

SMPTE STANDARD

Digital Cinema Distribution Master – Subtitle



Table of Contents	Page
Foreword	2
Intellectual Property	2
1 Scope	3
2 Conformance Notation	3
3 Normative References	3
4 Overview	4
5 DCDM Subtitle Structure (Normative).....	7
6 Subtitle Instances (Normative).....	12
7 Samples (Informative).....	37
8 XML Schemas (Normative).....	39
9 XML Diagram Legend (Informative).....	42
Annex A Bibliography (Informative)	45
Annex B Summary of Main Changes from SMPTE ST 428-7:2010 (Informative).....	46

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices, and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in its Standards Operations Manual.

SMPTE ST 428-7 was prepared by Technology Committee 21DC.

Intellectual Property

At the time of publication no notice had been received by SMPTE claiming patent rights essential to the implementation of this Standard. However, attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. SMPTE shall not be held responsible for identifying any or all such patent rights.

1 Scope

This document specifies the format of a DCDM Subtitle file. A DCDM Subtitle file contains a set of instructions for placing rendered text or graphical overlays at precise locations on distinct groups of motion picture frames. A DCDM Subtitle File is an integral component of a D-Cinema composition and may be present in Digital Cinema Package (DCP) file sets. Consequently, its design reflects features of other DCP file formats. The DCDM Subtitle file format is not intended for use in streaming applications.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:".

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document. The keyword "reserved" indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword "forbidden" indicates "reserved" and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; Tables shall be next; followed by formal languages; then figures; and then any other language forms.

3 Normative References

The following standards contain provisions that, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

World Wide Web Consortium (W3C) (2004, February 4). Extensible Markup Language (XML) 1.0 (Third Edition)

World Wide Web Consortium (W3C) (2004, October 28). XML Schema Part 1: Structures (Second Edition)

World Wide Web Consortium (W3C) (2004, October 28). XML Schema Part 2: Datatypes (Second Edition)

Internet Engineering Task Force (IETF) (1996, November). RFC 2396 — Uniform Resource Identifiers (URI): Generic Syntax

Internet Engineering Task Force (IETF) (2005, July). RFC 4122 — A Universally Unique Identifier (UUID) URN Namespace

ISO/IEC 15948:2004, Information Technology — Computer Graphics and Image Processing — Portable Network Graphics (PNG): Functional Specification

ISO/IEC 14496, Part 22, 2009 Information Technology — Coding of Audio-Visual Objects — Open Font Format

ISO/IEC 10646-1, 2000 Information Technology — Universal Multiple-Octet Coded Character Set (UCS)

Unicode Consortium, (2012, January 16). Unicode Standard Annex #9 – Unicode Bidirectional Algorithm.

4 Overview

A DCDM Subtitle file, depicted in block form in Figure 1, is a representation of a series of *subtitle instances*: rendered text or graphical overlays on a primary picture in a D-Cinema work, such as a motion picture, trailer, or advertisement. A subtitle track file contains a set of file-global metadata and a set of subtitle structures which encode the content and temporal and spatial locations of the subtitles to be displayed over the primary picture.

A virtual timeline provides the temporal dimension of the DCDM Subtitle file. The timeline is a contiguous set of *editable units*. Spatial positions are expressed as percentages of the primary picture's size relative to a chosen border or center axis. The primary picture source shall have constant size.

The DCDM Subtitle file may reference external font resources for rendered text, and external image resources for graphical overlays. The font and image resource formats are specified in this document's normative references.

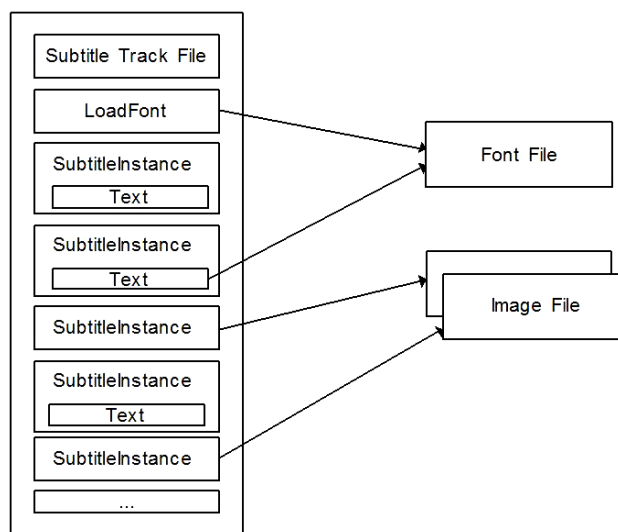


Figure 1 – Prototypical DCDM Subtitle File (Informative)

The structures defined in this document are represented using the *Extensible Markup Language* [XML 1.0], and specified using XML Schema [XML Schema Part 1: Structures] and [XML Schema Part 2: Datatypes].

This specification shall be associated with a unique XML namespace name [Namespaces in XML]. The namespace name shall be the string value “<http://www.smpte-ra.org/schemas/428-7/2014/DCST>”.

In addition, the namespace name defined in the 2010 version of this standard, the string value “<http://www.smpte-ra.org/schemas/428-7/2010/DCST>”, shall be supported by decoders and is available for use by content creators that do not wish to use the features specific to the 2014 schema. More information on the differences between these two schemas is presented in Section 8 below. To be clear, use of the 2010 schema is recommended for applications that do not need features of the 2014 schema and which are intent on compatibility with legacy implementations of this standard.

Table 1 lists the XML namespace names used in this specification. Namespace names are represented as Uniform Resource Identifier (URI) values [RFC 2396]¹.

The MIME type for a document containing a single `SubtitleReel` element as its root shall be `text/xml`.

Table 1 – XML Namespaces (URIs are Normative)

<i>Qualifier</i>	<i>URI</i>
dcst	http://www.smpte-ra.org/schemas/428-7/2014/DCST
dcst	http://www.smpte-ra.org/schemas/428-7/2010/DCST
xs	http://www.w3.org/2001/XMLSchema

An XML document must be parsed in its entirety before it can be considered valid, thus the format defined by this document is not suitable for streaming applications.

The URIs found in Table 1 are normative. The namespace qualifier values (also called namespace prefixes in XML jargon) used in Table 1 and elsewhere in this document, namely “dcst” and “xs”, are not normative meaning that they may be replaced in instance documents by any XML compliant namespace qualifier value. In other words, implementations shall expect any arbitrary XML compliant namespace qualifier value that is associated with a URI from Table 1.

Datatypes from other schemas that are used in this document will be prefixed with the appropriate namespace qualifier (e.g., `xs:dateTime`). See [XML Schema Part 2: Datatypes] and [XML-Signature Syntax and Processing] for further information about these types.

¹ Although an XML namespace URI value begins with a “method” element (“http” in this case), the value is designed primarily to be a unique string and does not necessarily correspond to an actual on-line resource. URI values are not intended to be resolved.

4.1 Deprecated Namespace Name

The namespace name listed below was defined by a previous version of this standard. Documents conforming to that version shall be considered conformant to this standard, and decoders conforming to this standard shall decode such documents, but this namespace name is not intended for new documents.

<http://www.smp-te-ra.org/schemas/428-7/2007/DCST>

4.2 Definitions

4.2.1

Primary Picture

The primary picture shall be the DCDM picture over which the rendered subtitle instances will be displayed. It has a height and a width, defining its size.

Note: The projected picture size could be different from the size of the primary picture (e.g., in the case of scaling).

4.2.2

Point

A “point” is a unit of length, commonly abbreviated as “pt”². Fonts shall be rendered so that a 72-point font shall be 1/11 primary picture height.

4.2.3

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.

4.2.4

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 6.5) shall be the baseline of the characters contained within the `Rb` element.

4.2.5

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 5.8) 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).

The `HH` field shall be encoded using two digits, with a value between 00 and 23 (inclusive). The `MM` and `SS` fields shall each be encoded using 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 5.9). For example:

² The concept of “point” comes from the printing industry.

- 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:

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

4.2.6

Instance Optional

An XML element or attribute that is not required to be present in a document instance conforming to this standard shall be identified by the clause "instance optional" and in the schemas by the attribute `minOccurs="0"` for elements, and by the attribute `use="optional"` for attributes.

A DCDM subtitle instance document does not need to contain any "instance optional" elements or attributes to be compliant with this standard. Unless otherwise specified, implementations shall decode "instance optional" items when present, and shall implement default behaviors when "instance optional" items are not present.

4.2.7

Current Font

The nearest ancestor `Font` element, or if no ancestor `Font` element is present, the font identified by the first `LoadFont` element.

5 DCDM Subtitle Structure (Normative)

A DCDM Subtitle file shall be an XML document consisting of a single `SubtitleReel` element as defined by this document.

5.1 SubtitleReel Element

The `SubtitleReel` element (Figure 2) shall be the top level container of XML data in a DCDM Subtitle file. The `SubtitleReel` element shall contain a set of elements which define the global parameters of the track file, followed by the `SubtitleList` element which shall contain the set of elements describing the individual subtitle instances.

5.1.1 IntrinsicPictureResolution Attribute [instance optional]

When present, the `IntrinsicPictureResolution` attribute of the `SubtitleReel` element shall be a string that shall indicate the resolution of the primary picture for which the PNG images, referenced by the `Image` elements present in the DCDM Subtitle file, were generated. For example, if 4K is specified, the decoded bitmap from the PNG referenced by the DCDM Subtitle file will match the expected size and position when displayed over a 4K picture.

The possible values of this attribute are outside the scope of this document and should be defined by accompanying specification (e.g., in an operational constraints document).

Note: This attribute is intended for mastering asset management purposes, not for playback processing.

5.2 Id Element

The `Id` element shall uniquely identify the reel for asset management purposes. It shall be encoded in the element body as a `urn:uuid` [RFC 4122].

5.3 ContentTitleText Element

The `ContentTitleText` element shall contain an unbounded string value encoding of the subtitle file's title (e.g., "Gone With the Wind"). It is meant strictly for display to the user.

5.3.1 Language Attribute [instance optional]

When present, the `language` attribute of the `ContentTitleText` element shall indicate the primary language represented by the text in the `ContentTitleText` element. It shall be encoded as an `xs:language` type. The default value shall be `en` (English).

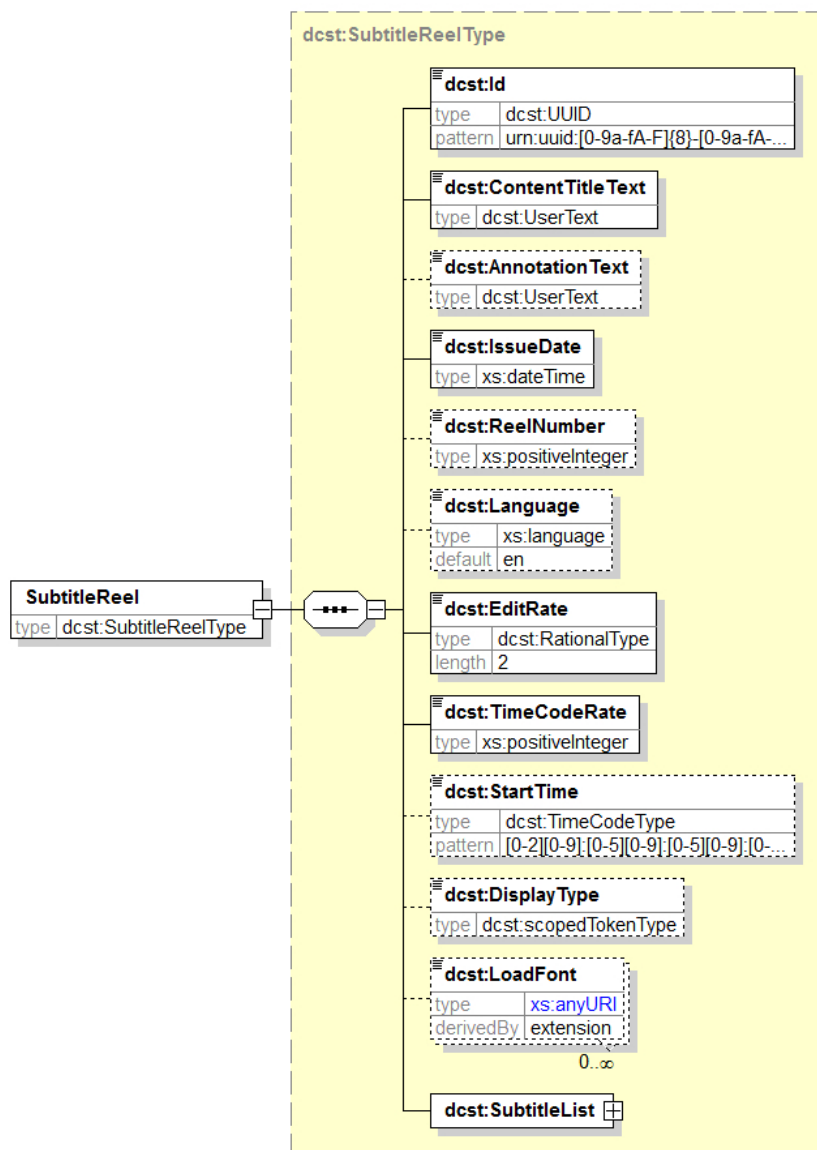


Figure 2 – DCDM Subtitle file structure. The dotted lines denote optional elements. (Informative)

5.4 AnnotationText Element [instance optional]

When present, the `AnnotationText` element shall contain an unbounded string value, which describes the subtitle reel essence. It is meant strictly for display to the user.

5.4.1 Language Attribute [instance optional]

When present, the `language` attribute of the `AnnotationText` element shall indicate the primary language represented by the text in the `AnnotationText` element. It shall be encoded as an `xs:language` type. The default value shall be `en` (English).

5.5 IssueDate Element

The `IssueDate` parameter shall indicate the time and date at which the subtitle file was created. It may be displayed to the user. It shall be encoded as an XML `xs:date` type.

5.6 ReelNumber Element [instance optional]

When present, the `ReelNumber` element shall denote the ordered placement of this file in a set of DCDM Subtitle files. The value shall only be used for informative purposes and shall not influence the reproduction of the subtitle instances. The value shall be a positive integer.

5.7 Language Element [instance optional]

When present, the `Language` element shall denote the primary language used in the text elements and/or image resources. The value shall be encoded as an `xs:language` type. The default value shall be `en` (English).

5.8 EditRate Element

The `EditRate` element shall encode the rate at which the virtual timeline progresses. It shall be expressed as a ratio of two integers giving the number of editable units per second of real time.

5.9 TimeCodeRate Element

The value of the `TimeCodeRate` element shall be equal to the value of the `EditRate` element rounded to the nearest integer. For example: If the value of the `EditRate` element is “24000 1001”, then the value of the `TimeCodeRate` element will be 24. If the value of the `EditRate` element is “200 11”, then the value of the `TimeCodeRate` element will be 18. In case where the `EditRate` ratio corresponds exactly to the mid-point of two integer values (e.g., 23.5), then the value of the `TimeCodeRate` element shall be equal to the value of the `EditRate` element rounded to the nearest higher integer (e.g., 24 in this example).

The `TimeCodeRate` element shall encode the count of subtitle editable units for each increment of the seconds field (`SS`) of a subtitle time code — see Section 4.2.5. It shall be expressed as a positive (non-zero) integer value.

5.10 StartTime Element [instance optional]

When present, the `StartTime` element shall specify the starting time of the DCDM Subtitle file’s timeline. The timeline extends forward in time in discrete increments (subtitle editable units).

The value of the `StartTime` element shall be encoded as a subtitle time code as defined in Section 4.2.5.

The default value of the `StartTime` element shall be `01:00:00:ZeroE`, where 'ZeroE' is the zero value encoded using the appropriate number of digits (e.g., '00' for a two-digit encoding, or '000' for a three-digit encoding — see Section 4.2.5 for more details).

5.11 LoadFont Element [instance optional]

When present, the `LoadFont` element shall be used to declare an OpenType [ISO/IEC 14496, Part 22] font resource for use within the DCDM Subtitle file. The font shall be identified by a `urn:uuid` value [RFC 4122] in the element's body. The mapping of `urn:uuid` values to actual font resources is beyond the scope of this document.

At no time shall the presence of a character outside the font's character set be visible in the rendered subtitle instance. For the purpose of this standard, Unicode [ISO/IEC 10646-1] control codes (the 65 characters in the ranges U+0000..U+001F and U+007F..U+009F) shall be considered outside the character set of any font (control codes shall not be displayed). Implementations shall support UTF-8 character encodings up to three bytes in length. Implementations may support four-byte UTF-8 character encoding. Note: legacy implementations might not render four-byte UTF-8 encoded characters. In the case where four-byte encodings are desired, it is strongly advised that prospective playback equipment be surveyed to assure support for this feature.

To achieve the behavior defined above, the rendering device may assume that the font conforms to the recommendation titled "First four glyphs in fonts" from [ISO/IEC 14496, Part 22], Section 7. Operation of the rendering device when using fonts otherwise constructed is not defined.

When one or more `Text` elements are present, at least one `LoadFont` element shall also be present. The first `LoadFont` element shall identify the font to be used for any text that is not explicitly styled with a `Font` element (see Section 6.4 below).

5.11.1 ID Attribute

The `ID` attribute shall be a string of one or more characters that shall define an identifier, unique within the scope of the parent `SubtitleReel` element, to be referenced by `Font` elements in the `SubtitleList`. Each `LoadFont` element in a DCDM Subtitle file shall have a distinct `ID` attribute value.

5.12 SubtitleList Element

The `SubtitleList` element shall contain the set of elements that defines subtitle instances. Any combination of `Font` and `Subtitle` elements may be present. Figure 3 illustrates this structure.

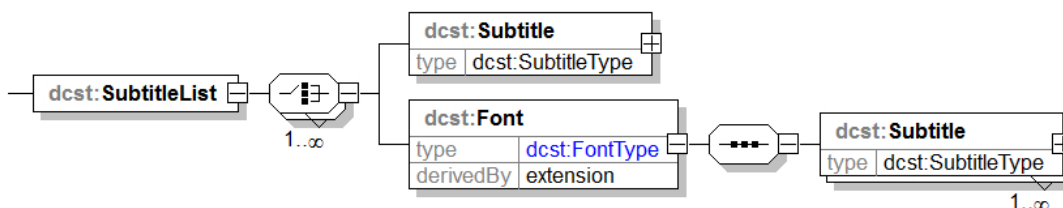


Figure 3 – subtitleList contents (Informative)

5.12.1 Ordering of Subtitle Elements

Subtitle elements shall be ordered within the `SubtitleList` element in ascending temporal order (least recent first). The `TimeIn` attribute of a given `Subtitle` element shall be equal to or greater than that of the preceding element. The first `Subtitle` element in the `SubtitleList` shall have a `TimeIn` value equal to or greater than the value of the `StartTime` element (or the default start time if the `StartTime` element is not present). This ordering shall not be affected by the presence of enclosing `Font` elements.

5.12.2 Display Precedence

Subtitle instances shall appear to be rendered in ascending temporal order. The instance having the most recent `TimeIn` value shall occlude those in the same screen area having earlier `TimeIn` values. In the case of two or more instances having the same `TimeIn` value, the instance that occurs latest in the track file shall be considered most recent.

In the case of two or more `Text` and/or `Image` elements within a given `Subtitle` instance, the element that occurs latest in the `Subtitle` instance shall be considered most recent and shall occlude the other element(s) in the same screen area.

The `Zposition` attribute and the content of any `LoadVariableZ` element shall not be used to determine display precedence.

Note: This specification allows using different `Zposition` and/or `LoadVariableZ/VariableZ` values for different `Text` and/or `Image` elements — whether located in concurrent `Subtitle` instances or within the same `Subtitle` instance. Constraints on `Zposition` and `LoadVariableZ/VariableZ` values can be defined in accompanying specification (e.g., in an operational constraints document). The display precedence rules listed above are independent of any `Zposition` or `LoadVariableZ/VariableZ` values — meaning that `Zposition` and `LoadVariableZ/VariableZ` values will not impact the definition of “more recent” element provided by the above display precedence rules.

5.13 DisplayType Element [instance optional]

When present, the `DisplayType` element shall indicate the intended use of the content of the XML document. The element value shall be a text string that is meant to be both human and machine-readable. An instance optional “`scope`” attribute with default URI value as indicated in the Sections 5.13.1 and 5.13.2 below shall determine the permissible values of the element.

When the `DisplayType` element is not present, the display affinity for that document is not defined.

5.13.1 2014 DisplayType Element

When the namespace name is `http://www.smpte-ra.org/schemas/428-7/2014/DCST`:

- the `scope` attribute shall have the following default URI value “`http://www.smpte-ra.org/schemas/428-7/2014/DCST#display-type`”.
- if the `scope` attribute is absent, or set to its default value, the content of the element shall be one of the values listed in Table 2.
- The value of the `scope` attribute shall not be “`http://www.smpte-ra.org/schemas/428-7/2010/DCST#display-type`”

Table 2 – 2014 Default scope Attribute: Possible DisplayType Values (Normative)

DisplayType Value	DisplayType Value Meaning
MainSubtitle	Subtitle content, either on-screen or off-screen (the prefix "Main" is present for historical reason only – it was used in SMPTE ST 428-7:2010)
Caption	Caption content, either on-screen or off-screen

5.13.2 2010 DisplayType Element

When the namespace name is `http://www.smpte-ra.org/schemas/428-7/2010/DCST`:

- the `scope` attribute shall have the following default URI value “`http://www.smpte-ra.org/schemas/428-7/2010/DCST#display-type`”.
- if the `scope` attribute is absent, or set to its default value, the content of the element shall be the value listed in Table 3.
- The value of the `scope` attribute shall not be “`http://www.smpte-ra.org/schemas/428-7/2014/DCST#display-type`”

Table 3 – 2010 DisplayType Default scope Attribute Value (Normative)

DisplayType Value	DisplayType Value Meaning
MainSubtitle	Subtitle content, either on-screen or off-screen.

6 Subtitle Instances (Normative)

A subtitle instance is an auxiliary image to be displayed over a series of identically sized primary pictures. Subtitle instance images are given either directly in externally referenced Portable Network Graphics [PNG] image resources, or are rendered from integral text using externally referenced OpenType font resources.

PNG images shall be authored with a 1:1 spatial relationship to the primary picture.

Note: If the primary picture is re-sized prior to projection on screen, the PNG images must be re-sized equivalently.

Subtitle instances in a DCDM Subtitle file shall be contained within the `SubtitleList` XML element (see Section 5.12). The `Subtitle` element shall be the XML container of a subtitle instance. Each `Subtitle` element shall contain one or more `Image`, `Text` or `Font` elements. `Font` elements shall be allowed to appear at any level within the `Subtitle` content hierarchy, except as a child of an `Image`, `Ruby`, `Rb`, `Rt`, `HGroup`, `Space`, `Rotate` or `LoadVariableZ` element, or as an immediate child of another `Font` element. Figure 4 illustrates this structure.

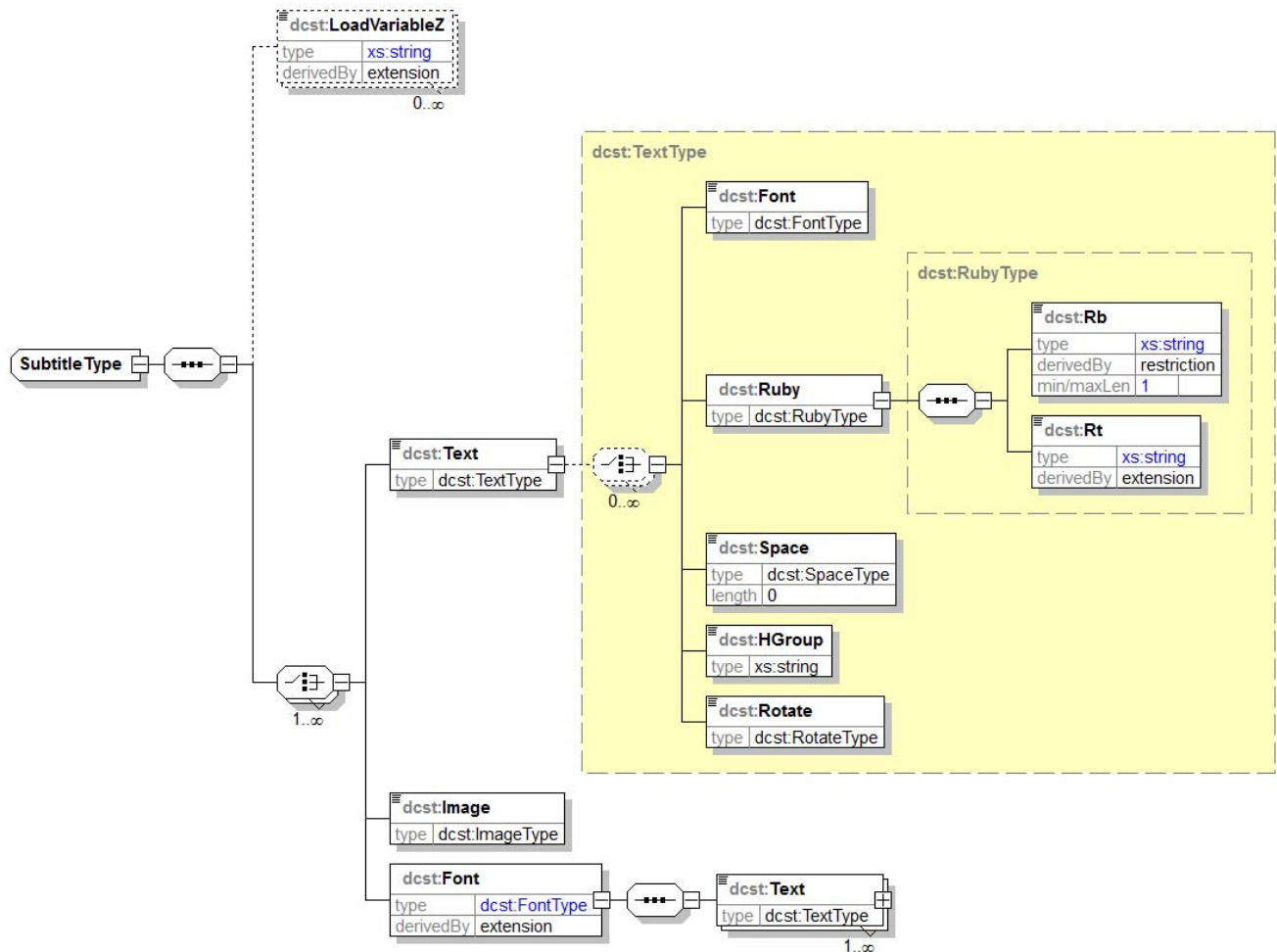


Figure 4 – Subtitle instance structure (Informative)

6.1 Subtitle Element

The `Subtitle` element shall have attributes, defined below, which specify the start time, duration, and fade characteristics of the subtitle instance. `Subtitle` elements must be ordered in ascending temporal order within the `SubtitleList` element, and could have overlapping temporal regions as defined in Section 5.12.1.

6.1.1 SpotNumber Attribute [instance optional]

When present, the `SpotNumber` attribute shall be a string value, which shall identify the instance in some other notation system, such as a spotting list.

6.1.2 TimeIn Attribute

The `TimeIn` attribute shall define the position on the DCDM subtitle timeline, expressed as hours, minutes, seconds and subtitle editable units (HH:MM:SS:E+, see Section 4.2.5), at which the subtitle instance shall begin to fade in. See additional `TimeIn` related provisions in Section 6.1.6.

Example 1:

Given the following set of DCDM Subtitle file parameters:

- The `EditRate` element value is equal to 24 1
- The `StartTime` element value is equal to 06:00:00:00
- The `TimeIn` attribute value of the first `Subtitle` element is equal to 06:00:05:00

Then, the first text or image of the DCDM Subtitle file is expected to appear 5 seconds from the start of its timeline.

Example 2:

Given the following set of DCDM Subtitle file parameters:

- The `EditRate` element value is equal to 24 1
- The `StartTime` element is not present, meaning that its default value (01:00:00:00) is used
- The `TimeIn` attribute's value of the first `Subtitle` element equals 02:00:05:00

Then, the first text or image of the DCDM Subtitle file is expected to appear 1 hour, 5 seconds from the start of its timeline.

6.1.3 TimeOut Attribute

The `TimeOut` attribute shall define the position on the DCDM subtitle timeline, expressed as hours, minutes, seconds and subtitle editable units (HH:MM:SS:E+, see Section 4.2.5), at which the subtitle instance shall complete the fade out. This value shall be greater than the `TimeIn` value of the same element. See additional `TimeOut` related provisions in Section 6.1.6.

6.1.4 FadeUpTime Attribute [instance optional]

When present, the `FadeUpTime` attribute shall define the duration, expressed as hours, minutes, seconds and subtitle editable units (HH:MM:SS:E+, see Section 4.2.5), of the fade of the subtitle instance from 100% transparent to the full specified opacity, with the possible overlap-related exception presented below in Section 6.1.6. 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' for a two-digit encoding, or '002' for a three-digit encoding — see Section 4.2.5), with the possible overlap-related exception presented below in Section 6.1.6.

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 is expected to have 0% of the full specified opacity
- Editable unit 01 is expected to have 50% of the full specified opacity
- Editable unit 02 is expected to have 100% of the full specified opacity

See additional `FadeUpTime` related provisions in Section 6.1.6.

6.1.5 FadeDownTime Attribute [instance optional]

When present, the `FadeDownTime` attribute shall define the duration, expressed as hours, minutes, seconds and subtitle editable units (HH:MM:SS:E+, see Section 4.2.5), of the fade of the subtitle instance from full specified opacity to 100% transparent, with the possible overlap-related exception presented below in Section 6.1.6. 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' for a two-digit encoding, or '002' for a three-digit encoding – see Section 4.2.5), with the possible overlap-related exception presented below in Section 6.1.6.

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 18 is expected to have 100% of the full specified opacity
- Editable unit 19 is expected to have 50% of the full specified opacity
- Editable unit 20 is expected to have 0% of the full specified opacity

See additional `FadeDownTime` related provisions in Section 6.1.6.

6.1.6 Subtitle Element Constraints

For any given `Subtitle` element, $(\text{TimeOut} - \text{FadeDownTime})$ shall be greater than or equal to $(\text{TimeIn} + \text{FadeUpTime})$.

When two subtitle instances have overlapping time windows, their respective `FadeUpTime` and `FadeDownTime` attributes values (or their default values if absent) may be considered by the rendering device to be equal to "00:00:00:ZeroE" (where 'ZeroE' is the zero value encoded using the appropriate number of digits — see Section 4.2.5), even if the fading operations are specified outside of the overlapping period.

When the time window of a subtitle instance does not overlap the time window of any other subtitle instance, its `FadeUpTime` and `FadeDownTime` attribute values (or their default values if absent) shall be respected.

Note: The overlapping of the time window (defined by the `TimeIn` and `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., in an 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.

6.2 Image Element

When present, the `Image` element shall identify an external image resource to be displayed over the primary picture. The image resource shall be identified by a `urn:uuid` value [RFC 4122] in the element's body. The mapping of `urn:uuid` values to actual image resources is beyond the scope of this document. Images shall be encoded in Portable Network Graphics [PNG] format, and shall have a size no larger than that of the respective primary picture. Transparent regions of images shall be rendered transparently. If there is a portion of the content from another `Text` or `Image` element located underneath such transparent region, it shall be visible – meaning that the transparent region shall not show only the primary picture. The image position over the primary picture shall be determined by the `Image` element attributes values.

Content creators should make sure that all the content enclosed within the `Image` element fits entirely within the primary picture boundary, especially when the `VariableZ` and/or `Zposition` attribute(s) are/is present (in which case both left- and right-eye images shall fit entirely within this boundary).

6.2.1 Halign Attribute [instance optional]

When present, the `Halign` attribute shall identify the intended horizontal alignment mode, which shall be the edge or center of the image and primary picture to be used when calculating position with `Hposition`. The value shall be one of [`left`, `center`, `right`]. The default value shall be `center`.

6.2.2 Hposition Attribute [instance optional]

When present, the `Hposition` attribute shall specify the distance between the image edge or center and the edge or the center, respectively, of the primary picture. The distance shall be expressed as a percentage of primary picture width. Table 4 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).

The rendering engine shall ignore the value of any stereoscopic attribute (e.g., `Zposition` or `VariableZ`) or element (e.g., `LoadVariableZ`) when calculating the horizontal position of a subtitle instance according to Table 4.

Table 4 – Halign and Hposition interaction for both Text and Image elements (Normative)

<i>Halign Value</i>	<i>Hposition Interpretation</i>
<code>left</code>	For <code>Image</code> elements and for horizontal <code>Text</code> elements: the horizontal distance between the left primary picture edge and the left text bounding box edge or image edge. For vertical <code>Text</code> elements: the horizontal distance between the left primary picture edge and the center of each character bounding box or <code>Hgroup</code> element's value. Negative values shall not be used.
<code>center</code>	The horizontal distance between the center of the primary picture and the center of the text bounding box or image. Negative values shall indicate left offset, positive values shall indicate right offset.
<code>right</code>	For <code>Image</code> elements and for horizontal <code>Text</code> elements: the horizontal distance between the right primary picture edge and the right text bounding box edge or image edge. For vertical <code>Text</code> elements: the horizontal distance between the right primary picture edge and the center of each character bounding box or <code>Hgroup</code> element's value. Negative values shall not be used.

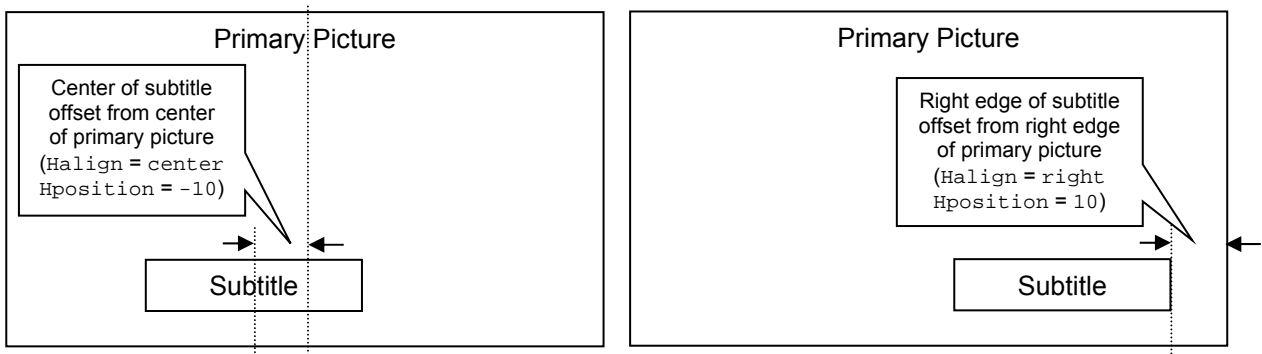


Figure 5 – Halign/Hposition Usage Illustration (Informative)

6.2.3 Valign Attribute [instance optional]

When present, the `Valign` attribute shall identify the intended vertical alignment mode, which shall determine the edge or center of the image and primary picture to be used when calculating position with `Vposition`. The value shall be one of [`top`, `center`, `bottom`]. The default value shall be `center`.

6.2.4 Vposition Attribute [instance optional]

When present, the `Vposition` attribute shall specify the distance between the image edge or center and the edge or the center, respectively, of the primary picture. The distance shall be expressed as a percentage of primary picture height. Table 5 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 5 – `valign` and `vposition` interaction for Image elements (Normative)

<i>Valign Value</i>	<i>Vposition Interpretation</i>
<code>top</code>	The vertical distance between the top primary picture edge and the top image edge. Negative values shall not be allowed.
<code>center</code>	The vertical distance between the center of the primary picture and the center of the image. Negative values shall indicate top offset, positive values shall indicate bottom offset.
<code>bottom</code>	The vertical distance between the bottom primary picture edge and the bottom image edge. Negative values shall not be allowed.

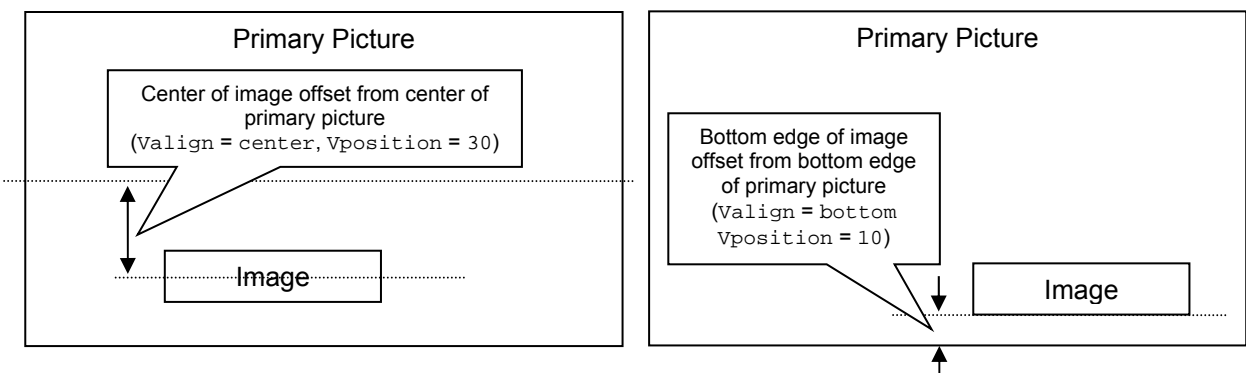


Figure 6 – `valign/vposition` Usage Illustration for Image Elements (Informative)

6.2.5 Zposition Attribute [instance optional]

When present, the `Zposition` attribute shall provide a value that specifies the horizontal distance between the “left eye” image center and the “right eye” image center - in order to generate a stereoscopic effect. The `Zposition` attribute value shall be expressed as a percentage of the primary picture width. The value shall be encoded as a signed decimal number having zero or more decimal places. Positive numbers shall move the image away from the viewer. Negative numbers shall bring the image closer to

the viewer. The default value shall be 0 (zero), which shall correspond to a perceived display in the plane of the screen.

In order to determine the positions of the left- and right- eye image parts, 50% of the `Zposition` attribute value shall be subtracted from the `Hposition` attribute value to derive the left-eye image position and 50% shall be added to the `Hposition` attribute value to derive the right-eye image position. Both the left- and right-eye images shall be displayed using the same vertical positioning information as defined by the associated `Valign` and `Vposition` attribute values (or their respective default values if absent).

The entire PNG image identified by the `Image` element shall be displayed for each of the left- and right-eye image positions.

Example 1:

`Zposition` = 2 (intended to produce a perceived display behind the screen plane)
`Hposition` = 0

left-eye `Hposition` = `Hposition` - (`Zposition` / 2) = -1
 right-eye `Hposition` = `Hposition` + (`Zposition` / 2) = 1

Example 2:

`Zposition` = -2 (intended to produce a perceived display in front of the screen plane)
`Hposition` = 0

left-eye `Hposition` = `Hposition` - (`Zposition` / 2) = 1
 right-eye `Hposition` = `Hposition` + (`Zposition` / 2) = -1

6.2.6 VariableZ Attribute [instance optional]

When present, the `VariableZ` attribute shall identify a vector of value-duration pair(s) (see `LoadVariableZ`, Section 6.12) to be used to generate a stereoscopic effect for the PNG image identified by the `Image` element. The `VariableZ` attribute value shall match the `ID` attribute value defined by a `LoadVariableZ` element present within the same `Subtitle` element. Support for control of stereoscopic rendering via the `VariableZ` attribute and `LoadVariableZ` element is optional.

When the `VariableZ` attribute is present, the `Zposition` attribute shall also be present in the same `Image` element. Devices that do not support control of stereoscopic rendering via `VariableZ` attribute and `LoadVariableZ` element shall instead use the value of the `Zposition` attribute in order to generate a stereoscopic effect. Devices that make use of the `VariableZ` attribute and its associated `LoadVariableZ` element shall ignore the `Zposition` attribute located in the same `Image` element.

When the `VariableZ` attribute is not present in an `Image` element, the implementation shall use the `Zposition` attribute value (or its default value if absent) of this `Image` 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.

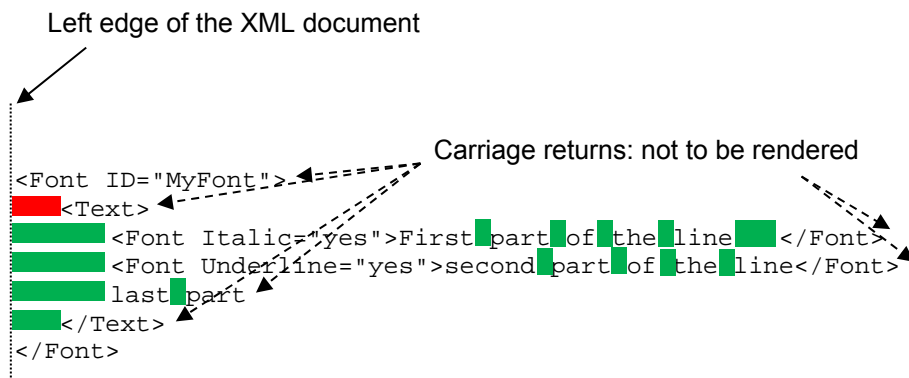
6.3 Text Element

When present, the `Text` element shall encapsulate a text string to be rendered into the subtitle instance for display over the primary picture. The `Text` element shall be permitted to contain child elements (*i.e.*, `Font`, `Ruby`, `Space`, `HGroup` and `Rotate`) that further specify properties of the rendered text string. Any character data inside the `Text` element and any of its child element(s), except any control code character (as identified in Section 5.11), shall be combined to form the displayed string. The displayed string may contain zero characters. The text string should be short enough to fit in a single line (or column - for vertically written text) on screen using the given font and position parameters – otherwise, the text string will only be partially visible on screen. Multiple lines (or columns - for vertically written text) of text to be displayed on screen simultaneously shall be encoded as sibling `Text` elements under the same `Subtitle` element.

The position and direction of the rendered text shall be controlled by attributes of the `Text` element.

All space characters present inside the content of a `Text` element shall be rendered, even when located in a `Font`, `HGroup` or `Rotate` child element – meaning that consecutive space characters shall not be collapsed. Space character(s) located outside of a `Text` element shall not be rendered. Also, as specified in Section 5.11, control codes (*e.g.*, carriage return, line feed, tab, etc...) shall not be rendered.

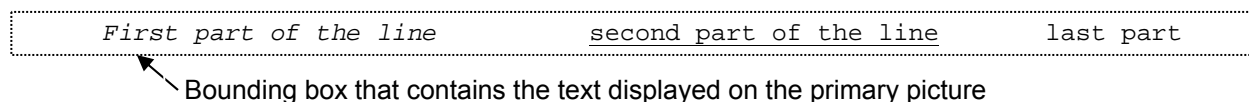
The example XML fragment below illustrates the above rules. Note: web browsers, XML file viewers and other authoring programs often employ character processing that will obscure the actual sequence of characters present in the XML, *e.g.*, by collapsing line feed, carriage return and space characters, or by inserting space to align elements for easier viewing. Users are advised to use a plain text editor or a “raw source” mode when viewing subtitle XML files so that the actual sequence of characters can be observed.



Legend:

- Space characters highlighted in green will be rendered
- Space characters highlighted in red will not be rendered

The rendered result of the above XML example is expected to produce something like:



It is the responsibility of the content creator to assure that all content enclosed within the `Text` element (taking into account all transformations specified by its attributes – including stereoscopic attributes – and child elements) will fit entirely within the primary picture boundary.

6.3.1 Halign Attribute [instance optional]

When present, the `Halign` attribute shall identify the intended horizontal alignment mode, which shall determine the edge or center of the text bounding box and primary picture to be used when calculating position with `Hposition`. The value shall be one of [`left`, `center`, `right`]. The default value shall be `center`.

6.3.2 Hposition Attribute [instance optional]

When present, the `Hposition` attribute shall specify the distance between the text bounding box edge or center from the edge or center of the primary picture as defined in Table 4. The distance shall be expressed as a percentage of primary picture width. Table 4 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).

The rendering engine shall ignore the value of any stereoscopic attribute (e.g., `Zposition` or `VariableZ`) or element (e.g., `LoadVariableZ`) when calculating the horizontal position of a subtitle instance according to Table 4.

6.3.3 Valign Attribute [instance optional]

When present, the `Valign` attribute shall identify the intended vertical alignment mode, which shall determine the edge or center of the primary picture to be used when calculating position with `Vposition`. The value shall be one of [`top`, `center`, `bottom`]. The default value shall be `center`.

6.3.4 Vposition Attribute [instance optional]

When present, the `Vposition` attribute shall specify the distance of the text's baseline from the edge or the center of the primary picture. The distance shall be expressed as a percentage of primary picture height. Table 6 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 6 – 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 the value of the <code>Direction</code> attribute of a <code>Text</code> element is <code>ttb</code> or <code>btt</code>, the baseline used shall be the baseline for the character displayed on top. Negative values shall not be used.</p> <p>Note: Specifying <code>Valign="top" 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 shall indicate top offset, positive values shall indicate bottom offset. When the value of the <code>Direction</code> attribute of a <code>Text</code> element is <code>ttb</code> or <code>btt</code>, then:</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 character string; – in case of an even number of characters, the baseline for the character located just before the center 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 the value of the <code>Direction</code> attribute of a <code>Text</code> element is <code>ttb</code> or <code>btt</code>, the baseline used shall be the baseline for the first character displayed on the bottom. Negative values shall not be used.</p> <p>Note: Specifying <code>Valign="bottom" Vposition="0"</code> will result in the descender of certain characters, like 'p', being drawn off the bottom of the screen.</p>

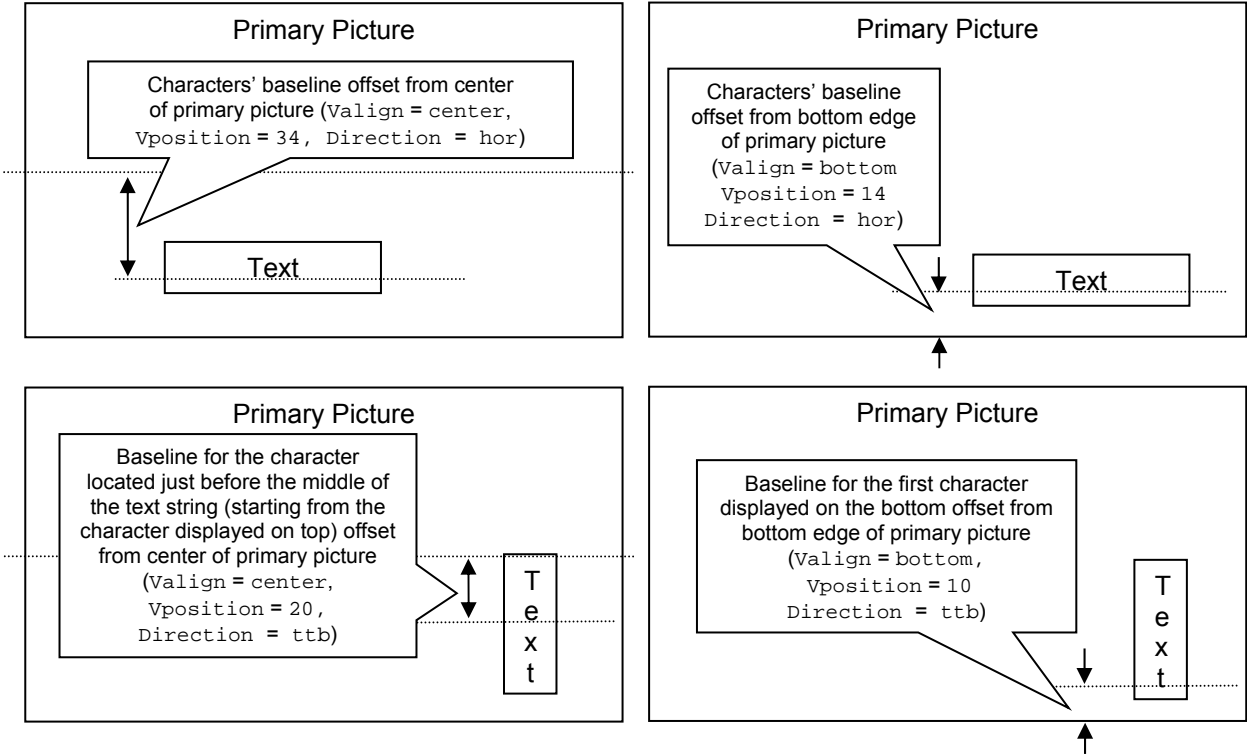


Figure 7 – Valign/Vposition Usage Illustration for Text Elements (Informative)

Note: DCDM Subtitle files authored in accordance with previous versions of this standard could have been created with a slightly different interpretation of `Valign` and `Vposition` than what is presented above, *i.e.*, based on the text border (or *bounding box*) instead of the baseline. The vertical position of text elements defined within such files will be slightly different when rendered by an implementation compliant with this revised standard.

6.3.5 Direction Attribute [instance optional]

When present, for horizontally rendered text (*i.e.*, for attribute's values `ltr`, `rtl` and `hor`), the `Direction` attribute shall specify the base writing direction. For vertically rendered text (*i.e.*, for attribute values `tbt` and `btt`), this attribute shall specify the direction in which the text shall be rendered. One of the values in Table 7 shall be used. The default value shall be `ltr`.

Table 7 – Direction attribute values (Normative)

<i>Direction Value</i>	<i>Direction of Rendered Text</i>
<code>ltr</code>	Horizontal. Text shall be rendered left to right. The Unicode Bidirectional Algorithm should be used. Note: Some legacy equipment will not support the Unicode Bidirectional Algorithm, while new implementations might support it.
<code>rtl</code>	Horizontal. Text shall be rendered right to left. The Unicode Bidirectional Algorithm should be used. Note 1: When using the Unicode Bidirectional Algorithm, implementations must not invert the character sequence in addition to application of the algorithm. Note 2: Some legacy equipment will not support the Unicode Bidirectional Algorithm, and would likely invert the character string prior to rendering.
<code>tbt</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 (Note: The <code>btt</code> value is deprecated, and should not be used in new DCDM Subtitle files).
<code>hor</code>	Horizontal. The Unicode Bidirectional Algorithm shall be used.

When Unicode right-to-left characters are 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 (*i.e.*, in the order of the character stream), in the direction specified by the value of the `Direction` attribute.

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 — unless modified by the use of the `Rotate` element.
- the horizontal center of each character's bounding box (or of the `HGroup` element's value) shall be at the position specified by the value of the `Hposition` attribute — see Table 4.

Note: The underlying thinking behind the definition of the `Direction` attribute given above is to determine the horizontal rendering direction using only the Unicode Bidirectional Algorithm, but without breaking legacy implementations that do not support it. New implementations are expected to rely only on the Unicode Bidirectional Algorithm when rendering horizontal text.

6.3.6 Zposition Attribute [instance optional]

When present, the `Zposition` attribute shall provide a value which specifies the horizontal distance between the “left eye” text center and the “right eye” text center in order to generate a stereoscopic effect. The `Zposition` attribute value shall be expressed as a percentage of the primary picture width. 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), which shall correspond to a perceived display in the plane of the screen.

In order to determine the positions of the left- and right-eye text parts, 50% of the `Zposition` value shall be subtracted from the `Hposition` value to derive the left-eye item position and 50% shall be added to the `Hposition` value to derive the right-eye item position. Both the left- and right-eye text item parts shall be displayed using the same vertical positioning information as defined by the associated `Valign` and `Vposition` attribute values (or their respective default values if absent), and shall use the same text rendering direction as defined by the respective `Direction` attribute (or its default value if absent).

All the content enclosed within the `Text` element shall be displayed for each of the left- and right-eye text item positions.

Note: See Examples in Section 6.2.5.

6.3.7 VariableZ Attribute [instance optional]

When present, the `VariableZ` attribute shall identify a vector of value-duration pairs (see `LoadVariableZ`, Section 6.12) to be used to generate a stereoscopic effect for all content enclosed within the `Text` element. The `VariableZ` attribute value shall match the `ID` attribute value defined by a `LoadVariableZ` element present within the same `Subtitle` element. Support for control of stereoscopic rendering via `VariableZ` attribute and `LoadVariableZ` element is optional.

When the `VariableZ` attribute is present, the `Zposition` attribute shall also be present in the same `Text` element. Devices that do not support control of stereoscopic rendering via `VariableZ` attribute and `LoadVariableZ` element shall instead use the value of the `Zposition` attribute in order to generate a stereoscopic effect. Devices that make use of 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, the implementation 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.

6.4 Font Element

When present, the `Font` element shall identify a previously loaded font resource (see `LoadFont`, Section 5.11 above) which shall be used to control the rendering of all text content enclosed within the `Font` element. When rendering text, the `Font` element closest in the `SubtitleReel` hierarchy (*i.e.*, the nearest ancestor) shall have precedence. Attributes of a `Font` element shall further define font properties. Attributes not given in a particular `Font` element instance shall be inherited from the nearest ancestor `Font` element.

The `Font` element shall be permitted to contain `Subtitle` or `Text` elements. `Font` elements shall also be permitted to contain text data in the case where the particular `Font` element has a `Text` element ancestor - and in this case, the `Font` element shall not contain any `Subtitle` or `Text` element as descendent.

A `Font` element shall have attributes as defined in the following paragraphs. Within the scope of a given `Font` element, the presence of additional `Font` elements shall be permitted, with the expectation that such elements would be intended to provide different `Font` attribute values — e.g., in the case of a `Font` element embedded within a `Text` element, itself embedded within another `Font` element.

The example below illustrates this principle. The example assumes that there is no ancestor `Font` element embedding the first `Font` element:

```
<Font Effect="shadow" Script="super" Underline="yes">
  <Text>In this area: Effect=shadow, Script=super, Underline=yes, Italic=no
    <Font Effect="none" Script="sub" Italic="yes">
      In this area: Effect=none, Script=sub, Underline=yes, Italic=yes
    </Font>
  In this area: Effect=shadow, Script=super, Underline=yes, Italic=no
</Text>
</Font>
```

6.4.1 ID Attribute [instance optional]

When present, the `ID` attribute shall identify the font to be used when rendering the element content. The value shall match the value of the `ID` attribute of a `LoadFont` element contained within the same document instance. If this attribute is missing, the current font shall be used (see Section 4.2.7).

6.4.2 Script Attribute [instance optional]

When present, the `script` attribute shall instruct 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 8 shall be used. The default value shall be `normal`.

If the value of the `Direction` attribute of a `Text` element is `ttb` or `btt`, the `Script` attribute shall be ignored within the scope of the `Text` element — meaning that its default value shall be used.

Note: DCDM Subtitle files authored in accordance with previous versions of this standard could have a `Script` attribute present when the value of the `Direction` attribute of an ancestor `Text` element equals `ttb` or `btt`. Implementations compliant with this revised standard are allowed to disregard the `Script` attribute in such a case.

The baseline for the text subject to superscript (*i.e.*, when the value of the `Script` attribute is `super`) shall be above the baseline of the characters as determined prior to applying the `Script` attribute. The baseline for the text subject to subscript (*i.e.*, when the value of the `Script` attribute is `sub`) shall be below the baseline of the characters as determined prior to applying the `Script` attribute. In both cases (meaning when `Script` attribute's value equals either `super` or `sub`), the absolute distance between the two baselines should be 0.5 em, and it shall not be less than 0.3 em, and not more than 0.7 em. This range is provided in order to accommodate legacy implementations (new implementations should use 0.5 em).

When the value of the `Script` attribute is `normal`, no offset shall be applied.

All other attributes of the `Font` element (*i.e.*, `Weight`, `Italic`, `Underline`, `Spacing`, `AspectAdjust`, `Color`, `Effect`, `Size`, `EffectColor`, `EffectSize` and `Feather`) shall be applied to the text without regard to the value of the `Script` attribute.

When the value of the `Script` attribute is `super` or `sub`, the size of the superscript or subscript text, respectively, shall be 0.6 em, *i.e.*, 60% of the text size as determined prior to the application of the `Script` attribute. When the value of the `Script` attribute is `normal`, the size of the text shall be 1 em, *i.e.*, the current size. Note: The size of any group of characters can be adjusted as desired by enclosing those characters within a `Font` element having the appropriate `Size` attribute value. The following example illustrates the use of `Script`:

```
<Font ID="MyFont">
  <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 8 – 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

6.4.3 Effect Attribute [instance optional]

When present, the `Effect` attribute shall enable a special effect to alter the contrast of the rendered text against the primary picture. One of the values in Table 9 shall be used. The default value shall be `shadow`.

Note: The default effect value has been changed from previous versions of this standard, which used `none`. As a result, documents authored using the namespaces <http://www.smp-te-ra.org/schemas/428-7/2010/DCST> and <http://www.smp-te-ra.org/schemas/428-7/2007/DCST> may not appear as expected when rendered by an implementation that is compliant with this standard.

Table 9 – Effect attribute values (Normative)

<i>Effect Value</i>	<i>Effect Description</i>
border	The text shall be rendered with a colored border conforming to the exterior contours of each character. The color of this border shall be determined by the value of the <code>EffectColor</code> attribute. The thickness of the border shall be determined by the value of the <code>EffectSize</code> attribute. The border may be feathered or not, depending on the value of the <code>Feather</code> attribute.
shadow	The text shall be rendered with a colored drop shadow behind each character, which shall be placed diagonally to the bottom and right, or bottom and left of each character as preferred by the implementation. The direction of the drop shadow shall be consistent for all characters in any given <code>Text</code> element. The color of the shadow shall be determined by the value of the <code>EffectColor</code> attribute. The thickness of the drop shadow shall be determined by the value of the <code>EffectSize</code> attribute. The drop shadow may be feathered or not depending on the value of the <code>Feather</code> attribute.
none	No effect shall be applied to the rendered text.

If the character is rotated (by use of the `Rotate` element — see Section 6.11), the shadow effect shall remain on the bottom and right (or bottom and left, see description in Table 9) of the character after rotation.

6.4.4 Italic Attribute [instance optional]

When present, the Italic attribute shall enable the oblique rendering of the font glyphs. The value shall be one of `yes`, `no`, `left` or `right`. The default value shall be `no`.

The values `yes`, `left` and `right` shall enable oblique rendering as defined below, while the value `no` shall disable oblique rendering.

When oblique rendering is enabled, it shall be determined by a skew transformation along the horizontal axis for horizontal text and along the vertical axis for vertical text (to be applied to each character individually) and the absolute angle value should be 12 degrees. This value shall however not be more than 14 degrees, and shall not be less than 10 degrees. This range is provided in order to accommodate legacy implementations - new implementations should use 12 degrees. The actual oblique direction to apply in each case is defined in Table 10.

Table 10 – Italic Attribute's Value and Oblique Direction (Normative)

<i>Italic Attribute's Value</i>	<i>Oblique Direction</i>
yes	The oblique direction is not defined: it may be to either the left or to the right and may depend on the rendered text (left-to-right, right-to-left, top-to-bottom or bottom-to-top), depending on the implementation.
no	Oblique rendering shall not be applied.
left	For horizontal text, the top of each character shall be slightly to the left of the bottom of the character. For vertical text, the left of each character shall be slightly above the right of the character.
right	For horizontal text, the top of each character shall be slightly to the right of the bottom of the character. For vertical text, the right of each character shall be slightly above the left of the character.

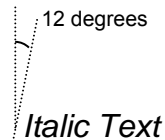


Figure 8 – Italic Attribute Usage Illustration: Value Equals right, Horizontal Text (Informative)

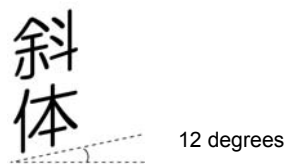


Figure 9 – Italic Attribute Usage Illustration: Value Equals right, Vertical Text (Informative)

Note: Specific oblique rendering different from the description above can be accomplished through the use of the `Image` element.

6.4.5 Underline Attribute [instance optional]

When present, the `Underline` attribute shall enable underlining of characters for horizontal text only. The value shall be one of `yes` or `no`. The default value shall be `no`.

When the value of the `Direction` attribute of a `Text` element is `ltr`, `rtl` or `hor` (or when its default value is in effect), and the value of the `Underline` attribute is `yes`, a line shall be rendered below the character(s). If the current font resource provides information about position and thickness of this line, rendering devices should use it. If no such information is provided in the font resource, or if the device is not able to take such information into account, then the line should be located 0.1 em below the character's baseline and its thickness should be 0.1 em. An `Underline` attribute value of `no` shall have no effect on the rendered text (*i.e.*, an underline shall not be drawn).

The following attributes of the nearest ancestor `Font` element shall be applied to the line added by the `Underline` attribute: `Size`, `Color`, `Effect` and `EffectColor`. All other attributes of the `Font` element shall not be applied when rendering the underline.

When the value of the `Direction` attribute of a `Text` element is `ttb` or `btt`, the `Underline` attribute shall be ignored, *i.e.*, no line shall be drawn below the character(s).

Note: A line added by use of this attribute can go through character descenders, if any. Specific underline rendering different from the description above can be accomplished through the use of the `Image` element.

6.4.6 Weight Attribute [instance optional]

When present, the `Weight` attribute shall specify the weight of the characters. The value shall be one of `bold` or `normal`. The default value shall be `normal`.

A value of `bold` shall cause the respective characters to be rendered with greater thickness, *i.e.*, with the strokes comprising the glyphs having increased width. Bold characters should be 5% thicker than non-bold characters, but shall not be more than 8% thicker, and shall not be less than 2% thicker (this range is provided in order to accommodate legacy implementations - new implementations should use 5%). A value of `normal` shall have no effect, *i.e.*, characters shall be rendered with normal weight.

Note: Use of this attribute could increase the length and the height of the text string as each character would be slightly thicker – while preserving the spacing between characters.

6.4.7 Color Attribute [instance optional]

When present, the color attribute shall define the color to be used for rendering text characters and effects specified to use the same color as the text, e.g., underline. The color shall be encoded as a string of eight (8) hexadecimal characters in the following format: AARRGGBB, where AA is an eight (8) bit alpha-blend value, and RR, GG, and BB represent eight (8) bits of red, green and blue, respectively. The default value shall be FFFFFFFF (opaque white).

Note: If the text is not fully opaque (*i.e.*, meaning if the alpha-blend value is not FF), blending can happen between the character and the background and/or its effect (drop shadow or border) when present.

6.4.8 EffectColor Attribute [instance optional]

When present, the EffectColor attribute shall define 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.

6.4.9 Size Attribute [instance optional]

When present, the Size attribute shall define the size of the rendered characters. It shall be expressed in decimal units of points. The default value shall be 42.

6.4.10 AspectAdjust Attribute [instance optional]

When present, the AspectAdjust attribute shall modify the aspect ratio of the rendered characters. This attribute can be used to increase or decrease the width of each rendered character without affecting height. This effect can be used to slightly lengthen or shorten a character string. The value shall be encoded as a decimal number. 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. The value shall be not less than 0.25 and not more than 4.0. The default value shall be 1.0.

6.4.11 Spacing Attribute [instance optional]

When present, the Spacing attribute shall provide additional spacing between the rendered characters. The value shall be encoded as a number in decimal units of em. 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. Typical usage would use values less than 1. Negative values are allowed but should be used with care as characters could overlap. Spacing shall not be less than -1.0. The default value shall be 0.0.

6.4.12 EffectSize Attribute [instance optional]

When present, the EffectSize attribute shall define the thickness of the shadow or border effect (as defined in Section 6.4.3). The EffectSize value shall be specified in decimal units of em. Negative values shall not be used. The default value shall be 0.01.

Note: A particularly small value of EffectSize could result in an effect that is not visible in the rendered text. Also, because the EffectSize value is expressed in "em", the effect will be proportional to the current font size, and therefore when using small font sizes the resulting effect may also not be visible.

The EffectSize attribute, if present, may not be rendered by some implementations.

6.4.13 Feather Attribute [instance optional]

When present, the `Feather` attribute shall specify that the respective effect (shadow or border effect as defined in Section 6.4.3) be rendered with a feathered edge so as to product a smooth transition between the effect and the primary picture. The value shall be one of `yes` or `no`. The default value shall be `no`. A value of `yes` shall indicate that the effect (when present) shall be feathered. A value of `no` shall indicate that the effect (when present) shall not be feathered. In the case where no effect is applied to the character, the `Feather` attribute shall be ignored.

When enabled, the feathering detail shall be implemented linearly from the specified opacity through fully transparent, with the specified opacity on the effect edge adjoining the character, and full transparency on the edge opposite to the character.

The `Feather` attribute, if present, may not be rendered by some implementations.

6.5 Ruby Element

When present, the `Ruby` element shall contain *ruby* (annotation, or context) *characters* and associated *base characters*, used to present some languages, e.g., 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, with the exception of space characters as constrained below. The `Ruby` element shall be an immediate child of a `Text` element.

The `Rb` element shall contain the *base characters*, to be annotated by the *ruby characters* contained within the sibling `Rt` element.

The `Ruby` element constitutes an integral container that shall be positioned according to the `Hposition`, `Vposition`, `Zposition` and `VariableZ` attributes of the ancestor `Text` element. This means that both `Rb` and `Rt` elements comprising the `Ruby` container shall be assembled (taking into account any adjustment due to the `Size`, `Position`, `Offset`, `Spacing` and `AspectAdjust` attributes of the `Rt` element) before aligning the resulting `Ruby` container as specified by the respective `Hposition`, `Vposition`, `Zposition` and `VariableZ` attributes. Section 4.2.4 defines the baseline to use for positioning the `Ruby` container.

The ruby characters shall be rendered offset from the base characters. The offset shall be determined by the value of the `Position` attribute, and adjusted by the `Offset` attribute (both defined below):

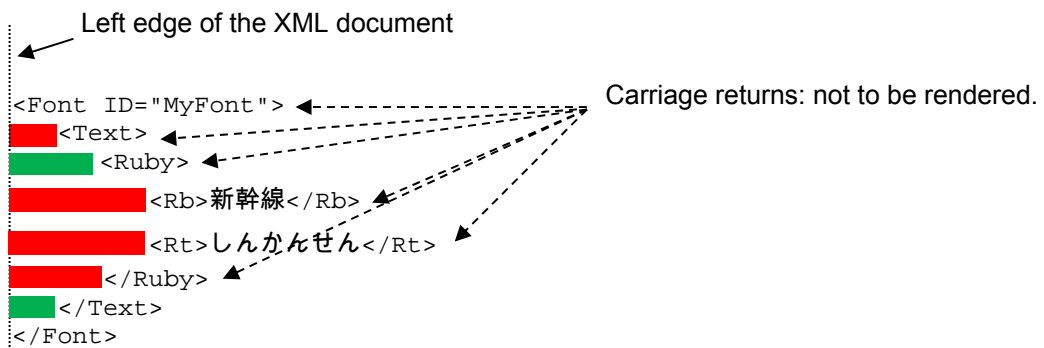
1. For horizontal Ruby text:
 - a. When the value of the `Position` attribute is `before`:
The baseline of the ruby characters shall be above the baseline of the respective base characters by a distance of 1.0 em, plus any `Offset` attribute adjustment.
 - b. When the value of the `Position` attribute is `after`:
The baseline of ruby characters shall be below the baseline of the respective base characters by a distance equal to the value of the `Size` attribute of the `Rt` element (or its default), plus any `Offset` attribute adjustment.
2. For vertical Ruby text:
 - a. If the value of the `Position` attribute is `before`:
The amount of space between the left edge of the bounding box of the rendered ruby characters and the right edge of the bounding box of the respective base characters shall be equal to the value of the `Offset` attribute or its default.

- b. If the value of the `Position` attribute is after:
 The amount of space between the right edge of the bounding box of the rendered ruby characters and the left edge of the bounding box of the respective base characters shall be equal to the value of the `Offset` attribute or its default.

Rules for rendering space characters presented in Section 6.3 shall apply inside a `Ruby` element, with the following constraints:

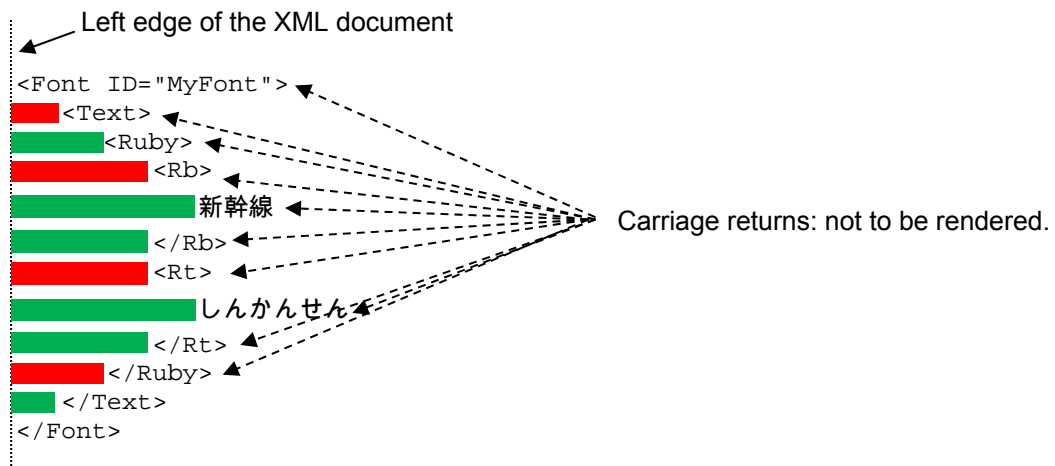
- Any space characters located inside the `Ruby` element but outside of the `Rb` and `Rt` elements shall not be rendered.
- All space characters located inside the `Rb` or `Rt` element shall be rendered per Section 6.3.

The examples below illustrate the rules for rendering space characters inside a `Ruby` element:



Legend:

- Space characters highlighted in green will be rendered
- Space characters highlighted in red will not be rendered



Legend:

- Space characters highlighted in green will be rendered
- Space characters highlighted in red will not be rendered

6.6 Rb Element

The `Rb` element shall contain the set of base characters with which the ruby characters in the sibling `Rt` element will be associated. 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: The 2007 and 2010 schemas do allow the `Rb` element to be empty. See Section 8.

6.7 Rt Element

The `Rt` element shall contain the ruby (annotation, or context) characters to be associated with the base characters contained within the sibling `Rb` element. No elements shall be present inside an `Rt` element. The `Rt` element shall have attributes as defined in the following paragraphs. The ruby characters shall be rendered with center alignment, horizontally or vertically (for horizontal or vertical text, respectively), with respect to the base characters (the content of the sibling `Rb` element).

6.7.1 Size Attribute [instance optional]

When present, the `Size` attribute shall indicate the size of the rendered ruby characters. The value shall be specified in decimal units of em and shall be a positive, non-zero number (*i.e.*, greater than 0). The default value of the `Size` attribute shall be 0.5, *i.e.*, half the size of the current font.

6.7.2 Position Attribute [instance optional]

When present, the `Position` attribute shall indicate the position of the rendered ruby characters with respect to the base characters. The value shall be one of `before` or `after`. A value of `before` shall indicate that the ruby characters shall be rendered above the base characters when the text direction is horizontal or to the right of the base characters when the text direction is vertical. A value of `after` shall indicate that the ruby characters shall be rendered below the base characters when the text direction is horizontal or to the left of the base characters when the text direction is vertical. The default value of the `Position` attribute shall be `before`.

6.7.3 Offset Attribute [instance optional]

When present, the `Offset` attribute shall indicate additional spacing between the set of rendered ruby characters and the respective set of base characters. The `Offset` attribute allows for adjusting the position of the ruby text above or below the default position (defined in Section 6.5 above). The offset shall be specified in decimal units of em. Offset shall not be less than -1.0. The default `Offset` value shall be 0.0.

A positive value shall move the ruby characters farther from the base characters, and a negative value shall move the ruby characters closer to the base characters. Note that larger negative values could cause the ruby text to overlap with the base text.

6.7.4 Spacing Attribute [instance optional]

When present, the `Spacing` attribute shall indicate additional spacing between the rendered ruby characters. The value shall be specified in decimal 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 slightly lengthen or shorten the ruby character string. Typical usage would have values less than 1. Negative values are allowed but should be used with care as characters could overlap. The value of `Spacing` shall not be less than -1.0. Note: spacing may also be specified in the `Font` element (see Section 6.4.11 above). The effect of both shall be cumulative. The default `Spacing` value shall be 0.0.

6.7.5 AspectAdjust Attribute [instance optional]

When present, the `AspectAdjust` attribute shall be 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 slightly lengthen or shorten the character string. Values greater than 1.0 shall 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 not be less than 0.25 and not more than 4.0. The default value of `AspectAdjust` shall be 1.0.

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="0.0">しんかんせん</Rt>
</Ruby>
</Text>
```

Is expected to produce something like this:

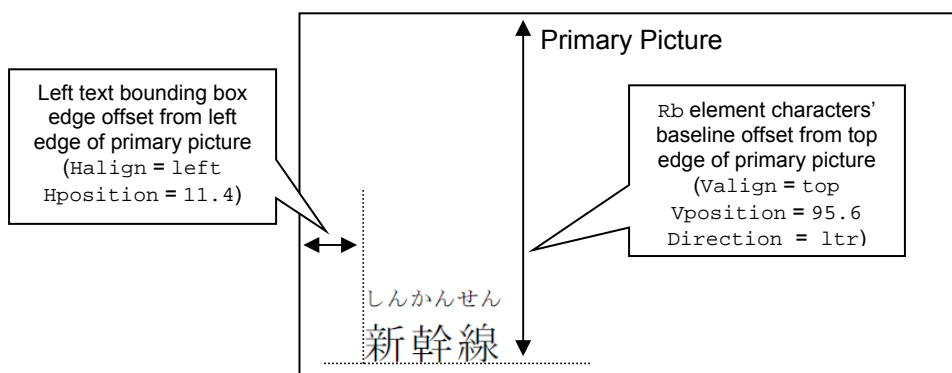


Figure 10 – Ruby Example 1 Illustration (Informative)

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>

```

Is expected to produce something like this:

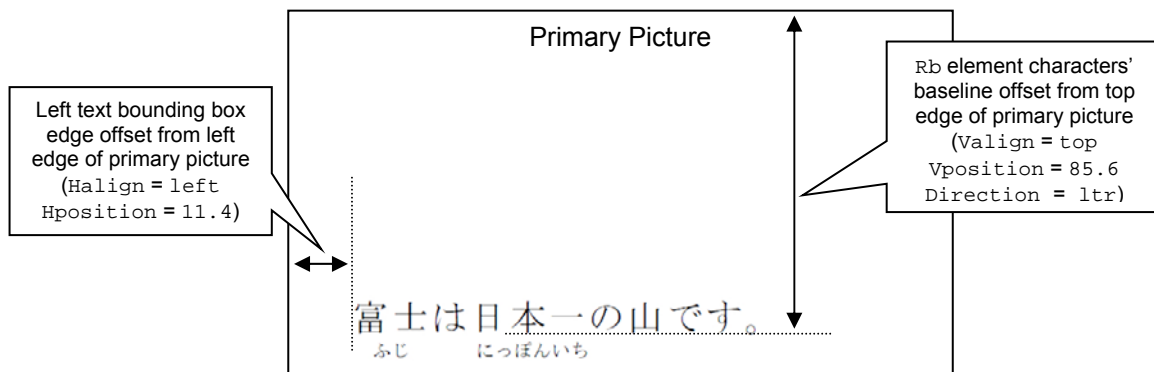


Figure 11 – Ruby Example 2 Illustration (Informative)

6.9 Space Element

When present, the `Space` element shall provide a mechanism to insert a variable amount of space in the middle of a rendered text string. The amount of space to insert shall be indicated by using the `Size` attribute described in the following section. The `Space` element shall be an immediate child of a `Text` element. No text or elements shall be present inside a `Space` element, *i.e.*, the `Space` element shall be empty.

Note: This element can be used for both horizontal text and vertical text.

6.9.1 Size Attribute [instance optional]

When present, the `Size` attribute shall indicate the size of the space to be inserted. The value of the `Size` attribute shall be specified in decimal units of em. Values for size will typically be positive numbers. Negative values shall be allowed but should be used with care as characters could overlap.

The value of the `Size` attribute shall not be less than -1.0. The default value of the `Size` attribute shall be 0.5.

6.10 HGroup Element

When present, the `HGroup` element shall contain a string of characters that are to be rendered horizontally (from left to right) within a `Text` element configured for vertical rendering, *i.e.*, where the `Direction` attribute of the `Text` element is either `ttb` or `btt`. This is most commonly used to render multi-digit numbers in a vertical Asian text string.

This element provides a temporary override for the selected text rendering direction.

The `HGroup` element shall be an immediate child of a `Text` element having a `Direction` attribute of either `ttb` or `btt`. The `HGroup` element shall contain only character data, *i.e.*, no elements shall be present in the content of the `HGroup` element.

The string of characters contained in the `HGroup` element shall be rendered with center alignment with respect to the associated vertical text.

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>
```

Is expected to produce something like this:

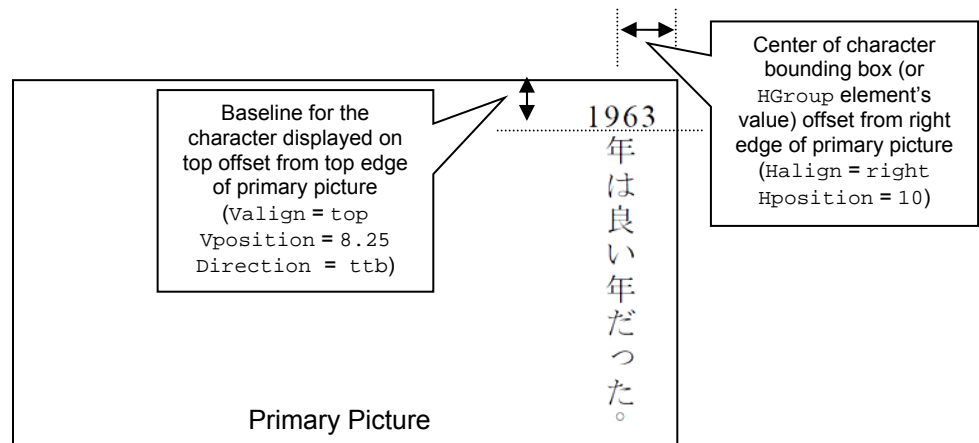


Figure 12 – `HGroup` Example Illustration (Informative)

6.11 Rotate Element

When present, the `Rotate` element shall contain a string of characters that are to be rotated either right or left by 90 degrees prior to rendering. This is most commonly used to present special characters in a text string that is being displayed vertically. The `Rotate` element shall be an immediate child of a `Text` element. The `Rotate` element shall allow one attribute described in Section 6.11.1.

The `Rotate` element shall contain only character data, *i.e.*, no element shall be present in the content of the `Rotate` element.

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

6.11.1 Direction Attribute [instance optional]

When present, the `Direction` attribute shall indicate the direction of the character rotation. The value shall be one of `left`, `right` or `none`. A value of `none` shall indicate that the characters shall not be rotated. A value of `right` shall indicate that the characters shall be rotated to the right or clockwise. A value of `left` shall indicate that the characters shall be rotated to the left or counter-clockwise. The default value shall be `none`.

Table 11 – Direction attribute values (Normative)

<i>Direction Value</i>	<i>Direction Value Description</i>
<code>left</code>	Each character shall be rotated to the left or counter-clockwise.
<code>right</code>	Each character shall be rotated to the right or clockwise.
<code>none</code>	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>
```

Is expected to produce something like this:

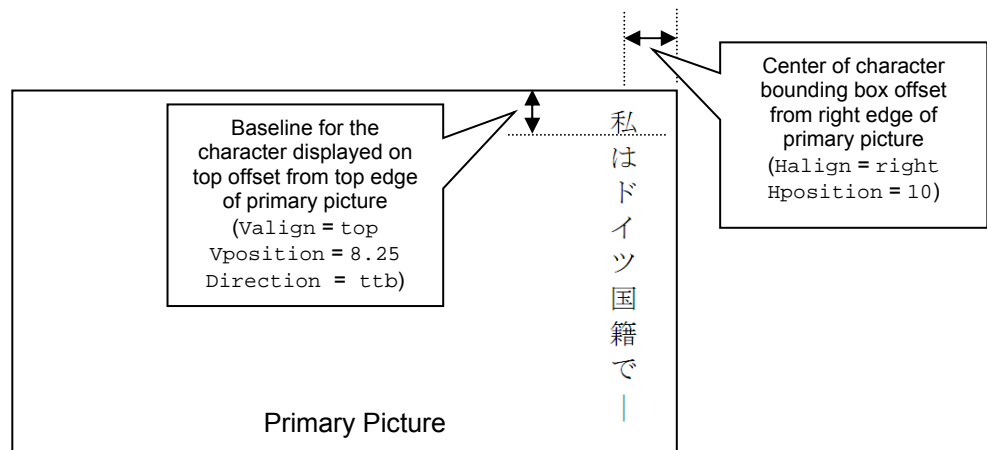


Figure 13 – Rotate Example Illustration (Informative)

6.12 LoadVariableZ Element [instance optional]

When present, the `LoadVariableZ` element shall provide an ordered list of one or more value-duration pairs, separated by one or more whitespace characters. Leading and trailing whitespace (*i.e.*, before and after the list) shall be allowed.

For the purpose of the above definition, the term “whitespace” shall correspond to any of the following characters: 0x09, 0x0a, 0x0d and 0x20.

Each of the value-duration pairs shall be formatted as follows:

```
Zvalue[:Duration]
```

where:

- `Zvalue` shall be identical in definition to the `Zposition` attribute defined above for the `Image` and `Text` elements (see Sections 6.2.5 and 6.3.6), and when present, `Duration` shall be a positive (non-zero) integer specifying the number of editable units during which the respective `Zvalue` shall be active. When absent, the default value of `Duration` shall be 1.

The first value-duration pair within the `LoadVariableZ` element shall be used for the first subtitle editable unit of the respective `Image` or `Text` element, for the number of editable units defined by the respective `Duration` field. The value-duration pairs shall be used one after the other, in the order presented.

Example:

The line below provides an example of a `LoadVariableZ` element compliant with the above definition:

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

When one or more `VariableZ` attributes 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 the content of a `LoadVariableZ` element, any differences shall be handled as follows:

- if the list of values provided by a `LoadVariableZ` element is shorter than the duration of the parent `Subtitle` element, then the rendering device shall use the last value of the `LoadVariableZ` element for the entire remaining duration of the subtitle instance, for all `Text` and/or `Image` elements having the same `Subtitle` ancestor and which reference the given `LoadVariableZ` element identifier.
- if the list of values provided by a `LoadVariableZ` element is longer than the duration of the associated subtitle instance, then the rendering device shall truncate the list of `LoadVariableZ` element values and ignore any values which extend beyond the duration of that subtitle instance, for all `Text` and/or `Image` elements having the same `Subtitle` ancestor and which reference the given `LoadVariableZ` element identifier.

6.12.1 ID Attribute

The `LoadVariableZ` element shall have a single attribute, `ID`, which shall define an identifier, unique within the scope of the ancestor `SubtitleReel` element, 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 value.

7 Samples (Informative)

The following DCDM Subtitle XML structure samples are valid instances of the 2014 SubtitleReel schema. These reels start at 00:00:00:00. A font is loaded and then three subtitle instances are displayed.

Monoscopic DCDM Subtitle XML Structure Samples:

- Monoscopic DCDM Subtitle Sample #1:

```
<?xml version="1.0" encoding="UTF-8"?>
<dcst:SubtitleReel xmlns:dcst="http://www.smpte-ra.org/schemas/428-7/2014/DCST">
  <dcst:Id>urn:uuid:b5965350-d323-4c35-91dd-630556daef04</dcst:Id>
  <dcst:ContentTitleText>Example</dcst:ContentTitleText>
  <dcst:AnnotationText>This is a test file</dcst:AnnotationText>
  <dcst:IssueDate>2012-07-14T21:52:02.000-00:00</dcst:IssueDate>
  <dcst:ReelNumber>1</dcst:ReelNumber>
  <dcst:Language>en</dcst:Language>
  <dcst:EditRate>24 1</dcst:EditRate>
  <dcst:TimeCodeRate>24</dcst:TimeCodeRate>
  <dcst:StartTime>00:00:00:00</dcst:StartTime>
  <dcst:DisplayType>MainSubtitle</dcst:DisplayType>
  <dcst:LoadFont ID="MyFont">urn:uuid:3dec6dc0-39d0-498d-97d0-928d2eb78391</dcst:LoadFont>
  <dcst:SubtitleList>
    <dcst:Font ID="MyFont" Color="FFFFFF" Weight="normal" Size="40">
      <dcst:Subtitle SpotNumber="1" TimeIn="00:01:34:17" TimeOut="00:01:40:20">
        <dcst:Text Valign="top" Vposition="10.00">These are not the droids you're looking for.</dcst:Text>
      </dcst:Subtitle>

      <dcst:Subtitle SpotNumber="2" TimeIn="00:01:41:10" TimeOut="00:01:45:20">
        <dcst:Text Valign="top" Vposition="30.00"><dcst:Font Italic="yes">[Trooper]</dcst:Font> These are not
the droids we're looking for.</dcst:Text>
      </dcst:Subtitle>

      <dcst:Subtitle SpotNumber="3" TimeIn="00:01:50:01" TimeOut="00:01:56:20">
        <dcst:Image Valign="top" Vposition="10.00">urn:uuid:0392ad89-30a2-471c-b289-c210ab8b371e</dcst:Image>
      </dcst:Subtitle>
    </dcst:Font>
  </dcst:SubtitleList>
</dcst:SubtitleReel>
```

- Monoscopic DCDM Subtitle Sample #2:

```
<?xml version="1.0" encoding="UTF-8"?>
<SubtitleReel xmlns="http://www.smpte-ra.org/schemas/428-7/2014/DCST">
  <Id>urn:uuid:9d19361f-72d5-4829-93b8-4a6b57e2f34c</Id>
  <ContentTitleText>Example</ContentTitleText>
  <AnnotationText>This is a test file</AnnotationText>
  <IssueDate>2012-07-14T21:52:02.000-00:00</IssueDate>
  <ReelNumber>1</ReelNumber>
  <Language>en</Language>
  <EditRate>24 1</EditRate>
  <TimeCodeRate>24</TimeCodeRate>
  <StartTime>00:00:00:00</StartTime>
  <DisplayType>MainSubtitle</DisplayType>
  <LoadFont ID="MyFont">urn:uuid:3dec6dc0-39d0-498d-97d0-928d2eb78391</LoadFont>
  <SubtitleList>
    <Font ID="MyFont" Color="FFFFFF" Weight="normal" Size="40">
      <Subtitle SpotNumber="1" TimeIn="00:01:34:17" TimeOut="00:01:40:20">
```

```

<Text Valign="top" Vposition="10.00">These are not the droids you're looking for.</Text>
</Subtitle>

<Subtitle SpotNumber="2" TimeIn="00:01:41:10" TimeOut="00:01:45:20">
  <Text Valign="top" Vposition="30.00"><Font Italic="yes">[Trooper]</Font> These are not the droids we're
  looking for.</Text>
</Subtitle>

<Subtitle SpotNumber="3" TimeIn="00:01:50:01" TimeOut="00:01:56:20">
  <Image Valign="top" Vposition="10.00">urn:uuid:0392ad89-30a2-471c-b289-c210ab8b371e</Image>
</Subtitle>
</Font>
</SubtitleList>
</SubtitleReel>

```

Stereoscopic DCDM Subtitle XML Structure Sample:

```

<?xml version="1.0" encoding="UTF-8"?>
<dcst:SubtitleReel xmlns:dcst="http://www.smpte-ra.org/schemas/428-7/2014/DCST">
  <dcst:Id>urn:uuid:8a7a7f98-1a80-430d-97d9-27340f2a471a</dcst:Id>
  <dcst:ContentTitleText>Stereoscopic Subtitle Example</dcst:ContentTitleText>
  <dcst:AnnotationText>This is a test file</dcst:AnnotationText>
  <dcst:IssueDate>2012-01-11T21:52:02.000-00:00</dcst:IssueDate>
  <dcst:ReelNumber>1</dcst:ReelNumber>
  <dcst:Language>en</dcst:Language>
  <dcst>EditRate>24 1</dcst>EditRate>
  <dcst:TimeCodeRate>24</dcst:TimeCodeRate>
  <dcst:StartTime>00:00:00:00</dcst:StartTime>
  <dcst:DisplayType>MainSubtitle</dcst:DisplayType>
  <dcst:LoadFont ID="MyFont">urn:uuid:3dec6dc0-39d0-498d-97d0-928d2eb78391</dcst:LoadFont>
  <dcst:SubtitleList>
    <dcst:Font ID="MyFont" Color="FFFFFF" Weight="normal" Size="40">
      <dcst:Subtitle SpotNumber="1" TimeIn="00:01:34:17" TimeOut="00:01:40:20">
        <dcst:LoadVariableZ ID="Zvector1"> -0.2:48 0.1:72 0.2 0.3:26 </dcst:LoadVariableZ>
        <dcst:Text Valign="top" Vposition="10.00" Zposition="-0.2" VariableZ="Zvector1">These are not the
        droids you're looking for.</dcst:Text>
      </dcst:Subtitle>

      <dcst:Subtitle SpotNumber="2" TimeIn="00:01:41:10" TimeOut="00:01:45:20">
        <dcst:Text Valign="top" Vposition="30.00" Zposition="0.6"><dcst:Font
        Italic="yes">[Trooper]</dcst:Font> These are not the droids we're looking for.</dcst:Text>
      </dcst:Subtitle>

      <dcst:Subtitle SpotNumber="3" TimeIn="00:01:50:01" TimeOut="00:01:56:20">
        <dcst:Image Valign="top" Vposition="10.00">urn:uuid:0392ad89-30a2-471c-b289-c210ab8b371e</dcst:Image>
      </dcst:Subtitle>
    </dcst:Font>
  </dcst:SubtitleList>
</dcst:SubtitleReel>

```

8 XML Schemas (Normative)

8.1 2014 XML Schema

The XML Schema document presented in this section normatively defines the structure of a DCDM Subtitle file belonging to the namespace <http://www.smpte-ra.org/schemas/428-7/2014/DCST> using a machine-readable language [XML Schema Part 1][XML Schema Part 2]. While this schema is intended to faithfully represent the structure presented in the normative prose portions (Sections 4, 5 and 6) of this document, conflicts in definition may occur. In the event of such a conflict, the normative prose shall be the authoritative expression of the standard.

Note: The schema has been removed from this document to facilitate maintenance and processing with tools suitable to the purpose. If you are reviewing this document, you should have been provided with an up-to-date version of the schema in electronic form. The name of the file should be "DCDMSubtitle-2014.xsd".

8.2 2010 XML Schema

The 2010 XML Schema document presented in this section normatively defines the structure of a DCDM Subtitle file belonging to the namespace <http://www.smpte-ra.org/schemas/428-7/2010/DCST> using a machine-readable language [XML Schema Part 1][XML Schema Part 2]. While it has been corrected from its original published version to faithfully represent the provisions of the ST 428-7:2010 prose, conflicts in definition may remain. Also, conflicts between the SMPTE ST 428-7:2010 prose and the prose of this document exist. In the event of a conflict, the normative prose of this document shall be authoritative, except as modified or constrained in this section.

Note: The schema has been removed from this document to facilitate maintenance and processing with tools suitable to the purpose. If you are reviewing this document, you should have been provided with an up-to-date version of the schema in electronic form. The name of the file should be "DCDMSubtitle-2010.xsd".

The following items have been identified as being different between the original 2010 XML Schema and this revised version:

- The edit unit field of the `TimeCodeType` XML type has been corrected to allow edit rate higher than "30 1", consistent with the provisions of this standard.
- The `TimeIn` and `TimeOut` attributes of the `Subtitle` element were erroneously defined in 2010 as time since the start of the subtitle timeline. These values shall be interpreted as defined in this document (see Section 6.1.2).
- The `Effect` attribute of the `Font` element has a default value that is intentionally different from the original 2010 schema, consistent with the provisions of this standard.
- The cardinality of the `LoadFont` element was in error in the original 2010 schema. This revised version has been corrected, consistent with the provisions of this standard.
- When child of a `Font` element, the `Text` element had a cardinality that was in error in the original 2010 schema. This revised version has been corrected, consistent with the provisions of this standard.
- The `Color` and `EffectColor` attributes of the `Font` element were missing their default values in the original 2010 schema. This revised version has been corrected, consistent with the provisions of this standard.
- The values of the `AspectAdjust` attribute of the `Font` element and of the `Size` attribute of the `Rt` element were missing their ranges in the original 2010 schema. This revised version has been corrected, consistent with the provisions of this standard.

- The default value of the `Spacing` attribute of the `Font` element was in error in the 2010 schema. Also, the value of this `Spacing` attribute was missing its range. This revised version has been corrected, consistent with the provisions of this standard.
- The default values of the `Spacing` and `Offset` attributes of the `Rt` element were in error in the original 2010 schema, and these attributes were missing their value ranges. This revised version has been corrected, consistent with the provisions of this standard.
- The absence of the `AspectAdjust` attribute of the `Rt` element was in error in the original 2010 schema. Its presence in this revised version is intentional, consistent with the provisions of this standard.
- The default value of the `Size` attribute of the `Space` element was in error in the original 2010 schema, and the value of the `Size` attribute was missing its range. The non-zero length of the value of the `Space` element was in error as well. This revised schema has been corrected, consistent with the provisions of this standard.
- The `Language` child element of the `SubtitleReel` element was missing its default value in the original 2010 schema. This revised version has been corrected, consistent with the provisions of this standard.

The following items shall be interpreted as follows, notwithstanding the description of similar or identical items appearing elsewhere in this standard:

- The default `scope` attribute of the `DisplayType` element shall be “`http://www.smpte-ra.org/schemas/428-7/2010/DCST#display-type`” and the associated `DisplayType` value shall be `MainSubtitle`.
- The `Italic` attribute of the `Font` element shall have one of the two following values only: `yes` and `no`.
- The `Rb` element shall be allowed to be empty.
- The `Direction` attribute of the `Text` element shall have one of the four following values only: `ltr`, `rtl`, `ttb` and `btt`.
- Implementation of the 2010 version of this standard may reject documents that contain one or more empty `Text` elements.
- The `ID` attribute of the `LoadFont` element shall be instance optional, and when present may be empty.

The following items are normative provisions of this document that are intentionally omitted from this revised 2010 schema because they are not supported by legacy implementations:

- The `IntrinsicPictureResolution` attribute of the `SubtitleReel` element
- The `LoadVariableZ` element
- The `VariableZ` and `Zposition` attributes of the `Text` and `Image` elements
- The `EffectSize` and `Feather` attributes of the `Font` element

8.3 2007 XML Schema

The 2007 XML Schema document presented in this section normatively defines the structure of a DCDM Subtitle file belonging to the deprecated namespace `http://www.smpte-ra.org/schemas/428-7/2007/DCST` using a machine-readable language [XML Schema Part 1][XML Schema Part 2]. While it has been corrected from its original published version to faithfully represent the provision of the SMPTE

ST 428-7:2007 prose, conflicts in definition may remain. Also, conflicts between the SMPTE ST 428-7:2007 prose and the prose of this document exist. In the event of a conflict, the normative prose of this document shall be authoritative, except as modified or constrained in this section.

Note: The schema has been removed from this document to facilitate maintenance and processing with tools suitable to the purpose. If you are reviewing this document, you should have been provided with an up-to-date version of the schema in electronic form. The name of the file should be "DCDMSubtitle-2007.xsd".

The following items have been identified as being different between the original 2007 XML Schema and this revised version:

- The edit unit field of the `TimeCodeType` XML type has been corrected in the 2014 Schema to allow edit rate higher than "30 1", consistent with the provision of this standard.
- The `TimeIn` and `TimeOut` attributes of the `Subtitle` element were erroneously defined in 2007 as time since the start of the subtitle timeline. These values shall be interpreted as defined in this document (see Section 6.1.2).
- The absence of the value "btt" in the enumeration of values for the `Direction` attribute of the `Text` element was in error. Its presence in this version of the 2007 schema is intentional.
- The `Effect` attribute of the `Font` element has a default value that is intentionally different from the original 2007 schema, consistent with the provisions of this standard.
- When child of a `Font` element, the `Text` element had a cardinality that was in error in the original 2007 schema. This revised version has been corrected, consistent with the provisions of this standard.
- The `Color` and `EffectColor` attributes of the `Font` element were missing their default values in the original 2007 schema. This revised version has been corrected, consistent with the provisions of this standard.
- The value of the `Size` attribute of the `Rt` element was missing their ranges in the original 2007 schema. This revised version has been corrected, consistent with the provisions of this standard.
- The default values of the `Spacing` and `Offset` attributes of the `Rt` element were in error in the original 2007 schema, and these attributes were missing their value ranges. This revised version has been corrected, consistent with the provisions of this standard.
- The absence of the `AspectAdjust` attribute of the `Rt` element was in error in the original 2007 schema. Its presence in this revised version is intentional, consistent with the provisions of this standard.
- The default value of the `Size` attribute of the `Space` element was in error in the original 2007 schema, and the value of the `Size` attribute was missing its range. The non-zero length of the value of the `Space` element was in error as well. This revised schema has been corrected, consistent with the provisions of this standard.
- The `Language` child element of the `SubtitleReel` element was missing its default value in the original 2007 schema. This revised version has been corrected, consistent with the provisions of this standard.

The following items shall be interpreted as follows, notwithstanding the description of similar or identical items appearing elsewhere in this standard:

- The `LoadFont` element shall be present.
- The `Italic` attribute of the `Font` element shall have one of the two following values only: `yes` and `no`.
- The `Rb` element content shall be allowed to be empty.

- The `Direction` attribute of the `Text` element shall have one of the four following values only: `ltr`, `rtl`, `ttb` and `btt`.
- Implementation of the 2007 version of this standard may reject documents that contain one or more empty `Text` elements.
- The `ID` attribute of the `LoadFont` element shall be instance optional, and when present may be empty.

The following items are normative provisions of this document that are intentionally omitted from this revised 2007 schema because they are not supported by legacy implementations:

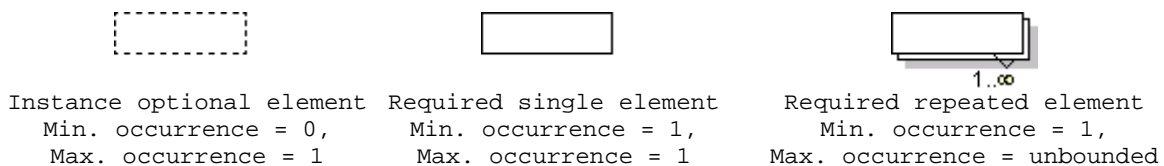
- The `IntrinsicPictureResolution` attribute of the `SubtitleReel` element.
- The `DisplayType` element.
- The `VariableZ` and `Zposition` attributes of the `Text` and `Image` elements.
- The `LoadVariableZ` element.
- The `AspectAdjust`, `Spacing`, `EffectSize` and `Feather` attributes of the `Font` element.

9 XML Diagram Legend (Informative)

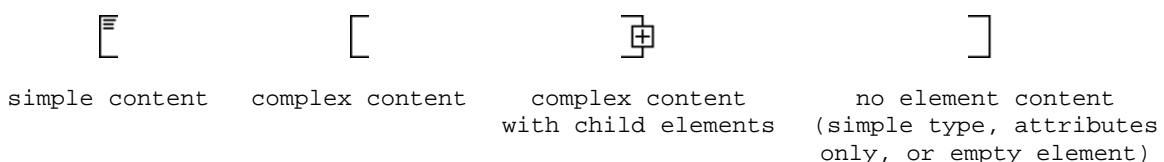
The following provides a legend for notation used in diagrams depicting XML structures.

9.1 Element Symbols

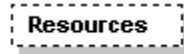
In the schema design diagrams, presented above in this document, only the elements are drawn. Attributes are not visible. The cardinality of the element (0..1, 1 exactly, 0..n, 1..n) is indicated by the border of the elements. Instance optional elements are drawn with a dashed line, required elements with a solid line. A maximum occurrence greater than one is indicated by a double border.



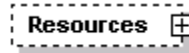
The content model of elements is symbolized on the left and right side of the element boxes. The left side indicates whether the element contains a simple type (text, numbers, dates, etc.) or a complex type (further elements). The right side of the element symbol indicates whether it contains child elements or not:



9.1.1 Examples



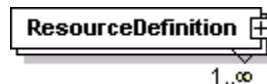
Instance optional single element without child elements. Minimum Occurrence = 0, Maximum Occurrence = 1, content = complex.



As above, but with child elements. The "plus" at the right side indicates the presence of one or more undisplayed child elements.



Mandatory single element. Minimum Occurrence = 1, Maximum Occurrence = 1, content = complex, no child elements (i.e. this denotes an *empty element*). The gray or green text below the element displays the xml-schema annotation associated with the element.



Mandatory multiple element containing child elements (content = complex). This element must occur at least once (Minimum Occurrence = 1) and may occur as often as desired (Maximum Occurrence = unbounded).



Mandatory single element with containing simple content (e.g., text) or mixed complex content (e.g., text with xhtml markup). Minimum Occurrence = 1, Maximum Occurrence = 1, type = xsd:string (for example), content = simple. The three lines in the upper left corner are used for both text and numeric content.

9.2 Model Symbols ("compositors")

A sequence of elements. The elements must appear exactly in the sequence in which they appear in the schema diagram.



A choice of elements. Only a single element from those in the choice may appear at this position.

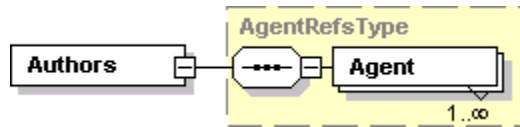


The "all" model, in which the sequence of elements is not fixed.

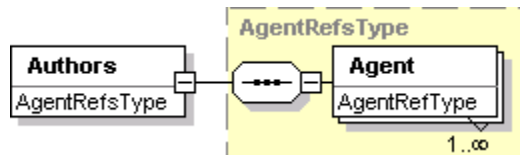


9.3 Types

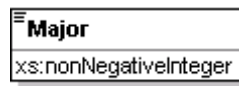
If an element refers to a complex global type, the type is shown with a border and a gray or shadowed background.



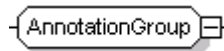
The type name may be shown in the line below the element name:



In that case, the type names of simple types are shown as well:

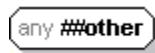


9.4 Model Groups and References

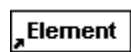


An *element group* is a named container with one or several elements. The group of elements can be reused at multiple places in the schema. Model groups are invisible in the instance document. Model groups have been used sparingly since they do not map to a feature in object-oriented programming languages (unless they support multiple inheritance).

Import note on reading the diagrams for model groups: If the model group symbol is drawn with simple lines (i.e. not dashed), this does not imply that the elements in the model group are required. The instance optionality of the group depends on the instance optionality of elements contained in the model group. (Model groups can be made instance optional, e.g., to make a model group with required elements instance optional in some cases, but this has not been used.)



The *"any"* group is a special kind of model group. It is a placeholder for elements not defined in the schema. The "any" element defines points where the schema can be extended. After the "Any" keyword the namespace from which the elements may come is defined, for example, "##other" specifies that the extension elements may come from any namespace, except from the current schema namespace.



Element references are indicated through a link arrow in the lower left corner. They are similar to references to model groups within a schema, but instead of refining the model group, they directly refer to a single global element. The global element can then be reused in multiple places.

Annex A Bibliography (Informative)

Note: All references in this document to other SMPTE documents use the current numbering style (e.g., SMPTE ST 428-1:2006) although, during a transitional phase, the document as published (printed or PDF) may bear an older designation (such as SMPTE 428-1-2006). Documents with the same root number (e.g., 428-1) and publication year (e.g., 2006) are functionally identical.

SMPTE ST 428-1:2006, D-Cinema Distribution Master (DCDM) — Image Characteristics

Internet Engineering Task Force (IETF) (1994, December). RFC 1738 — *Uniform Resource Locators (URL)*. <http://www.ietf.org/rfc/rfc3986.txt>

RDDL — *Resource Directory Description Language* J. Border and T. Bray 2002. <http://www.rddl.org/>

World Wide Web Consortium (W3C) — Namespaces in XML, <http://www.w3.org/TR/REC-xml-names/>

World Wide Web Consortium (W3C) — *QA Framework: Specification Guidelines, Formal Languages*, <http://www.w3.org/TR/2004/WD-qaframe-spec-20041122/>

World Wide Web Consortium (W3C) — *XML Schema Primer*, <http://www.w3.org/>

World Wide Web Consortium (W3C) — *Ruby Annotation*, <http://www.w3.org/TR/ruby/>

Annex B Summary of Main Changes from SMPTE ST 428-7:2010 (Informative)

This document has been revised from its 2010 version in order to add support for stereoscopic subtitles, and also to eliminate ambiguities and address issues raised since its previous publication in 2010. The purpose of this section is to highlight the main differences for the benefit of the reader. Only brief descriptions are provided in each case and the intent of this section is not to list every single change.

Note that it is likely that legacy implementations will not support the new XML elements, XML attributes and certain XML element and attribute values associated with the new `http://www.smptra.org/schemas/428-7/2014/DCST` namespace name. Content creators are encouraged to use the new namespace name only when the associated new features are required for the content, and in that case it is strongly advised that prospective playback equipment be surveyed to assure support for this namespace name.

The main differences are listed below:

- A new `http://www.smptra.org/schemas/428-7/2014/DCST` namespace name has been introduced in Section 4.
- A definition Section 4.2 has been created in order to group and clarify the definitions of primary picture, point, em, baseline, subtitle time code, instance optional and current font. The subtitle editable units field (E+) of the subtitle time code has also been expanded to allow support for higher `EditRate` values (e.g., “120 1”).
- An instance optional `IntrinsicPictureResolution` attribute has been added in Section 5.1 to provide information about the PNG image(s) referenced by a DCDM Subtitle file.
- `EditRate` and `TimeCodeRate` element definitions were clarified in Sections 5.8 and 5.9.
- The `LoadFont` element is now instance optional in Section 5.11 as it is not needed when `Text` elements are not present. 4-byte UTF-8 encoded characters support is now optional.
- The expected formulation of font resources was clarified in Section 5.11.
- A display precedence rule was added in Section 5.12.2 for `Text` and `Image` elements located within the same `Subtitle` instance to extend the display precedence rule already present for overlapping `Subtitle` instances.
- New default `DisplayType` values associated with the new namespace were defined in Section 5.13 to accommodate the two display types identified at the time this standard was revised.
- `TimeIn` and `TimeOut` attribute definitions were corrected in Sections 6.1.2 and 6.1.3.
- `Subtitle` element constraints were added in Section 6.1.6 to clarify use of the `StartTime` element, and to make dual fading operations optional.
- Guidance regarding the use of transparency information in a PNG image referenced by an `Image` element was added in Section 6.2 to clarify the expected display.
- The definition of the `Text` element was clarified in Section 6.3.
- The rendering of space characters was clarified in Sections 6.3 and 6.5.
- `Zposition` and `VariableZ` attributes were added in Sections 6.2 and 6.3 to accommodate stereoscopic subtitle support. `Halign`, `Hposition`, `Valign` and `Vposition` definitions were further defined to work properly with these two new attributes. `Halign`, `Hposition`, `Valign` and `Vposition` definitions were also clarified and the definitions of `Valign` and `Vposition` attributes associated with `Text` elements were changed in order to rely on characters’ baseline –

rather than on the border of the rendered text – to avoid possible misalignment due, for example, to the presence of descenders or capital letters.

- The definition of the `Direction` attribute of the `Text` element was revised in Section 6.3.5 to clarify rendering directions and to encourage the Unicode Bidirectional Algorithm usage. A new `Direction` attribute value (“`hor`”) was added.
- The definition of the `Font` element attributes was clarified in Section 6.4 to avoid nesting behaviors and to enforce consistent processing of attribute values.
- Section 6.4.2 provides additional provisions to further specify superscript and subscript text rendering.
- The `Effect`, `Italic`, `Underline` and `Weight` attributes were further specified (in Sections 6.4.3, 6.4.4, 6.4.5 and 6.4.6 respectively) to increase consistency between implementations. The `Effect` attribute’s default value was also changed to `shadow` in order to match current practice as exhibited by existing implementations at the time this standard was revised, and two new `Italic` attribute values were added: `left` and `right`.
- The definition of the `EffectColor` attribute was revised to accommodate implementations not able to handle two alpha-blend values (see Section 6.4.8).
- The `EffectSize` and `Feather` attributes were added in Sections 6.4.12 and 6.4.13, respectively.
- The rendering of Ruby text was clarified in Sections 6.5, 6.6 and 6.7, and the `Rb` element is now required to contain at least one character. The definition of the `Rt` element’s `Offset` attribute has been clarified in Section 6.7.3.
- Conflicts between the XML examples, the XML schema and the prose were fixed in Sections 6.8, 6.10, 7 and 8. New DCDM Subtitle XML Structure Samples were added.
- The provisions of the `HGroup` and `Rotate` elements were clarified (in Sections 6.10 and 6.11 respectively).
- The `LoadVariableZ` element was added in Section 6.12 to support stereoscopic subtitles.
- Section 8 has been updated to include revised schemas for the <http://www.smpte-ra.org/schemas/428-7/2007/DCST> and <http://www.smpte-ra.org/schemas/428-7/2010/DCST> namespace names. The schemas allow continued use of these legacy formats in compliance with this standard.