

[ACES - TF - Academy LUT Format] Input and Output Descriptor (ColorSpace Tag)

Walter Arrighetti 15 dic 2014, 02:25

a ACES

Just to expand on Jim's comment, my proposal is to have `InputDescriptor` and `OutputDescriptor` nodes to both encapsulate an `aces:ColorSpace` node like in the below examples. The first ("minimal" shows just a minimum set of info to include :

- `<abbr>` -- the abbreviated colorspace name (useful for showing in a futurable applications' UI info-window for the LUT)
- `<channels>` -- the name (and number) of input/output channels (+ a boolean for whether or not an *alpha* channel is included in the LUT)
- `<primaries>` and `<illuminant>` -- the primaries' and white-point's chromaticities

The second example shows a more extended set of tags for input/output colorspace definitions. All of the tags in it can be considered either *recommended* (i.e. applications are allowed to discard them when parsing the LUT and/or ignore while writing one) or for future CommonLUT extensions.

There are tags for specifying:

- **gamut boundary** definitions for one or more of the following types (especially useful for full-to-legal LUTs [& vice-versa], as well as for LUTs with *faux-couleurs* to highlight Out-of-Gamut codevalues):
 - minimum/maximum allowed codevalues
 - "legal"-range codvalues
 - (hard)-clipping codevalues' extremes
- apparent gamma (if present),
- additional input-referred (like ISO and CCT) or output-referred parameters (like target or viewing environment lighting parameters, expected adaption, etc.)

In my opinion, and in case the group feels going this way for the CommonLUT spec, we can go for mandatory `<abbr>` tag (so that colorspace are at least defined from a list of values like already specified in the CommonLUT doc and in our ACES documentation), `<primaries>/<illuminant>` (at least for color spaces whose primaries are known), plus optional/recommended other tags

Please also note that, for consistency's sake, `aces` namespace should be used for all the tags, not just for top-level ones.

Walter Arrighetti mer 25 apr, 17:25

a Joachim

Hi JZ.

Thanks for getting back to me: yes, the things you report are mostly correct as we said at the phone conversation (and I'll reference the posts I wrote about in ACESCentral, where possible). Besides, I had a second thought and I feel I could add a few things. Hope you don't mind if this is becoming large text.

As you can see, most of the items, despite being inherently technical, ultimately relate to ACES UX in order to improve its *effective usability*.

Where needed, I add additional-indent comments to further clarify my point of view.

1. ACES documentation should be improved a lot. That includes whitepapers and mid-to-low tech nice brochures, not just a thorough set of deeply technical documents.
 - o I believe we need to re-publish ACES as an *updated* SMPTE standard again. I also evaluate [ETSI](#) and/or ITU as possible additional standardization bodies. [ETSI](#) is almost unknown in US, but is a very important standardization body in Europe.
2. [...]
3. [...]
4. [...]
5. Despite DITs and post companies should be let free to develop their own workflow, as well as VFX houses to have their own pipelines, more gaps in the workflow need be filled, otherwise each company uses stuff in the middle which screws up interoperability. Under this view I believe we should look in particular at getting more specifications for **ACESclip** and **CommonLUT**. More specifically:
6. **ACESclip** needs to have a stronger logical reference to clips and, since it's most useful when linked to original camera-raw files than to ACES EXRs, the Academy should publish Guidelines for vendors to link each specific file formats' Unique ID fields (from the respective file headers) as candidates for **ClipID**'s `ClipName` and `Source_MediaID` XML elements. Since there's no regulation on this "last-mile" aspect of ACESclip, everyone may or may not use it and, even then, the unconstrained usage will lead to lack of interoperability -- which ultimately leads to lack of usage again. [[Cross-referencing footage with ACESclip thread](#)]. For example:
 - o **MXF** — `ClipName` = UUID; `Source_MediaID` = UMID (of the underlying video *essence*, not of the MXF container).
 - o **ARRIRAW** — `ClipName` = 8-chars 'Reel ID'; `Source_MediaID` = 'UUID' or 'SMPTE UMID' (both from Camera Device Information header).
 - o **R3DCODE** — `ClipName` = 8-chars 'Reel ID'; `Source_MediaID` = 'Reel ID Full'.
 - o **OpenEXR** — both `ClipName` and `Source_MediaID` reference custom, same-name metadata (of `string` type) in the EXR header; `Source_MediaID` is a UUID or UMID.
 - o **DPX** — `ClipName` = tape/clip name from DPX header; `Source_MediaID` = film reel's first-frame `KeyCode` (if present, and laid out as string, e.g. "KN728-0000-0000"), otherwise *unused*.
7. [...]
8. [...]

9. There should also be Guidelines for **linking** an **ACESclip** XML file to a specific content, be it either single-clip file (.R3D, .MOV, .MXF, ...) or **frame-per-file** sequences (.ARI, .EXR, .DPX, .TIFF, ...). [[Cross-referencing footage with ACESclip thread](#)]
10. **CommonLUT** format needs to be further expanded. First, it should be possible to incorporate it in ACESclip sidecar file (it makes sense as per things said above). Second, it should provide additional information on the ***color pedigree*** of a clip, i.e. the color-processing the clip underwent through its history (Input Transform, LMTs, intermediate rendering to output-space files, Output Transforms intended for display/reference) and, if ever, the output colorimetry. This helps Product Partner honoring CommonLUT (and ACESclip) metadata to properly process the clip for viewing. [[ACES Feature requests thread](#)]
11. [...]
12. [...]
13. [...]
14. [...]
15. [...]
16. [...]
17. [...]
18. [...]
19. [...]
20. ACES "file" data components *deserve* official **file-type icons**; this would improve usability. A few people (including myself) started proposing some graphic-design [drafts](#) for ACESclip in the past, but this always stopped. Better resume that [[ACESclip icon ima](#)]