

ICC Votable Proposal Submission

Title of proposal

Amendment to the `cicpType`: interpretation of `ColourPrimaries` value 2 (Unspecified) via an embedded array of custom chromaticity coordinates.

Proposers

Luke Wallis (Apple)

Approved

19-10-2026

1. Introduction

A summary

This proposal amends the definition of the `cicpType` (ICC.1:2022, 10.3) so that, when the `ColourPrimaries` field of a `cicpTag` is equal to 2 (Unspecified), the tag may optionally be extended with an array of eight `s15Fixed16Number` values — $[x_R, y_R, x_G, y_G, x_B, y_B, x_w, y_w]$ — giving the CIE 1931 chromaticity coordinates of the red, green and blue source colour primaries and of the associated white point directly.

This turns “Unspecified” into a precisely-defined, self-contained escape hatch: a profile author can signal custom primaries that have no Recommendation ITU-T H.273 `ColourPrimaries` code point, without depending on any other profile tag or on the profile’s class. The mechanism works identically whether the `cicpTag` appears in an Input, Display, or ColorSpace profile (see 9.2.17, as amended by the HDR Profiles Amendment Proposal, 4.4).

The problem to be solved

The `cicpTag` added to ICC.1:2022 (9.2.17) provides the long-needed bridge between ICC profiles and video pipelines compliant with Recommendation ITU-T H.273. However, the `ColourPrimaries` field of the `cicpType` is an 8-bit enumerated value whose non-reserved, non-unspecified entries (Recommendation ITU-T H.273, Table 2) enumerate only a finite, fixed set of chromaticities:

- 1 — Rec. ITU-R BT.709 / sRGB / sYCC
- 4 — Rec. ITU-R BT.470-6 System M
- 5 — Rec. ITU-R BT.470-6 System B, G / Rec. ITU-R BT.601-7 625
- 6 — Rec. ITU-R BT.601-7 525 / SMPTE ST 170
- 7 — SMPTE ST 240
- 8 — Generic film (Wratten filters)
- 9 — Rec. ITU-R BT.2020 / Rec. ITU-R BT.2100
- 10 — SMPTE ST 428-1 (CIE XYZ)
- 11 — SMPTE RP 431-2 (DCI P3)
- 12 — SMPTE EG 432-1 (Display P3)
- 22 — a specific non-industry set

Many ICC profiles have source colour primaries that correspond to none of these:

- camera characterisation profiles whose primaries are derived from sensor spectral sensitivities,
- scanner profiles,
- custom-calibrated display profiles whose primaries deviate from the nominal gamut due to ageing, panel-to-panel variation or calibration to a non-standard white,
- wide-gamut working spaces such as Adobe RGB (1998), ROMM RGB, eciRGB, ACESeg, etc., that have no `ColourPrimaries` code point assigned in Recommendation ITU-T H.273.

In the current ICC.1:2022 text, an author of such a profile who wishes to include a `cicpTag` (for example to convey a matching

`TransferCharacteristics` and `VideoFullRangeFlag`, or to be recognised by a video-style processor as per the original CIEP Votable Proposal) has no correct value to put in the `ColourPrimaries` field:

- Choosing the numerically nearest enumerated value silently misrepresents the gamut and produces colour errors in any receiver that takes the `cicpTag` at face value.
- Choosing 0 (Reserved) is non-conforming.
- Choosing 2 (Unspecified) is, under the current ICC.1:2022 text, the only defensible choice, but it leaves the receiver without any way to recover the primaries, which makes the rest of the `cicpTag` effectively useless for gamut mapping.

An earlier version of this proposal addressed this gap by having the receiver recover the primaries from the profile's own `redMatrixColumnTag`, `greenMatrixColumnTag` and `blueMatrixColumnTag`. That approach has two drawbacks. First, it ties the interpretation of `ColourPrimaries` = 2 to a specific profile shape — a three-component matrix-based profile (ICC.1:2022, 8.3.3 / 8.4.3) — so a profile class that does not carry matrix column tags, such as a `ColorSpace` profile (8.7), now also permitted to carry `cicpTag` (see 9.2.17, as amended), has no equivalent tag to recover primaries from. Second, recovering chromaticity coordinates from CIE XYZ tristimulus values requires undoing the profile's white-point adaptation (`mediaWhitePointTag`, `chromaticAdaptationTag`, Annex F.3) — machinery a receiver would not otherwise need just to read a `cicpTag`.

The solution

This proposal's mechanism is not a novel invention: the Headroom Adaptive Gain Curve (HAGC) tag already defines an almost identical embedded custom-primaries mechanism (Gain Curve Chromaticities Mode = 3), itself based on SMPTE ST 2094-50, which encodes a set of custom chromaticities directly within the tag rather than deriving them from another profile tag. This proposal adopts the same pattern for `cicpType`: both mechanisms exist precisely to solve the same problem (a profile-independent way to signal primaries with no H.273 code point), so reusing an already-established, already-implemented pattern from an adjacent, actively-used standard is lower-risk than inventing a new one.

The proposal amends clause 10.3 (`cicpType`) of ICC.1:2022 so that, when `ColourPrimaries` is 2, the tag may be followed immediately by eight additional `s15Fixed16Number` (ICC.1:2022, 4.6) values encoding the CIE 1931 (x, y) chromaticity coordinates of the red, green and blue source colour primaries and of the associated white point, in the order $x_R, y_R, x_G, y_G, x_B, y_B, x_w, y_w$ — the same field order used by the Headroom Adaptive Gain Curve (HAGC) tag's custom-primaries mechanism (Gain Curve Chromaticities Mode = 3) and by SMPTE ST 2094-50, so that a value computed for one may be carried into the other without re-deriving it from first principles.

`s15Fixed16Number` is chosen over HAGC's `uInt16Number`-plus-scale-factor encoding for two reasons: it is signed, so it can represent the negative chromaticity coordinates that occur in some very-wide-gamut primary sets (for example ACES AP0), which HAGC's unsigned encoding cannot; and its native resolution ($1/65536 \approx 1.5 \times 10^{-5}$) is finer than HAGC's ($1/50000 = 2 \times 10^{-5}$), so a value already quantised to HAGC's precision round-trips through `s15Fixed16Number` and back without loss.

Because `ColourPrimaries` = 2 is already a legal value under ICC.1:2022, and every ICC tag's length is declared independently in the profile's tag table (ICC.1:2022, 7.3), a receiver that does not implement this amendment reads the `cicpType` tag exactly as it does today — it already knows the tag's total length from the tag table entry and is not required to, and need not, consume any bytes past the existing 12-byte structure — and continues to treat `ColourPrimaries` = 2 as "unknown". A receiver that implements this amendment gains access to the exact chromaticities when the extension is present.

2. Summary of proposed changes to ICC specification

The proposal amends ICC.1:2022 clause 10.3 (`cicpType`) to add:

- an optional 32-byte extension to the `cicpType` tag, present only when `ColourPrimaries` is 2, encoding eight `s15Fixed16Number` values ($x_R, y_R, x_G, y_G, x_B, y_B, x_w, y_w$);
- a normative statement of what these values mean when present;
- an informative note relating the field order and choice of encoding to the HAGC tag's custom-primaries mechanism and to SMPTE ST 2094-50.

The proposal also amends ICC.1:2022 clause 9.2.17 (`cicpTag`) to add a short cross-reference NOTE pointing to the new interpretation in 10.3.

No existing `cicpTag` / `cicpType` encoding is invalidated. Profiles with `ColourPrimaries` set to any H.273-defined chromaticity value, and existing profiles with `ColourPrimaries` = 2 and no extension present, continue to behave exactly as before. The additional interpretation applies only when `ColourPrimaries` = 2 **and** the extension is present.

There are no required changes for CMMs that do not act on the `cicpTag`. A CMM that already implements `cicpTag`-driven behaviour and encounters `ColourPrimaries` = 2 requires the small additional code path described in clause 4 below to read the appended array when present.

3. Nature of the proposal

This proposal amends the normative text of an existing Version 4 tag type (`cicpType`) and adds a cross-reference NOTE to an existing tag (`cicpTag`) in ICC.1. It does not define any new tag, tag type, or enumerated value, and it does not impact ICC registries.

It extends, but does not alter, the existing 12-byte `cicpType` structure: the four existing fields (`ColourPrimaries` , `TransferCharacteristics` , `MatrixCoefficients` , `VideoFullRangeFlag`) keep their byte positions and encodings unchanged. The extension is present only when `ColourPrimaries` = 2, and its presence or absence does not change the meaning of any other field. Because every ICC tag's length is declared in the profile's tag table independently of the tag type's own internal structure (ICC.1:2022, 7.3, Table 24), a pre-amendment parser reading a post-amendment `cicpType` with the extension present reads the first 12 bytes exactly as it does today and is not required to, and need not, read past them; it falls back to its existing "Unspecified" handling (typically by using the profile's matrix/TRC transform, or its `AToB0Tag` / `BToA0Tag` pair, in the conventional way). Post-amendment parsers gain an additional, well-defined option.

4. Votable Proposal

4.1 Scope

The amendment applies to `cicpTag` / `cicpType` data in an ICC profile of any profile class permitted to carry the `cicpTag` — currently Input and Display (ICC.1:2022, 9.2.17), and ColorSpace (as amended by the HDR Profiles Amendment Proposal, 4.4) — whose data colour space in the profile header is RGB or YCbCr.

NOTE `ColourPrimaries` is not meaningful for the XYZ data colour space, since an XYZ-encoded profile has no RGB primaries to describe; this amendment accordingly does not apply when the data colour space is XYZ.

Unlike the profile-class-specific mechanism it supersedes, this amendment does not depend on, and does not require, any other tag — in particular it does not require `redMatrixColumnTag` , `greenMatrixColumnTag` , `blueMatrixColumnTag` , `mediawhitePointTag` , or `chromaticAdaptationTag` — to be present. It therefore applies uniformly regardless of whether the profile class in question happens to define those tags.

4.2 Proposed normative text (addition to ICC.1:2022, 9.2.17 `cicpTag`)

Append the following NOTE to clause 9.2.17, after the paragraph beginning "This tag may be present when the data colour space in the profile header is RGB, YCbCr, or XYZ...":

NOTE When `ColourPrimaries` is equal to 2 (Unspecified), the `cicpTag` may optionally convey the source colour primaries and white point directly, via an extension to the `cicpType` structure; see 10.3.

4.3 Proposed normative text (amendment to ICC.1:2022, 10.3 `cicpType`)

Replace Table 32 (`cicpType` encoding) with:

Table 32 — *cicp* Type encoding

Byte position	Field length (bytes)	Content	Encoded as
0 to 3	4	'cicp' (63696370h) type signature	
4 to 7	4	Reserved, shall be set to 0	
8	1	ColourPrimaries	uInt8Number
9	1	TransferCharacteristics	uInt8Number
10	1	MatrixCoefficients	uInt8Number
11	1	VideoFullRangeFlag	uInt8Number
12 to 15	4	x_R (present only when ColourPrimaries = 2)	s15Fixed16Number
16 to 19	4	y_R (present only when ColourPrimaries = 2)	s15Fixed16Number
20 to 23	4	x_G (present only when ColourPrimaries = 2)	s15Fixed16Number
24 to 27	4	y_G (present only when ColourPrimaries = 2)	s15Fixed16Number
28 to 31	4	x_B (present only when ColourPrimaries = 2)	s15Fixed16Number
32 to 35	4	y_B (present only when ColourPrimaries = 2)	s15Fixed16Number
36 to 39	4	x_w (present only when ColourPrimaries = 2)	s15Fixed16Number
40 to 43	4	y_w (present only when ColourPrimaries = 2)	s15Fixed16Number

Append the following paragraphs to clause 10.3 immediately after the existing paragraph beginning “The fields ColourPrimaries, TransferCharacteristics, MatrixCoefficients, and VideoFullRangeFlag shall be...”:

When `ColourPrimaries` is equal to 2 and the tag's declared length in the profile's tag table (see 7.3) is greater than 12 bytes, bytes 12 to 43 shall be present and shall encode, as `s15Fixed16Number` values, the CIE 1931 (x, y) chromaticity coordinates of the red, green and blue source colour primaries and of the associated white point, in the order $x_R, y_R, x_G, y_G, x_B, y_B, x_W, y_W$, as given in Table 32.

NOTE 1 When `ColourPrimaries` is not equal to 2, the `cicpType` tag is exactly the 12-byte structure above; bytes 12 onward are not present, and the tag's declared length is 12 bytes.

NOTE 2 When `ColourPrimaries` is equal to 2 and the tag's declared length is 12 bytes (i.e. the extension is absent), `ColourPrimaries` retains its original Recommendation ITU-T H.273 meaning: "Image characteristics are unknown or are determined by the application". This preserves the behaviour of every existing profile that uses `ColourPrimaries` = 2 today.

NOTE 3 The field order and the choice of an explicit white point, alongside the three primaries, matches the custom-primaries mechanism of the Headroom Adaptive Gain Curve tag (Gain Curve Chromaticities Mode = 3) and of SMPTE ST 2094-50, so that a chromaticity value derived for one may be carried into the other without re-derivation. Unlike HAGC's `uInt16Number`-based encoding, `s15Fixed16Number` is signed and of finer native resolution; a value expressed in HAGC's encoding round-trips through `s15Fixed16Number` and back without loss.

NOTE 4 These chromaticity coordinates have the same semantic role as the values listed in Table 2 of Recommendation ITU-T H.273 — i.e. they are the chromaticity coordinates of the source colour primaries and white point — and may be used in any downstream processing that would otherwise consume the chromaticity pair listed for a defined `ColourPrimaries` enumerated value.

NOTE 5 This mechanism does not depend on, and does not require, any other tag. In particular, it does not require `redMatrixColumnTag`, `greenMatrixColumnTag`, `blueMatrixColumnTag`, `mediaWhitePointTag`, or `chromaticAdaptationTag` to be present, and no chromatic-adaptation computation is needed to interpret the encoded values: they are CIE 1931 (x, y) chromaticity coordinates directly, not CIEXYZ tristimulus values requiring white-point recovery.

5. Applications and Workflows

The original CICP Votable Proposal targeted frame-grabs and ICC-tagged stills that can be re-inserted into a video timeline. This amendment broadens the scope of that same workflow to a large family of ICC profiles that today fall outside the reach of the `cicpTag` :

- **Camera DCP-style profiles.** A digital-cinema or photography workflow can emit an ICC profile whose primaries correspond to the camera's actual colour response, and use the `cicpTag` to simultaneously signal an H.273 transfer function (for example PQ or HLG) and matrix encoding, without being forced to misrepresent the primaries as BT.709 or BT.2020.
- **Custom-calibrated reference monitors.** Display profiles produced by instrumentation typically differ from the nominal BT.2020 / DCI-P3 primaries by measurable amounts. With this amendment, the `cicpTag` can carry the correct transfer and matrix meta-data together with the primaries themselves, encoded directly in the tag's own extension — no separate profile tag is needed to recover them.
- **Wide-gamut working spaces with no assigned CICP code point.** Profiles for working spaces such as Adobe RGB (1998), ROMM/ProPhoto, ACEScg, ITU-R BT.2020 scaled variants, etc., can carry a meaningful `cicpTag` today only by lying about the primaries; with this amendment, they can correctly declare `ColourPrimaries` = 2 together with the custom-primaries extension and still be consumed by a video-style processor.
- **Round-tripping through graphics tools.** An image edited in any working space whose profile is permitted to carry a `cicpTag` — matrix- or LUT-based alike — can be re-introduced into a video pipeline at the correct transfer function and matrix without the primaries being silently coerced to a nearby enumerated gamut.
- **ColorSpace-class profiles with custom primaries.** Because this mechanism does not depend on matrix column tags, it applies equally to ColorSpace-class profiles (8.7), which have none. The HDR Profiles Amendment Proposal relies on exactly this: an

HDR Profile (8.10.1) may declare `ColourPrimaries = 2` provided the `cicpType` extension defined here is present, removing the profile-class dependency that its earlier, matrix-tag-based mechanism had.

Issues

- A receiver that implements the `cicpTag` but not this amendment will see `ColourPrimaries` equal to 2, treat the primaries as unknown, and either refuse the tag or fall back to a default (typically BT.709). This is the same failure mode such receivers already exhibit today for any value of 2; this proposal does not worsen it.
- The custom primaries and white point depend entirely on the accuracy with which the profile author populates the extension array. Authors who want tight agreement with the CIEP interpretation should encode these values with their full available precision.
- Unlike the mechanism this proposal supersedes — which derived its recovered primaries directly from the profile’s own `redMatrixColumnTag` / `greenMatrixColumnTag` / `blueMatrixColumnTag`, and so could never disagree with them — this extension is independently authored. A three-component matrix-based Input or Display profile (8.3.3/8.4.3) always carries its own matrix column tags regardless of whether it also carries a `cicpTag`, so such a profile could in principle declare `ColourPrimaries = 2` with this extension present while its matrix column tags encode different primaries. This proposal defines no rule for which a CMM should trust in that case; profile authors should ensure the two agree wherever a profile carries both, and absent that, which value a CMM uses is at its discretion. A ColorSpace-class profile (8.7) cannot exhibit this particular disagreement, since it has no matrix column tags to disagree with in the first place (see 4.1).
- If `ColourPrimaries` is equal to 2 and the tag’s declared length is 12 bytes (i.e. the extension is absent), the amendment does not apply and `ColourPrimaries` retains its existing “unknown” meaning. This preserves behaviour for every profile — of any class or data colour space — that already uses `ColourPrimaries = 2` today without this extension.
- Extending an existing tag type’s structure with additional trailing data, gated by an already-legal enumerated value, is backward compatible because ICC.1:2022 already requires every tag’s length to be declared independently in the tag table (7.3); a parser has never been required to assume a fixed length for any tag type from its signature alone. This amendment relies on that existing guarantee rather than introducing a new one.
- This amendment supersedes the profile-class-specific recovery mechanism (via `redMatrixColumnTag` / `greenMatrixColumnTag` / `blueMatrixColumnTag`) proposed in an earlier version of this document. That mechanism is withdrawn in favour of the self-contained extension defined here.

6. Normative references

- ICC.1:2022, *Image technology colour management — Architecture, profile format and data structure*. In particular clauses 4.6 (`s15Fixed16Number`), 7.3 (Tag table), 8.7 (ColorSpace profile), 9.2.17 (`cicpTag`), 10.3 (`cicpType`).
- ICC HDR Profiles Amendment Proposal, *Addition of clause 8.10 High Dynamic Range (HDR) Profiles to ICC.1...* Amends 9.2.17 to permit the ColorSpace profile class; that amendment is what brings ColorSpace-class profiles within scope of this proposal (see 4.1).
- Headroom Adaptive Gain Curve Tag (HAGC) Votable Proposal (v10), *headroomAdaptiveGainCurveTag* / *headroomAdaptiveGainCurveType*. Source of the field order and `uInt16Number`-based custom-primaries encoding (Gain Curve Chromaticities Mode = 3) that this proposal’s field order and value semantics are aligned with, using `s15Fixed16Number` instead (see 4.3, NOTE 3).
- SMPTE ST 2094-50, *Dynamic metadata for colour volume transform — Application #5*. Source of the underlying custom-chromaticities convention referenced above.
- Recommendation ITU-T H.273 (07/2021), *Coding-independent code points for video signal type identification*.

7. Bibliography

- ICC CIEP Votable Proposal (2021-01-13), *Coding-independent code points (CIEP) for video signal type identification*.

Annex 1 (informative): Deriving a D50-adapted primary matrix from the custom-primaries extension

Some receivers may want to convert the chromaticities carried in the `cicpType` custom-primaries extension (see 4.3) into the same D50-adapted, PCS-relative matrix form conventionally carried in `redMatrixColumnTag`, `greenMatrixColumnTag` and `blueMatrixColumnTag` (ICC.1:2022, 9.2.46, 9.2.31, 9.2.4) — for example, to build a fallback matrix/TRC profile from the same primaries, to compare against a companion matrix-based profile, or to feed a CMM implementation whose internal colour engine is built around the classic matrix/TRC model. This conversion is entirely optional and is not required for conformance with this proposal.

Because this proposal’s custom-primaries encoding shares its field order and chromaticity semantics with the Headroom Adaptive Gain Curve (HAGC) tag’s own custom-primaries mechanism (Gain Curve Chromaticities Mode = 3), itself based on SMPTE ST 2094-50, the same three-step procedure below applies unchanged to a set of custom primaries authored via HAGC — an implementer does not need a second derivation for that case.

A1.1 Overview

Given the eight extension values $x_R, y_R, x_G, y_G, x_B, y_B, x_w, y_w$ (see 4.3), the corresponding D50-adapted primary matrix M_D50 is obtained in three steps:

1. Convert the three primaries and the white point to CIEXYZ tristimulus values, and solve for the 3×3 matrix M_native that maps linear RGB (relative to the given white point) to CIEXYZ (also relative to that white point).
2. Compute the linear Bradford chromatic-adaptation matrix M_adapt (ICC.1:2022, Annex E) from the given white point to the PCS adopted white D50 (ICC.1:2022, 7.2.16).
3. $M_D50 = M_adapt \cdot M_native$. The three columns of M_D50 are the values a conforming `redMatrixColumnTag`, `greenMatrixColumnTag` and `blueMatrixColumnTag` triple would carry for the same primaries.

A1.2 Step 1 — chromaticities to a native-white matrix

For each primary $i \in \{R, G, B\}$, and for the white point w :

$$\begin{aligned} X_i &= x_i / y_i \\ Y_i &= 1 \\ Z_i &= (1 - x_i - y_i) / y_i \end{aligned}$$

Form the matrix of unit-luminance primaries and the white-point vector:

$$P = \begin{bmatrix} X_R & X_G & X_B \\ Y_R & Y_G & Y_B \\ Z_R & Z_G & Z_B \end{bmatrix} \quad W = \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$$

Solve $P \cdot S = W$ for the 3-vector $S = [S_R, S_G, S_B]$, then scale each column of P by the corresponding element of S :

$$\begin{aligned} S &= P^{-1} \cdot W \\ M_native &= P \cdot \text{diag}(S) \end{aligned}$$

M_native maps linear RGB (relative to the white point x_w, y_w) to CIEXYZ relative to that same white point; it is not yet adapted to D50.

A1.3 Step 2 — Bradford adaptation to D50

ICC.1:2022, Annex E.3, Equation (E.1) gives the linear Bradford cone-response matrix:

$$M_BFD = \begin{bmatrix} 0.8951000 & 0.2664000 & -0.1614000 \\ -0.7502000 & 1.7135000 & 0.0367000 \\ 0.0389000 & -0.0685000 & 1.0296000 \end{bmatrix}$$

The PCS adopted white (ICC.1:2022, 7.2.16) is D50, with nCIEXYZ $(X, Y, Z) = (0.9642, 1.0000, 0.8249)$. Following ICC.1:2022, Annex E.2, Equation (E.2):

$$\begin{aligned} \text{rho_src} &= M_BFD \cdot [X_w, Y_w, Z_w] && \text{(white point from step 1)} \\ \text{rho_D50} &= M_BFD \cdot [0.9642, 1.0000, 0.8249] \end{aligned}$$

$$M_adapt = M_BFD^{-1} \cdot \text{diag}(\text{rho_D50} / \text{rho_src}) \cdot M_BFD$$

A1.4 Step 3 — combine

$$M_{D50} = M_{\text{adapt}} \cdot M_{\text{native}}$$

Each column of M_{D50} , encoded as three `s15Fixed16Number` values, is the value a conforming `redMatrixColumnTag`, `greenMatrixColumnTag` or `blueMatrixColumnTag` would carry (ICC.1:2022, 9.2.46, 9.2.31, 9.2.4) for a matrix/TRC-based profile built from the same primaries.

A1.5 Illustrative implementation (informative)

The following self-contained Python implements A1.2 to A1.4 directly. It is illustrative only and not itself normative; any correct implementation of the same linear algebra is conforming.

PYTHON

```

def xy_to_XYZ(x, y, Y=1.0):
    return (x * Y / y, Y, (1.0 - x - y) * Y / y)

def mat3_mul(A, B):
    return [[sum(A[i][k] * B[k][j] for k in range(3)) for j in range(3)]
            for i in range(3)]

def mat3_vec(A, v):
    return [sum(A[i][k] * v[k] for k in range(3)) for i in range(3)]

def mat3_inv(A):
    a, b, c = A[0]
    d, e, f = A[1]
    g, h, i = A[2]
    det = a*(e*i - f*h) - b*(d*i - f*g) + c*(d*h - e*g)
    cof = [
        [ (e*i - f*h), -(d*i - f*g), (d*h - e*g)],
        [-(b*i - c*h), (a*i - c*g), -(a*h - b*g)],
        [ (b*f - c*e), -(a*f - c*d), (a*e - b*d)],
    ]
    return [[cof[j][i] / det for j in range(3)] for i in range(3)]

def mat3_solve(A, b):
    return mat3_vec(mat3_inv(A), b)

def primaries_to_native_matrix(xr, yr, xg, yg, xb, yb, xw, yw):
    Xr, Yr, Zr = xy_to_XYZ(xr, yr)
    Xg, Yg, Zg = xy_to_XYZ(xg, yg)
    Xb, Yb, Zb = xy_to_XYZ(xb, yb)
    P = [[Xr, Xg, Xb], [Yr, Yg, Yb], [Zr, Zg, Zb]]
    W = list(xy_to_XYZ(xw, yw, 1.0))
    S = mat3_solve(P, W)
    return [[P[i][j] * S[j] for j in range(3)] for i in range(3)]

M_BFD = [
    [ 0.8951000, 0.2664000, -0.1614000],
    [-0.7502000, 1.7135000, 0.0367000],
    [ 0.0389000, -0.0685000, 1.0296000],
]
D50_XYZ = (0.9642, 1.0000, 0.8249) # ICC.1:2022, 7.2.16

def bradford_adapt_to_D50(xw, yw):
    Ws = list(xy_to_XYZ(xw, yw, 1.0))
    rho_src = mat3_vec(M_BFD, Ws)
    rho_dst = mat3_vec(M_BFD, list(D50_XYZ))
    D = [[rho_dst[i] / rho_src[i] if i == j else 0.0 for j in range(3)]
         for i in range(3)]
    return mat3_mul(mat3_inv(M_BFD), mat3_mul(D, M_BFD))

def custom_primaries_to_matrixColumnTags(xr, yr, xg, yg, xb, yb, xw, yw):
    # Returns a 3x3 matrix whose columns are the red/green/blue
    # matrixColumnTag values, in that order, for the given cicpType
    # extension chromaticities.
    M_native = primaries_to_native_matrix(xr, yr, xg, yg, xb, yb, xw, yw)
    M_adapt = bradford_adapt_to_D50(xw, yw)
    return mat3_mul(M_adapt, M_native)

```

A1.6 Worked example

Applying this procedure to Recommendation ITU-R BT.709 primaries $(x_R, y_R) = (0.6400, 0.3300)$, $(x_G, y_G) = (0.3000, 0.6000)$, $(x_B, y_B) = (0.1500, 0.0600)$, with CIE Illuminant D65 white $(x_w, y_w) = (0.3127, 0.3290)$ — the same primaries and white point as ColourPrimaries = 1 in Recommendation ITU-T H.273, Table 2 — produces:

```
[ 0.4360413  0.3851129  0.1430458 ]
[ 0.2224845  0.7169051  0.0606104 ]
[ 0.0139202  0.0970672  0.7139126 ]
```

which matches, to within a few `s15Fixed16Number` least-significant bits, the widely published sRGB-to-XYZ-D50 matrix used by existing sRGB ICC profiles:

```
[ 0.4360747  0.3850649  0.1430804 ]
[ 0.2225045  0.7168786  0.0606169 ]
[ 0.0139322  0.0971045  0.7141733 ]
```

NOTE Small (sub-visible) differences of a few `s15Fixed16Number` least-significant bits between independent implementations of this procedure are expected and immaterial for colour-management purposes; they arise from the precision each implementation carries for intermediate white-point values, not from any ambiguity in the procedure itself. This worked example was independently verified against both the implementation of A1.5 and a numerical linear-algebra library, with the two agreeing exactly.

The procedure also handles primaries whose chromaticity coordinates are negative, which is exactly the case this proposal’s choice of a signed `s15Fixed16Number` encoding (rather than HAGC’s unsigned `uInt16Number`) was intended to support (see the Introduction, “The solution”): applying it to a set of primaries with a negative blue-primary y coordinate — the shape of ACES AP0’s blue primary — completes without any special-casing.

Last updated Sep 14, 2026 at 12:52:52 PM