

PNG iCCP/iCCN/iCCM investigation

Problem

In past PNG WG meetings, some questions came up:

- Do we explicitly need a separate iCCN chunk or can we just say the iCCP chunk is ICC v4?
- Can & should we add iccMAX (perhaps in an iCCM chunk)?

There was a question about if this deviated from the previously established plan or if these things simply had not been considered. To answer that question, I read through the [HDR in PNG requirements doc \(and history\)](#), the [ColorWeb-CG pull requests](#), [ColorWeb-CG issues](#), and [public-colorweb@ emails all the way back to 2017](#).

At the bottom of this doc is a section titled “Related Mentions (chronological order)” which includes every mention that seemed related. This will save others the time if they want to confirm these findings. There is also a summary of those related mentions.

Investigation Result

At no point was there mention for or against repurposing the iCCP chunk to be ICC v4. However, there is clear acknowledgement that the existing iCCP chunk effectively already is ICC v4 (May 6, 2021, <https://github.com/w3c/ColorWeb-CG/commit/c0d909ec716a997ff345a49e9773d620c0c6b747>). This indicates a separate iCCN chunk is not needed and iCCP can indeed be ICC v4.

There were also several mentions of iccMAX being desirable. However, there were also concerns about security and limited support. The security concern comes from a scripting language included in iccMAX. Although the scripting language is designed to be safe, it is not yet widely adopted and thus not trustworthy.

We do not want to risk contentious items in the new PNG spec. Including an iccMAX chunk (iCCM) would endanger the other changes we want to land. I think it is best for us to delay the iCCM chunk for the next PNG spec edition. This reduces the number of changes and buys time for iccMAX to gain both adoption and trustworthiness.

Summary of Related Mentions (chronological order)

- Early on, iccMAX was proposed as a solid solution.

- There was push back because Chrome and other CMMs did not yet handle it well (nor did they indicate plans to do so). However, the same thought suggested ICC v4 was also too difficult. And ICC v4 was added to Chrome.
- Security concerns in iccMAX's calculator functions. And ICC v4 didn't have wide adoption 15 years after its spec, so expect iccMAX adoption to take a while.
- iccMAX has a broad range of functionality even though a small subset is required for any given application domain. This adds to the difficulty of adoption.
- There was a question if ICC v4 would be sufficient.
- The security concerns of iccMAX were addressed (should be fine). And the adoption time of ICC v4 (plus relating it to iccMAX) was addressed as ICC v4 didn't introduce new features that people needed.
- The iccMAX security issues were rekindled. It may be safe but hasn't been tested to confirm such.
- There was mention that the cICP tag and iCCP tag serve different purposes. cICP specifies the nature of the pixels while iCCP specifies the rendering. There should be a defined behavior if both are provided.

Related Mentions (chronological order)

- May 18, 2017
 - <https://lists.w3.org/Archives/Public/public-colorweb/2017May/0001.html>
 "It seems clear that a vastly better way to encode BT.2100 still images in PNG would be to embed an ICCMax profile that correctly describes the EOTF and the primary chromaticities, and has a correct peak luminance value. I assume that the flaws noted above are due to limitations of ICC v.4?"
 - <https://lists.w3.org/Archives/Public/public-colorweb/2017May/0004.html>
 "It would of course be ideal if PNG in the long-run could directly signal non-power transfer functions, but this would require creating and/or modifying chunks, which seemed undesirable in the short-term since this might cause incompatibilities with the installed base of decoders. As it stands, a PNG file that conforms to the proposal should be displayable on all PNG decoders -- some more accurately than others evidently."

"Hope this provides additional background. Looking forward to exploring more promising avenues, if they exist."
- May 19 2017
 - <https://lists.w3.org/Archives/Public/public-colorweb/2017May/0005.html>
 "Have you created such an ICCMax profile?
 How do you encode PQ EOTF?"

1. I assume the profile is to be used for color conversions, and not only to signal a color space. The current profile works in tested contemporary CMMs and apps. Those same CMMs fail on ICCMax profiles. I am not aware of any plans to change that.
2. No way to map the content to SDR media as the 10,000 nits value is useless for this. The current ICC profile provides a fallback for SDR on existing systems.
3. Did you find any flaws related to ICC v4? It seems the mismatch between your expectations and the current implementation are not due to flaws in ICC v4, so we can simply update the ICC profile to address your needs.”

- <https://lists.w3.org/Archives/Public/public-colorweb/2017May/0006.html>
“While ICCMax does have proper ways to describe PQ, it seems to do so in a way that is not easy to understand and implement. Recent work to support ICC profiles in chrome has focused on ICC version 2, because ICC version 4.x is too difficult to implement efficiently, and ICCMax looks much more difficult. Since we don't know what the uptake on ICCMax will look like, having a special bit for HDR content might not be a bad idea. (However, the proposed magic string solution doesn't sound like a good idea.)

Another problem altogether is that PQ as a transform is defined in absolute lumens. Since non-HDR images are all in screen-relative brightness, it's impossible to make an PQ image which looks the same as a non-HDR image. To observe this problem in practice, hook up a recent Win10 machine with an HDR-capable graphics card to an HDR-capable tv and turn on HDR. Suddenly all legacy apps become dark and gray, as windows translates them to 80 nits. Most people quickly turn HDR off again.

A better solution is probably to use hybrid-log-gamma, which uses screen-relative brightness. I'm not sure if there is an ICC(Max) profile that describes HLG accurately though.”

- May 29, 2018
 - <https://lists.w3.org/Archives/Public/public-colorweb/2018May/0003.html>
“Storage of subtitles and other still images - currently still formats like PNG can only store a gamma value in the header files - the only way to store non-gamma encoded images is to embed an ICC Profile. Currently for PQ HDR, there's a draft proposal that looks for a given ICC Profile filename in the header which then over-rides both the header and profile. We would prefer to have the correct ICC profiles available.”
- June 6, 2018
 - <https://lists.w3.org/Archives/Public/public-colorweb/2018Jun/0002.html>
“None of this can be directly accomplished using a single V4 profile. V4 uses display relative with a D50 PCS (with chromatic adaptation applied). By playing around with defining the media white point one can achieve a level of luminance scaling using absolute intent with some possible interoperability issues to

contend with. The Hlg algorithm cannot be directly encoded in v4, and LUTs and inverse LUTs using interpolation need to be populated for a fixed scene luminance condition with different profiles created for different scene luminances.”

- June 10, 2018
 - <https://lists.w3.org/Archives/Public/public-colorweb/2018Jun/0006.html>
“BTW, the HLG scene profile is trivially implementable in V4. The reference display profile is a little bit more complex in V4, but doable.”
- June 25, 2018
 - <https://lists.w3.org/Archives/Public/public-colorweb/2018Jun/0029.html>
“Well, that’s a problem since many CMMs will choose to avoid the calculator feature of ICC.Max for security concerns. I would think that calculator would be a last resort for any new profile that is created.”
- October 1, 2018
 - <https://lists.w3.org/Archives/Public/public-colorweb/2018Oct/0001.html>
“1. What the security concerns are with using ICCmax calculator functions? Any implementation of ICC should be considered insecure and a hacker target. Implementations that accept scripts as user input such as the calculator are inherently more insecure than pure data tables.

2. How far are we from widespread deployment of ICCmax? Be prepared to wait. 15 years since the first spec came out, V4 still is not fully supported.”
- October 2, 2018
 - <https://lists.w3.org/Archives/Public/public-colorweb/2018Oct/0002.html>
“The main problem is that iccMAX has a very broad range of functionality and typically only a subset of this functionality is required for any given application domain.”
- October 15, 2018
 - <https://lists.w3.org/Archives/Public/public-colorweb/2018Oct/0005.html>
“It sounds like there is a possibility that we could use an ICC v4 profile? We would need ITU-R BT.2020 primaries, D65 White point and the transfer functions applied on luminance.”
 - <https://lists.w3.org/Archives/Public/public-colorweb/2018Oct/0006.html>
“And as you say, the question is whether a v4 profile can be constructed, or whether this really needs ICC Max.”
- October 21, 2018
 - <https://lists.w3.org/Archives/Public/public-colorweb/2018Oct/0010.html>
“Lars is correct in asserting that any implementation of ICC is a hacker target. And ultimately, implementers are responsible to ensure that they have done their very best to ensure the security of their products. However, the ICC would have been negligent if it had not taken security into consideration in the design of the calculator element. And great consideration was put into the development of the specification for the calculator element in iccMAX (please note proper

capitalization). Two significant features/limitations of the iccMAX calculator scripting language were put into place to help establish security from a specification point of view. The first is static addressing of memory - The iccMAX calculator script is formed by a sequence of operator codes that each have static parameters associated with them. The second is that the script can be represented as a tree structure (no loops) and therefore every possible flow of operations along with all possible associated memory accesses related to the script (defined using static parameters) can be determined using a recursive analysis of the script BEFORE THE SCRIPT TRANSFORM IS APPLIED. The iccMAX specification specifies that this SHALL be done to ensure that all operators are valid and only valid memory use is performed by any branch of the script execution path. Any profile with either invalid operators or invalid memory uses is by definition an invalid profile and should be flagged by the CMM and not used. Additionally, the ICC is providing profiles that test each and every operator to purposely access invalid memory for implementers to test against to ensure that they are correctly checking for these conditions. Every effort has been made to try to achieve as close as possible the same level of security in an iccMAX calculator script as a data table.”

“I beg to differ with Lars on this for two reasons. The first is that V4 did not provide much functional difference over V2 (everything that could be done in V4 could be done in V2). The second is that there was no reference implementation. The whole point of iccMAX is that it provides for functionality that cannot be completely achieved using V4/V2 profiles. Once iccMAX profiles are being used to do things that cannot be done with V4/V2 profiles there will be greater pressure for support of iccMAX to be implemented. The iccMAX specification is currently up for FDIS ballot concluding early in November. If it passes this ballot then it will be an ISO standard. Interoperability Conformance Specification (ICS) documents are presently in the works (awaiting having an ISO standard for iccMAX) to define subsets of iccMAX for specific workflows.”

- <https://lists.w3.org/Archives/Public/public-colorweb/2018Oct/0011.html>
“[Sorry to have this discussion “in public”, but since it has come up] Max – as an implementor of one of two most widely attacked (and therefore security conscious) applications in use today (aka Adobe Acrobat/Reader), I can say with quite a bit of authority that while I appreciate your (and the ICC’s) efforts to create a “secure” calculator feature, it is not! None of the members of the ICC are security experts (nor would I expect them to be) and the material was not reviewed by any such experts. And because you chose to create a completely new “language” which requires completely new parsers – it makes it even more open for possible security holes.”
- September 16, 2019
 - <https://lists.w3.org/Archives/Public/public-colorweb/2019Sep/0004.html>

“PNG does support ICC profiles but currently references a >20-year-old v2 specification. A file format for images on the web should ideally support both v2 and v4, and possibly iccMAX for more advanced applications.”

- January 11, 2021
 - <https://github.com/w3c/ColorWeb-CG/issues/9#issuecomment-758037964>
“...but ICC Max is currently not well deployed and it is not clear that ICC v.4 will do the job (it may, but that needs to be explored).”
- February 24, 2021
 - <https://docs.google.com/document/d/1s9HTTo8CiPk-hkgGcBfDEbtlijW0xXEeB16xx982fsY/edit>
“If both iCCP and cICP chunks can be present, then decoder behavior should be specified. Perhaps specifying decoder behavior can be deferred until after requirements.
Alternatively, only one of iCCP and cICP chunks can be present
ICC profiles and H.273 code points serve different purposes: ICC specifies rendering whereas H.273 code points specify the nature of the pixels.”
- March 16, 2021
 - <https://docs.google.com/document/d/1amMOdpPJ6Q4NyKv6qozO8Ai4x-aGLS8C4S0ZCw9RX4o/edit#heading=h.ftmmlg8rrae7>
“Decoder behavior should be specified when both iCCP and cICP are present
 - Well-defined behavior is beneficial to the web
 - cICP should not always take precedence over iCCP since they apply to different use cases and cICP may contain information not in iCCP”
- March 31, 2021
 - <https://docs.google.com/document/d/1-YhccaT2MxSZahRYv4rDB79GVMYghrdtU8S-Y6OyGi4/edit>
 - “Proposal #1: For the purpose of rendering, the PNG decoder shall use the iCCN chunk if present.
How do we define “rendering”?
Image + ICC profile for a specific target cannot be repurposed to a different target In a browser, the browser does not know the ultimate display when decoding the image.
 - Proposal #2: the iCCN chunk shall take precedence over the cICP chunk (that does not mean that the cICP does not contain invalid information).
iCCP is used if cICP is not understood.
This does not work if the cICP chunk is also present and matches the target display but the iCCN chunk does not match the target display

Use cases:

 - HDR image to be rendered to SDR display (benefit from ICC)
 - HDR image rendered to HDR display (ICC profile is an issue)
 - Editing HDR image, which is out-of-scope

- Proposal #3: The iCCN chunk shall take precedence over the cICP chunk if the display/graphics plane does not support the codepoint specified in cICP. Provide examples.
 - Consensus to integrate Proposal #3”
 - <https://github.com/w3c/ColorWeb-CG/pull/19>
“@svgeesus “...unless the intent is specifically to exclude iccMAX, the following should be added:
[ICC-2019] Specification ICC.2:2019 (iccMAX) Image technology colour management - Extensions to architecture, profile format, and data structure”
 - @Irosenthol “The intent is to exclude iccMAX since there are still outstanding security concerns that I would like to avoid.”
 - <https://github.com/w3c/ColorWeb-CG/pull/19#issuecomment-811487208>
“iCCN chunk should take precedence over cICP chunk.”
- April 6, 2021
 - <https://github.com/w3c/ColorWeb-CG/blob/cce74b7a52145fd58da3c520fc0a74c149977ee9/hdr-in-png-requirements.md>
First HDR in PNG requirements doc created.
cICP and iCCP coexist, behavior undefined.
iCCN chunk proposed as well.
cICP should be used over iCCN if able. iCCN should be used over iCCP if able.
iCCP and iCCN shouldn't coexist.
- May 6, 2021
 - <https://github.com/w3c/ColorWeb-CG/commit/c0d909ec716a997ff345a49e9773d620c0c6b747>
Acknowledged that iCCP is v2 (even though it is treated as v4 by many) and iCCN is explicitly v4.
 - <https://github.com/w3c/ColorWeb-CG/commit/d683518e4e97e06683b7f4bae5fc48ef70019aed>
cICP and iCCP chunks allowed to coexist, no longer undefined behavior.
 - <https://github.com/w3c/ColorWeb-CG/commit/a28f958a8f9b262ec6a708fcff2a6322f0c17754>
Clarify that cICP and sRGB chunks act as labels which specify a color space. And iCCN & iCCP contain embedded profiles. Is this similar to the previously mentioned “cICP specifies the nature of the pixels and iCCN specifies the rendering”?