Proposed COLR revisions to ISO/IEC 14492 5th Edition Working Draft

Peter Constable, Microsoft
Dominik Röttsches, Google
Jonathan Kew, Mozilla

October 10, 2022

These are proposed changes to the WD reflected in ISO/IEC JTC 1/SC 29/WG 03 N0631. The main focus of these proposed revisions is on clause 5.7.11, COLR – Color Table, based on insights gains from vendors implementing support for enhancements to the COLR table introduced in Amendment 2 to ISO/IEC 14496-22:2019.

Summary of proposed revisions:

- _In 5.7.11.1.1, delete obsolete content (from an early draft of 4th ed./Amd 2 that was meant to be dropped but overlooked).
- *In 5.7.11.1.2.2, clarifications are added regarding interaction between color line definitions and font variations. Also, guidance is provided for implementers working with application platforms that have limiting constraints on color stop offsets.*
- *In 5.7.11.1.2.3, guidance is provided for implementers working with application platforms that support linear gradients but that have limiting constraints on color stop offsets.*
- *In 5.7.11.1.2.4, details are added regarding expected behaviour for radial gradients when variation deltas are applied to the gradient definition—specifically, when deltas applied to circle radii result in negative values. Also, guidance is provided for implementers working with application platforms that support radial gradients but that have limiting constraints on color stop offsets.*
- *In 5.7.11.1.2.5, it has been found that the description of sweep gradients was confusing for implementers, resulting in non-interoperable implementation behaviours. This was due in part to the semantics for the sweep gradient data diverging from the semantics of analogous data in other existing libraries and specifications, such as CSS. Changes are proposed to the semantics for gradient data described in 5.7.11.1.2.5 to better align with other specifications, and which implementers of OFF have found clearer and more conducive to interoperability.*
- *Other editorial revisions are proposed.*

Instructions for proposed changes are presented in italic style. Blocks of proposed text content are presented in normal style.

---

*In **5.7.11.1.1**, make the following changes:*

Rev 1) In the second paragraph ("The CPAL color data..."), add the following sentence at the end:

If the palette entry index is 0xFFFF, the alpha value in the COLR structure is multiplied into the alpha value of the text foreground color.

Rev 2) Delete the third paragraph, "Two color index record formats..."

---

*In **5.7.11.1.2.2**, make the following changes:*

Rev 3) Replace the third paragraph ("A color stop is defined...") with the following:

A color stop is defined by a real number, the *stop offset*, and a color. A color line shall have at least one color stop. Stop offsets are represented using F2DOT14 values, therefore color stops can only be specified within the range [-2, 2). In a variable font, however, stop offsets can be variable, and with the application of deltas can lie outside that range. (See 5.7.11.2.5 for format details.) If only one color stop is specified, that color shall be used for the entire color line; at least two color stops are needed to create color gradation.

Rev 4) After the fourth paragraph ("Color gradation is defined..."), insert the following paragraph:

Color stops shall be used in their *stop offset* order, which can be different from the order in which they are defined in the font. Also, in a variable font, stops shall be ordered according to offset values after variation deltas are applied. Therefore the ordering of colors in the color line definition can change between one variation instance and another instance.

Rev 5) Replace the eighth paragraph ("If there are multiple...") with the following:

If there are multiple color stops defined for the same stop offset, the first one given in the font shall be used for computing color values on the color line below that stop offset, and the last one shall be used for computing color values at or above that stop offset. All other color stops for that stop offset shall be ignored. In a variable font, if two color stops have different default offset values but vary to have the same offset value for a given variation instance, the stops shall be applied to the color line in the order in which they are given in the font, as described above.

Rev 6) After Figure 5.17, replace NOTE 2 ("Special considerations apply...") with the following:

NOTE: Given the above specifications for how color lines are defined, and also allowing for the application of variation deltas, the definition of a color line can span an interval greater than, narrower than, or in other ways different from the interval [0, 1]. However, some application platforms that provide an implementation for gradients can require a color line to be defined for exactly the interval [0, 1]. Implementations can work around such constraints by *normalizing* the offsets of the defined color stops. For example, if constrained to the range [0, 1], map the lowest stop offset to 0, map the largest offset to 1, and linearly interpolate normalized offsets for other stops in between. Then, for correct alignment to the geometry of a gradient definition, interpolate or extrapolate alternate geometric values to align with the normalized 0 and 1 stop offsets. For radial gradients, special geometry considerations can also apply; see 5.7.11.1.2.4 for more information.

---

*In **5.7.11.1.2.3**, make the following changes:*

Rev 7) Insert the following at the end of the clause:

Note on implementation of linear gradients

Some applications platform can provide an implementation for linear gradients that requires a color line to be specified for exactly the interval [0, 1]. If an application is using such a platform implementation but the font has a linear gradient with a color line specified over a different interval, the application can work around the platform constraint by normalizing the color line to the interval [0, 1]. This would require deriving alternate geometric points for the linear gradient definition. For the defined points p_0, p_1 and p_2, alternate points p_0', p_1'

and p_2' can be derived as follows: if o_{min} is the minimum stop offset in the defined color line, o_{max} is the maximum offset, and vector **v** = p_1 - p_0, then let

- $p_0' = p_0 + o_{min}\mathbf{v}$
- $p_1' = p_0 + o_{max}\mathbf{v}$
- $p_2' = p_2 + o_{min}\mathbf{v}$

If an implementation derives a single vector with point p_3, as described above, then alternate points p_0 and p_3 can be derived as follows: with vector **w** = p_3 - p_0, let

- $p_0' = p_0 + o_{min}\mathbf{w}$
- $p_3' = p_0 + o_{max}\mathbf{w}$

---

*In **5.7.11.1.2.3**, make the following changes:*

Rev 8) Replace the third paragraph ("The drawing algorithm...") with the following:

The drawing algorithm for radial gradients follows the HTML WHATWG Canvas specification for createRadialGradient(), but adapted with alternate color line extend modes and more flexibility for color stop offsets. Radial gradients shall be rendered with results that match the results produced by the following steps.

Rev 9) Following Figure 5.34, after the first paragraph ("The artifacts seen..."), insert the following:

While the Paint table formats use unsigned values to define the circle radii, application of deltas in variable fonts can result in negative radii values. While a negative radius does not seem to make sense intuitively, negative radii are not a problem for the rendering algorithm described above: given the formula $r(\omega) = (r_1 - r_0)\omega + r_0$, $r(\omega)$ will have positive and negative values for different values of ω when the values of r_0 and r_1 are both positive, but also if either r_0 or r_1, or both, are negative. For example, if $r_0 = -1$ and $r_1 = -2$, then $r(\omega) = \omega - 2$, and $r(\omega) \geq 0$ for $\omega \geq 2$. There is one notable change, however, in the case of $r_0 = r_1$ (hence, $r(\omega) = r_0$): if the values are positive, a cylinder is painted (since $r_0 > 0$), but if the values are negative, then nothing is painted (since $r_0 < 0$).

NOTE: If r_0 is different from r_1, the geometry defined by the two circles can be understood as a 2D projection of a cone. The cone has a *tip* where $r(\omega) = 0$, and the cone is painted on the side of the tip for which $r(\omega) \geq 0$. In the general case,

- $\omega_{tip} = r_0/(r_0 - r_1)$
- if $r_1 > r_0$, $r(\omega) > 0$ for all $\omega > r_0/(r_0 - r_1)$
- if $r_1 < r_0$, $r(\omega) > 0$ for all $\omega < r_0/(r_0 - r_1)$

Rev 10) At the end of the clause, insert the following:

Note on implementation of radial gradients

Some application platforms can provide an implementation for radial gradients that require the circle radii to be non-negative. If an application is using such a platform implementation but a variable font instance has a radial gradient with a negative circle radius, the application can work around the platform constraint by deriving an alternate gradient definition that uses only circles with non-negative radii and that produces the

same visual appearance. This would also require deriving an alternate color line definition that preserves the same color line extend mode behaviors.

Similarly, if the platform provides an implementation for radial gradients that requires color stops to be defined for exactly the interval [0, 1] but the the font has a radial gradient with a color line specified over a different interval, the application can work around that platform constraint by normalizing the color line to the interval [0, 1]. But the normalized color line would require deriving alternate circle specifications, and a simple normalization can result in a circle with a negative radius.

Thus, handling of negative circle radii and normalization of the color line need to be considered together. The following observations will be helpful:

- If r_0 and r_1 are positive and $r_0 = r_1$, a simple normalization of the color line can be used.
- If the color stops that define the color line all have offsets ω such that $r(\omega) \geq 0$, then a simple normalization of the color line can be used with alternate circles that have non-negative radii.
- For pad extend mode, if some color stops have offsets ω such that $r(\omega) < 0$, the same visual result can be obtained by using a truncated color line definition in which all stops with offsets ω such that $r(\omega) < 0$ are replaced by a single stop for ω_{tip} with an interpolated color value. (See Interpolation of Colors in 5.7.12 for requirements on how colors are interpolated.)
- For pad and repeat extend modes, the extended color line has a repeating pattern with offset wavelength λ, and any color line definition that is a slice of the color line of length λ will produce the same visual results. Thus, the application can derive an alternate color line definition in which all stops have offsets ω such that $r(\omega) \geq 0$, normalize those stop offsets to [0, 1], then derive alternate circles that align to the normalized 0 and 1 offsets.

---

*In **5.7.11.1.2.4**, make the following changes:*

Rev 11) In the first note ("NOTE: The following figures..."), delete the word "circular".

Rev 12) In the paragraph after the note following figure 5.29 ("A sweep gradient is defined..."), replace "starting and ending angles" with "start and end angles".

Rev 13) For the next six paragraphs — from the paragraph that begins "The color line is aligned..." to the note that begins "NOTE: Because the sweep gradient is drawn..." — and interleaved figures 5.36 – 5.38, replace that content with the following:

For sampling and interpolating colors, the color line is aligned to an infinitely spiraling circular arc around the center point, with arbitrary radius. The position 0° on the arc means the direction of the positive x-axis, and 360° means one full counter-clockwise rotation. The ColorLine's stop offset 0 is aligned with the start angle, and stop offset 1 is aligned with the end angle, independent of which angle is larger.

Outside the defined interval of the color line, the color value of a position on the color line is filled in depending on its extend mode. See 5.7.11.1.2.1 Color Lines for more details. In effect, this means that the spiraling circular arc can be sampled for colors outside the defined color line interval.

To draw the sweep gradient, for each position along the circular arc starting from from 0° up to but not including 360°, a ray from the center outward is painted with the color of the color line where the ray intersects the circular arc at that particular angle. Angle positions below 0° and equal to or above 360° are not sampled for drawing the rays.

If the color line's extend mode is reflect or repeat and start and end angle are equal, nothing shall be drawn.

NOTE: When the sweep gradient's color line uses the repeat or reflect extend mode and the angular distance between start and end angle is small, this results in very high spatial-frequency transitions that can lead to Moiré patterns or other display artifacts. (See Figure 5.34 where this effect is shown for a similar case involving radial gradients).

Figure 5.36 illustrates a sweep gradient with the drawing direction progressing from 0° to 360°. The gradient is specified with a start angle of 110° and end angle of 230°. The color line is specified with a yellow stop at offset 0 (aligned to the start angle) and a red stop at offset 1 (aligned to the end angle). The pad extend mode is used, hence the color for angles below 110° is yellow, and the color for angles above 150° is red.

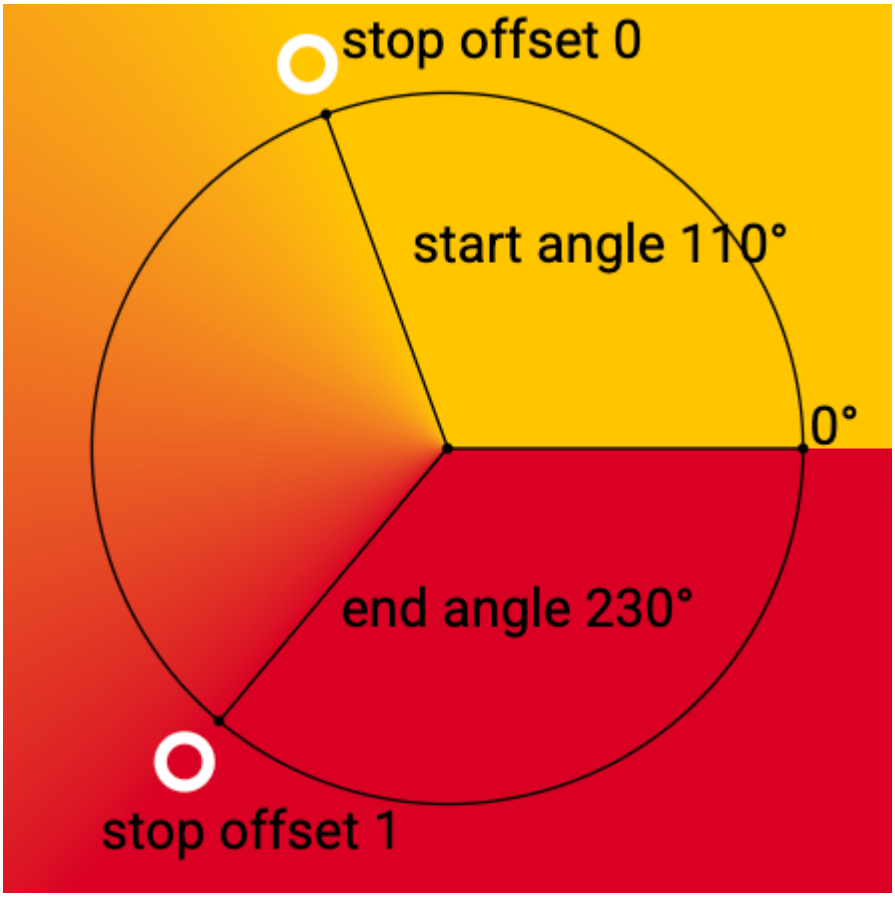

Figure 5.36 A sweep gradient with start angle of 110° and an end angle of 230°, using a `ColorLine` with color stops 0 and 1 for yellow and red and pad extend mode.

NOTE: If the start angle is less than or equal to the end angle, the color line progresses from smaller to larger offsets in the counter-clockwise direction along the circular arc. If the start angle is larger than the end angle, however, the color line is inverted and progresses in the clockwise direction, as illustrated in figure 5.37.

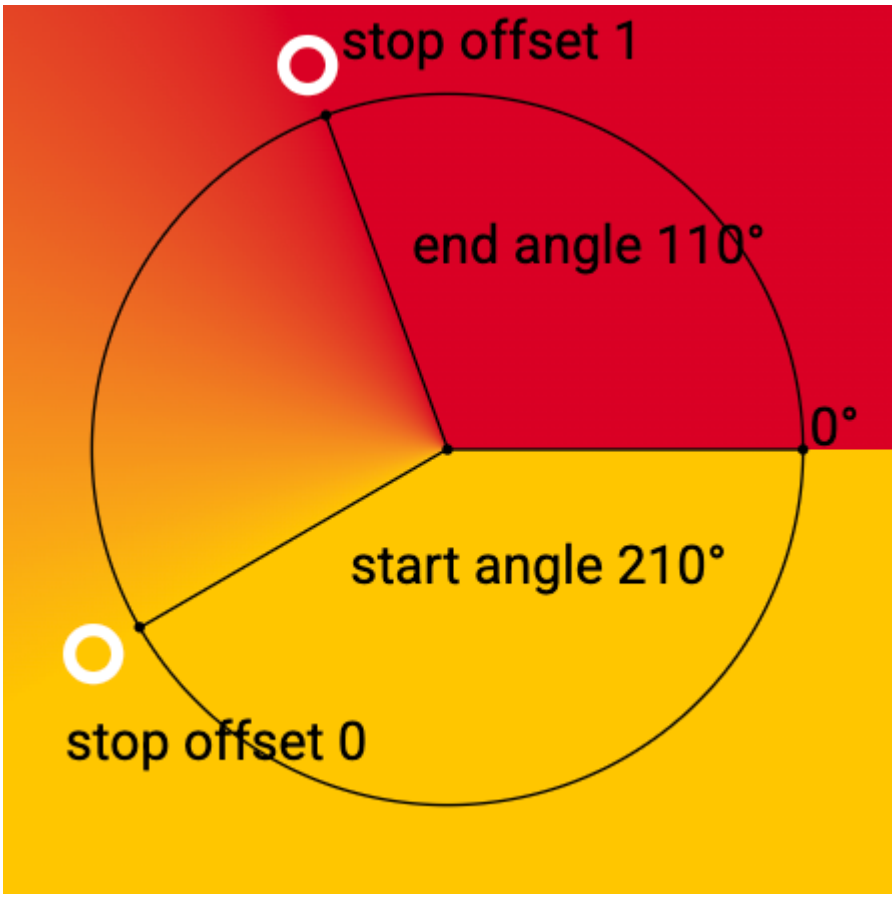

Figure 5.37 A sweep gradient, from yellow to red, with start angle of 210° and end angle of 110°, showing the inversion of the color line when the start angle is larger than the end angle.

Not more than one full rotation is drawn and there is no overlap in drawing for angles outside the 0° and 360° range. However, start and end angles may be positioned at angles below 0° or above 360°. Through that, and through how wide the color line interval is defined, color stops can lie outside the 0° to 360° circle. This has an effect on the computation of the gradient colors inside the interval of 0° to 360°, but colors are not sampled from outside this interval. See figure 5.38 for an example: The color line goes from yellow to red with a start angle of 330° and an end angle of 400°. The color for angles lower than 330° is yellow due to the pad extend mode, then the color line transitions from yellow at 330° to red at 400°, but only the color values up until 360° are sampled for drawing.

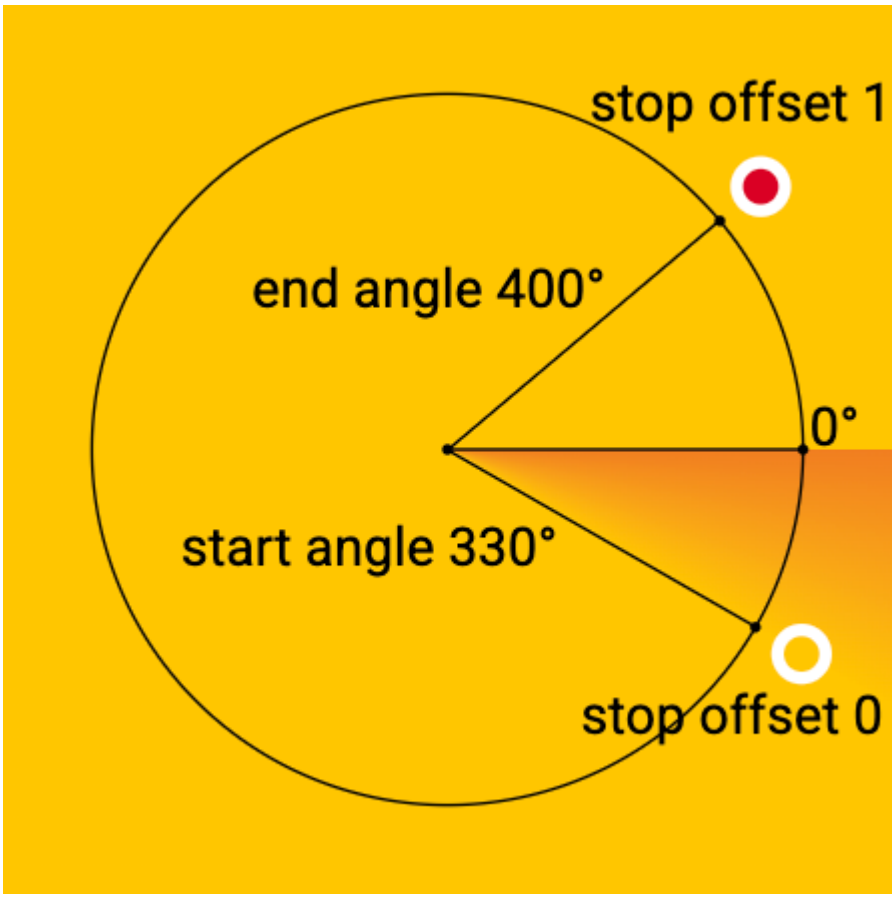

Figure 5.38 A sweep gradient with start angle of 330° and an end angle of 400°, using a color line with color stops 0 and 1 for yellow and red and extend mode pad.

NOTE: Because the sweep gradient is drawn from 0° to 360° a sharp transition can occur at 0°. This can be mitigated by adjusting the color stops at the 0° and 360° positions on the arc to have the same color. The location of the transition axis can also be shifted by nesting the sweep gradient inside a a rotation transformation (see 5.7.11.1.5).

Rev 14) Replace the following note 3 ("When a sweep gradient is combined...") with the following:

When a sweep gradient is combined with a transformation (see 5.7.11.1.5), the appearance will be the same as if (i) a circle of some non-zero radius was constructed with points on the circle computed from the start and end angles; (ii) the center point, circle and points on the circle were transformed; (iii) the color line was aligned to the transformed circle; and then (iv) a gradient was derived from the result, with rays from the transformed center point passing through the transformed color circle. When aligning the color line to the transformed circle, stop offset 0 would be aligned to the transformed point derived from the start angle, with stop offset 1 aligned to the transformed point derived from the end angle. Thus, a transform can result in the color line progressing in the opposite direction compared to the non-transformed gradient.

---

*In **5.7.11.2.5**, make the following changes:*

Rev 15) In the third paragraph ("The alpha value..."), replace "clipped" by "clamped".

Rev 16) In the sixth paragraph ("Two color stop record formats..."), delete the second sentence, "The format supporting variations..."

Rev 17) After the table describing the VarColorStop record, insert the following paragraph:

The VarColorStop format uses a base/sequence scheme to index into mapping data. Also, applying variation deltas to an F2DOT14 value requires special handling. See 5.7.11.4 for details.

---

*In **5.7.11.2.6.2**, make the following changes:*

Rev 18) At the end of the clause, insert the following paragraph:

The PaintVarSolid format uses a base/sequence scheme to index into mapping data. Also, applying variation deltas to an F2DOT14 value requires special handling. See 5.7.11.4 for details.

---

*In **5.7.11.2.6.5**, make the following changes:*

Rev 19) In the tables describing the PaintSweepGradient and PaintVarSweepGradient, replace the first sentence of the descriptions for startAngle by the following:

Start of the angular range of the gradient: add 1.0 and multiply by 180° to retrieve counter-clockwise degrees.

Rev 20) In the tables describing the PaintSweepGradient and PaintVarSweepGradient, replace the descriptions for endAngle by the following:

End of the angular range of the gradient: add 1.0 and multiply by 180° to retrieve counter-clockwise degrees.

Rev 21) Following the table for PaintVarSweepGradient, replace the first paragraph ("The PaintVarSweepGradient format...") by the following:

The PaintVarSweepGradient format uses a base/sequence scheme to index into mapping data. Also, applying variation deltas to an F2DOT14 value requires special handling. See 5.7.11.4 for details.

Rev 22) Add the following note at the end of the clause:

NOTE: To allow for a representation of +360°, a bias of 1.0 is used in the representation of start and end angles of sweep gradients. For example, an F2DOT14 value of -2.0 (0x8000) represents -180°; an F2DOT14 value of 0.0 (0x0000) represents +180°; an F2DOT14 value of 0.25 (0x1000) represents +225°; an F2DOT14 value of 1.0 (0x4000) represents +360°. However, a bias is not used for representation of angles in rotate or skew transforms (5.7.11.2.5.11, 5.7.11.2.5.12).

---

*In **5.7.11.2.6.8**, make the following changes:*

Rev 23) Following the table describing the VarAffine2x3 table, replace the first paragraph ("The VarAffine2x3 format...") by the following:

The VarAffine2x3 format uses a base/sequence scheme to index into mapping data. Also, applying variation deltas to a Fixed value requires special handling. See 5.7.11.4 for details.

---

*In **5.7.11.2.6.10**, make the following changes:*

Rev 24) Following the table describing the PaintVarScaleUniformAroundCenter table, replace the first paragraph ("The PaintVarScale,...") by the following:

The PaintVarScale, PaintVarScaleAroundCenter, PaintVarScaleUniform, and PaintVarScaleUniformAroundCenter formats use a base/sequence scheme to index into mapping data. Also, applying variation deltas to an F2DOT14 value requires special handling. See 5.7.11.4 for details.

---

*In **5.7.11.2.6.11**, make the following changes:*

*Rev 25) Following the table describing the *PaintVarRotateAroundCenter table, replace the first paragraph ("The PaintVarRotate...") by the following:*

The PaintVarRotate and PaintVarRotateAroundCenter formats use a base/sequence scheme to index into mapping data. Also, applying variation deltas to an F2DOT14 value requires special handling. See 5.7.11.4 for details.

---

*In **5.7.11.2.6.12**, make the following changes:*

*Rev 26) Following the table describing the *PaintVarSkewAroundCenter table, replace the first paragraph ("The PaintVarSkew...") by the following:*

The PaintVarSkew and PaintVarSkewAroundCenter formats use a base/sequence scheme to index into mapping data. Also, applying variation deltas to an F2DOT14 value requires special handling. See COLR table and OpenType Font Variations for details.

---

*In **5.7.11.4**, make the following changes:*

Rev 27) In the second paragraph ("Variation data is provided..."), replace "Item Variation Store table" by "ItemVariationStore table".

Rev 28) In the third paragraph ("In a variable font..."), replace three occurrences of "Item Variation Store" by "item variation store".

Rev 29) In the fifth paragraph ("For example..."), replace "has eight variable fields" by "has six variable fields".

Rev 30) In the ninth paragraph ("If the COLR table does not..."), replace "Item Variation Store subtable" by "ItemVariationStore subtable".

Rev 31) In the tenth paragraph ("If the COLR table contains..."), replace "Item Variation Store" by "item variation store".

Rev 32) In the eleventh paragraph ("For variable fonts..."), insert the following sentence at the end:

Also, special considerations apply for color lines and for radial gradients. See 5.7.11.1.2.2 and 5.7.11.1.2.4 for details.

Rev 33) After the eleventh paragraph, insert the following paragraph:

Some COLR version 1 formats allow for variation deltas to be applied to F2DOT14 or Fixed values. While the F2DOT14 and Fixed types have fractional components, variation deltas are integer values. Applying deltas to F2DOT14 or Fixed values requires special handling in which these values are treated like integers. See 7.2.3.4 for details.