

# 2023-05-15

May 15, 2023

Host: Carol Payne

Secretary: Carol Payne

Attendees:

- ☒ Rémi Achard (TSC) - DNEG
- ☒ Mark Boorer (TSC) - Industrial Light & Magic
- ☒ Mei Chu (TSC) - Sony Pictures Imageworks
- ☐ Sean Cooper (TSC ACES TAC Rep) - ARRI
- ☒ Michael Dolan (TSC) - Epic Games
- ☐ Patrick Hodoul (TSC) - Autodesk
- ☐ John Mertic - Academy Software Foundation / Linux Foundation
- ☒ Carol Payne (TSC Chair) - Netflix
- ☐ Mark Titchener (TSC) - Foundry
- ☐ Carl Rand (TSC) - Weta Digital
- ☒ Doug Walker (TSC Chief Architect) - Autodesk
- ☒ Kevin Wheatley (TSC) - Framestore
- ☒ Zach Lewis - Method Studios
- ☒ Thomas Mansencal - WetaFX

Apologies:

- Name

## OCIO TSC Meeting Notes

- Items for 2.2.2 and possibly 2.1.4
  - Doug and Cedric planning on a 2.2.2 release (and 2.1.4)
  - If there's important things, please DM Doug or post in the #dev channel
  - No release date set, but soon-ish
  - Mark B: did we ever put in a fix for the xml throwing errors for unknown entities?
    - Doug, no, but please put in a ticket and we'll take a look
- Config merge feature, continued discussion
  - Doug - made some adjustments to the deck, based on feedback
  - Added: should be possible to re-use spaces if they are **already** in the user's config.
  - Added: new top level attribute to the config format: "allow\_config\_merging"
  - Added: introduce a new "ociom" file format - to allow for on-the-fly merging sort of similar to the #include approach
  - Added: ability to remove items via ociomerge
  - Mark: not very comfortable with the proposed strategy at all - merging overall
    - These changes influence how people interact with the library
    - On board with use case #1
    - Use case #2 - we already have methods for applications to do this via roles or otherwise. If that's too complicated, would rather keep the black box of what the DCCs are doing if they need specific colorspace added.
    - Use case #3, also already have solutions for - i.e. LUT search paths
    - Overall unease with the workflows that this enables - not the direction OCIO should go
  - Doug - yes, with use case #3, would imagine some sort of templated workflow, the configs working group would need to discuss further.
  - Doug - with use case #2 - roles work to a point, but breaks down when there are many colorspaces to support in-app - such as support of a camera SDK etc
  - Kevin: for use case #2 - we're providing a common mechanism vs. recreating the wheel every time. Making it more transparent vs. black box. Already have too many roles with the current approach because naming is specific - many times the colorspaces are the same!
  - Thomas: this feature definitely has the possibility to be far-reaching for sure. We'll need to be really careful in best practices, guides, config updates, etc.
  - Mark: that's part of my objection - the point of OCIO is complete control of color management of applications. Concern is this merging feature is removing control from the authors and adding increase for risk. Would like to see a better example re: camera log spaces examples.
  - Doug: often in Nuke common space is acescg, but not necessarily the case for finishing tracks/timelines. These timelines might be working in many different colorspaces.
  - Carol: the root of what we're speaking about in a lot of ways is adding new features and new tools to the core functions OCIO was designed to handle. It's good to have concern, but OCIO is growing and the apps that want to use it are also growing, it does make sense that we may have to add some new features to help the project grow
  - Mark: concern is the end user - already see issues in production throwing configs around. Right now this gets caught, when it errors on colorspaces not being found. With merging - this could become endemic & uncontrollable. Still don't understand the need for actual config modification

- Doug - the concern is valid around changing who has authoring power and when. As an open standard, anyone can modify, and changes the chain of control. Should there be some way that an application should be checking for config correctness? Some sort of checksum or additional validation workflow?
- Doug - use of merged configs generated by apps in other DCCs is not necessarily the goal - it could be possible but wasn't original intention.
- Mark: don't see how this feature really solves anything for use case 2.
- Doug: we are committing to the implementation and documentation of this feature.
- Kevin: only bit of this that modifies the library is the ociom - could be a separate library
- Mark: objections aren't the technology, I just don't want people merging configs
- Remi: how would we enforce/regulate guidelines around apps adding spaces that are duplicate/bloating configs?
- Kevin: this helper actually helps that - it makes the tooling more transparent
- Mark: what about unpublished/proprietary math that is used in a DCC? We can't expect that everyone publishes everything openly.
- Kevin: technically - you could already do this... grab a processor and serialize the transform.
- Michael: in general, thinks this feature would be useful. Understand why this is controversial, but current workflows are a bit limiting and black box. App devs would likely be more comfortable merging vs. using two configs or adding colorspace to a users config.
- Remi: is there a way to encourage proper use of the APIs by DCCs? To encourage adding colorspace vs. other black box things?
- Remi: maybe a soft-wrapper that wouldn't actually merge, but provide what would be added to the application for use in the DCC? But this wouldn't enable these spaces to show up in menus, etc
- **Doug: propose that Cedric and Doug go ahead implementing this, as their apps need it regardless. And as we move along, we can work through issues as they come up. If it can get into 2.3, great, if not, this is ok. Treat it as a spike. Ok with this path?**
  - **Mark, Thomas, Carol ok with this approach.**
- Kevin: what happens in the case a config is set not to merge but an app needs it?
- Doug: want to enable use case 1, but not use case 2 with the merging? tough to control the flow
- Mark - exactly, and if the DCC can't add the colorspace they want, aren't we back to where we were?
- MaterialX color transforms as graphs:
  - For discussion: <https://github.com/AcademySoftwareFoundation/MaterialX/pull/1352>
  - Doug: for FYI, Autodesk made a PR that was not merged by MaterialX to support OCIO in MaterialX - <https://github.com/AcademySoftwareFoundation/MaterialX/pull/840>
  - This proposal PR would support basic color transforms expressible in graph form
  - Would require OCIO to output MaterialX graphs
  - Please take a look at this PR, and then we'll come up with our opinion as OCIO next TSC meeting.