



3 Barker Avenue – White Plains, NY 10601
Tel: 914-761-1100 – Fax: 914-761-3115
Web Site: www.smpte.org

SMPTE Technical Report

TTML Features for IMF Data Essence

SMPTE is developing standards for the Interoperable Master Format (IMF). IMF is a file-based framework designed to represent high-quality versions of a given finished multimedia work destined for distribution across multiple channels, including broadcast and Internet. It is intended for professional applications worldwide.

IMF plans to base its data essence format on SMPTE ST 2052-1, which is in turn based on W3C TTML. This data essence format would be used for captions, subtitles, karaoke, commentary, etc., and for both delivery to client devices and burn-in at the time of content encoding. In the latter scenario, where creative approval is sometimes necessary, burned-in data essence is rendered consistently across implementations.

In order to support IMF requirements, SMPTE ST 2052-1 needs to be extended with additional features, which are listed in the "Data Essence Features" section. The "ST 428-7 Excerpts" section contains excerpts from the draft revision to SMPTE ST 428-7¹ (dated 27/09/2012), which describes in detail a current implementation of these features.

Data Essence Features

3D

See Sections 6.3.6, 6.3.7 and 6.12 at ST 428-7 Excerpts.

A `zValue` style parameter shall be applicable to individual `<p>` elements for the purpose of defining the rendering of the element in terms of left- and right-eye stereoscopic images.

`zValue` shall be the horizontal distance between the origin of the element as rendered in the left eye and right eye images. Half of the `zValue` shall correspond to the horizontal distance between the origin of the element as rendered in either the left or right images, and the origin of the element in the absence of a `zValue`. Positive (negative) `zValue` values shall move the element away from (closer to) the viewer.

`zValue` shall be a signed decimal number and shall be expressed as a percentage of the width of the root container region.

¹ SMPTE ST 428-7 is an XML-based subtitling standard, which is in worldwide use in D-Cinema applications



3 Barker Avenue – White Plains, NY 10601
Tel: 914-761-1100 – Fax: 914-761-3115
Web Site: www.smpte.org

The vertical position of the element shall be unchanged by the application of `zValue`, and shall be identical in both left- and right-eye images.

`zValue` shall apply to the entire content of the `<p>` element.

The `zValue` style parameter shall be animatable, without interpolation. In order to avoid markup duplication, it shall be possible to define a given sequence of `<set tts:zValue="...">` elements once and apply it to multiple `<p>` elements.

tts:fontSize as percentage of container dimensions

See Section 6.4.9 at *ST 428-7 Excerpts*, and Section 8.2.9 at *TTML*².

It shall be possible to express `tts:fontSize` dimensions as a percentage of the root container dimensions.

Note: TTML currently allows percentage values for `tts:fontSize`, but defines such values as "relative to parent element's font size" (see Section 8.2.9 in TTML).

Note: this allows the same TTML document to be used to burn-in subtitles on a range of container sizes, e.g. 1080p, 2K, 4K, without modifying `tts:fontSize` for each output.

<p> origin and extent

See Sections 6.3.1-6.3.4 at *ST 428-7 Excerpts*, and Sections 8.2.7 and 8.2.14 at *TTML*.

It shall be possible to specify the origin and extent of an individual `<p>` element relative to its parent region, which may be the root container region, without requiring the definition of a unique region for each such element.

<p> fade-up and -down

See Sections 6.1.4-6.1.6 at *ST 428-7 Excerpts*, and Section 8.2.13 at *TTML*.

It shall be possible to apply the `tts:opacity` style parameter to an individual `<p>` element. The parameter shall be animatable.

In order to avoid markup duplication when `tts:opacity` is used to create fade-up and -down effects, it shall be possible to either (a) define a given sequence of `<set tts:opacity="...">` elements once and apply it to multiple `<p>` elements, or (b) allow that linear interpolation be applied between `<set>` values.

Ruby text

See Sections 6.5-6.8 at *ST 428-7 Excerpts*, and *Ruby Annotation* at <http://www.w3.org/TR/Ruby/>.

² <http://www.w3.org/TR/taf1-dfxp/>



3 Barker Avenue – White Plains, NY 10601
Tel: 914-761-1100 – Fax: 914-761-3115
Web Site: www.smpite.org

A Ruby element shall consist of two sequences of characters: a base character sequence and an associated Ruby character sequence, as illustrated in Figure 1. It shall be considered an inline block for formatting and flow.

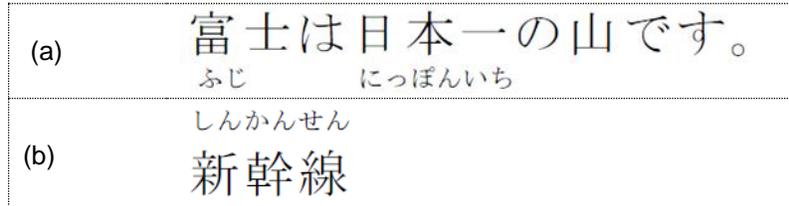


Figure 1. Ruby text examples.

Note: Ruby text is typically used with logographic characters such as Chinese or Japanese as pronunciation guides for characters that are likely to be unfamiliar to the reader.

The baseline of a Ruby text shall be the baseline of its base sequence.

All style parameters applicable to shall be applicable to both the base and Ruby sequences individually. The following additional style parameters shall be applicable to the Ruby sequence.

Position

It shall be possible to specify the position of the Ruby sequence relative to the base sequence.

A position of "before" shall indicate that the Ruby characters are rendered above the base characters for horizontal progression, or to the right of the base characters for vertical progression.

A position of "after" shall indicate that the Ruby characters are rendered below the base characters for horizontal progression, or to the left of the base characters for vertical progression.

Offset

It shall be possible to specify the offset, in em, between the rendered Ruby and base sequences. The permissible negative offset shall be no less than -1.0em.

Mixed vertical-horizontal progression direction

See Section 6.10 at ST 428-7 Excerpts, and Section 8.2.24 at TTML.

It shall be possible to specify a horizontal progression direction for a element within a container where progression direction is otherwise vertical.

Note: This is most common for special characters in an Asian text string that is being displayed vertically, as illustrated in Figure 2.



3 Barker Avenue – White Plains, NY 10601
Tel: 914-761-1100 – Fax: 914-761-3115
Web Site: www.smpte.org

1963
年
は
良
い
年
だ
っ
た
。

Figure 2. Examples of mixed progression directions.

Forced display

A forcedDisplay parameter shall be applicable to <p> elements and shall indicate that the element is to be displayed even if not explicitly activated. The forcedDisplay parameter shall be animatable.

Note: IMF Data Essence may be selected for presentation based on user preferences, as a result of user action, or by default. Such Data Essence is however displayed only if either (a) explicitly activated as a result of user action or (b) identified as “forced”.

Individual character rotation

See Section 6.11 at ST 428-7 Excerpts.

It shall be possible to specify that each rendered character within a element be individually rotated right or left around its center by 90 degrees.

Note: This is most common for special characters in an Asian text string that is being displayed vertically.

Bottom-to-top text direction

See Section 6.3.5 at ST 428-7 Excerpts, and Section 8.2.24 at TTML.

It shall be possible to specify bottom-to-top progression direction. When text is vertically rendered, each individual character shall retain its horizontal orientation - the placement of the subsequent character shall be vertically positioned in relation to it.

Superscript and subscript character style

See Section 6.4.2 at ST 428-7 Excerpts.



3 Barker Avenue – White Plains, NY 10601
Tel: 914-761-1100 – Fax: 914-761-3115
Web Site: www.smpte.org

It shall be possible to apply a superscript/subscript style parameter to a element, for the purpose of introducing an offset from the baseline of horizontal text. The parameter value shall be one of "sub", "super" or "normal".

For a value of "normal", no modifications to the rendered text shall be made.

For a value of "sub" and "super", the baseline of the element shall be offset vertically by – 0.5 em and 0.5 em, respectively. In both cases, the font size shall be 0.6 em.

The parameter is not applicable when vertical progression is used.

Drop Shadow character style

See Section 6.4.3, 6.4.8 and 6.4.12 at ST 428-7 Excerpts.

It shall be possible to apply a drop shadow style parameter to a element for the purpose of altering the contrast of the rendered text against the primary picture.

The application of the parameter shall result in the drawing of a drop shadow behind each character, falling diagonally to the bottom and right of each character.

It shall be possible to specify the color (including alpha channel) and thickness of the drop shadow. The thickness shall be expressed as a positive decimal number in units of em.

The drop shadow effect shall remain on the bottom and right of each character, whatever the progression direction.

Feathering of shadow and outline effects

See "Drop Shadow character style" above, Section 6.4.13 at ST 428-7 Excerpts and Section 8.2.20.

It shall be possible to specify whether drop shadow and outline effects applied to a element are feathered or not, for the purpose of smoothing the effect.

The feathering shall be applied linearly through the entire effect thickness. Maximum feathering shall be on the edge of the shadow or outline, and minimum feathering shall be at the edge of the character.

Character spacing, i.e. letter-spacing

See Section 6.4.11 at ST 428-7 Excerpts.

It shall be possible to introduce a constant amount of additional spacing between characters within a element. The spacing shall be expressed as a decimal number in units of em and shall be no less than -1 em.

Note: this can be used to lengthen or shorten the extent of the element slightly.



3 Barker Avenue – White Plains, NY 10601
Tel: 914-761-1100 – Fax: 914-761-3115
Web Site: www.smpte.org

Inline space

See Section 6.9 at ST 428-7 Excerpts.

It shall be possible to introduce inline space of arbitrary linear dimension in the current progression direction. The space shall be expressed in units of em and shall be no less than -1 em.

ST 428-7 Excerpts

IMPORTANT NOTICE

The excerpts in this section are Copyright ©2012 by THE SOCIETY OF MOTION PICTURE AND TELEVISION ENGINEERS (3 Barker Avenue, White Plains, NY 10601) and are made available solely for review. They are subject to change without notice and shall not be referred to as a SMPTE Standard.

Definitions

Point

A “point” is a unit of measurement, commonly abbreviated as “pt”. Fonts shall be rendered as if the primary picture frame size is 11 inches in height, so a 72 point font shall be 1/11 active frame height.

em

An “em” is a unit of measurement. 1 em shall be equal to the currently specified point size, meaning that 1 em in a 16 point typeface shall be 16 points. Therefore, 2 em in a 16 point typeface shall be 32 points and 0.5 em in a 16 point typeface shall be 8 points.

Baseline

For a given font, the baseline shall be the line upon which most letters “sit” and below which descenders extend (e.g. the following lower case letters have descenders: g, j, p, q, y).

The baseline of a `Ruby` container (see Section 0) shall be the baseline of its `Rb` element’s character(s).

Subtitle Time Code

Subtitle time code shall be a value of the form `HH:MM:SS:E+`, or, hours, minutes, seconds and subtitle editable units, respectively. It is a representation of a count of subtitle editable units. The `HH:MM:SS` portion is an approximation of real time - it matches real time only if the `EditRate` element’s ratio (see Section **Error! Reference source not found.**) corresponds to an integer (e.g. “24 1” corresponds to 24 which is an integer, while “24000 1001” corresponds to 23.98 which is not an integer).

`HH` field shall be encoded on two digits, with a value between 00 and 23 (inclusive). `MM` and `SS` fields shall each be encoded on two digits, with a value between 00 and 59 (inclusive).

The subtitle editable units field (`E+`) of a subtitle time code shall have a value between zero and “`TimeCodeRate - 1`” (inclusive). The number of digits to use for the `E+` field shall match the minimum number of digits required to encode the value “`TimeCodeRate - 1`” (see Section **Error! Reference source not found.**). For example:

- if the `TimeCodeRate` element’s value is between 11 and 100 (inclusive), the `E+` field shall be encoded using two digits – allowing the encoding of a value between 00 and 99 (inclusive).
- if the `TimeCodeRate` element’s value is between 101 and 1000 (inclusive), the `E+` field shall be encoded using three digits – allowing the encoding of a value between 000 and 999 (inclusive).

All ‘`E+`’ fields within a given DCDM Subtitle document shall be encoded using the same number of digits.

The maximum value that can be expressed by a subtitle time code shall be `23:59:59:MaxE`, where ‘`MaxE`’ is one number less than the value of `TimeCodeRate` element - encoded with the appropriate number of digits as defined above.

Note: The formula to convert the above HH:MM:SS:E+ encoded value into the actual count of subtitle editable units is provided below:

$$\text{Count of subtitle editable units} = (((((\text{HH} \times 60) + \text{MM}) \times 60) + \text{SS}) \times \text{TimeCodeRate}) + \text{E} +$$

Section 6.1.4 – FadeUpTime attribute

The duration, expressed as hours, minutes, seconds and subtitle editable units (HH:MM:SS:E+, see Section 0), of the fade of the subtitle instance from 100% transparent to the full specified opacity, with the possible overlapping related exception presented in Section 0. The default value shall be 00:00:00:TwoE, where TwoE is the value '2' encoded using the appropriate number of digits (e.g. '02' on two digits, or '002' on three digits - see Section 0), with the possible overlapping related exception presented in Section 0.

For example (in the case of a subtitle instance that does not overlap any other subtitle instance), if FadeUpTime="00:00:00:02", FadeDownTime="00:00:00:02", TimeIn="00:00:00:00" and TimeOut="00:00:00:20", then:

- . Editable unit 00 should have 0% of the full specified opacity
- . Editable unit 01 should have 50% of the full specified opacity
- . Editable unit 02 should have 100% of the full specified opacity

See additional FadeUpTime related provisions in Section 0.

Section 6.1.5 – FadeDownTime attribute

The duration, expressed as hours, minutes, seconds and subtitle editable units (HH:MM:SS:E+, see Section 0), of the fade of the subtitle instance from full specified opacity to 100% transparent, with the possible overlapping related exception presented in Section 0. The default value shall be 00:00:00:TwoE, where TwoE is the value '2' encoded using the appropriate number of digits (e.g. '02' on two digits, or '002' on three digits - see Section 0), with the possible overlapping related exception presented in Section 0.

For example (in the case of a subtitle instance that does not overlap any other subtitle instance), if FadeUpTime="00:00:00:02", FadeDownTime="00:00:00:02", TimeIn="00:00:00:00" and TimeOut="00:00:00:20", then:

- . Editable unit 20 should have 0% of the full specified opacity
- . Editable unit 19 should have 50% of the full specified opacity
- . Editable unit 18 should have 100% of the full specified opacity

See additional FadeDownTime related provisions in Section 0.

Section 6.1.6 – Subtitle Element Constraints

(TimeOut - FadeDownTime) shall be greater than or equal to (TimeIn + FadeUpTime).

Access to the content of a subtitle instance may be ruled by information specified outside of this DCDM Subtitle XML file - e.g. using a Composition Playlist's EntryPoint and/or Duration element(s) within the scope of Digital Cinema. Such external subtitle instance access rules may invalidate a portion of a subtitle instance's time window (defined by its TimeIn/TimeOut values). In that case, the entire content of the subtitle instance shall be displayed during the remaining (valid) portion of the subtitle instance.

When two subtitle instances have overlapping time windows, their respective FadeUpTime and FadeDownTime attributes values (or their default values if absent) may be considered equal to "00:00:00:ZeroE" (where 'ZeroE' is the zero value encoded using the appropriate number of digits - see Section 0) by the rendering device – even if the fading operations are specified outside of the overlapping period.

When a subtitle instance's time window does not overlap any other subtitle instance's time window, its `FadeUpTime` and `FadeDownTime` attributes values (or their default values if absent) shall be enforced.

Note: The overlapping of time window (defined by `TimeIn/TimeOut` values) of subsequent subtitle instances is allowed by this specification. Constraints on the number of simultaneous subtitle instances can be defined in accompanying specification (e.g. Operational Constraints document). A persistent subtitle instance (meaning a subtitle instance meant to stay on screen while more than one subsequent subtitle is displayed and removed) can be an issue for some equipment. The effect of a persistent subtitle can be accomplished by repeating the persistent text or image into subsequent subtitle instances.

Section 6.3.1 – Halign attribute

Horizontal alignment mode. Determines the edge or center of the text and primary picture's frame to be used when calculating position with `Hposition`. The value shall be one of [`left`, `center`, `right`]. The default value shall be `center`.

Section 6.3.2 – Hposition attribute

Specifies the distance of the text from the edge or the center of the primary picture's frame. The distance is expressed as a percentage of primary picture width. **Error! Reference source not found.** lists the values of `Halign` and the resulting interpretation of `Hposition`. The value shall be encoded as a signed decimal number having zero or more decimal places. The default value shall be 0 (zero).

Section 6.3.3 – Valign attribute

Vertical alignment mode. Determines the edge or center of the primary picture's frame to be used when calculating position with `Vposition`. The value shall be one of [`top`, `center`, `bottom`]. The default value shall be `center`.

Section 6.3.4 – Vposition attribute

Specifies the distance of the text's baseline from the edge or the center of the primary picture's frame. The distance shall be expressed as a percentage of primary picture height. **Table 1** lists the values of `Valign` and the resulting interpretation of `Vposition`. The value shall be encoded as a signed decimal number having zero or more decimal places. The default value shall be 0 (zero).

Table 1 – Valign and Vposition interaction for Text elements (Normative)

Valign Value	Vposition Interpretation
top	<p>The vertical distance between the top primary picture edge and the baseline for the characters drawn - if <code>Direction="ttb"</code> or <code>"btt"</code>, the baseline used shall be the baseline for the character displayed on top. No negative values allowed.</p> <p><u>Note:</u> Specifying <code>Valign="top"</code> <code>Vposition="0"</code> will result in characters being drawn off the top of the screen – and the descender of certain characters, like 'p', will remain visible.</p>
center	<p>The vertical distance between the center of the primary picture and the baseline for the characters drawn. Negative values indicate top offset, positive values indicate bottom offset.</p> <p>When <code>Direction="ttb"</code> or <code>"btt"</code>:</p> <ul style="list-style-type: none"> - in case of an odd number of characters, the baseline used shall be the baseline for the character displayed in the middle of the characters string - in case of an even number of characters, the baseline for the character located just before the middle of the text string (starting from the character displayed on top) shall be used.
bottom	<p>The vertical distance between the bottom primary picture edge and the baseline for the characters drawn - if <code>Direction="ttb"</code> or <code>"btt"</code>, the baseline used shall be the baseline for the first character displayed on the bottom. No negative values allowed.</p> <p><u>Note:</u> Specifying <code>Valign="bottom"</code> <code>Vposition="0"</code> will result in the descender of certain characters, like 'p', being drawn off the bottom of the screen.</p>

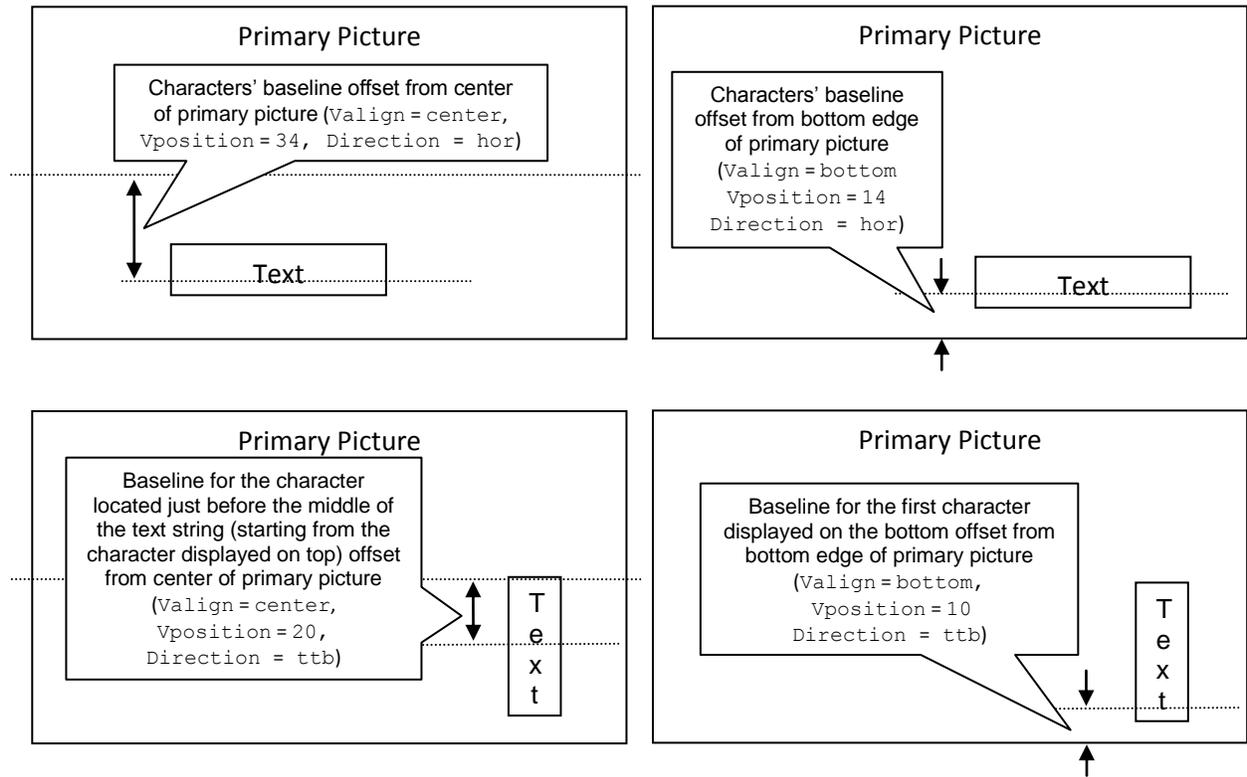


Figure 3 – Valign/Vposition Usage Illustration for Text Elements (Informative)

Note: DCDM Subtitle files belonging to a deprecated namespace (as listed in Section **Error! Reference source not found.**) could have been created with a slightly different Valign/Vposition

expected interpretations – based on text border instead of baseline. Implementations claiming compliance with this revised version of ST 428-7 are allowed to apply the `Valign/Vposition` interpretations listed in **Table 1** above, even when handling a legacy file belonging to one of these deprecated namespaces.

Section 6.3.5 – Direction attribute

For horizontally rendered text (i.e. for attribute's values `ltr`, `rtl` and `hor`), this attribute specifies the base writing direction. For vertically rendered text (i.e. for attribute's values `ttb` and `btt`), this attribute specifies the direction in which the text shall be rendered. One of the values in Table 2 shall be used. The default value shall be `ltr`.

Table 2 – Direction attribute values (Normative)

<i>Direction Value</i>	<i>Direction of Rendered Text</i>
<code>ltr</code>	Horizontal. Value meant for text to be rendered in the order that is left to right. Unicode Bidirectional Algorithm should be used. Some legacy equipment may not support this algorithm, while new implementations should use it.
<code>rtl</code>	Horizontal. Value meant for text to be rendered in the order that is right to left. Unicode Bidirectional Algorithm should be used, without re-inverting characters order due to <code>rtl</code> value usage. This should lead to a behavior similar to <code>ltr</code> value usage. Some legacy equipment may not support the Unicode Bidirectional Algorithm, and may just write the text enclosed in the <code>Text</code> element from right to left instead, while new implementations should use the Unicode Bidirectional Algorithm only.
<code>ttb</code>	Top-to-bottom. The Unicode Bidirectional Algorithm shall not be used.
<code>btt</code>	Bottom-to-top. The Unicode Bidirectional Algorithm shall not be used.
<code>hor</code>	Horizontal. The Unicode Bidirectional Algorithm shall be used.

When bidirectional text is present in vertically rendered text, the Unicode Bidirectional Algorithm shall not be used and the characters shall be rendered in the order in which they appear in the `Text` element (when reading its content from left to right), in the direction specified by this `Direction` attribute's value. When text is vertically rendered, each individual character shall retain its horizontal orientation - the placement of the subsequent character shall be vertically positioned in relation to it.

Note: The underlying thinking behind the above `Direction` attribute's values definition is to determine the horizontal rendering direction using the Unicode Bidirectional Algorithm only, without breaking legacy implementations that do not support it at the time this standard was revised. It is expected that horizontal rendering direction would be determined by this algorithm usage only in the future.

Section 6.3.6 – Zposition attribute

Provides a value corresponding to the horizontal distance between the “left eye” text center and the “right eye” text center at the screen plane level - in order to generate a stereoscopic effect. The `Zposition` attribute value shall be a percentage of the horizontal primary picture's size. The value shall be encoded as a signed decimal number having zero or more decimal places. Positive numbers shall move the text away from the viewer. Negative numbers shall bring the text closer to the viewer. The default value shall be 0 (zero) – it shall correspond to a perceived display in the plane of the screen.

In order to get the left and right eye text positions, half of the `Zposition` attribute value shall be used in both horizontal directions around what would be the center of the text if positioned without taking into account the `Zposition` and `VariableZ` attributes. Both eye texts shall be displayed using the same

vertical positioning information as defined by the associated `Valign/Vposition` attributes values (or their respective default values if absent) and using the same direction as defined by the associated `Direction` attribute (or its default value if absent).

All the content enclosed within the `Text` element shall be displayed at both left and right eye text resulting positions.

Section 6.3.7 – VariableZ attribute

The `VariableZ` attribute identifies a previously loaded vector of value-length pair(s) (see `LoadVariableZ`, Section 0) to be used in order to generate a stereoscopic effect for all content enclosed within the `Text` element, on devices that are able to handle `LoadVariableZ` elements. The `VariableZ` attribute value shall match one of the `ID` attribute values defined by the `LoadVariableZ` element(s) present in the same `Subtitle` element.

If the `VariableZ` attribute is present, then the `Zposition` attribute shall also be present in the same `Text` element. Devices that are not able to handle `LoadVariableZ` elements shall use the `Zposition` attribute value instead of the `LoadVariableZ` element referenced by the `VariableZ` attribute in order to generate a stereoscopic effect. Devices that are using the `VariableZ` attribute and its associated `LoadVariableZ` element shall ignore the `Zposition` attribute located in the same `Text` element.

If the `VariableZ` attribute is not present in a `Text` element, all devices shall use the `Zposition` attribute value (or its default value if absent) of this `Text` element in order to generate a stereoscopic effect.

Note: It is recommended that the `Zposition` attribute value associated with a `VariableZ` attribute be the most forward `Zvalue` present within the `LoadVariableZ` element referenced by this `VariableZ` attribute, while content creators remain free to choose the value that appears the most appropriate in each case.

Section 6.4.2 – Script attribute

Instructs the processor to render text offset from the baseline for horizontal text only. The value of the attribute determines the direction of the offset if any. One of the values in Table 3 shall be used. The default value shall be `normal`.

If the `Direction` attribute's value is `ttb` or `btt`, the `Script` attribute shall not be present – meaning that its default value shall be used.

Note: Document belonging to a deprecated namespace listed in Section **Error! Reference source not found.** could have a `Script` attribute present when the `Direction` attribute's value equals `ttb` or `btt`. Implementations claiming compliance with this version of ST 428-7 are allowed to disregard the `Script` attribute in such case.

The baseline for the text subject to superscript (meaning when `Script` attribute's value equals `super`) shall be above the baseline for the character(s) displayed without any superscript/subscript feature. The baseline for the text subject to subscript (meaning when `Script` attribute's value equals `sub`) shall be below the baseline for the character(s) displayed without any superscript/subscript feature. In both cases (meaning when `Script` attribute's value equals either `super` or `sub`), the absolute distance between the two baselines should be 0.5em, and it shall not be less than 0.3em, and not more than 0.7em. This range is provided in order to accommodate legacy implementations (new implementations should use 0.5em).

When `Script` attribute value equals `normal`, no offset shall be applied – meaning that the character's baseline shall match the baseline for the character(s) displayed without any superscript/subscript feature.

All other `Font` element's attributes shall be applied to the text (e.g. `Weight`, `Italic`, `Underline`, `Spacing`, `AspectAdjust`, `Color`, `Effect`, `Size`, `EffectColor`, `EffectSize`, `Feather`

attributes), whether the `Script` attribute's value equals `super`, `sub` or `normal`.

When the `Script` attribute's value equals `super` or `sub`, the size of the superscript or subscript text respectively shall be $0.6em$ – the reference font size being defined by the `Font` element's `Size` attribute or its default value, taking into account the `Font` attributes' superseding rule defined in Section **Error! Reference source not found.** When the `Script` attribute's value equals `normal`, the size of the text shall be $1em$ – taking into account the same previous reference font size. Note that this text's size can be adjusted if needed by using a `Font` element having a different `Size` attribute's value as illustrated in the example below (the example below assumes that there is no other `Font` element embedding the elements listed):

```
<Font ID="Arial">
  <Text Valign="top" Vposition="30.00">Normal text at default size: 42pt
    <Font Script="super">Superscript text at size: "42pt x 0.6"</Font>
    <Font Script="super" Size="30">Superscript text at size: "30pt x 0.6"</Font>
    <Font Script="sub">Subscript text at size: "42pt x 0.6"</Font>
  </Text>
</Font>
```

Table 3 – Script attribute values (Normative)

<i>Script Value</i>	<i>Offset Mode</i>
super	offset above baseline of text
sub	offset below baseline of text
normal	no offset - on the baseline of text

Section 6.4.3 – Effect attribute

Enables a special effect to alter the contrast of the rendered text against the primary picture. One of the values in Table 4 shall be used. The default value shall be `shadow`.

Note: Documents belonging to one of the deprecated namespaces listed in Section **Error! Reference source not found.** could have been created with `none` as the expected default `Effect` attribute's value. Implementations claiming compliance with this revised version of ST 428-7 are allowed to consider the default `Effect` attribute's value be `shadow` even in the case of these deprecated namespaces.

Table 4 – Effect attribute values (Normative)

<i>Effect Value</i>	<i>Effect Description</i>
border	Draw colored border around the four sides of each character. The color is determined by <code>EffectColor</code> attribute. The thickness of the border is determined by the <code>EffectSize</code> attribute. This border may be feathered or not depending on the <code>Feather</code> attribute.
shadow	Draw colored drop shadow behind each character – falling diagonally to the bottom and right of each character, independently of the <code>Direction</code> attribute's value. The color is determined by <code>EffectColor</code> attribute. The thickness of the drop shadow is determined by the <code>EffectSize</code> attribute. This drop shadow may be feathered or not depending on the <code>Feather</code> attribute.
none	No effect is applied to the rendered text.

Note: The shadow effect will remain on the bottom and right of each character, even if the text is written from right to left. Also, if the character is rotated (through `Rotate` element usage – see Section 0), the shadow effect will remain on the bottom and right of the character after rotation.

Section 6.4.8 – EffectColor attribute

The color to be used for rendering the effect chosen by the `Effect` attribute. The format shall be identical to that used by the `Color` attribute of this element. The default value shall be `FF000000` (opaque black).

If the rendering device is only able to use one alpha-blend value at a time, then the alpha-blend value of the `Color` attribute shall be used for both the text and its effect. However, the remaining values provided by the `EffectColor` attribute shall be applied to the effect in all cases.

Section 6.4.9 – Size attribute

The size of the characters. It shall be expressed in points. The default value shall be 42.

Section 6.4.11 – Spacing attribute

Provides additional spacing between the rendered characters. The spacing is specified in units of em. The value shall be encoded as a decimal number. This attribute can be used to increase or decrease the amount of space between adjacent characters. This can be used to lengthen or shorten the character string slightly. Typical usage would have values less than 1 em. Negative values are allowed but should be used with care as characters could overlap. Negative spacing shall be limited to no more than -1.0 em. The default value shall be “0.0”.

Section 6.4.12 – EffectSize attribute

Defines the thickness of the shadow or border effect (as defined in Section 0). The `EffectSize` value shall be specified in units of em. The value shall be encoded as a decimal number. Negative values shall not be used. The default value shall be “0.01”.

Note: A too small `EffectSize` attribute's value could result in a non-visible effect. As the `EffectSize` value is expressed in "em" (meaning dependent on the font used), a given `EffectSize` attribute's value associated with a font having a small size would generate a smaller effect than if it was associated with a font having a bigger size.

Section 6.4.13 – Feather attribute

Specifies if the effect (shadow or border effect as defined in Section 0) shall be feathered or not. Its purpose is to smooth the effect. The value shall be one of `yes` or `no`. The default value shall be `no`. A value of `yes` requires the effect (if present) to be feathered. A value of `no` requires the effect (if present) to remain un-feathered. In case no effect is applied to the character(s), the `Feather` attribute shall be disregarded.

When present, the feathering shall be implemented linearly through the entire effect thickness – maximum feathering shall be on the edge of the shadow or border, and feathering shall decrease until the edge of the character on which the effect is applied.

Section 6.5 – Ruby element

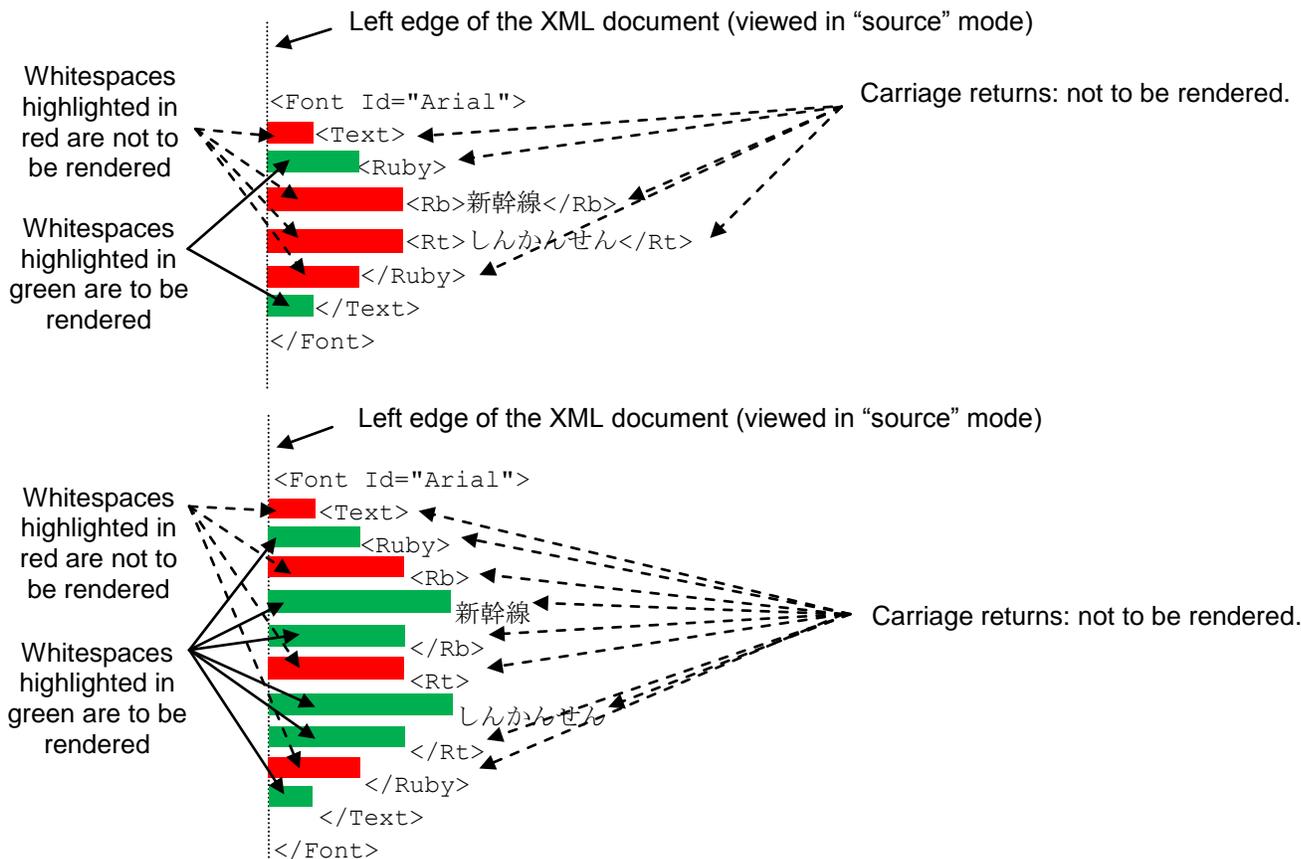
The `Ruby` element is a container used to specify ruby characters that are to be associated with a set of base characters in Asian text, specifically Japanese. The `Ruby` element shall contain exactly one (1) `Rb` element, and one (1) `Rt` element. No other text or elements shall be present inside the `Ruby` element. The `Ruby` element shall not be present inside elements other than a `Text` element.

As a container, the whole `Ruby` element shall be positioned according to the related `Hposition`, `Vposition`, `Zposition`, `VariableZ` attributes. This means that both `Rb` and `Rt` elements constituting the `Ruby` container shall be assembled (taking into account any adjustment due to the `Size`, `Position`, `Offset`, `Spacing` and/or `AspectAdjust` attributes of the `Rt` element) before aligning the resulting `Ruby` container on the values provided by the related `Hposition`, `Vposition`, `Zposition`, `VariableZ` attributes. Section 0 defines the baseline to use for such `Ruby` container.

The whitespace(s) rendering rule presented in Section **Error! Reference source not found.** is applicable inside a `Text` element containing a `Ruby` element, except concerning the portion located inside the `Ruby` element itself. Inside the `Ruby` element, the following rules shall be applied:

- All whitespaces located inside the `Ruby` element but outside of the `Rb` and `Rt` elements shall not be rendered.
- All whitespace(s) located inside the `Rb` or `Rt` element shall be rendered – without being collapsed.

The example below corresponds to portions of XML files viewed in “source” mode. They illustrate the above rules:



Section 6.6 – Rb element

The `Rb` element is a container used to specify the set of base characters that ruby text will be associated with in Asian text, specifically Japanese. The `Rb` element shall be present only inside a `Ruby` element. There shall be one (1) `Rb` element inside a `Ruby` element. No elements shall be present inside an `Rb` element.

The `Rb` element shall not be empty, meaning that at least one character shall be present in its value.

Note: Legacy DCDM Subtitle files belonging to a deprecated namespace (see Section **Error! Reference source not found.**) could contain empty `Rb` element(s). They are considered compliant with the standard as stated in Section **Error! Reference source not found.**

Section 6.7 – Rt element

The `Rt` element is a container used to specify the actual ruby characters that will be associated with the base characters specified in a corresponding `Rb` element in Asian text, specifically Japanese. The `Rt` element shall be present only inside a `Ruby` element. No elements shall be present inside an `Rt` element. There shall be one (1) `Rt` element inside a `Ruby` element. The `Rt` element shall have five attributes, `Size`, `Position`, `Offset`, `Spacing`, and `AspectAdjust`. These are described in the following sub sections. The ruby characters shall be centered with respect to the base characters in all cases.

Size attribute [optional]

Indicates the size of the rendered ruby characters. Character sizes shall be specified in units of em. Values for size shall be positive numbers (greater than 0). The default `Size` attribute's value shall be "0.5".

Position attribute [optional]

Indicates the position of the rendered ruby characters with respect to the base characters. Value shall be one of `before` or `after`. Specifying `before` shall indicate that the ruby characters shall be rendered above the base characters if the text direction is horizontal on screen. Specifying `before` shall indicate that the ruby characters are rendered to the right of the base characters if the text direction is vertical on screen. Specifying `after` shall indicate that the ruby characters are rendered below the base characters if the text direction is horizontal on screen. Specifying `after` shall indicate that the ruby characters are rendered to the left of the base characters if the text direction is vertical on screen. The default `Position` attribute's value shall be `before`.

Offset attribute [optional]

Indicates the offset or amount of space between the rendered ruby characters and the base characters. The offset shall be specified in units of em. Negative offset shall not exceed -1.0em. The default `Offset` value shall be "0.0".

Spacing attribute [optional]

Indicates additional spacing between the rendered ruby characters. The spacing shall be specified in units of em. This attribute can be used to increase or decrease the amount of space between adjacent ruby characters. This can be used to lengthen or shorten the ruby character string slightly. Typical usage would have values less than 1em. Negative values are allowed but should be used with care as characters could overlap. Negative spacing shall be limited to no more than -1.0em. Note: spacing can be specified here as well as in the `Font` element (see Section 0 above). The effect of both shall be cumulative. The default `spacing` value shall be "0.0".

AspectAdjust attribute [optional]

Used to adjust the aspect ratio of the rendered ruby characters. This attribute can be used to increase or decrease the width of each character. This can be used to lengthen or shorten the character string slightly. Values greater than 1.0 will widen each character and result in a longer string. Values less than 1.0 will narrow each character and result in a shorter string. Values shall be limited to not less than 0.25 and not more than 4.0. The default `AspectAdjust` attribute's value shall be "1.0".

Section 6.8 – Ruby Examples

Example 1:

```
<Text Direction="ltr" Halign="left" Hposition="11.4" Valign="top"
Vposition="95.6">
```

```
<Ruby>
```

```
<Rb>新幹線</Rb>
```

```
<Rt Size="0.5" Position="before" Offset="0.5" Spacing="0">しんかんせん</Rt>
```

```
</Ruby>
```

</Text>

Should produce something like this:

しんかんせん
新幹線

Example 2:

```
<Text Direction="ltr" Halign="left" Hposition="11.4" Valign="top"
Vposition="85.6">
<Ruby>
<Rb>富士</Rb>
<Rt Size="0.5" Position="after" Offset="0.5">ふじ</Rt>
</Ruby>
は
<Ruby>
<Rb>日本一</Rb>
<Rt Size="0.5" Position="after" Offset="0.5">にっぽんいち</Rt>
</Ruby>
の山です。
</Text>
```

Should produce something like this:

富士は日本一の山です。
ふじ にっぽんいち

Section 6.10 – HGroup element

The `HGroup` element is a container used to specify a string of characters that are to be rendered horizontally (from left to right) in a subtitle that has text direction set to vertical (meaning that the `Text` element's `Direction` attribute is set to either `ttb` or `btt`). This is most common for multi-digit numbers in an Asian text string.

This is a temporary override for the text direction specified as an attribute in the `Text` element. The `HGroup` element shall not be present inside elements other than `Text` elements. No elements shall be present inside an `HGroup` element.

The `HGroup` element shall not be present in a `Text` element that has a `Direction` attribute's value set to `ltr`, `rtl` or `hor`.

Example:

```
<Text Direction="ttb" Halign="right" Hposition="10.0" Valign="top"
Vposition="8.25"> <HGroup>1963</HGroup>年は良い年だった。</Text>
```

Should produce something like this:

1963
年
は
良
い
年
だ
っ
た
。

Section 6.11 – Rotate element

The `Rotate` element is a container used to specify a string of characters that are to be rotated either right or left by 90 degrees before they are rendered on screen. This is most common for special characters in an Asian text string that is being displayed vertically. The `Rotate` element shall not be present inside elements other than `Text` elements. The `Rotate` element shall not be present inside a `Font` element which is inside a `Text` element, and `Font` and `Rotate` elements shall not be present inside a `Rotate` element. The `Rotate` element has one optional attribute, `Direction` which is described below. No elements shall be present inside a `Rotate` element.

Each character in the `Rotate` element shall be rotated individually around its center before being rendered. If a border effect is present, it shall rotate with the character.

Direction attribute [optional]

Indicates the direction of the character rotation. Value shall be one of `left`, `right` or `none`. Specifying `none` indicates that the characters are not rotated. Specifying `right` shall indicate that the characters are rotated to the right or clockwise. Specifying `left` shall indicate that the characters are rotated to the left or counter-clockwise. The default `Direction` attribute's value shall be `none`.

Table 5 – Direction attribute values (Normative)

<i>Direction Value</i>	<i>Direction Value Description</i>
left	Indicates that the characters shall be rotated to the left or counter-clockwise.
right	Indicates that the characters shall be rotated to the right or clockwise.
none	Indicated that no rotation shall be applied to the characters.

Example:

```
<Text Direction="ttb" Halign="right" Hposition="10.0" Valign="top"
Vposition="8.25"> 私はドイツ国籍で<Rotate Direction="right">—</Rotate></Text>
```

Should produce something like this:

私
は
ド
イ
ツ
国
籍
で
—

Section 6.12 – LoadVariableZ element [optional]

Provides a succession of one or more value-length pair(s) separated using one (or more) whitespace(s). This succession of value-length pair(s) can be preceded by zero or more whitespace(s) and can be followed by zero or more whitespace(s). Each of these value-length pairs shall be formatted according to the following template:

Zvalue[:Length]

where:

- Zvalue shall correspond to the horizontal distance between the “left eye” text or image center and the “right eye” text or image center at the screen plane level - in order to generate a stereoscopic effect. Each Zvalue shall be a percentage of the horizontal primary picture’s size and shall be encoded as a signed decimal number having zero or more decimal places. Positive numbers shall move the text or image away from the viewer. Negative numbers shall bring the text or image closer to the viewer. The default value shall be 0 (zero) – it shall correspond to a perceived display in the plane of the screen.

For each Zvalue, in order to get the left and right eye positions, half of the Zvalue shall be used in both horizontal directions around what would be the center of the text or image if positioned using only the Halign/Hposition, Valign/Vposition and Direction attributes values (or their respective default values if absent) of the concerned Image or Text element – note that the Direction attribute does not exist for Image elements. For each Zvalue, both eye texts or images shall be displayed using the same vertical positioning information as defined by the associated Valign/Vposition attributes values (or their respective default values if absent). For a Text element, both eye texts shall be displayed using the same direction as specified by the associated Direction attribute’s value (or its default value if absent).

For each Image or Text element referencing the LoadVariableZ element, all the content enclosed within this Image or Text element shall be displayed at both left and right eye images or texts (respectively) resulting positions.

- “:Length” is optional. Length shall be a positive (non-zero) integer corresponding to the number of subtitle editable units during which the paired Zvalue shall be used in order to generate the stereoscopic effect. If absent, the default Length shall be 1.

The first value-length pair of the LoadVariableZ element shall be used for the first subtitle editable unit of the concerned Image or Text element - and for the duration defined by its Length field. The value-length pairs shall be used one after the other, in the order they are stored in the LoadVariableZ element, when reading the LoadVariableZ element from left to right.

Example:

The line below provides an example of LoadVariableZ element complying with the above definition:

```
<LoadVariableZ ID="Zvector1"> -0.7:10 0.1:14 0.2 0.3:7
</LoadVariableZ>
```

When one or more VariableZ attribute(s) are present in Text and/or Image element(s) within a given Subtitle element, at least one LoadVariableZ element shall be present within the same Subtitle element.

While it is expected that there is no difference between the number of subtitle editable units within the Subtitle instance and the number of subtitle editable units covered by LoadVariableZ element values, any differences shall be handled as presented below:

- if the list of values provided by a LoadVariableZ element is shorter than the duration of the associated Subtitle instance, then the processing device shall use the last value of the LoadVariableZ element for the entire remaining portion of the Subtitle instance – and this, for all its Text and/or Image element(s) referencing this LoadVariableZ element.
- if the list of values provided by a LoadVariableZ element is longer than the duration of the associated Subtitle instance, then the processing device shall truncate the list of

LoadVariableZ element values and ignore the value(s) going beyond the duration of the Subtitle instance – and this, for all its Text and/or Image element(s) referencing this LoadVariableZ element.

ID attribute

The LoadVariableZ element accepts a single attribute, ID, which creates an internal (to the parent SubtitleReel) identifier to be referenced by VariableZ attribute(s) located inside the same Subtitle element. The ID attribute shall be present in each LoadVariableZ element, and each LoadVariableZ element in a DCDM Subtitle file shall have a distinct ID attribute's value.