Grading HDR Video on a Rec.709 Monitor (For YouTube & Beyond)

HDR YouTube video created for Austin Evans, entirely graded and mastered using this workflow.

The Backstory...

I created this workflow for an upcoming Austin Evans YouTube project, that is currently in post production.

Without giving too much away, we knew that we would have an opportunity to get some awesome footage.

We had a picturesque location (the Nevada Desert), and some sports cars...what more could you want?

After perhaps a bit too much inspiration from a certain shared-favorite car show on Amazon Prime, we realized that our video would be perfect for HDR...

But we had no way of monitoring an hdr color grade.

First off, buying or even renting a professional HDR grading monitor was out of the question, financially.

Secondly, even though there is an HDR TV in the office (with a Chromecast Ultra for watching HDR YouTube videos), as of writing this, Windows (my OS) doesn't support HDR display output.

Apparently Windows has support on the individual application level, but it hasn't yet been implemented OS-wide. Ken (Austin's head of video production) and I tried everything, but we couldn't get any computer and the TV to jive in HDR.

And before you say it: no, Mac OS doesn't support it either.

So this where you give up and accept fate, right?

NOPE! Instead, I embarked on a "mission from God" to create a usable workflow for grading HDR video on a Rec.709 monitor.

And if I may...

It's Pretty Freakin' Cool.

Is it the "technically correct" way to do this? No. Not at all...but it WORKS!

If you are working on a large, commercial project and have the means, of

course you should go to a color facility with proper HDR gear and a workflow to match.

That said, I've actually been really surprised at just how well this workflow has been performing. After hammering out the kinks (and trust me, there were a few), I'm honestly finding it just as reliable as a normal Rec.709 workflow.

SO, whether you're a smaller production outfit, or just someone who likes playing with the latest in video tech...

I hope this write-up can help you out!

(plus it's filled it with lots of fun pictures!)

Tech Specs

Gear used to create this workflow, and Austin's video

CameraS

SONY PXW-FS7 (x2)

Negative Format

3840 x 2160 / 23.976fps
10 bit 4:2:2 XAVC-I
S-Log3 S-Gammut3.Cine

HDR TV

LG - 55" E6 OLED 4K HDR Smart TV (OLED55E6P)
Chromecast Ultra

Grading Monitor

EIZO CX271

Grading Workstation

Lenovo P70
Intel Xeon E3-1505M v5
NVIDIA Quadro M4000M
Windows 10 Pro

Grading Software

Davinci Resolve Studio 12.5

Plan of Attack

So How exactly are we going to do this?

As this workflow was originally designed for a YouTube video project, it is based on the specifications outlined in YouTube's HDR "whitepaper". You can find that full post by clicking here.

If you are just looking to do an HDR grade for a video that isn't intended for YouTube, this workflow will still work perfectly well for you. There are just a few steps that you won't have to complete.

YouTube-Specific HDR

In a perfect world, you would create two separate grades: one for HDR, and one for SDR (Standard Dynamic Range) displays. HOWEVER, YouTube doesn't have any way to upload two different grades for the same video.

Q: So what do we do to ensure that people can watch our videos on both HDR and SDR displays?

A: Well, there are two options...

When you upload an HDR video to YouTube, an SDR version of your video will be automatically created by YouTube's back-end processing. This is the version that people using regular SDR displays will see.

It's honestly a pretty good conversion, but if you want to take control of things, you can package a custom LUT in with your video file. YouTube will see it, and use it to do the HDR to SDR conversion, instead of it's own algorithms.

That second option is what I have been doing for Austin's video and my own tests, and it's worked perfectly. I'll guide you through that process later on in this post.

Here's an outline of the workflow:

1. Setting-up your Davinci Resolve Project
2. Grading Tips
3. Creating the Video File
    1. Rendering out of Resolve
    2. SDR Conversion LUT Creation
    3. Inserting Required Metadata
4. Uploading to YouTube

I've designed the workflow for Davinci Resolve Studio, running on a Windows-based PC.

Everything should work in the free version of Resolve as well, but I can't guarantee how well things will translate over to other software packages/grading applications.

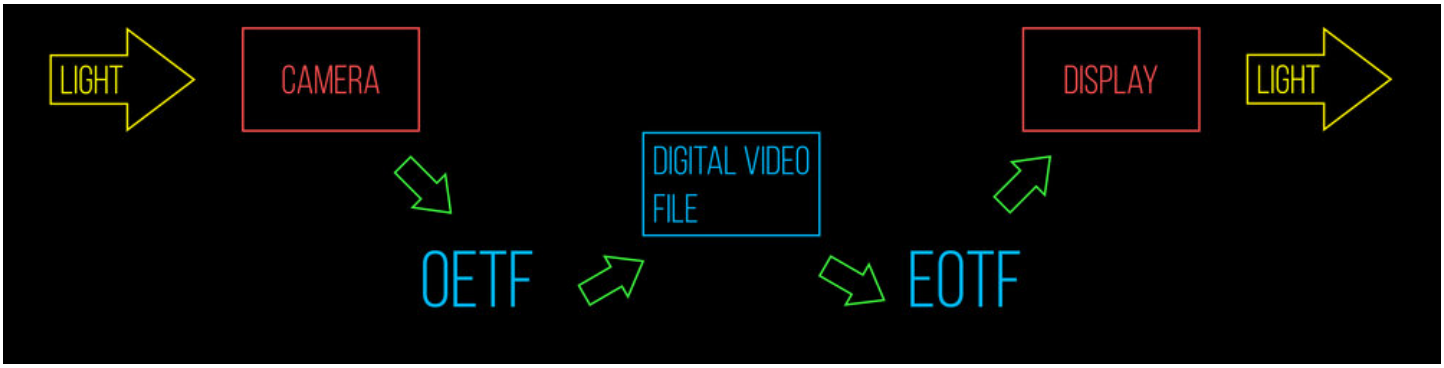
There are also some elements of this workflow that are based more on understanding how consumers watch web video, rather than technical standards (I'll point them out as they come up).

> ***We will be grading in Rec.2020, using an ST.2084 EOTF***
>
> ***But first things first...***

Some Background

A quick glossary to clear up some common confusions

Feel free to skip this if you're too cool for school



- EOTF stands for "Electro-Optical Transfer Function"
  - Mathematical function that defines how digital imaging data is translated to brightness on a display
  - Opposite of an OETF

- OETF stands for "Opto-Electrical Transfer Function"
  - Defines how light going into a camera is translated to digital data.
  - Examples found in today's cinema cameras include:
    - Canon's Cannon Log
    - Sony's S-Log, S-Log 2, S-Log 3
    - Arri's Log C

- EOTF can be thought of as the "gamma curve" of the video's intended display color space
  - Similar to how the Rec.709 colorspace uses a 2.4 gamma curve, or the BT.1886 gamma function (the newest recommended standard)...
    - ...or even a 2.2 gamma curve, if you're referring to the older variant designed for CRTs

- There are two EOTFs being used in HDR today
  - **PQ (ST.2084)**
    - "Perceptual Quantizer"
  - **HLG**
    - "Hybrid-Log Gamma"

The Two Big Standards: HDR10 vs. Dolby Vision

(the main differences)

HDR10

- 10 bit color
- Utilizes a PQ (ST.2084) EOTF
- Open standard
- Theoretical max brightness: 4,000 nits
- Current max brightness: 1,000 nits
- Supported by Amazon Prime Video & YouTube

Dolby Vision

- 12 bit color
- Utilizes a PQ (ST.2084) EOTF
- Requires proprietary Dolby equipment for file creation
- Theoretical max brightness: 10,000 nits
- Current max brightness: 4,000 nits
- Supported by Amazon Prime Video
- YouTube support currently unclear

The biggest difference on the consumer end, is that Dolby Vision certified display devices can also display HDR10 content. This is NOT true in the opposite direction.

This is why, if you can, it is better to get a Dolby Vision certified device rather than one that is just HDR10 compatible.

EOTFs:
PQ vs. HLG

(The Main Differences)

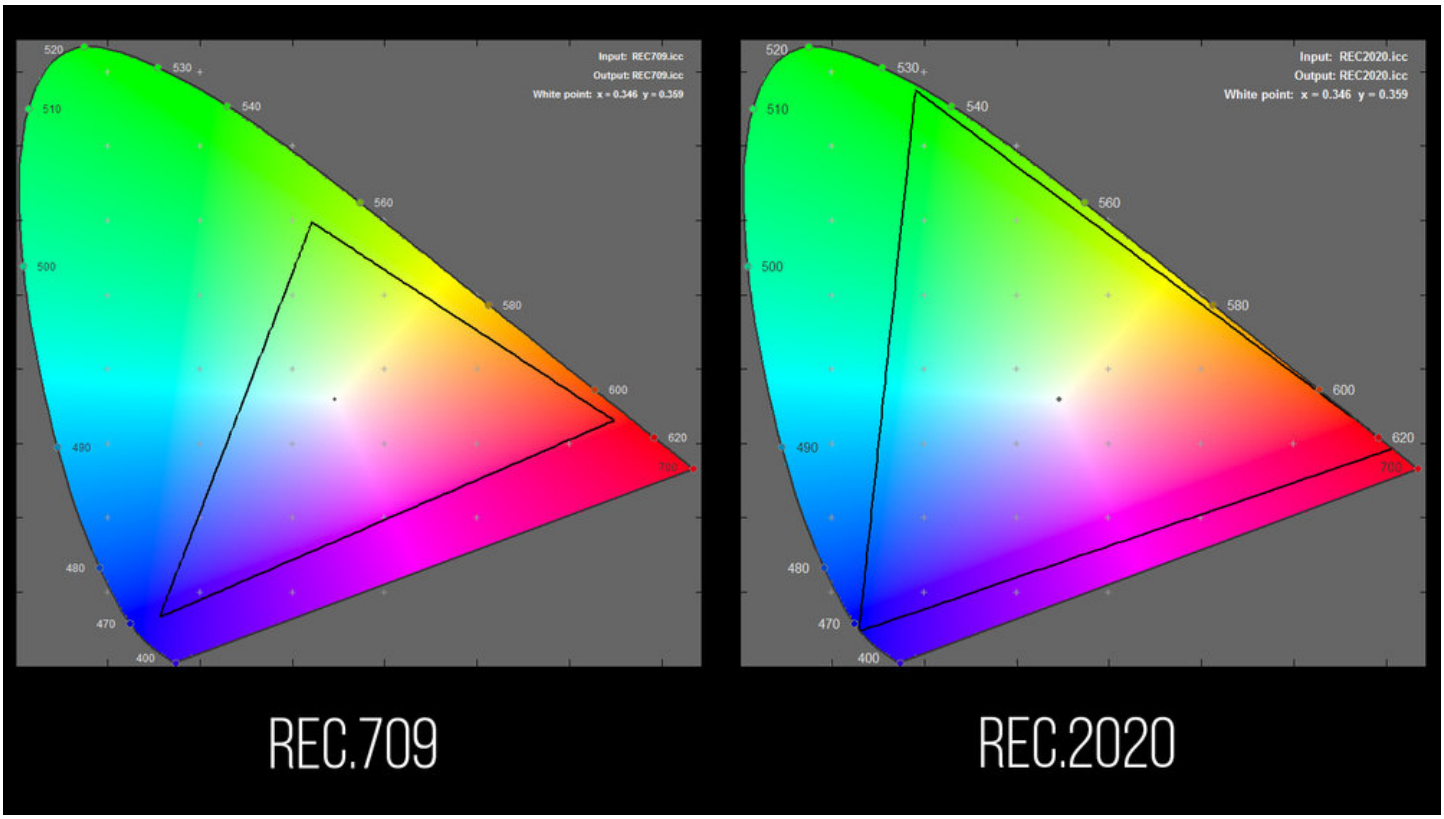
PQ
("perceptual quantizer")

- Just an EOTF
- Also referred to as: "ST.2084"
    - From it's SMPTE standard name: "SMPTE ST-2084"
- Used by both HDR10 and Dolby Vision
- Dominant EOTF being used in the United States, for HDR Video
- Currently supported by all of the major HDR display manufacturers
- Not backwards compatible with SDR displays
- Recommended for YouTube

HLG
("Hybrid Log-Gamma")

- An entire HDR standard, with it's own unique EOTF
    - Also referred to as: "ARIB STD-B67"
        - Standard name from the Association of Radio Industries and Businesses in Japan
    - Utilizes Rec.2100
- Co-developed by the BBC and NHK
- Only really in use in the UK
- Not yet supported by most of the major HDR display manufacturers
- Backwards compatible with SDR displays
- Not recommended for YouTube

Colorspaces: Rec.709 vs Rec.2020

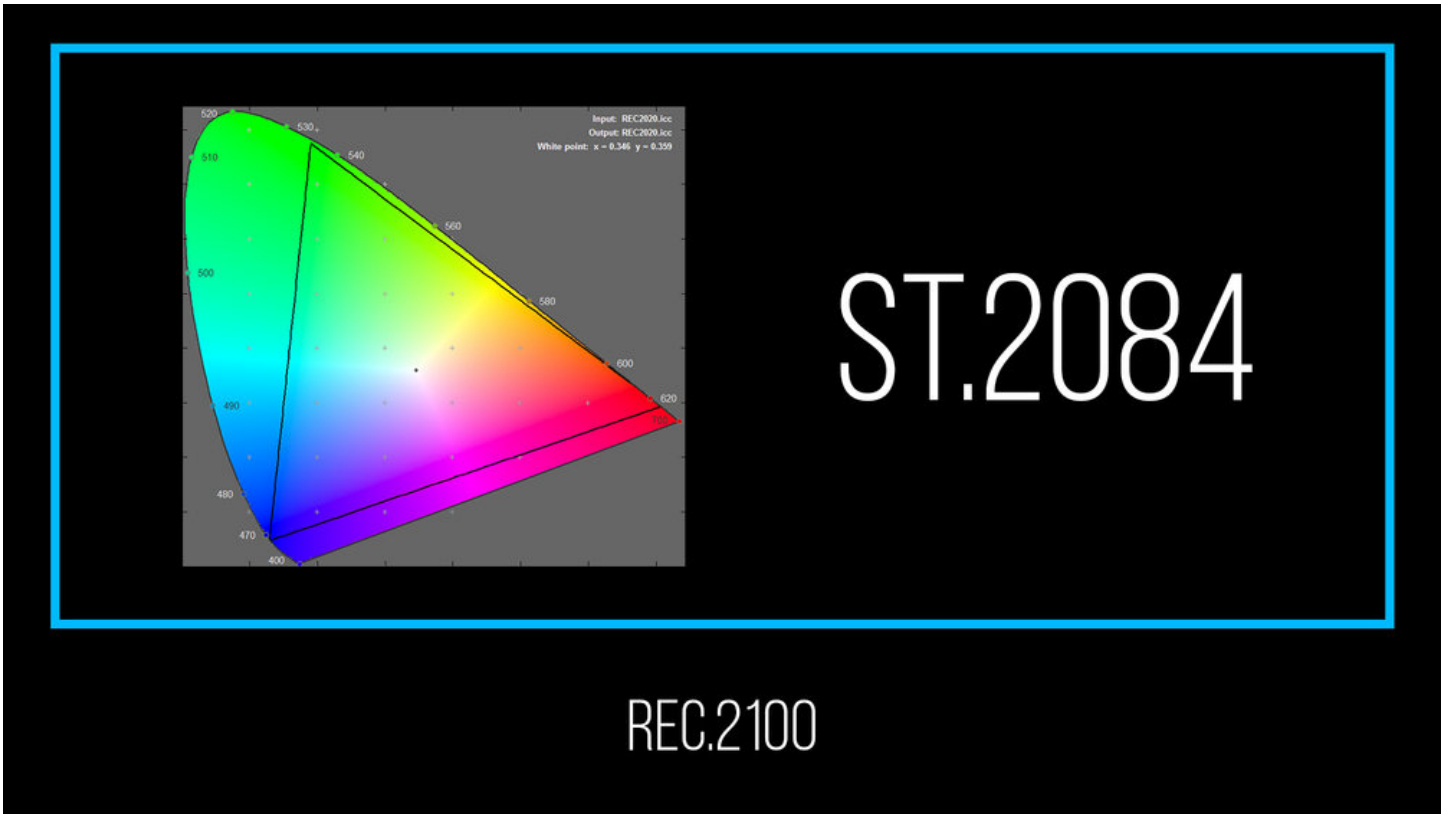


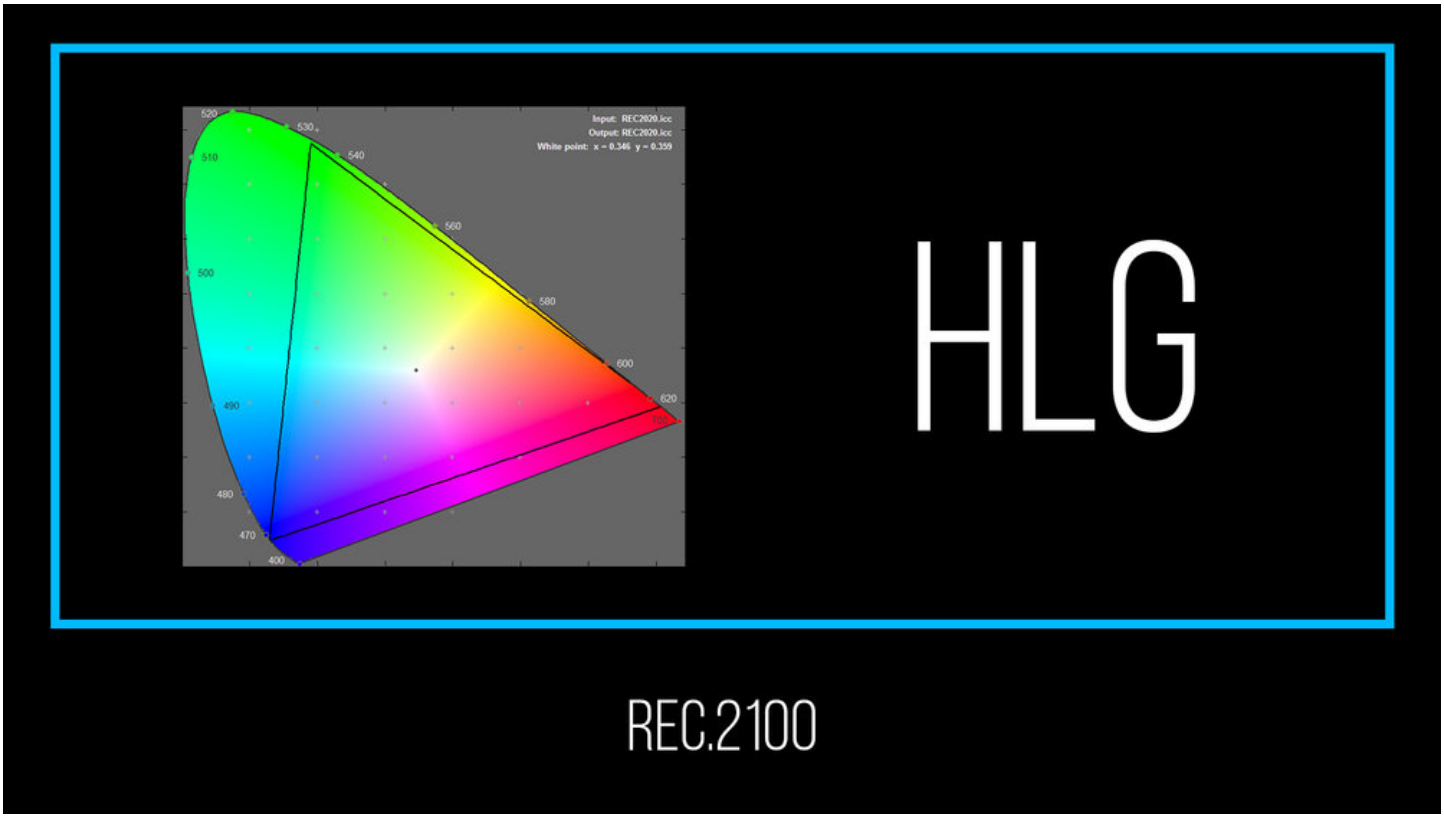
Rec.2020 is the colorspace used to grade and display HDR video.

As seen above, It has the same D65 whitepoint as Rec.709, but has a much larger gamut.

Rec.2020 vs. Rec.2100

- Rec.2020 is just a color space, with a defined set of color primaries and a white point
- Rec.2100 takes things a step further, and packages Rec.2020 with an EOTF.
    - Can include either an ST.2084 or HLG EOTF





Time for some examples!
(as seen in Davinci Resolve)

Rec.2100 ST2084

Rec.2020 with an ST2084 EOTF
(for HDR10 or Dolby Vision color grading)

Rec.2100 HLG

Rec.2020 with an HLG EOTF
(for HLG color grading)

So, again

We will be grading in HDR10 Rec.2100 ST.2084

We'll also "define" our grade as intended for 1000 nit max displaying

This is all per YouTube recommendations. Capisce?

Monitor

While using this workflow, I suggest keeping your monitor calibrated to Rec.709, with a gamma of 2.4, and at a brightness between 200-300 nits.

This is way brighter than Rec.709 is supposed to be, but I suggest 200-300 because:

1. It gets you closer to the brightness intended for HDR (1000 nits, in our case), which will help you grade more accurately
2. Most current HDR TVs can only output 300-500 nits anyways
3. This number strikes a nice balance between current HDR TV brightness,

and the usual brightness of the devices most people watch YouTube videos on (smartphones, laptops, etc.)...hardly anyone is ever in-spec.

Shooting Tips

I'm not going to go into too much detail here. In general, there a few things to aim for when shooting for an HDR grade:

1. Record 10 bit, 4:2:2 at minimum
    - To avoid banding & artifacting
2. Shoot Log/maximum dynamic range available on your camera
3. Watch your highlights and speculars
    - Clipping is VERY noticeable in HDR
    - Exposure differences can also be very apparent, from shot to shot
    - Polarizers are always a good idea when outside
4. Stay as close to Native ISO as possible
    - HDR grades are prone to bringing out digital noise, as a lot of signal range is given to the shadows
    - Especially important if you plan to lift shadows in post

Proficiency

While I've done my best to keep this write-up as clear as possible (and full of fun colors and pictures, after reading many bland reports, myself), it does assume a bit of proficiency on your end.

This is in regards to working in Resolve, as well as color grading, video tech specs, and HDR in general.

That said, if you're just in this for the sheer joy and excitement of reading

workflow breakdowns, read-on, my friend.

Simulated HDR

This workflow utilizes a LUT that I designed to help visualize how the video will look on an HDR display, as well as a a very lightweight GUI that I built for some software (all of which I will ask you download below).

Their file names or displayed version numbers may vary from what I have written/in the screenshots below.

Rest assured that they will still work exactly as described, just with some minor updates on the back end (that I've made since beginning this write-up).

Checking your grade

While the point of this workflow is to be able to do everything in a Rec.709 colorspace, I've found that having an HDR TV to check your work can be handy (by uploading to YouTube and playing back on the TV via a Chromecast Ultra, or built-in YouTube app with HDR support).

Not in the sense of checking every shot, but just as a general double check before making the video "public" on YouTube, or to check minor details in the grade.

It's also good to have one on hand when you are starting out, and still learning how HDR behaves. While the Monitor LUT used in the workflow does a good job of simulating how things will look once in HDR, some details (like highlight roll-off) are much more defined in HDR than can be simulated.

Thing's you'll need

There are a few.....

Tools will be back online soon!
Currently making a few minor tweaks.

Updated November 2017, with new and improved LUTs!

In the Zip folder you'll find three files. One is a program called HDR_metaJECTOR, and the two others are LUTs. Keep the program and LUTs nearby for later.

Place a copy of the LUT with "HDR Monitor" in the title in Resolve's LUT directory.

Open the "Windows" folder, then either "32bits" or "64bits" folder (depending on your processor), and download the "mkvmerge.exe" and "mkvinfo.exe" files.

Keep 'em handy.

The Workflow

Getting Started

Setting up Your Davinci Resolve Project

SO, here is how you should set up the "Color Management" section of your Resolve project:

- Input Color Space: **Rec.709 Gamma 2.4**
- Timeline Color Space: **Rec.2100 ST2084**
- Output Color Space: **ST2084 1000 nit**

If you don't have the option to change all of these drop-downs, make sure that the "Color science" drop-down in the "Master Projects Settings" tab is set to: "DaVinci YRGB Color Manage".

Remember the Monitoring LUT that I mentioned earlier (the one to help visualize how the video will look on an HDR display)? This is where we are going to put it to work.

Under the "Lookup Tables" section of the "Color Management" tab:

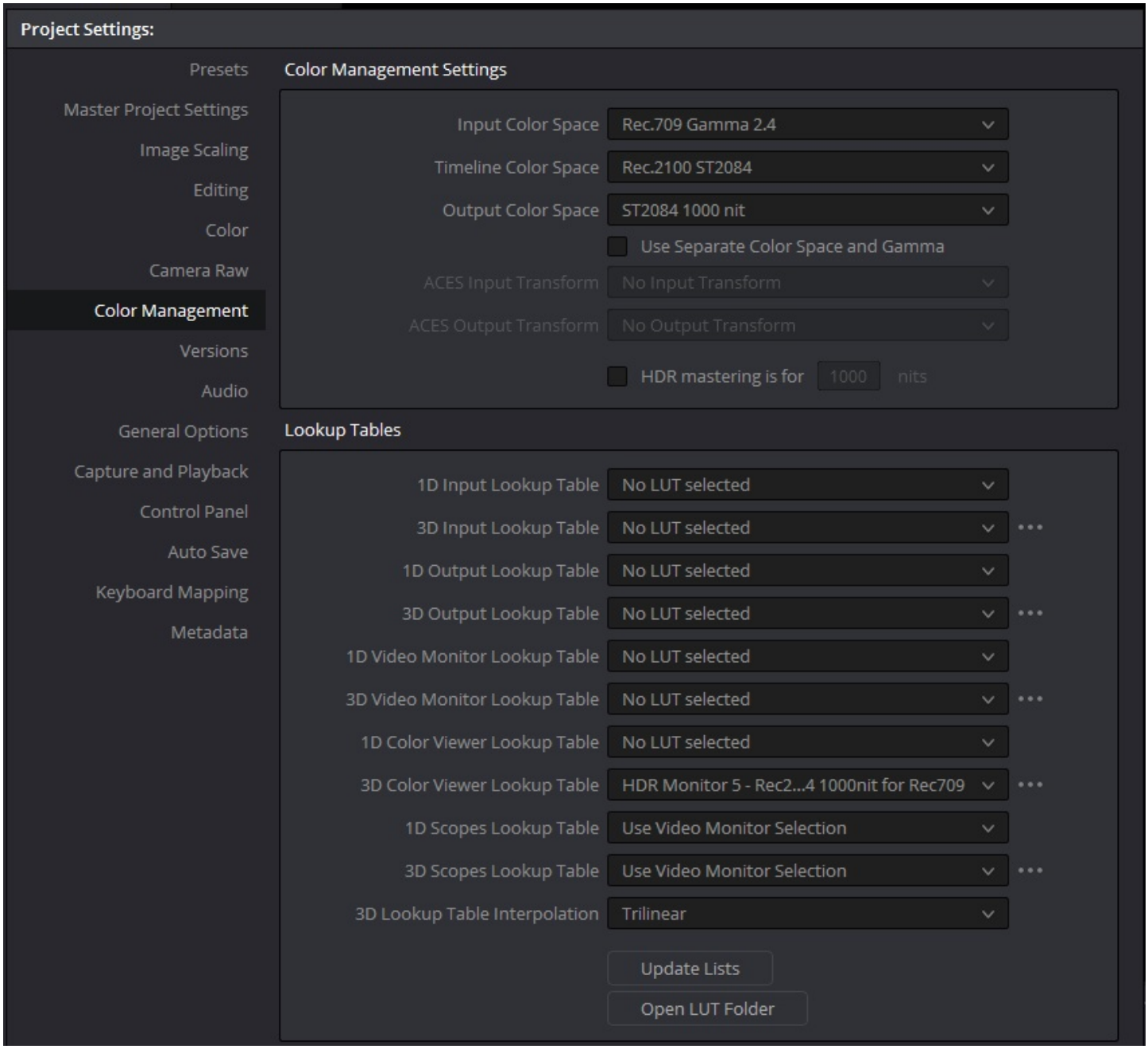
- 3D Color Viewer Lookup Table: **HDR Monitor 5 - Rec2100 ST2084 1000nit for Rec709**

Now for the last step. Head over to the "Color" tab, and select "**Enable HDR Scopes for ST.2084**" (indicated by the blue arrow to the right).

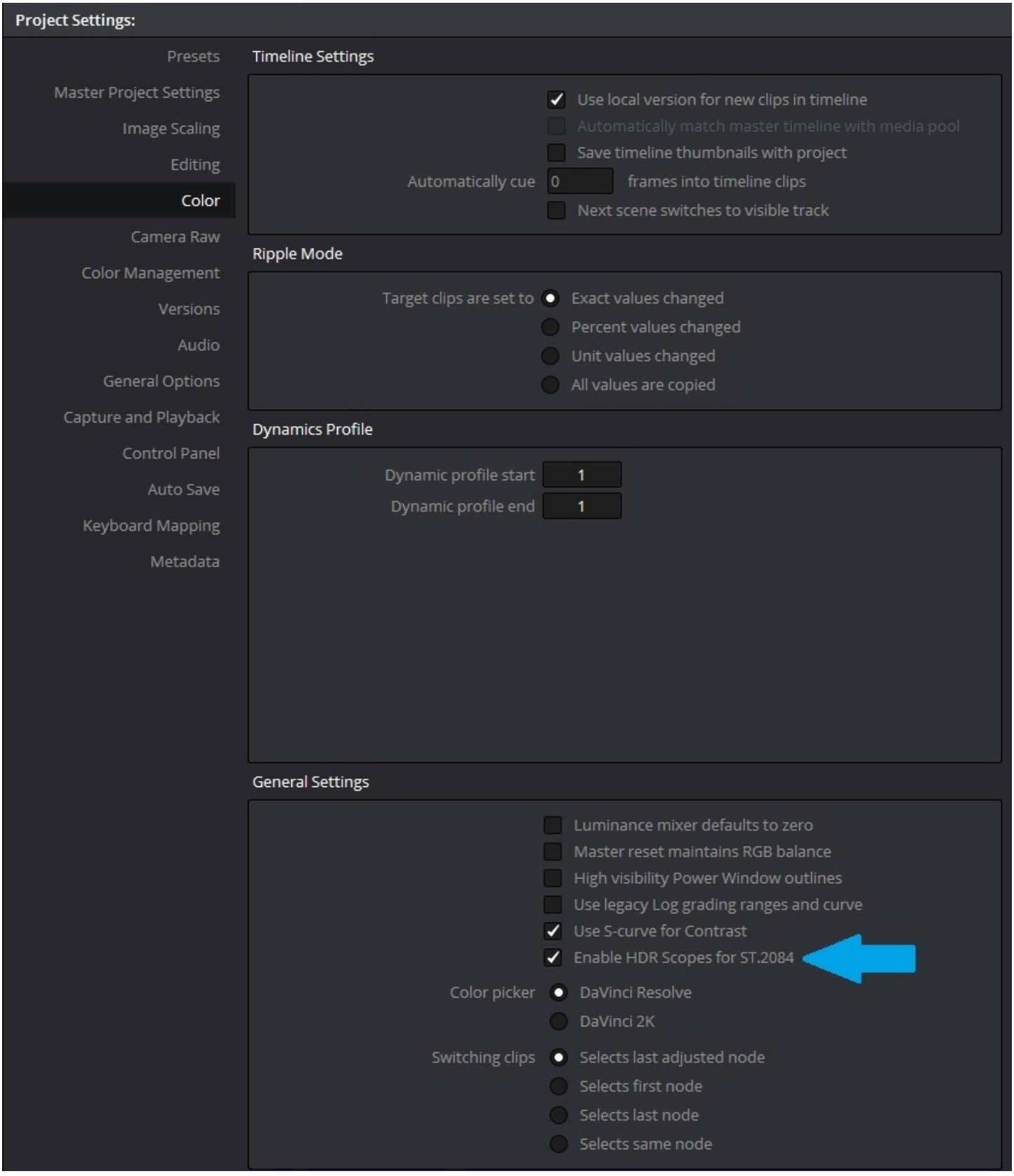
Once all is set, hit "Apply" at the bottom of the Project Settings window.

Immediately, you should notice that the scale of your waveform as changed. Instead of being the Y-Axis reading 10bit data values from 0-1024, it will now show brightness (in "nits") from 0-10,000.

You will also notice that the scale is not uniform, but increases seemingly exponentially. This is one reason why I can't stress enough that (check out this awesome segway)...



"Color Management" tab can be found by going to "File" > "Project Settings"



"Color" tab of the "Project Settings" window

The Scopes Are Your Friend

HDR Grade WITH Monitor LUT

HDR Grade WITHOUT Monitor LUT

grading the footage

As creating a good grade is highly subjective and varies from project to project, I'm not going to be delving too much into the creative side of things.

The biggest thing to remember is to **always watch your scopes**.
Note just how much room on the waveform is dedicated to the 0-100 nit range (out of our 1000 nit max).

The Monitor LUT that I had you implement in the previous section is designed to give a really solid "simulation" of what your grade will look like on an HDR display.

The LUT will only effect what you see in the grading window/"Color Viewer". It has no effect on the scopes, thus allowing you to monitor the actual data levels with the waveform.

So feel free to grade away as normal!

This being said, you will notice that the LUT causes highlights towards the very top of the range to start clipping in the grading window before they actually appear to be clipping in the waveform...

...and that's OKAY!

As long as you keep an eye on the waveform, and see that nothing is clipping...nothing will be clipping in the actualy video data, itself.

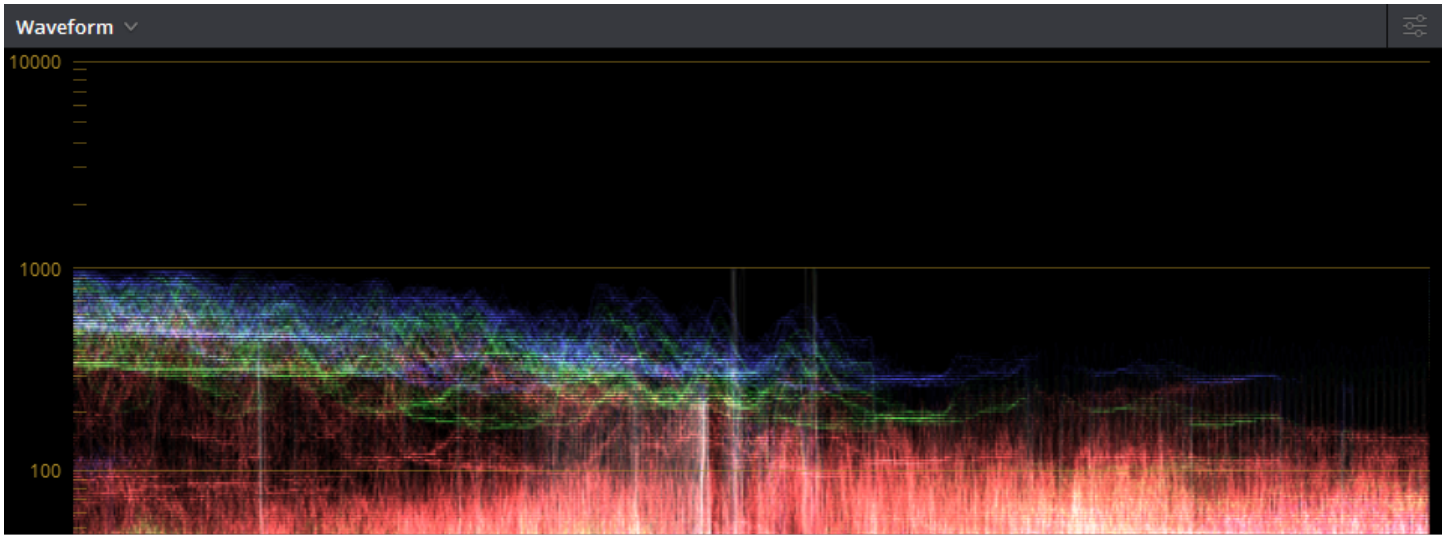
If you are ever not 100% sure, simply disabling the Monitor LUT and re-checking the grading window will give you a visual confirmation (as seen to the left). I do this a bunch.

Don't worry if you have a lot of grades where highlights seem to be clipping in the grading window.

Remember that you can always make a custom HDR to SDR conversion LUT that reins back the highlights for your SDR viewers on YouTube.

I have been doing exactly this with my YouTube HDR to SDR LUT for Austin's video (more on this later).

Taming the highlights has been as simple as bringing them down using the "Highlight" wheel in the "Log" section of the Color Wheels.



Waveform of the Pure HDR Grade, Itself, of the Three Images Below

HDR Grade without any LUTs Applied

HDR Grade with a Custom YouTube HDR to SDR Converstion LUT

HDR Grade with the Monitor 5 LUT

The Power of Windowing

For anyone who gets that pun...my apologies...it was right there.

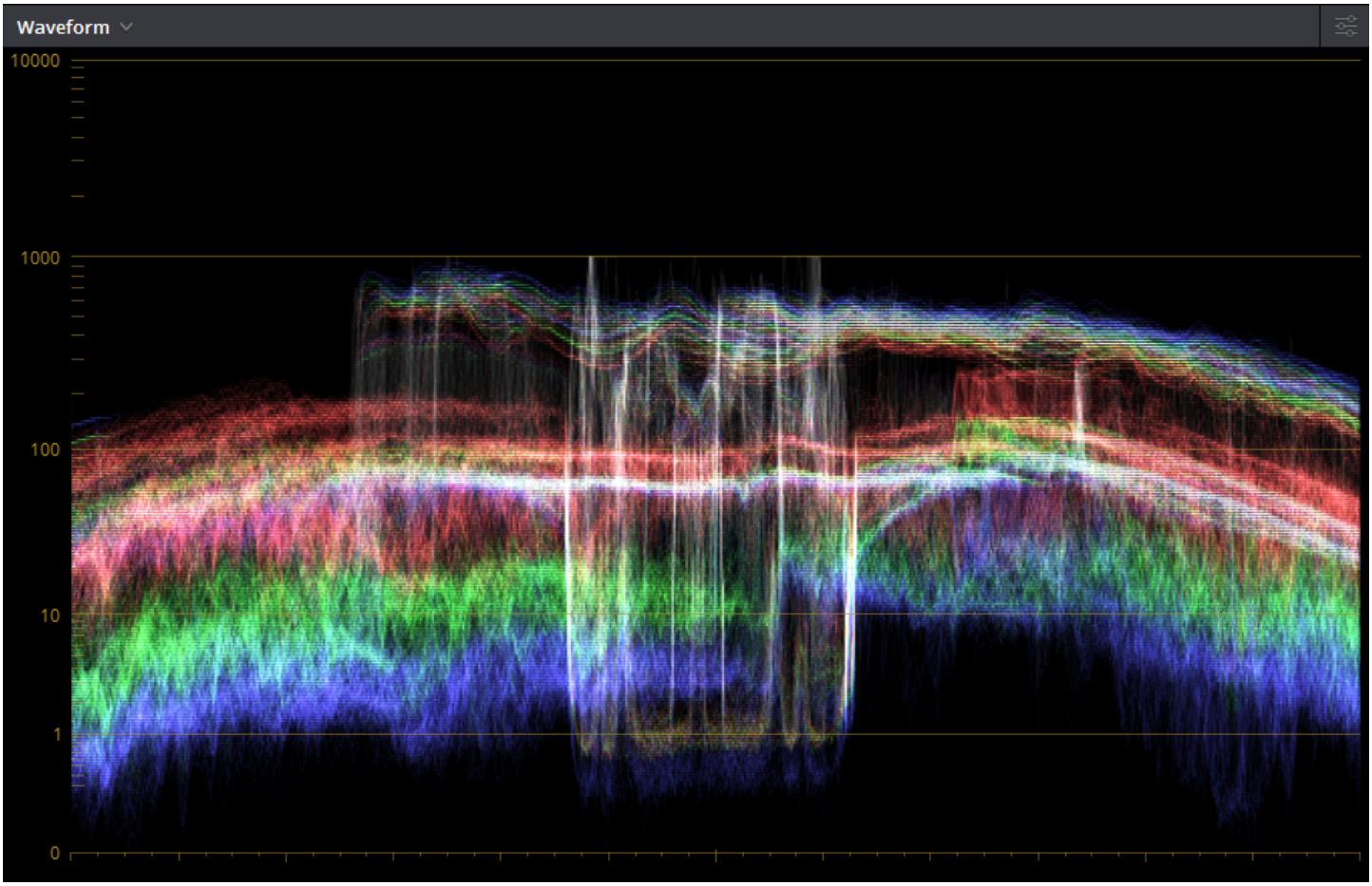
One thing that I've noticed with HDR, is that highlight clipping and exposure changes from shot-to-shot are VERY apparent.

For this reason, and the fact that you now have more latitude to play with, windows are your friend.

Below are some samples from the trailer for Austin's video that have some heavy windowing going on to control both the sky and ground separately.

Using a Power Window to Seperately Control the Sky and Ground

Complete Grade with the Monitor 5 LUT Applied



Waveform of the HDR Grade Above

Take Your Time

When you first start out grading HDR video, you'll notice that some controls, such as the Primary Wheels, effect the image a little bit differently than they do during a Rec.709 grade.

There is an "HDR Mode" that you can apply to individual nodes, but I haven't found it to be of any help, really.

Chances are that your creative LUTs (if you have any) will also not work as you'd normally expect.

My best advice is to take your time, and play around with some test footage before jumping into a big project (as you would with any new video tech).

It doesn't take long to get the hang of HDR grading, especially if you're already comfortable doing a Rec.709 grade. Once you get the hang of things, you'll find that it really is pretty straightforward.

Rendering: Output pt. 1

The rendering/output process is where things are going to get very specific to YouTube. But as I mentioned before, the resulting file should be good to go for other HDR applications as well (just skip pt.2 of this section, if you're in that camp).

There are three main steps here, and the first is the obvious one: rendering out of Resolve.

Here are the video specs that need to be in place when rendering:

- Format: **Quicktime**
- Codec: **DNxHR HQX**

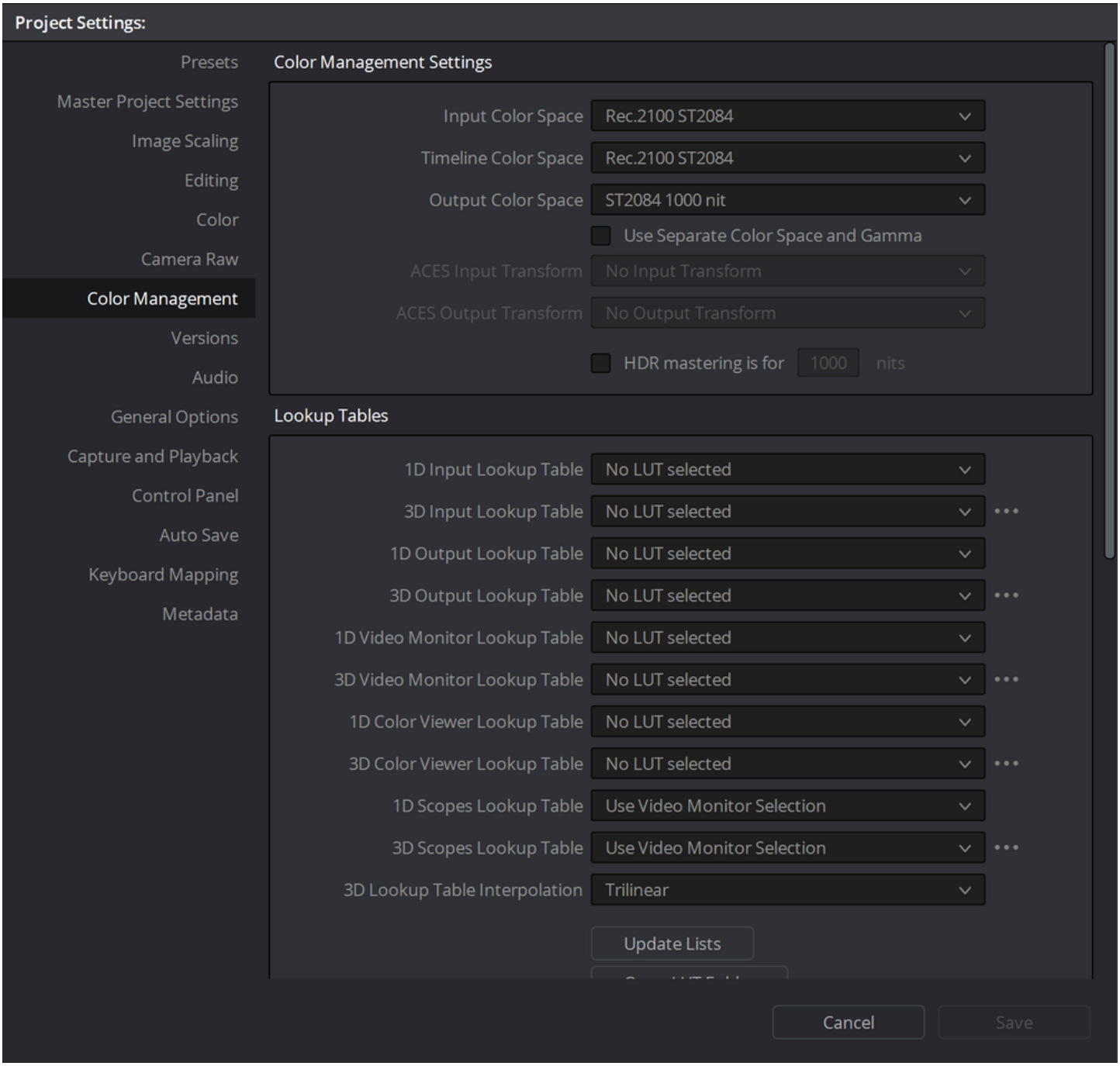
- Pixel aspect ratio: **Square**
- Data Levels: **Video**
- Data burn-in: **Same as project**

THIS WILL CREATE AN MOV FILE

For audio, while YouTube recommends using the AAC codec, I've had no problems going with PCM.

Although they also say that H264 can be used, I've run into issues with using it not triggering YouTube's HDR detection, so I wouldn't recommend it.

ProRes 422 and 4444 are supported as per the documentation, but I have not personally tested them (as my workstation is a PC). That said, if you're on a Mac, give it a shot, and let me know how it goes!



Color Management Settings for Creating a Custom HDR to SDR Conversion LUT

If you wish to create a custom HDR to SDR conversion LUT:

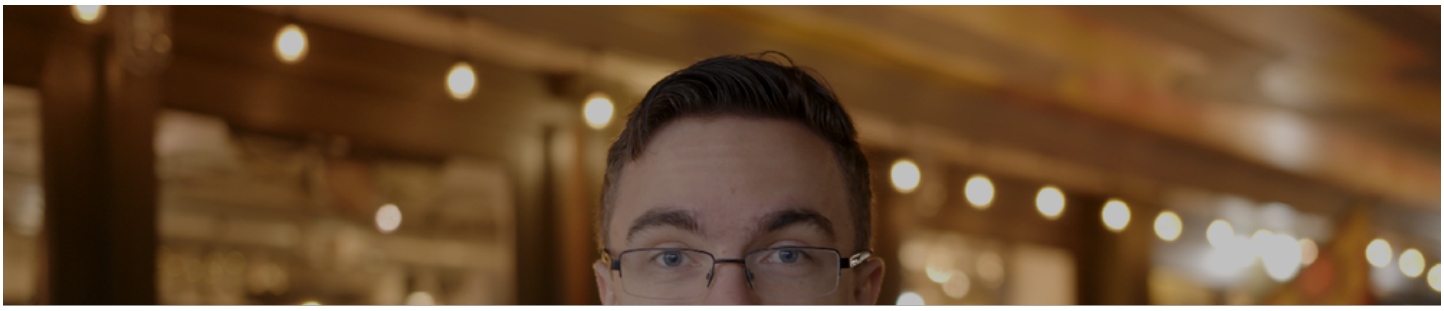
1. Bring your rendered DNx file back into Resolve, and put it in a new Timeline
2. Change the "Input Color Space" to: **Rec.2100 ST2084**

3. Change the "Output Color Space" to: **Rec.2100 ST2084**
4. Remove the Monitor LUT from the "3D Color Viewer Lookup Table" section
5. Create a grade that you like (that works for your entire video)
6. Generate the 3D CUBE LUT

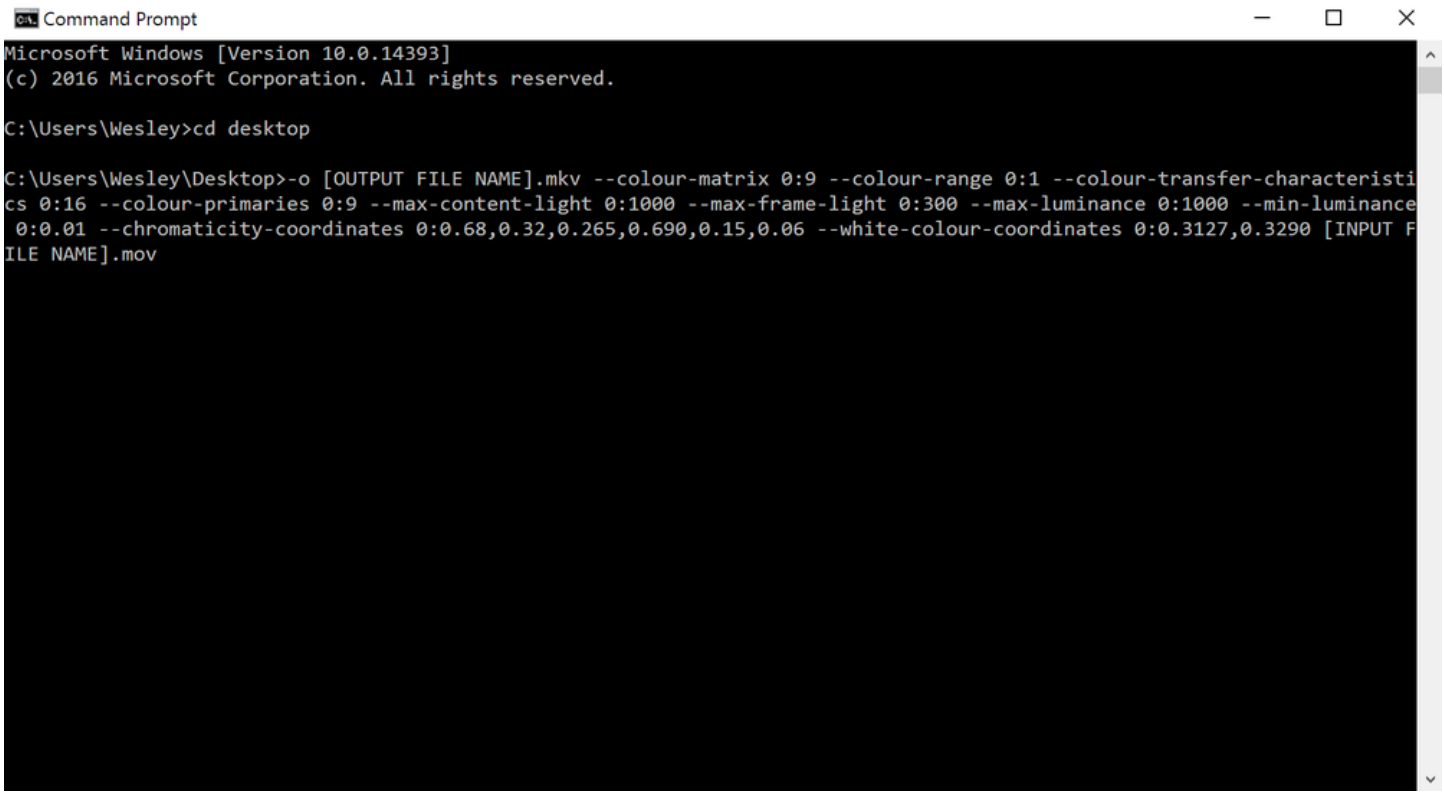
As seen in the image to the right, the LUT can be generated by right clicking on the clip (while inside the "Color" page), and selecting: "Generate 3D LUT (CUBE)".

And that's it!

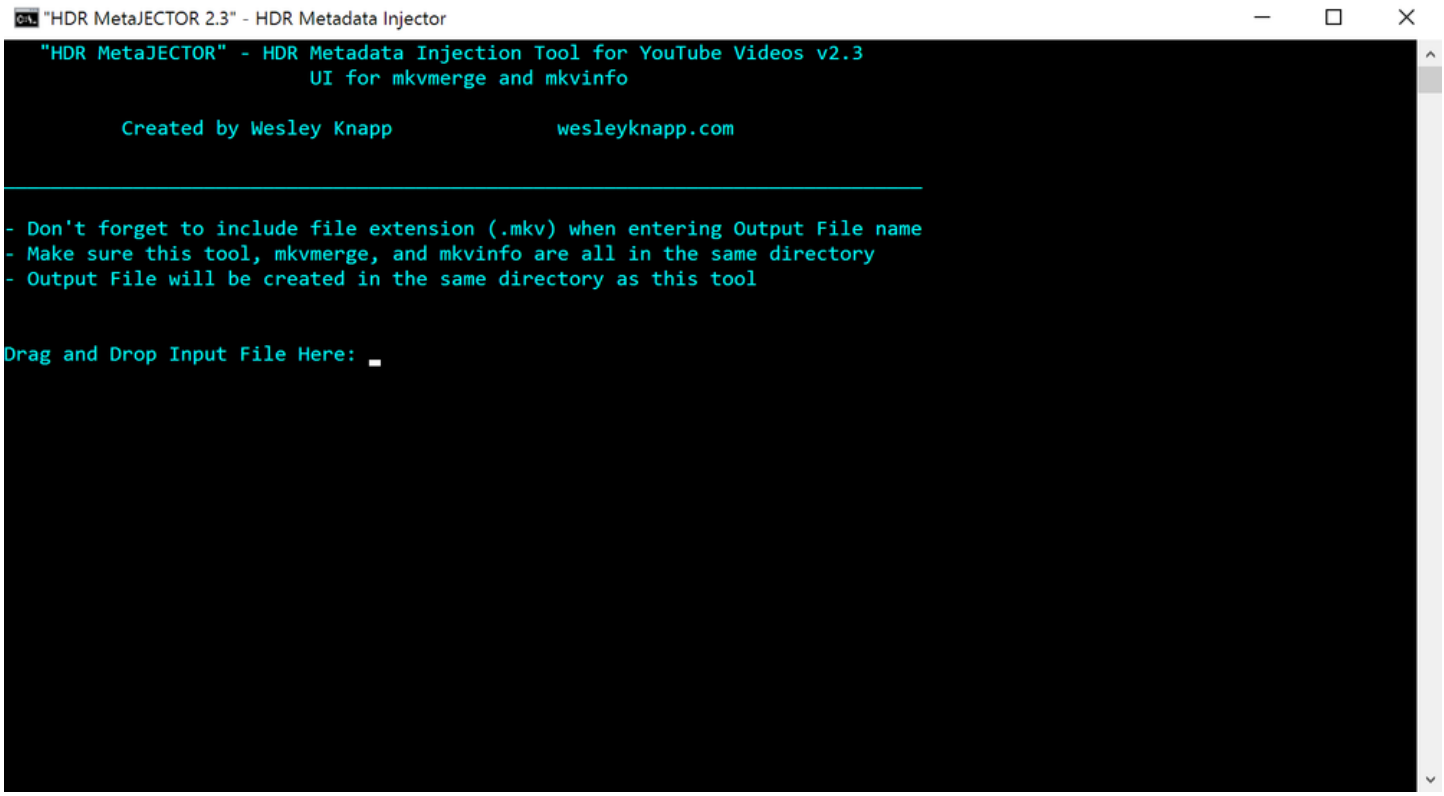
If you'd prefer to just use the Monitor LUT ("HDR Monitor..."), it will work perfectly fine. You can also use the second LUT that I included in the "Wes's HDR Tools" download ("YouTube-HDRtoSDR..."), which essentially just adds a little contrast and tones down saturation.



HDR Grade with the "HDR Monitor 5" LUT Applied



mkvmerge.exe with the Recommended Flags



MetaJECTOR v2.3: A Simple UI for mkvmerge.exe and mkvinfo.exe

MetaJECTOR should be pretty straight forward to use, but just in case, here's how you operate it:

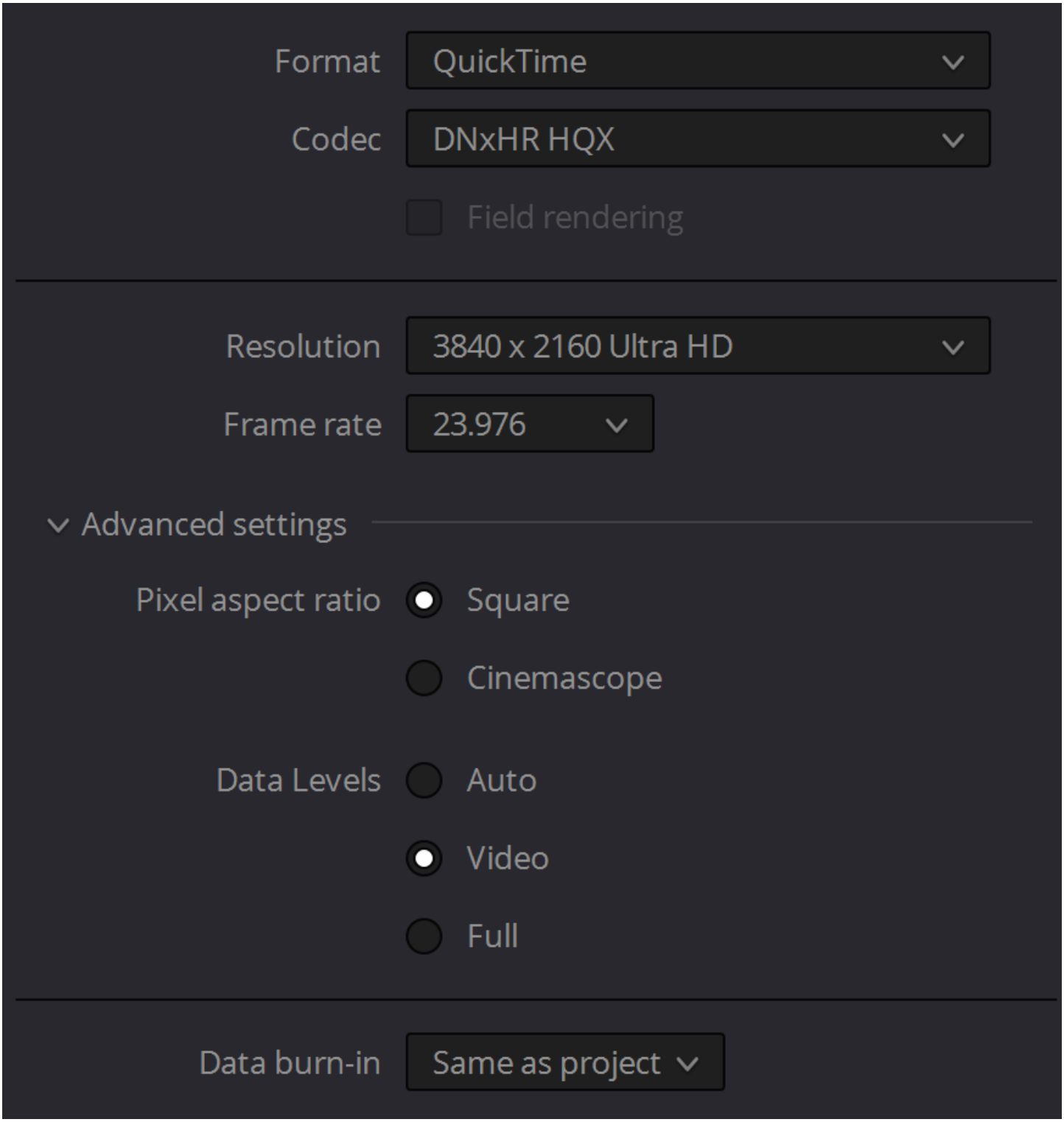
1. Double click on the MetaJECTOR program to open it
2. Drag and drop the Rendered DNx file onto the window, and press ENTER
3. Type in the desired name for the new output file (don't forget to include the .mkv extension on the end), and press ENTER
4. Type: "Y" if you wish to include a custom HDR to SDR LUT, or "N" if you don't (and want to use YouTube's automated processing). Press ENTER
5. If you typed: "Y", now is when you drag and drop the LUT onto the window that you'd like to use (you won't see this step if you typed: "N"). Press ENTER
6. Double check that the file paths look correct. If all looks good, press ENTER, and watch the magic happen.

Red text will appear on the screen with a progress percentage at the bottom. This is mkvmege.exe creating the .mkv file.

The time it takes to complete will vary greatly depending on length of video, and computer hardware specs. That said, it should be relatively quick.

Once the file creation is complete, the text will go green, and display a whole bunch of metadata in the file. This is mkvinfo.exe being run, and is a great time to give everything a last look.

Press any key to exit, and you're done!



custom HDR to SDR conversion: Output pt.2

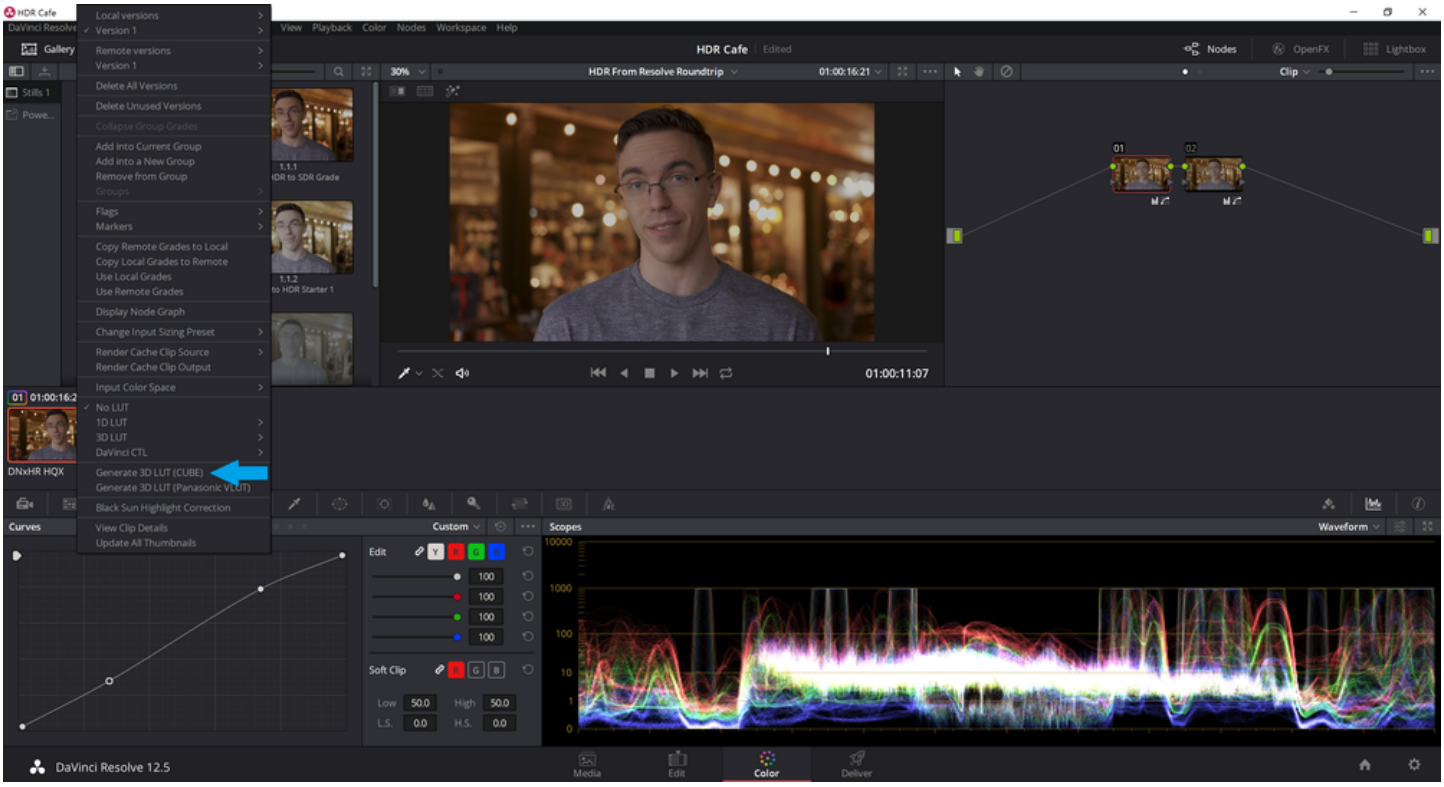
By default, when YouTube detects that you've uploaded an HDR video, it will automatically create an SDR (or "Standard Dynamic range") version of the video.

This is what people with SDR (non-HDR) displays will see when viewing your video, and honestly, the conversion isn't too shabby.

In an ideal world, YouTube would allow us to upload an entirely different video file to show to these viewers (one graded for Rec709)...but alas, they don't.

What they DO allow us to do, though, is upload a custom LUT with the video. This will then be used instead of their automated process to create the SDR version of the video.

Pretty nifty!



Creating a Custom HDR to SDR Conversion LUT in Davinci Resolve

As I mentioned earlier, Austin's video employ's a different HDR to SDR LUT for the YouTube upload. That said, I've done MANY test uploads with the Monitor LUT, with perfect results each time.

HDR Grade with an Early Version of the Custom HDR to SDR LUT, for YouTube

The Funky Part: Output pt.3

As things currently stand, Resolve will output the video with all of the correct color management, but it doesn't yet have the ability to encode all of the metadata YouTube needs.

There is no manual way to tell YouTube Uploader that the video you are uploading is HDR. Because of this, you need to inject the missing metadata, so as to trigger YouTube's HDR processing.

The way to do this is via the "mkvmerge" application (that I had you download earlier in the "Things You'll Need" section). This tool packages the rendered DNx file and a custom HDR > SDR LUT (optional) into an MKV file that YouTube will read as being HDR.

Unfortunately for us Windows users, this tool is only available to us in an .exe

to be run from command line with a bunch of flags.

THIS BEGAN TO BECOME A BIT OF A NUISANCE, SO...

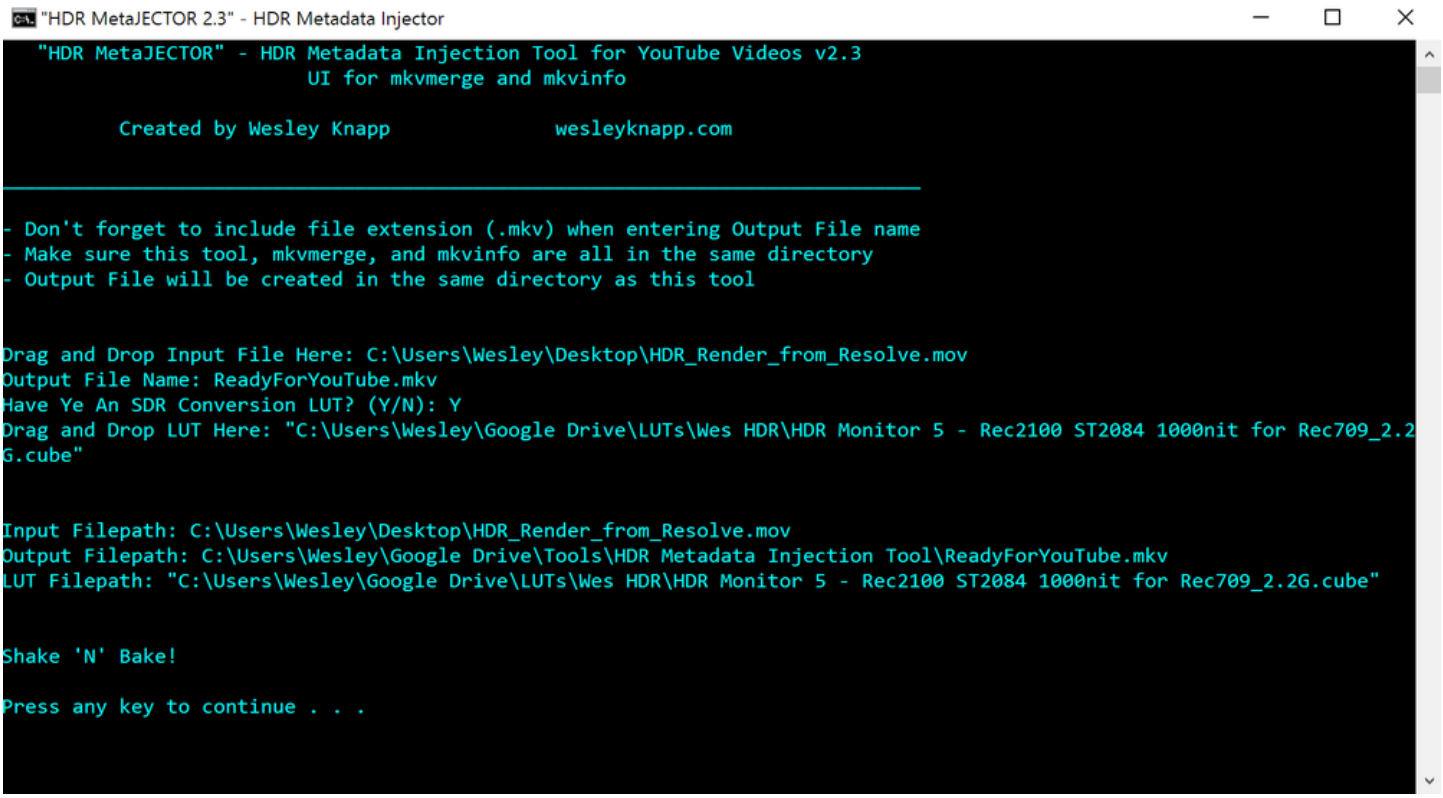
I created a drag and drop ui called metajector

It's a lightweight Windows Batch Script that I've included in the "Wes's HDR Tools" zip folder that I had you download earlier.
It's super simple to use, and has some funky fresh colors!

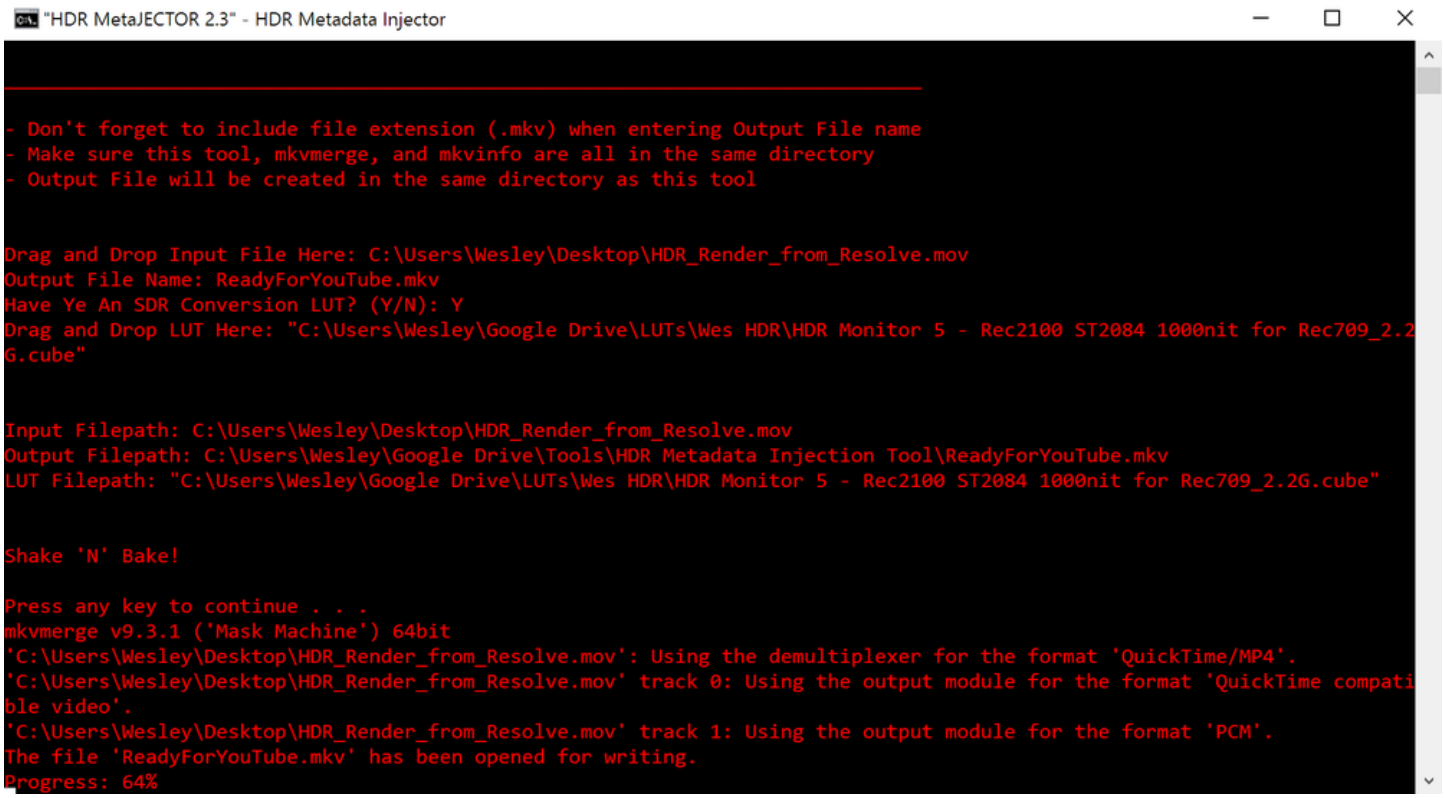
Eventually I'll upload it to the official GitHub where mkvmerge and mkvinfo are located...

Just make sure the tool (called: HDR_MetaJECTOR_vX-Y) is in the same folder as mkvmerge.exe and mkvinfo.exe, and you're good to go!

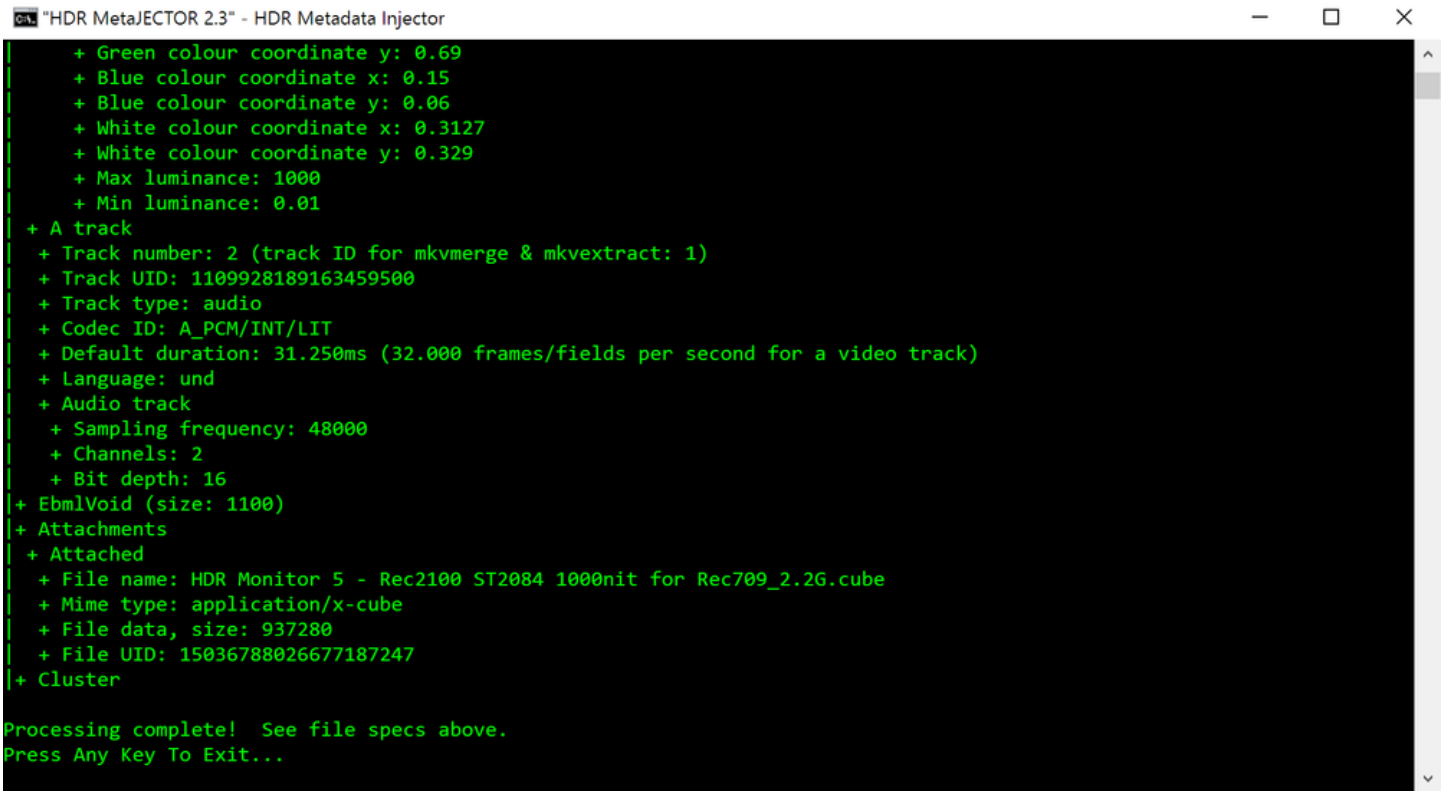
THIS FOLDER IS WHERE METAJECTOR WILL PLACE THE NEW FILE IT CREATES.



MetaJECTOR v2.3, Ready to Create an .mkv File. Look at that Color! How Fun!



MetaJECTOR v2.3, Creating the .mkv File using mkvmerge. Oh Wow, it's Red Now!



MetaJECTOR v2.3, Displaying the Output File's Metadata using mkvinfo. Hoory, it's Green!

You now have a video file ready for upload!!

Upload your new .mkv file to youtube, as normal

And while you wait, go hang out with friends or something...after all, you are uploading a (most likely) huge video file to YouTube and have some time to kill.

Tell your them about your awesome new HDR video.

Trust me, they'll think you're SUPER cool.

Feel free to give me a shout if this worked for you, or if you have any questions!

Email: wesleyknappfilm@gmail.com

Twitter: @KnappInColor

Or just drop a comment below!

Q&A

Answering some of the questions that I've received...and you may have had while reading this

(Updated as I get 'em)

Q: *Wes, why did you select "Video Levels" when rendering out of Davinci?*

A: It seems that as with Rec.709 video, YouTube still likes HDR videos to be uploaded with legalized video levels. Extended range (labeled as "Data Levels" in Resolve) will result in crushed blacks. This conclusion came after a bit of testing, by creating and uploading custom stepped gray charts, and seeing how YouTube treated them after processing.

Q: *Wes, is Ken as sassy in person as he seems on camera?*

A: It's worse. Much worse. Don't give him coffee...a box of White Cheddar Cheez-It's can dampen things, but don't hold out hope.

Q: *Why did you put the Monitor LUT in the "3D Color Viewer" section?*

A: Putting it there means that the LUT will ONLY be applied to what is being seen in the grading window. It WILL NOT effect the scopes, which is what you want (so you can see the actual data levels on the scopes). It also won't be in the rendered output. Setting the Monitor LUT as the "3D Video Monitor Lookup Table" (while removing it from all other drop-down menues), will result in it only being applied to the scopes. Setting it as the "3D Output Lookup Table" (and removing from all other drop-downs) will cause the LUT to be

applied to BOTH the grading window and scopes, as well as any render you might do.

14 Likes