



## ACES for the Rest of Us

The fact that ACES is more than just another buzzword in the VFX industry should be pretty obvious by now. Anyone working at a major studio in a reasonably up-to-date pipeline will have had some contact with ACES. But what about the one-person-dancing-bands, the freelancers with mobile workstations and those who like to dance on several weddings? This article should make it a little easier for them to get started.

by **Christoph Zapletal**

**W**hy ACES? This question is, of course, completely justified. Why should I change my own workflow to ACES? Probably the best and most important reason is to bring yourself up to speed before everybody else does, because one thing is clear by now: ACES is here to stay and has established itself as the new standard. But beyond that, there are also very practical reasons for ACES. ACES is „Scene Referred“ - earlier workflows were that already, but with ACES this now happens in a nearly open standard.

Whereas with „Display Referred“ one's own display was the measure of all things and one could therefore by default only work for one output format, namely that of one's own monitor, „Scene Referred“ means that with the aid of colour management the settings that apply to the scene in question are equally valid in all the different output formats and on the most diverse playback devices. Everyone sees the same picture. This is ensured by a shared library of LUTs. Especially for users who are finishing or compositing footage from multiple

camera types, it will be an enormous relief to know that all the material is in the same, lossless color space. But also for artists who have different deliverables, from web content to HDR masters, ACES is the best and, in view of future developments, also the most sustainable architecture on which to base their workflow upon.

### A little refresher course

This article will focus on the practical setup of an ACES workflow entirely without Pipeline TD. We will clearly focus on the Post-Production aspect. However, before we get into Media Res here, we'll refresh the basics a bit here.

First of all, let's clarify the most important question: What actually is ACES? ACES is, contrary to what many first assume, not a color space, but a collection of conversions, standards and yes, also color spaces. By the way, ACES stands for „Academy Color Encoding System“. Within ACES there are also color spaces that contain the name „ACES“, for example ACEScg or ACES

2065-1. More about that later. Already you can see that one of the hardest parts of ACES to learn are the abbreviations. That's why we'll get to the two most important ones right away: IDT stands for „Input Device Transform“ and ODT stands for „Output Device Transform“. The IDT is responsible for the conversion of the footage from the colour space of the camera to the working colour space. There are two groups of working spaces, the APO and the AP1 primaries.

The APO Primaries, also known as ACES 2065-1, are defined as Colour Space so far and open that all light perceivable by the human eye and even more is covered. This means that not only the signal from every camera currently on the market fits into this Colour Space, but also that of all future cameras.

Not quite as wide as the APO Primaries, but still larger than all classic Spaces are the AP1 Primaries, consisting of ACEScg, ACEScc and ACEScct. ACEScg, as the name

#### ACES Abbreviations

- ▷ ACES – Academy Color Encoding System
- ▷ AP – ACES Primary
- ▷ IDT – Input Device Transform
- ▷ RRT – Reference Rendering Transform
- ▷ ODT – Output Device Transform
- ▷ OCN – Original Camera Negative
- ▷ OCIO – OpenColor Input/Output
- ▷ LUT – Lookup Table
- ▷ LMT – Look Modifying Transform
- ▷ CTL – Color Transformation Language

suggests, is primarily intended for CGI data and therefore relies on a linear transformation of the brightness information. ACEScc, emulating the behavior of classical grading tools, uses a logarithmic transformation. Last but not least, ACEScct is a variant of ACEScc, which emulates the behavior in the lower ranges, i.e. the „toe“, for which the t stands, in a slightly different way. So much for the IDTs.

The ODT in turn converts the processed image for playback into the colour space required by the respective playback device, i.e. monitor, projector, tablet, etc.. Strictly speaking, in addition to the ODT, an RRT (Reference Rendering Transform) is also responsible for this, but this is usually applied under the hood by the software without the user intervening here. For the sake of simplicity we will only talk about ODT from here on. For example, for rendering in SDR this would be rec.709, for displaying web content on a classic computer monitor it would be sRGB and for displaying fancy HDR on a state of the art monitor it would be P3-D65 ST.2084(PQ), whereby for HDR you also have to specify the possible NITS of the monitor.

These three terms, i.e. IDT, Working Colour Space and ODT, will be thrown at us in almost all the software when we start to set our pipeline to ACES. However, you should not let this scare you too much. Anyone who has already used active or passive colour management in software will also have had to deal with input and output luts and a working colour space. The terms may be different, but many processes and principles are quite similar.

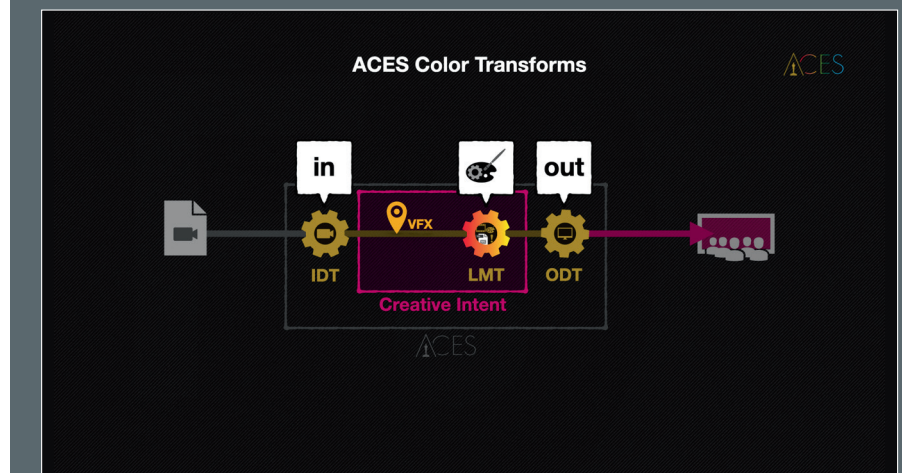
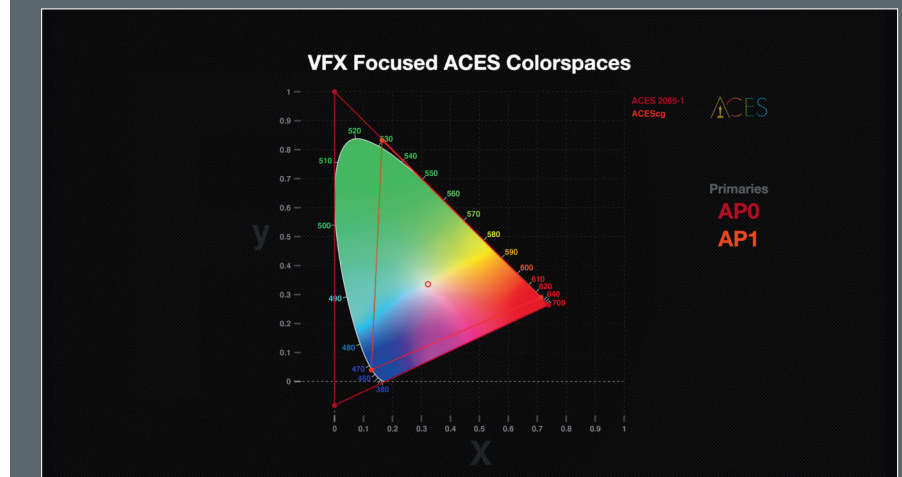
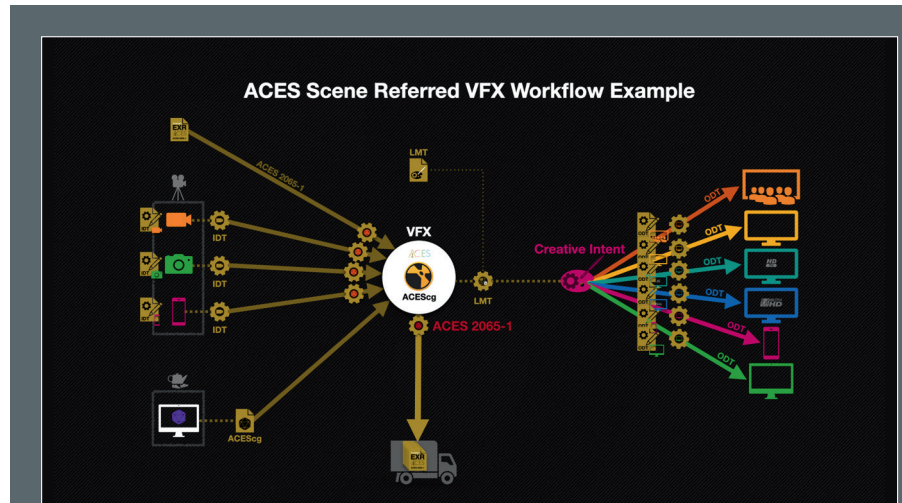
Before we leave the theory part behind, we should talk about the different ACES versions. ACES is an active standard that is intended to grow and, as described above, to support future cameras and output devices. Therefore, new versions of ACES are released at regular intervals - currently ACES 1.3 is the state of the art. It should be noted that these different standards are backwards compatible. No or only minimal changes are made to the actual Colour Spaces, rather new ones are added.

The focus of these new ACES versions, however, is on the IDTs and ODTs. This is only logical, after all the ACES colour spaces are defined so openly to support future sources and playback possibilities. The IDTs and ODTs updated and added in the version updates are the key to this. Ergo : If a software for example only supports ACES 1.1, this is no obstacle at all to reuse

the resulting files in a software that uses ACES 1.3. What can happen, however: For example, if a software only supports ACES 1.1, but you get footage from the very latest Sony or Blackmagic camera, the corresponding IDT may not yet be included. This should be taken into account when setting up your workflow.

## The practical part

In this workshop we will focus on some widely used VFX tools, namely DaVinci Resolve, Nuke, Flame and After Effects. For those who miss one or the other CG renderer, the focus will clearly be on applications that have to deal with different footage sources. For a CG renderer like Arnold it is relatively easy to set the output to ACEScg, if necessary you would have to convert supplied textures to ACES.





This article also ignores pure editing tools like Premiere or AVID, so the focus is clearly on finishing and compositing. We will look at Resolve and Flame as ingest, grading and finishing tools and Nuke and After Effects as comp tools to and from which ACES roundtripping is possible. But, also worth noting here: the principles are similar everywhere, and since ACES is an open standard, a lot more software will support ACES than this article can cover.

And one more thing: This article is not a one-to-one instruction on how an ACES workflow should look like. Rather, it's meant to provide a safe starting point for the most common apps on the market - how you use these different apps together in ACES, whether or not Flame or After Effects, for example, show up in your equation - that's your choice and completely adaptable.

## Resolve

The top dog of grading tools is well prepared for ACES, but Resolve inherently loves its own „DaVinci YRGB“ color science. From the get-go we have two options for how we want to set up our workflow: Do we rely on Resolve's colour management or do we prefer to do it ourselves on a scene-by-scene basis. If Resolve is only used for pulling plates and grading, but the finishing is done in another system, then it might make sense to leave the colour management to DaVinci.

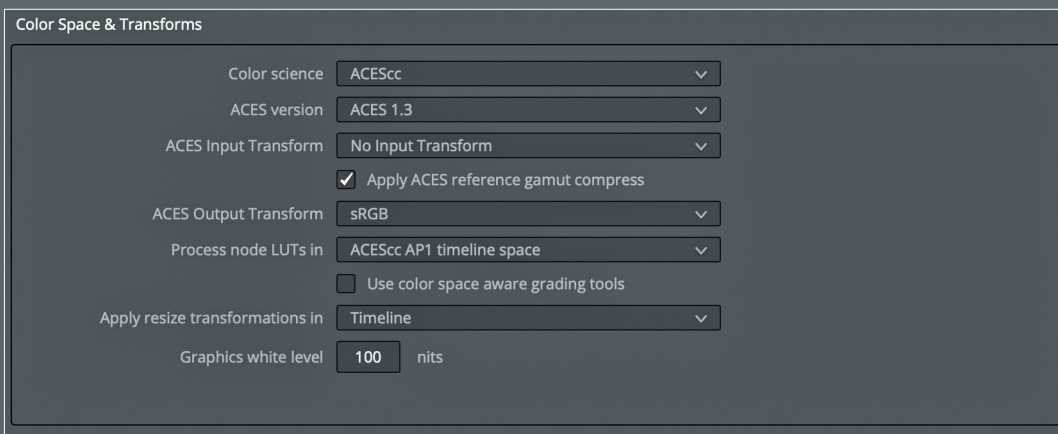
However, if we also want to finish in DaVinci, a manual workflow is the way to go - especially when working with titles and graphics, it's easier if Color Management doesn't interfere.

## Color Management through Resolve

In the Preferences, which we reach via the cogwheel at the bottom right, we select the „Color Management“ section in our current project and select ACESc.

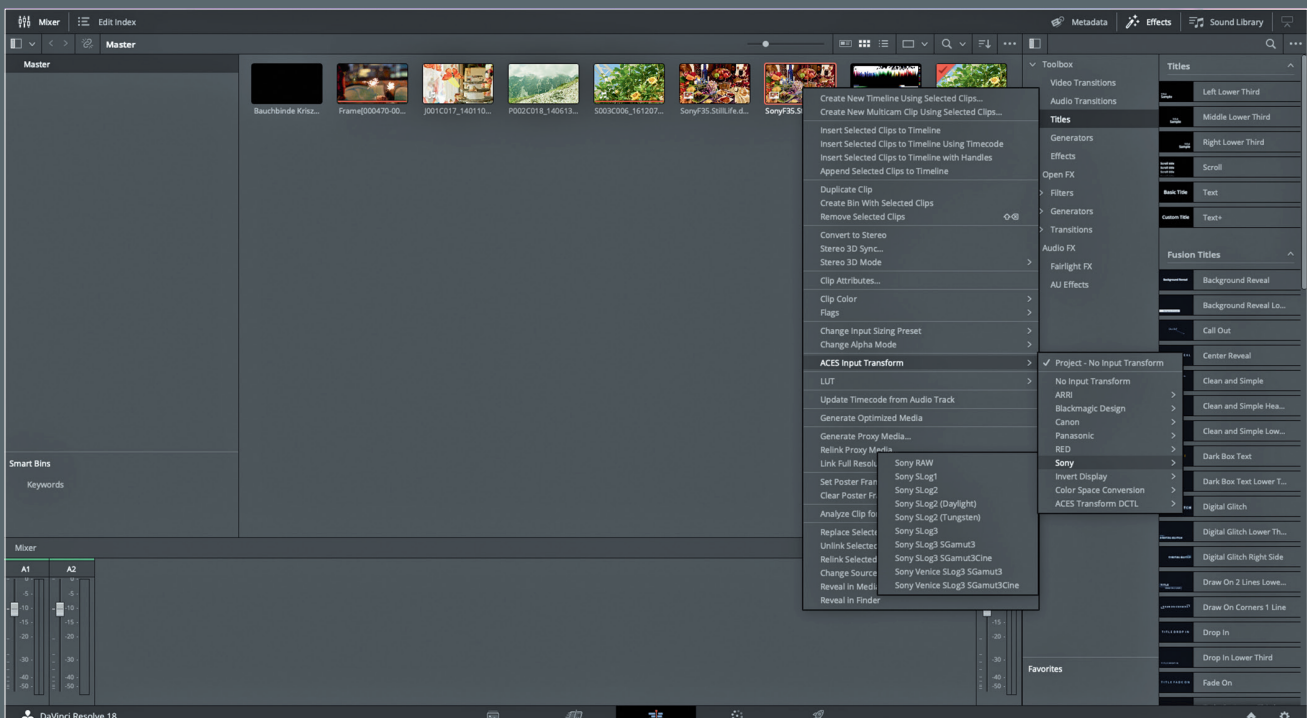
Alternatively, we can also use ACEScct. Technically, it makes no difference which Colour Science we choose here. Next, we select our ACES version. In Resolve 18, ACES 1.3 is the current version. Now we can already select a predefined IDT. This makes sense if you only have footage from one and the same source in the project.

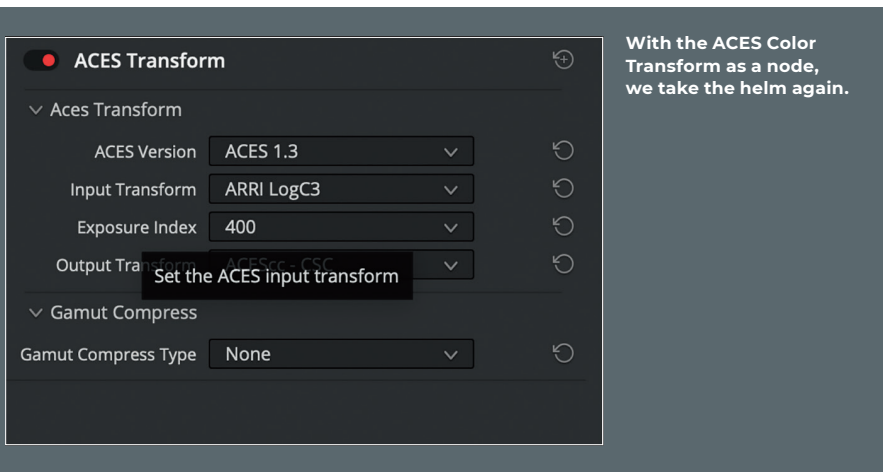
That way you save yourself the trouble of manually tagging when importing later. However, this also avoids the automatic detection, where Resolve correctly tags the colour space based on the metadata. Ex-



With these Project Settings, Resolve takes care of most of the color management.

Even though Resolve takes over the management, the artist still has to intervene with the IDTs.





ceptions are RAW files, where Resolve does the input mapping on its own in any case. However, this explicitly does not mean the setting of the debayer, exposure and colour temperature settings.

To start with, we don't select an Input Transform and move on to the ODT. Unfortunately, Resolve is a bit awkward here, as it uses the same Output Device Transform for the display on the monitor as well as for the export. For us, this means that we first select the ODT that we need for the correct display on our grading monitor. Before exporting, however, we must then return here and select the correct ODT for export. Depending on our individual setup, we now select sRGB, rec.709 or another suitable ODT.

So that's it for now with the preferences. But if we don't have any IDTs selected in the preferences, how can we tag our clips? The easiest way is to right-click on the thumbnail in the Colour tab. Here we can select the appropriate IDT for our footage under „ACES Input Transform“. The usual suspects like ARRI Alexa, Sony Venice, Black Magic etc. are available here.

## User Managed Color

Just because we take color management into our own hands doesn't mean we have to do without ACES. We can leave the preferences on default for the time being, the preset „DaVinci YRGB“ and an output transform corresponding to our setup are perfectly fine here.

The step that is usually done by the Colour Management, namely the conversion from and to ACES, we simply move to the Node Level. In the „Color“ tab we find the ACES Transform Node among the OFX plugins. This offers us the same options as the Colour Management, only on a clip or time-

line level and not for our whole project.

This means that we can also select our ACES version, our IDT and our ODT. Depending on the footage we now select the corresponding Input Transform and choose ACES as Output. But our image will still be displayed incorrectly in the viewer, because we have linearized our image, but our display is not linear. Unfortunately, unlike Flame and Nuke, Resolve does not have a dedicated Viewer Transform. This isn't a major problem though, for this we can take the timeline level of our Colour Tab (top right dropdown menu from „Clips“ to „Timeline“) and select an ACES Transform again.

This time we select „ACES“ as Input Transform and the Colour Space of our monitor as Output Transform, i.e. rec.709, s. RGB or whatever fits to the device on the desk. This Transform now lies like an Adjustment Layer on our entire Resolve Timeline and thus ensures a correct display on our monitor - it is only important that this Transform is switched off again before the export, otherwise it will be included in the export and we will not export a correct ACES.

By the way, if you find it too tedious to go through each clip individually and manually distribute aces colour transforms, here's a tip. Resolve has the wonderful option of combining various clips into groups. If you have several clips from the same camera in a group, you can define not just one, but two transforms for the group - a Pre Grade and a Post Grade.

That means for example the PreGrade group can contain the ACES Colour Transform Node with the transform from Camera Colour Space to ACES and Post Grade then the conversion from ACES to the out-

put format. This then allows a lot of the work to be automated even without Colour Management. However, it leaves us the freedom to leave ACES again at a certain point if this is necessary for our workflow.

## Export of the files from Resolve

No matter whether using the managed or the manual workflow - Resolve does not have a Viewer Transform and so either our Output Transform has to be set to „No Output Transform“ in the Preferences or our ACES Transform has to be deactivated on the Timeline Level. Once this is done we can switch to the Deliver tab and prepare our export.

The best export format is OPEN EXR in RGB Half PIZ compressed. Half stands for 16 bit, which is more than sufficient for our footage files and PIZ as a lossless compression is read by Nuke without any problems. In any case we should render in Source Resolution. In the Advanced Settings we select „ACES APO“ as Color Space Tag and check the boxes „Disable sizing and blanking output“ and „Force Debayer to highest quality“.

## Who needs LUTs anymore?

We've been doing Colour Space conversions for a while now and haven't needed a single LUT yet. Well, that's not quite true, because the IDTs and ODTs we've been using all this time are, strictly speaking, also LUTs. But, and this is an important distinction, this time we don't want to do a technical conversion to a different color space, we want to make the creative look created in Resolve visible and usable in other software.

### Further Reading

#### Resolve/Nuke

▷ ACES - from Resolve to Nuke by Lars Wemmje

<https://youtu.be/JW3z--WICvw>

▷ DaVinci Resolve to Nuke Roundtrip using ACES by Finn Jaeger

<https://youtu.be/8SG80SSkyGU>

#### After Effects

▷ Color Managed Workflows in After Effects

<https://youtu.be/Ins6AuJzBnI>

#### Flame

▷ Color Management in Flame

<https://youtu.be/mE8sRax78ul>

▷ ACES in 8 Minutes

<https://youtu.be/bsqESUhr7vw>



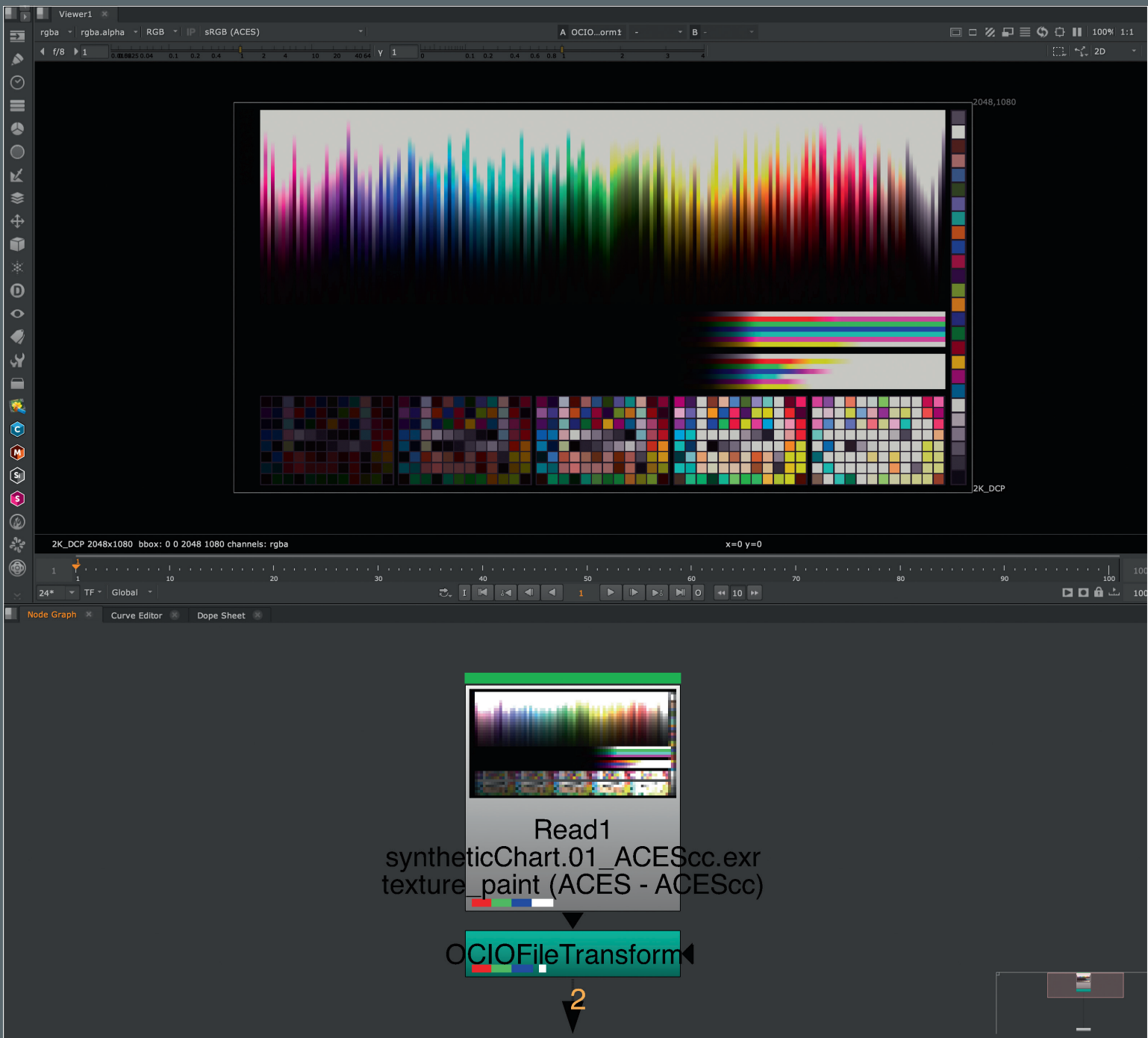
This is meant for when we either want to do the grading ourselves in Resolve, or have a colorist deliver LUTs to us for our ACES pipeline. Here's the thing to keep in mind - any corrections that use masks, gradients, blurs, or similar limitations, for example, will be ignored by our LUTs. But if really only the color value of a pixel is changed, these changes can be stored in a LUT and passed to another tool. For example, to see directly in Nuke what the comp looks like with a rough grade applied, this is a fine way that can save unnecessary

roundtripping. To export a LUT from Resolve, we activate the nodes in the Node Tree of the Color Tab that should be part of the LUT, right-click on the thumbnail of our clip and select „Generate 3D LUT(65 Point Cube)“. Select the destination folder and a name - done.

### So what about graphics?

Now that we talk about graphics, we have to make a basic distinction. Are the graphics part of a scene, for example a screen insert that is tracked into a smartphone, or

are we talking about titles, lower thirds and other graphics that take place quasi outside the scene. Let's assume the first scenario first: We get a screen insert in sRGB, for example as a tiff file. We can apply an IDT from sRGB to ACES and then work with it normally within the scene in which it is to be used. This is not a problem because we have a common denominator for the black and white values within the scene and can then match them. So we also work here „Scene Referred“.



Thanks to OCIO, we are now fully ACES compliant in Nuke.

Things get more complicated when we have graphics that have no context to the scene, take for example the lower third mentioned earlier. Its whiteness has no relation to the scene it sits on, but it is supposed to be white for the end user. ACES is not an output format and the biggest problem when dealing with graphics is the incredibly wide range of brightness values that one display can have compared to another. A white font in rec.709 has a brightness of 100 nits. But with HDR displays we quickly reach 1000 nits - the 100 nits of rec.709 are a pale grey. As you can already

guess, this is very much about the display, so we are working with „Display Referred“, something that ACES is simply not designed for.

And this brings us to the main reason why the ACES workflow managed by Resolve is not practical for mastering. Because if you select ACES Colour Management in the Preferences, every timeline within the project is considered ACES. In the preferences you have the possibility to set a Graphics White Level, for example. This defines the brightness of titles, colour areas

and gradients generated in DaVinci, but in practice this function fails the moment you import external graphics. The bigger problem comes when mastering, when you always have to set this white level value in the preferences to create different output masters. It's just very difficult to understand and error-prone.

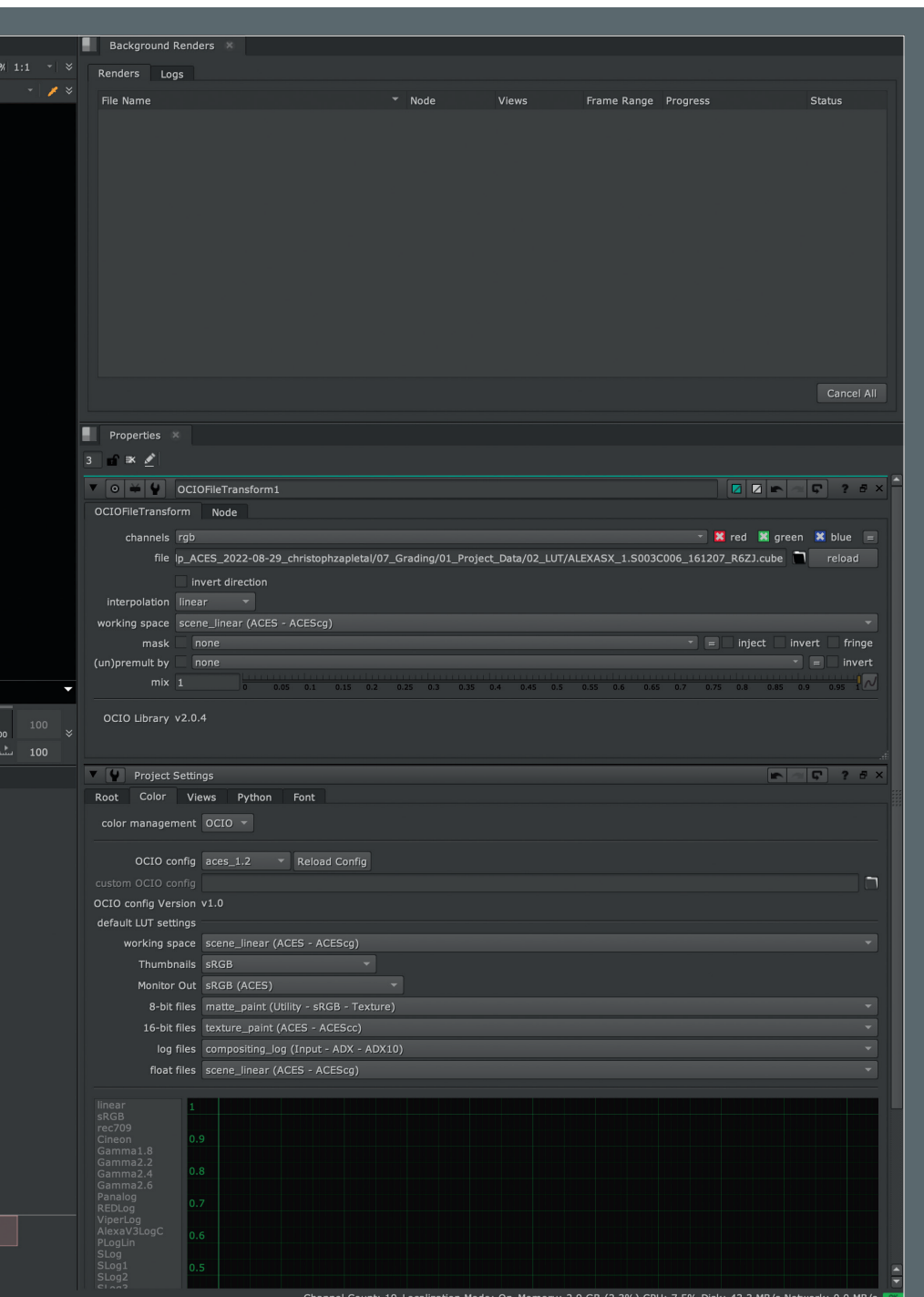
This is the strength of the manual workflow, where thanks to the ACES Colour Transforms the transformations do not take place globally, but locally from scene to scene. Here you can finalise your grade on a master timeline and then move it via the corresponding ODT into the colour space of the respective deliverable. If you then put graphics and titles on the timeline afterwards, you can do this in the best possible quality for the respective output medium. And if you need to create multiple deliverables, you can copy the timeline, adjust the ODTs and put the graphics on it accordingly.

## Roundtripping to Nuke

Color management systems are like opinions - everyone has one and far too many keep them to themselves. So Resolve has „DaVinci YRGB“ and Nuke has - very creatively - „Nuke“. This is the name of the in-house color management, which in principle has ensured since Nuke's youngest days, that all scenes in Nuke are first linearized, in order to comp them. However, Nuke only takes the gamma curve into account, the color gamut is completely disregarded. Fortunately, since version 11, Nuke comes with OpenColorIO (OCIO) profiles that we can pass colour management to. And in the current Nuke 13.2 these also support ACES in version 1.2.

This means that we first go to the Scene Settings in Nuke (park the cursor in the Node Graph and press „S“) and switch to the Color tab. Here we select OCIO instead of Nuke. Directly underneath we can select the OCIO Config File and choose ACES 1.2. Underneath we see suggested behavior, which types of files should be provided with which IDT by default. Of course we can adapt these to our workflow, but for the start the defaults here will work just fine. One last thing we need to do: Unlike Resolve, Nuke has its own Viewer Transforms for ACES, and so we need to set our viewer to the correct Colour Space. For a standard PC or Mac monitor this would be sRGB(ACES).

Now we can load our exports from Resolve into Nuke via a Read Node. If we



also have access to the previously exported LUT, we can load this via the „OCIO File Transform“ node. It is important that we select the same Colour Space as in Resolve, i.e. ACESc. Now the output should be exactly the same as in Resolve (please remember to turn on the ODT back on in Resolve that was deactivated for the export).

Of course, we can also create the OCIO File Transform Node as a Viewer Input to make it more quickly usable at any point in our comp. Otherwise, the work from here on is not much different than in the classic Nuke workflow. Only when exporting from Nuke we now have different options in the Write Node, which OCIO makes available to us.

In the end, an ODT is also applied in the Write Node, and so we can set our colour space for an EXR sequence to ACESc or, if we want to follow the rules by the book, to ACES 2065-1 for roundtripping back to Re-

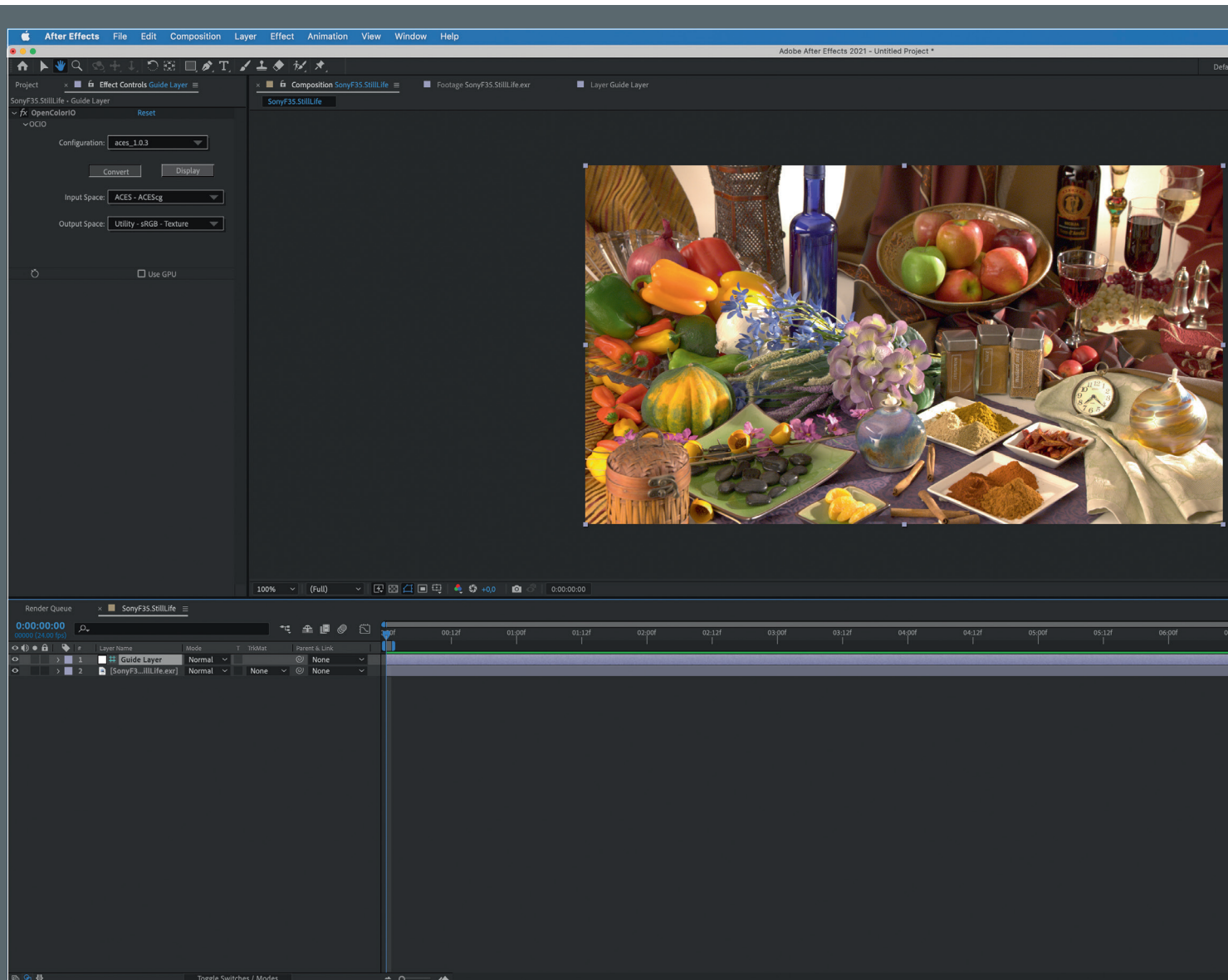
solve. In principle, however, there is nothing to stop you from rendering a file in rec.709 or sRGB directly from Nuke for reviews, as long as you select the correct color space in the Write Node.

## ACES in After Effects

We've already talked a lot about graphics and Scene Referred vs Display Referred. So it should come as no surprise that we're talking about Scene Referred work in After Effects, i.e. working on the plate. Unfortunately, After Effects is not very well equipped for ACES. For one thing, After Effects loves to do a lot of colour management itself. On the other hand, it doesn't even come with the required OCIO plugin and the corresponding config files by default. But we can download them under [is.gd/ocio\\_in\\_ae](#) and [is.gd/ocio\\_config](#).

After that, it's time to get rid of all the color management in After Effects. In the Project Settings under the Color tab we set

the Bit Depth to 32 and the Working Space to „None“. Now we can start importing our exports from Resolve. However, After Effects still wants to interfere with us, so we right-click on our footage under „Interpret Footage“ and check „Preserve RGB (disable color management for this item)“. The image will look quite dark now, after all we are looking at a linear image without a viewer transform on a non-linear display. So we need a Viewer Transform, which After Effects doesn't have by default. So again, we're going to use a trick that we've used before in the manual workflow in Resolve. There we applied an ODT to the timeline level. We're going to do the same thing in After Effects with an Adjustment Layer. So we create an Adjustment Layer above our footage and name it „Viewer Transform“. Now we select „Guide Layer“ by right clicking on it - this way it will be ignored later when rendering and we won't accidentally bake our Viewer Transform into the export. Then we search for our newly installed OpenColorIO Plugin.





As Configuration we choose ACES 1.2. The Input Transform should be the same as our export from Resolve, i.e. ACES 2065-1. As Output Transform we take rec.709, because by default After Effects always uses rec.709. And by the way - we can also apply our LUT from Resolve in exactly the same way, i.e. via Adjustment Layer and OpenColorIO plugin, if we want to. If we now want to use graphics - which is quite common in After Effects - our OpenColorIO plugin is used again - this time not as an adjustment layer but directly on the file. We set the input to Utility - sRGB - Texture and the output to ACES 2065-1. From here we can continue comping in After Effects as usual.

For rendering, we now put our scene on the Render Queue. In the Output Module Options we select an OpenEXR Sequence and the already familiar PIZ compression in the Format Options. And in Color Management, we check the „Preserve RGB“ box again - otherwise After Effects will try to mess with our picture again.

## Tagging and Input Rules - ACES in Flame

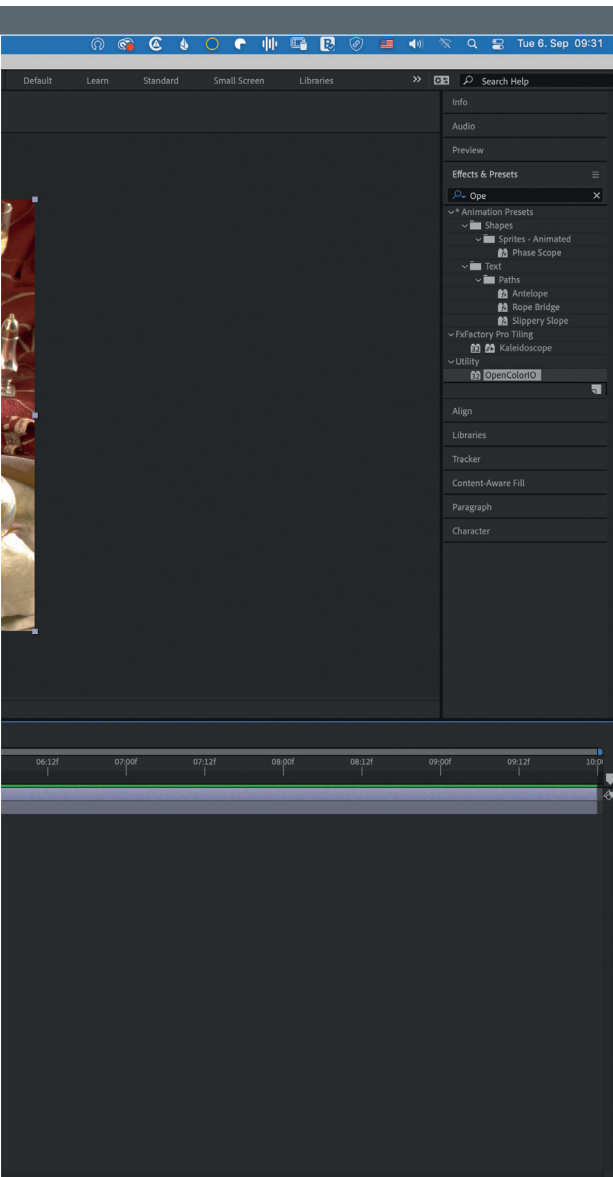
If Resolve's Color Management wants to make it as easy as possible for the artist by doing as much of the transformations under the hood as possible, Flame has a diametrically different approach. Flame's color management is designed to be adjusted, modified and changed at every point. This makes it a bit confusing and intimidating at first, but also leaves us with a lot of freedom and options not available to us in Resolve. In the conform and grading phase, it's not that interesting for us yet. But that changes abruptly when it comes to finishing and exporting.

Let's start with the very basics: We create our project. Under the „Color Policy“, „ACES 1.1“ is available as an option in Flame 2023. Together with our other presets like frame rate and resolution we choose this.

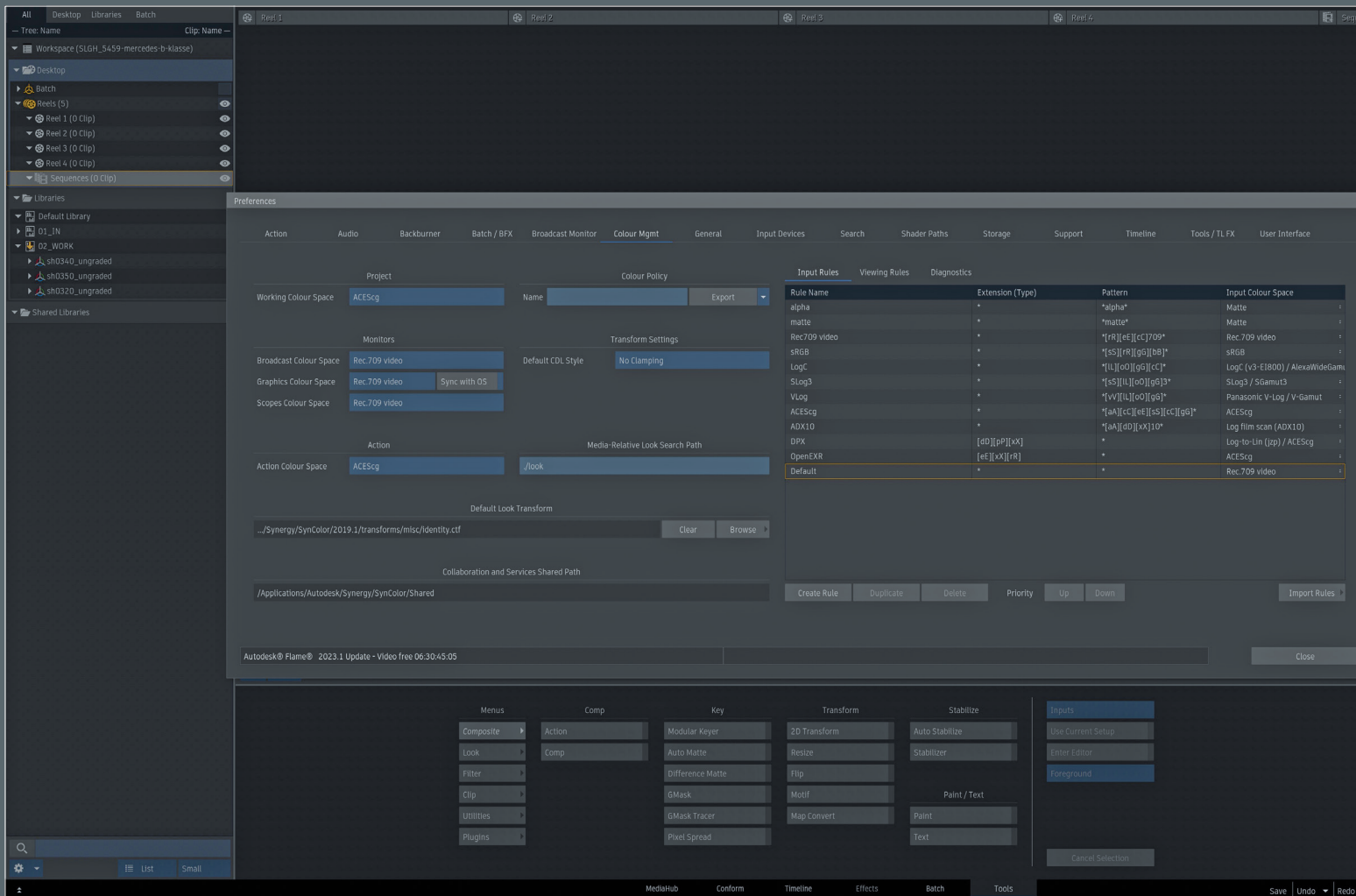
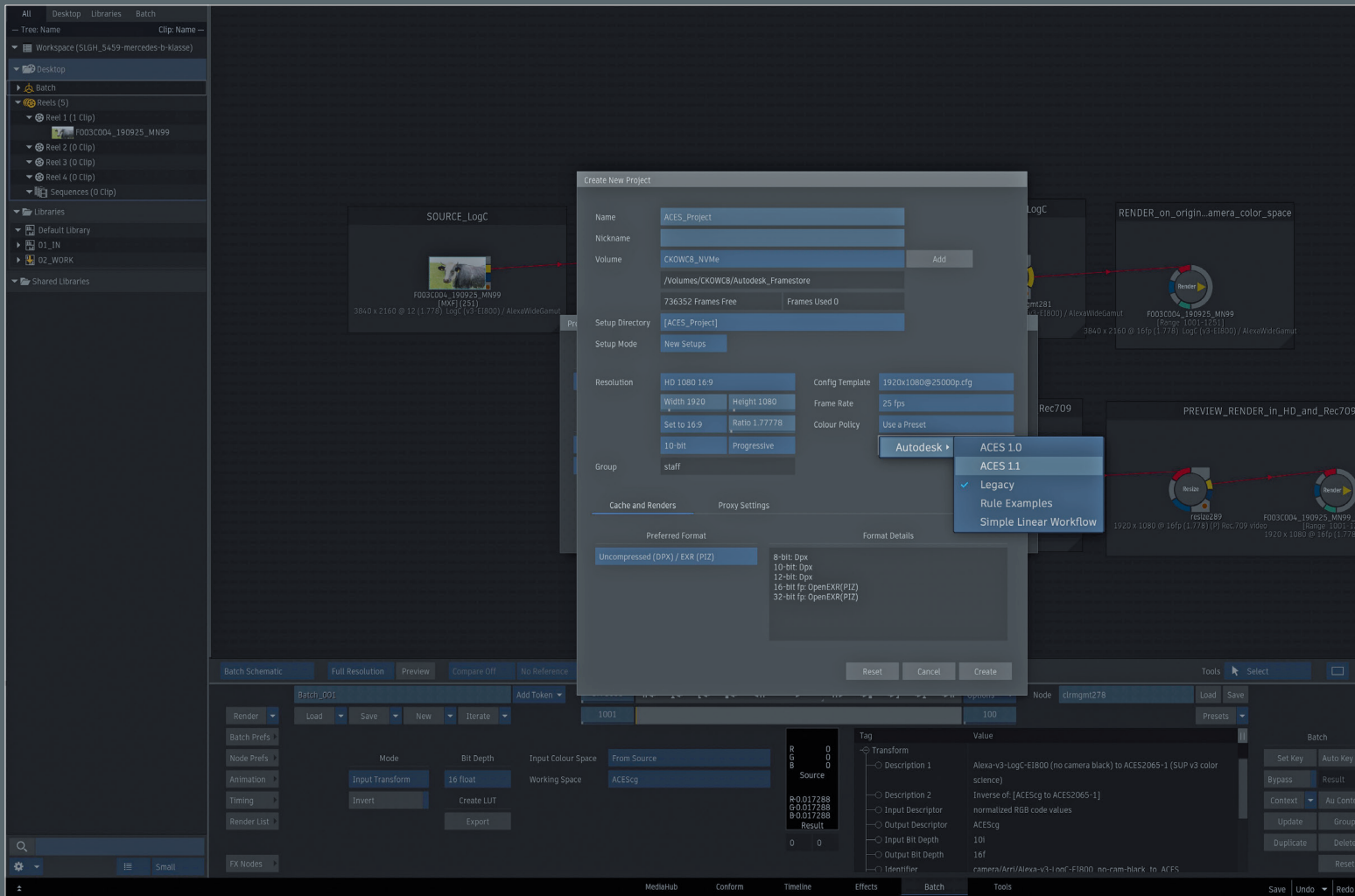
All of these settings, and that includes Color Management, are considered by Flame to be a starting default rather than set in stone. That is, Flame allows us to create a rec.709 composite or sRGB timeline even in an ACES project. These Project Settings only ensure that ACES is selected as the default for new nodes and timelines and that some import and viewing rules are created that are helpful for the ACES workflow, which we will briefly discuss later. When importing clips in the Media Hub, Flame offers us two options that we should briefly discuss: First, we can tag clips, and second, we can convert clips. Tagging means that the actual media is not converted, but the clip is marked within Flame as to which color space it is in. Converting converts the clip from its native color space to a working color space. Both ways of working have their merits, but just because Flame offers the possibility of color space conversion in so many places, we should at least opt for tagging to begin with - if only it helps us gain more understanding and practice in using Flame's color management.

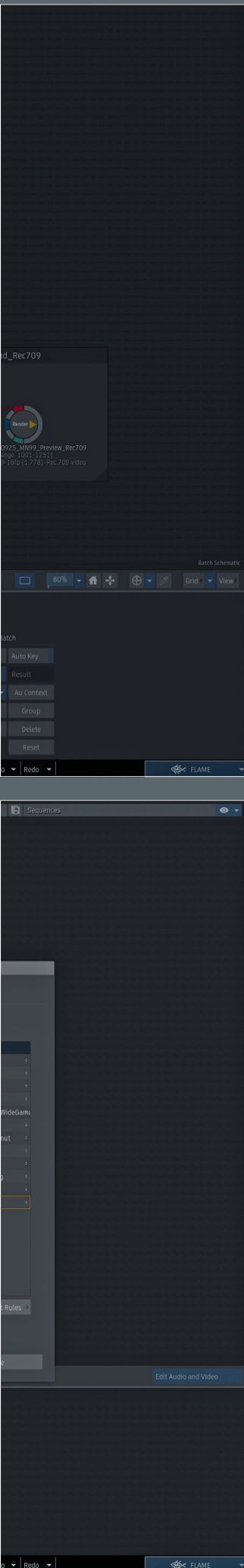
Next to the pulldown with the previously mentioned options, there is another pulldown menu with the color space to be tagged (or converted, if applicable). By default this is set to „From File or Rules“, which means that Flame will try to detect the color space either by metadata (from File) or by the previously mentioned „Input Rules“. So now, if we exit the Media Hub for a moment and go into Preferences, under the Color Management tab, we can see these rules - at least the ones that Autodesk has attached to the ACES Color Policy. In short, tokens can be used to create automations, so that, for example, files that have a certain name, a certain extension or are located within a certain folder, are automatically assigned a certain color space.

This is enormously powerful when building your own workflow, but it's also definitely an advanced feature. For now, we're going to tag our clips manually. It's relatively snappy and it's just more transparent as to what's happening where. So now when we open up the pulldown, we can now select the Color Space of the footage and drag it into our Library. The tagged color space is now literally attached to the clip, on every thumbnail, in every infobox we will now be able to see it. And just to avoid any misunderstandings - the clip is not yet in ACES color space.



**You have to persuade After Effects a bit, but finally everything runs in ACES here, too.**





Flame is probably the most flexible, but also the most comprehensive color management among the finishing tools.

Once you're comfortable with ACES in Flame, you can automate a lot of things with the Input Rules.

There are various places in Flame where a clip can be converted, and Flame is also very lenient about allowing clips of different „coloeur“ to coexist. So a batch setup can contain several clips of different color spaces and also a timeline where a RAW clip lives between an ACES clip and an sRGB clip are possible in principle - but with restrictions.

Let's start on the Timeline: How does Flame manage to correctly display these three different Color Spaces side by side? Well, Flame has one thing that we sorely missed in Resolve: the View Transform. This is basically an ODT for our playback display. But it's not rigid, so it doesn't just convert from ACEScct to rec.709 for example. It's also not applied to the export by default - unlike in Resolve - but switched „live“ to the viewer. The View Transform invokes a second set of rules - the Viewer Rules. We can look at these too in the Preferences under the Color Management tab. They are used to define how each material is converted and on what type of display.

And that's why tagging is so important in our workflow - it gives Flame the information about how the material is supposed to look and how it has to be converted. By the way, we don't need to create our own viewing rules - the set provided by Autodesk should be more than sufficient for most of us.

In Batch, we can process these images from different Color Spaces together, but the Color Science will definitely be wrong - we need to bring the images into a common Color Space. Like almost everything in Batch, we handle this via a node, namely the „Color Management“ node. Here we can do - among some other „legacy“ options - two Transforms that will become important for us: The View Transform and the Input Transform.

You might think that Input Transform is Flame Lingo for IDT and View Transform for ODT, but that's not quite true. To make an IDT, we use an Input Transform in any case and select „From Source“ under „Tagged Color Space“, because we have already tagged our source correctly. Under „Working Space“ we then select our ACES Color Space. This can be ACEScct or ACEScg, as in Resolve, but also ACEScg, which is very helpful in a comp-focused app like Flame. With the „Invert“ button this process is also reversible and can be used to create an ODT.

The View Transform on the other hand is purely for the ODT and works according

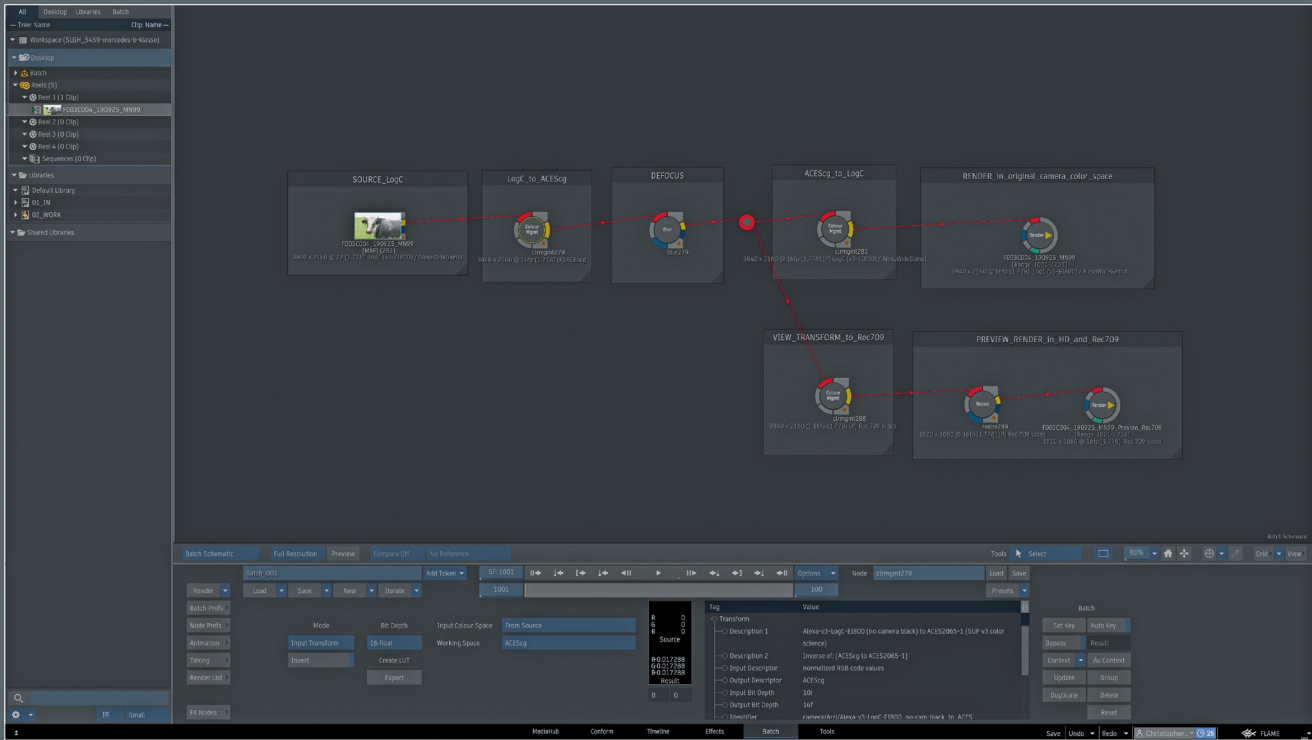
to the WYSIWYG principle. Here Flame looks at the active Viewing Rules and takes them as the ODT to apply. Very handy if you want to export a clip or a timeline quickly - but rather unsuitable as a conversion in batch. So we stick with the Input Transform and convert the clips to the ACES flavor of our choice. From now on we can combine all clips in Action - Action in this instance behaves like in any other Color Policy - either it takes over the Color Space of the inserted background or jumps back to a default defined in the Preferences - in Flame's ACES Color Policy this ACES is CG - this is also important when generating things like lens flares or particles in Action.

The IDTs and ODTs just described in the form of View Transform and Input Transform are found in half a dozen places in Flame. We can set them on the media level, we can set them on the incoming clip in batch, we can do it in a separate node as just described, we can muddle it in within the render node, we can apply color management on the timeline or on export. This all makes sense, but it's way too much to get started with. We limit ourselves to three ways: Inside the flowgraph in Batch (Which we have already discussed), on the Timeline level and on export.

Now let's move on to the timeline level. We were already able to admire how Flame wonderfully plays three different color spaces in a row, but we haven't talked about the limitations that come along with it: Blending between Slog3 and sRGB, for example, that won't work. If the timeline is not at least 16 bit, we will have to deal with clipping. Certain effects, which can also be applied to the timeline, are calculated differently or sometimes incorrectly depending on the color space - a defocus, for example, may only work with linear material. So we have our correctly tagged material coming from different Color Spaces - how do we get all this material into a common Color Space? The remedy is a timeline effect called „Color Management“, which is built exactly the same way as the node of the same name in Batch - in fact it's the same operation and the same source code as in Batch, just applied in a different part of Flame.

However, there are two effects with the same name, one in green and one in Flame-typical grey. The green one is applied before all resizes and other effects, so the conversion is done on a source level - especially before the image effect, which is used when grading in Flame.





Nowhere else can you experiment with IDTs and ODTs as well as in Batch.

However, the Color Management effect can also be applied as a Gap FX, i.e. as a kind of Adjustment Layer on the Timeline. Wait a minute, there was something in our manual Resolve workflow with an ODT that we applied to the timeline level? Exactly! Here, for example, we can take the View Transform to bring an entire Timeline of ACES clips into a Color Space like rec.709. Of course, this timeline must also be able to handle ACES, so it must be at least 16-bit float, even if our deliverable is rec.709 like in this example. But what's really exciting is that this effect is simply a layer on our timeline. A layer that we can put other layers on top of. With graphics and lower thirds and titles...in the correct target color space.

And the View Transform really lends itself to that, because it allows us three settings: The Tagged Color Space, in this case just our used ACES color space. Then the View Transform, where we can specify the exact Color Science behind the conversion. For example „ACES 1.1 SDR-Video“. And under Display, following the WYSIWYG principle I mentioned earlier, rec.709. On the timeline itself we won't see any visible difference, but if we switch between the layer our ACES clips are on and the one with our Color Management gap FX, then we'll see in the lower left corner of the display how the Color Space jumps. And if you want to be extra safe, click on the „Active“ button in the viewer and bypass the

ODT on the display. Then the difference should be clearly visible.

Finally, let's talk about the export: We won't go into every detail here, because as mentioned before, the basic structure of the Color Managements is the same everywhere in Flame. If you want to bring your timeline into the target color space only when exporting, select the somewhat old-fashioned „Use LUT“ under the Advanced Options.

From here on you can assign the same ODT to the individual export as we have just done with the timeline. But in principle, I would advise against this, because the export settings, unlike a timeline effect or a node in the batch setup, are no longer attached to the clip. Therefore, especially for the beginning, I would like to advise everyone that the transformations should be done where they can still be traced afterwards: In batch and on the timeline.

### Limitations

Fortunately, ACES is now widely supported. Boris FX has made many of its products compatible with OCIO in the last releases and so you can also use these plugins like Mocha Pro or Silhouette in an ACES workflow. But not every manufacturer is as far along and we have already noticed it a bit in this workshop - Adobe is unfortunately not really ACES Ready by design. With After Effects this is a bit of a nuisance, but it

can be solved with a few downloaded config files and a few changed settings - and with the next iteration of After Effects we can expect native ACES support.

The whole thing looks more difficult with the industry standard Photoshop. Photoshop is unfortunately the definition of a „display-referred“ software, the concept of Color Spaces, at least as we know them from the video world, is simply not known here. To make matters worse, not all of Photoshop's features work in 32-bit. Daniel Brylka has described a hack on his really very good blog [www.toodee.de](http://www.toodee.de) on how to use the proofs in Photoshop to work logarithmically on ACES looks, but it's far from intuitive - definitely some catching up to do here on Adobe's part.

If you are serious about ACES, you should definitely create a small test project and put your own pipeline through its paces before sending the first real job through there. If you don't want to download test footage from ten different camera manufacturers, this page could be really helpful:

<https://acescentral.com/knowledge-base-2/using-aces-reference-images/>

Here you can find test images in all kinds of color spaces, which should definitely point out any bugs in the pipeline. And it gives us the wonderful opportunity to experience what ACES promises - that our images really do look the same in any software.



Essential for Deliverables - Color Management as Gap FX

## Conclusion

Using ACES outside of a large pipeline is now absolutely doable - even if support varies across tools, good roundtripping is possible. ACES is here to stay and there are good reasons to use ACES. Future-proofing your own work, no longer bending Color Science but using it for better images, and using the maximum possible color space are all good reasons to take the step towards ACES. Finally, I would like to say thank you because this article would not have been possible without the great work of Finn Jaeger, Daniel Brylka, Chris Kasten, Grant Kay, Lars Wemmje and Randy McEntee who have already shared their knowledge with the community. I've put their respective articles or videos in the link box and would highly recommend them to anyone interested.



Christoph Zapletal ist seit mehreren Jahren als freiberuflicher Compositing Artist und VFX-Supervisor in Hamburg tätig. Sowohl mit Autodesk Flame als auch The Foundry's Nuke arbeitete er bereits an diversen Werbe- und Spielfilmprojekten mit. Darüber hinaus ist er als Dozent für fxphd und an der HFF tätig. Weitere Infos unter [www.christophzapletal.de](http://www.christophzapletal.de)