of converting a scene, and isn't appropriate here.

A note about the Vray Frame Buffer: The Vray Frame Buffer doesn't take into account the Gamma and LUT display settings in the 3ds Max preferences. To enable correct previewing, you have to turn on the sRGB button on the bottom toolbar of the Vray Frame Buffer.

finalRender: The only change you need to make to the finalRender settings is the AA Gamma located in the Anti-Aliasing rollout in the Render Settings.

To access the Anti-Aliasing Settings, click the hammer icon in the rollout.

Change AA Gamma to 2.2 and you're all set.

So, 3ds Max knows how to process our images correctly and our render engine knows how to process the renders. So how do we save out our wonderfully linear images?

PART 3: File Formats

The discussion surrounding file formats can be a difficult one. Individual artists and studios have different preferences and applications have different supported formats.

When it comes to a linear workflow, what matters most is saving to a lossless high bit format, preferably float.

A few of your options are: TIF, TGA, and EXR. Of course, feel free to research other formats that may be suitable for you.

Personally I use the OpenEXR format, which 3ds Max can export to. It is supported by almost every professional imaging application. Developed by ILM, OpenEXR is a great format with support for 16-bit floating-point, 32-bit floating-point and 32-bit integer pixels, lossless compression, and multichannel support. This enables you to have all of your render passes stored in a single file at the highest quality. For more information, visit <u>www.openexr.com</u>

So how do you render out to the very awesome .EXR format? Simply select OpenExr from the list of available file formats in 3ds max when saving your renders.

As far as options go I have a couple of suggestions, but feel free to fire up the 3ds Max help or Google if you are uncertain of other aspects.

What Bit Depth?

The 16bit vs. 32bit argument can go in favor of either bit depth, but when it comes down to it 16bit is suitable 99% of the time. If you make sure your raw renders are at least somewhat close to your desired final composite output (i.e. no extreme exposure changes) and you don't have a client who is always requesting crazy changes in a short time period, 16bit will give you more than enough data for your compositing and color correcting needs. 16bit also has the added benefit of significantly smaller file sizes than 32bit and faster load times without too much loss of flexibility.

32bit is fantastic for those times when you *need* the extra bit depth, either for extreme color correction or in cases where certain render passes will benefit from it. Although I mostly render to 16bit half float, I do render my motion vector passes and z-depth passes out in 32bit float.

16bit vs. 32bit can also depend on your project, so I certainly wouldn't rule out 32bit totally. It is all about what works best for you, and it may be beneficial for you to test both formats.

Compression Options

There are quite a few compression options to choose from, and we could spend days getting into what each one does and where it may be applicable. What I suggest is to use ZIPS. I find it is a good tradeoff between compression ability and reading speed. This is important, because although you *could* compress the file more to have smaller file sizes, your compositing application will have to decompress them to display and work with them. The more compression, the longer it takes to decompress and the more lag (read times) you will have during compositing. Of course this is also dependent on the speed of your hard drives or network, but that is a whole other can of worms.

Storage Type

Most compositing applications can work with either Scanline or Tile storage types, but there are preferable settings for different applications. If you are compositing using After Effects, set your storage type to Store Image as Tiles. But if you are using The Foundry Nuke, Autodesk Composite or Eyeon Fusion, it is best to set your storage type to Store Image as Scanlines. As an example in Nuke, storing the .EXR by Scanline can give you a 5-10% speed increase. It may not sound like much, but once you begin working with a large composition *any* speed increase you can gain is well appreciated.

Note about OpenEXR and Vray: Vray doesn't render directly to .EXR stored as scanlines due to the way it stores buckets while rendering. You will need to convert the files to be stored as scanline after rendering is complete if you want to take advantage of the significant speed increases in Nuke. Vray provides a utility for this called vrimg2exr. You can find more information about this in the Vray help. There are a couple of other considerations when it comes to outputting .EXR from Vray, all of it available through a simple Google search.

Render Elements

Whether you decide to use multichannel .EXR or render to individual .EXRs per pass is completely up to you, your workflow and requirements. Generally, I will render each render layer and its associated render elements into their own multichannel .EXR files. For example, say I have a character lying on a daybed in a garden. I might use 3 render layers for this: the character, the daybed and the garden. Each render layer has the following render elements: Diffuse, Shadow, Reflection, Specular, Alpha, GI, ObjectID, Z-depth, Motion Vector and Ambient Occlusion. There may even be extra render elements depending on the render layer, such as SSS for the character.

So pretty quickly we are rendering up to more than 30 individual .EXR files. But if I render each render layer out to a multichannel .EXR, I only have 3 files to deal with for that shot. Then in my compositing application I can access each render element from the multichannel .EXR per render layer as needed. This makes file management a lot easier, while not getting bogged down with disk read times that occur when using a single .EXR containing a large amount of render elements.

Of course, test out the various options (one large multichannel .EXR, multiple multichannel .EXRs or individual .EXRs for every pass) and see what works best for you.

Okay, okay, so I have setup 3ds Max correctly, set up my Render Engine, chosen my File Format and rendered out my project. How the hell do I composite this??

PART 4: Compositing with a Linear Workflow

There are certain workflow changes that need to be taken into consideration when using a Linear Workflow for compositing. I will give you a walkthrough and some hints for After Effects, Nuke and Composite.

Adobe After Effects- Layer Based Compositor

After Effects' implementation of color spaces is incomplete with CS5.5. Even though the features are there, they aren't fully useable or intuitive at all. Here I will show you the method that works for me, as well as link you to alternative methods. I recommend you try out various methods to find what works best for you.

In your Project Settings (File- Project Settings), you want to set your Depth to 32 bits per channel (float). After Effects doesn't support 16bit float, so to work with it without artifacts and issues, you must set your depth in the project settings to 32 bits per channel (float). As is evident by the options in the dropdown, After Effects doesn't support 32bit Integer either, so you will need to keep that in mind when rendering to 32bit .EXR in 3ds Max.

You want to set your Working Space to sRGB. Options may vary on your system, as After Effects takes into account your monitors gamma correction.

You also want to tick Linearize Working Space. This means After Effects will do any blending calculations based on a gamma of 1.0 to help prevent artifacts.

Project Settings
Time Display Style
 Timecode
Footage Start Time: Use Media Source 🔻
⊖ Frames
Use Feet + Frames: 35mm 🔻
Frame Count: Timecode Conversion
Color Settings Depth: 32 bits per channel (float)
Working Space: sRGB IEC61966-2.1
Linearize Working Space
Match Legacy After Effects QuickTime Gamma Adjustments
Compensate for Scene-referred Profiles

What we have told After Effects to do is display and output with a gamma of 2.2 and perform most operations with a gamma of 2.2, except those involving pixel blending which will be calculated with a gamma of 1.0.

Now, you can import your .EXR files and start a new composition, and everything will be displayed as expected. When rendering out your composition, gamma 2.2 will automatically be applied.

This is one process that worked for me, but rarely do I work with After Effects. If you do, I recommend checking out these links:

http://www.fxguide.com/featured/After Effects 32-bit Workflow In-Depth/

<u>http://prolost.com/blog/2006/2/7/linear-color-workflow-in-ae7-part-1.html</u> - search Linear Color Workflow on this blog for the other 5 parts in the series

Autodesk Composite- Node Based Compositor

Composite (used to be called Toxic) comes with 3ds Max, so is a great alternative for those who don't want/need to spend extra money on additional software. The fact that it is node based and fully compatible with a 16bit/32bit Linear Workflow makes me recommend it over After Effects.

It is very simple to set up Composite to work in a linear fashion.

First of all, you need to add a sRGB tool to your Player (step by step is provided in the documentation) and set the output depth to Input Depth. This means all operations, calculations and rendering will be performed with a gamma of 1.0, but your Player will display your composition with a gamma of 2.2.

Display - sRGB			
	Output Depth	Input Depth	
	Exposure	0.00	

The great part is, after you have applied the sRGB tool to the player once, it will automatically be applied for each project from then on. Of course, you can remove it too if you do a composite involving files with a 2.2 gamma baked in.

Doing this has told Composite to calculate everything with a gamma of 1.0, put display it with a gamma of 2.2. If you would like to bake gamma 2.2 into your final output, simply add a Remap Color tool just before your Output node, and set the Gamma to 2.2 for the R, G and B Values. The preview in your Player will look washed out, because the Player is taking the output of the Remap Color which changed the gamma to 2.2 and adding a display gamma of 2.2. This doesn't affect rendering though; your files will be saved with gamma 2.2.

Remap Color tool settings.

The Remap Color is added at the end of the node tree, right before Output.

The Foundry Nuke- Node Based Compositor

Nuke is the industry standard in compositing applications. Its huge feature list, flexibility and raw power have made it the only choice for studios around the world.

So, it can only be expected that Nuke works perfectly and is even optimized for an .EXR based Linear Workflow. Of course, you can also use other image extensions if you so choose.

All you have to do in Nuke is set Colorspace to Linear on your Read node (which is the default), and sRGB for your Viewer Process Operation (which is also the default in Nuke).

Read Sequer	nce Node				
file	C:/Users/Sonny/Desktop/render.exr				
cache locally	auto 🝷				
format	720x405 • =				
proxy					
proxy format	root.proxy_format 1024x778 - =				
frame range	1 hold - 1 hold -				
frame	expression 👻				
original range	1 1				
missing frames	error 🔹 reload				
colorspace	default (linear) 🔻 📕 premultiplied 📕 raw data 📕 auto alpha				
exr Options	X offset negative display window				

Read node settings

Viewer Process Operation settings

That's it! Now your images will be processed with a gamma of 1.0, but displayed with a gamma of 2.2.

If you want to bake gamma 2.2 into your render output, on your Write node simply change Colorspace to either sRGB or Gamma 2.2 (very slight differences, may depend on your monitor calibration).

▼ (○ ₩ ₩	Write1 🔽 🔽 🖛 🗠 😨 ? 🗗 🗙	
Write Pythor	n Node	
channels file	rgb 🔹 🗶 red 🗶 green 🗶 blue 📄 none 🔹 😑	
proxy	expression *	
colorspace	sRGB	
views	main 🔻 =	
file type		
render order	render order 1 Render frame range 1 1 Imit to range read file	
missing frames	error Reload	

Write node settings. Choose either sRGB or Gamma 2.2

PART 5: Final Thoughts and Comments

Using a Linear Workflow is a great skill to have in your arsenal, and once you have become accustomed to it you won't want it any other way.

What artists who are new to Linear Workflow get most annoyed at is the different working experience. To achieve the results you have been previously, you have had to cheat even if you don't realize it. Adding ambient lights to brighten areas, increasing GI or Final Gather settings to get the light to bounce further around the scene, setting light multipliers to high values, not using light falloff due to areas blowing out to white, using exposure controls to prevent clipping of white values... the list goes on.

It might take a short amount of time to get used to not cheating reality with your renders, but once you do you will realize what you have been missing out on all these years.

So stick with it and you will be extremely happy with the results.

EXTRAS:

If you would like to see more of my tutorials, please visit my blog and YouTube pages:

YouTube: <u>http://www.youtube.com/coldsidedigital</u>

Blog: <u>http://coldsidedigital.tumblr.com</u>

I can also be contacted through both of those websites, or you can email me:

Email: daniel@inthefleshart.com

I ask that you please don't redistribute this document, but rather share the link where you downloaded this from.

I hope you have enjoyed this tutorial, and found it helpful.

Regards,

Daniel Dye