EBU

OPERATING EUROVISION AND EURORADIO

TECH 3360

EBU-TT, PART 2
MAPPING EBU STL (TECH 3264) TO EBU-TT SUBTITLE FILES

VERSION 1.0

SOURCE: SP/MIM – XML SUBTITLES

Geneva
May 2017

There are blank pages throughout this document. This document is paginated for two sided printing

Conformance Notation

This document contains both normative text and informative text.

All text is normative except for that in the Introduction, any section explicitly labelled as 'Informative' or individual paragraphs which start with 'Note:'

Normative text describes indispensable or mandatory elements. It contains the conformance keywords 'shall', 'should' or 'may', defined as follows:

'Shall' and 'shall not':	Indicate requirements to be followed strictly and from which no deviation is permitted in order to conform to the document.
'Should' and 'should not':	Indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others. OR indicate that a certain course of action is preferred but not necessarily required. OR indicate that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.
'May' and 'need not':	Indicate a course of action permissible within the limits of the document.

Default identifies mandatory (in phrases containing "shall") or recommended (in phrases containing "should") values that can, optionally, be overwritten by user action or supplemented with other options in advanced applications. Mandatory defaults must be supported. The support of recommended defaults is preferred, but not necessarily required.

Informative text is potentially helpful to the user, but it is not indispensable and it does not affect the normative text. Informative text does not contain any conformance keywords.

A conformant implementation is one which 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.

Documentation Conventions

If a reference to an element type is used in this specification and the name of the element type is not namespace qualified, then the namespace http://www.w3.org/ns/ttml applies.

If a reference to an attribute type is used in this specification and the name of the attribute type is not namespace qualified, then the appropriate namespace for TT Parameter, TT Style or TT Metadata applies[1].

[1] See EBU Tech 3350 [2], p. 11

Contents

Status of this document

This document is a stable document and may be used as reference material or cited from another document.

This document is part of a series of EBU-TT (EBU Timed Text) documents. The full list of published and planned EBU-TT documents is given below.

Part 1: EBU-TT Subtitling format definition (EBU Tech 3350)
Introduction to EBU-TT and definition of the XML based format.

Part 2: STL (Tech 3264) Mapping to EBU-TT (EBU Tech 3360)
How EBU-TT provides backwards compatibility with EBU STL.

Part 3: EBU-TT in Live Subtitling applications: system model and content profile for authoring and contributions (EBU Tech 3370)
How to use EBU-TT for the production and contribution of live subtitles.

EBU-TT WebSocket Carriage Specification (EBU Tech 3370s1)
Carriage of EBU-TT Part 3 over WebSocket

EBU-TT, Part D (EBU Tech 3380)
EBU-TT content profile for TTML that can be used for the distribution of subtitles over IP based networks.

Carriage of EBU-TT-D in ISOBMFF (EBU Tech 3381)
How EBU-TT-D can be stored using the storage format of the ISO Base Media File Format (ISO/IEC 14496-12).

EBU-TT, Part M: Metadata Definitions (EBU Tech 3390)
Definition of metadata elements and attributes for use in EBU-TT documents

EBU-TT Annotation
How EBU-TT can be used in future scenarios for ‘authoring of intent’.

EBU-TT User Guide
General guide (‘How to use EBU-TT’).

EBU-TT Part 2

EBU STL (Tech 3264) Mapping to EBU-TT

EBU Committee	*First Issued*	*Revised*	*Re-issued*
TC	2013	2017	

Keywords: subtitling, STL, XML, W3C, TTML, DFXP, captions, 3264

Scope

The EBU STL subtitle file format (EBU Tech 3264 [1]) is currently widely used as an archiving and exchange format. The STL[2] format is a binary format that allows little scope for variation in implementation of a stored subtitle file. This rigid nature facilitates interchange of subtitle files between organisations. In contrast, The EBU-TT format, specified in EBU Tech 3350 [2], uses TTML as a foundation technology and consequently a certain 'freedom of expression' is possible. Stated differently, it is technically possible to author subtitle files in EBU-TT that differ markedly from each other in internal XML structure yet should result in exactly identical visual outcomes when rendered / played. This document makes recommendations for a strategy for converting an STL document into a valid EBU-TT document, such that conversions from STL produce EBU-TT documents with similar internal structures, thus facilitating interchange.

In practise, not all STL files conform precisely to the EBU Tech 3264 specification. Industry accepted practises exist that have extended the STL format, e.g. using 'dummy subtitles' to carry additional metadata or comments. In addition, certain EBU Tech 3264 defined values for metadata fields have been extended to cater for subsequent developments in the broadcast industry (e.g. frame rates). Where possible, the more common STL practises are covered by this document.

This document is intended to outline strategies that may be used to convert documents that fully comply with the STL specification (as defined by EBU Tech 3264) or with only minor deviations in interpretation of the STL specification (e.g. vertical position signalling, or the use of CR/LF). A summary of the conversion strategies is provided in Annex G. The conversion of STL documents that do not adhere to the specification,[3] or 'STL' documents that are internally inconsistent, is not covered, although the conversion strategies described in this document may still be valid.[4]

Several fields within the GSI block of the STL file format are not relevant to the described conversion strategies; consequently non-conformance in these fields should not preclude conversion (see 3.2 GSI fields that are not mapped into EBU-TT).

[2] All references to STL in this document refer to the subtitle file format specified in EBU Tech 3264.

[3] E.g. Values are used in GSI or TTI fields that are outside the range of valid values as defined by the STL specification.

[4] STL documents may have inconsistencies between values in the GSI fields and the content of the TTI blocks, e.g. the number of TTI blocks may not match the number of TTI blocks present, or the time code values used in the TCO / TCI fields of the TTI blocks are inconsistent with the time-base identified in the GSI block (e.g. frame counts values above 25 exist although the DFC field value signals 25 fps). An implementation may be able to determine a strategy to convert these 'non-conformant' documents, perhaps 'by examination', or a Document Processing Context may provide appropriate direction.

Definition of Terms

Captions and subtitles

The term “captions”, especially in the USA, describes on-screen text for use by deaf and hard of hearing audiences. Captions include indications of the speakers and relevant sound effects. The term “subtitles”, especially in European use, describes on-screen text for both translation and transcription (hearing loss) purposes. For easier reading only the term “subtitles” is used in this specification since the EBU-TT file representation for captions and subtitles is identical. In this specification the term “captions” may be used interchangeably for the term “subtitles” (except where noted).

Active video

The term “active video” (known alias: Production Aperture) refers to the portion of the video signal that is used to carry picture information, as specified in SMPTE ST 2016-1:2009 Chapter 4 [3].

Active image

The term “active image” refers to the portion of the video picture area that is being utilized for program content, as specified in SMPTE ST 2016-1:2009 Chapter 4. The active image excludes letter box bars and pillar box bars.

Subtitle Safe Area

The use of safe areas (for graphics, subtitles and important ‘on screen’ action) in television production ensures that the most important parts of the picture are seen by the majority of viewers. Older televisions can display less of the video outside of the safe area than ones made more recently. As a result of the deliberate ‘over-scan’ in analogue television sets, the presentation of any STL subtitle over the active video is assumed to be restricted to a ‘Subtitle Safe Area’ within which the 23 addressable rows (for subtitle text) of Teletext characters (excluding the ‘page header’[5] row and ‘Fast Text’ row[6]) are rendered. The 25 rows of Teletext characters are typically drawn ‘inside a horizontal and vertical margin’ over the active video.[7]

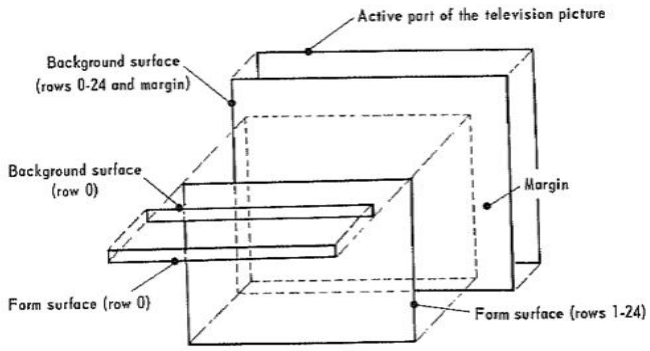

Figure 1: Diagram from Tech 3241 [4] (Fig 9 Presentation of the visual service message)

The position and exact proportion of the active video that could be covered by the 25 rows of Teletext characters of a typical Teletext implementation is undefined by the Teletext specifications. Many Teletext implementations used ROM based characters with a 10 x 12 (v /h) pixel size, consequently a typical implementation may have 25 rows of 20 video lines per character (since interlaced video system teletext decoders effectively scan the character ROM twice as the video lines are displayed for each interlaced field), i.e. 500 lines vertically were used for the

[5] Also known as the ‘service row’ or ‘row 0’ (this row is never displayed for subtitle ‘pages’).

[6] This is also confusingly known as ‘Row 24’. Early Teletext implementations had 24 rows (numbered 0 to 23), each page comprising of a page header (row 0) and 23 rows of content. The ‘Fast Text’ row (row 24) was added when early Teletext was standardised to 25 rows (Rows numbered 0 to 24 inclusive).

[7] Early Teletext Specifications (e.g. Figure 9 of Tech 3241; published by the EBU in 1982) describe a ‘background surface’ which includes a ‘margin’, inside of which two vertically adjoining ‘form surfaces’ are positioned; a ‘form surface for row 0’ (the ‘service / page header row’) and a second ‘form surface’ for the rows 1 to 24.

Teletext display area. Thus the **25 rows** of Teletext characters would cover **~87%** (500/576) of the display vertically. This display area is typically offset by 30 - 40 lines from the top of the active video – thus roughly centring the Teletext display vertically within the 576 video lines of the active video area.[8]

Unfortunately, it is not possible to determine a contemporary horizontal size for the area covered by Teletext characters in a similar manner because the contemporary horizontal active video resolution (704 pixels) was effectively an outcome of the pixel sampling rate of an analogue system.[9] However, given the 12 pixel wide characters implemented in ROM by most electronic Teletext decoders, it is evident that a 40 column display would generate Teletext characters covering a maximum of 480 'clocked pixels' horizontally. If these pixels were also co-incident with the contemporary underlying active video pixels then they would cover just **~68%** of the horizontal resolution (480/704).[10] However, it is clear from images and recordings of early Teletext pages on analogue television sets that contemporary Teletext presentation was not constrained to such a small central region of the visible picture, but instead used a horizontal margin similar in proportion of the active video to the vertical margin.[11]

Note: A subsequent revision of the Teletext standard (the level 2.5 version) introduced the option of an additional 16 character 'sidebar' positioned horizontally alongside the original 40 characters. This would occupy an additional 192 pixels (16 x 12), resulting in a total of 672 'clocked pixels' displayed over the 704 horizontal pixel resolution (with the result that text would occupy the majority of the horizontal screen width, i.e. ~95% if the pixels were co-incident (672/704)).

Subtitle zero

Many STL subtitle files contain a 'subtitle zero' that is used to convey metadata that cannot be carried within the GSI block. This 'subtitle zero' is typically the first subtitle in the file with time code in and out values outside of the range of the associated video file. (E.g. Time Code In = 00:00:00:00 Time Code Out = 00:00:00:08).[12]

Document Processing Context

The implied (or explicit) external context or environment internal to a content processor in which document processing occurs, and in which out-of-band protocols or specifications may define certain behavioural defaults, such as the most appropriate document conversion strategy.

The conversion of a source STL file by a content processing implementation may require examination of the source file to determine the most appropriate conversion strategies. This examination could be human mediated or fully automated. The most appropriate conversion strategies can also be influenced by other information external to the processing context, e.g. company-specific default settings, house guidelines, etc. This examination of source files, external settings and the determinations made is termed the 'Document Processing Context' by this specification.

1. STL to EBU-TT conversions

EBU-TT is intended as general purpose exchange format for subtitles. As an exchange format EBU-TT intrinsically also is an archiving format (see Figure 2). Opportunities for conversion from STL to EBU-TT occur at both the creation and archiving situations. The streaming of subtitles that may be created live in real time for live broadcast programmes is covered by EBU Tech 3370 (EBU-TT Part 3) [5].

[8] **Note: this assumes European PAL analogue video resolution (576 by 704).**

[9] The introduction of digital video 'standardised' the horizontal resolution of active video at 704 pixels within a 720 pixel horizontal carriage (8 'blanking' pixels each side) to facilitate a 'mod16' representation.

[10] Teletext character 'pixels' are **not required** to be horizontally co-incident with active video pixels.

[11] I.e. the 'clocked Teletext pixels' were typically wider than a video pixel.

[12] Typical practise is to include a **single** subtitle zero to contain information that is not intended for display. However, multiple subtitles with this characteristic might exist.

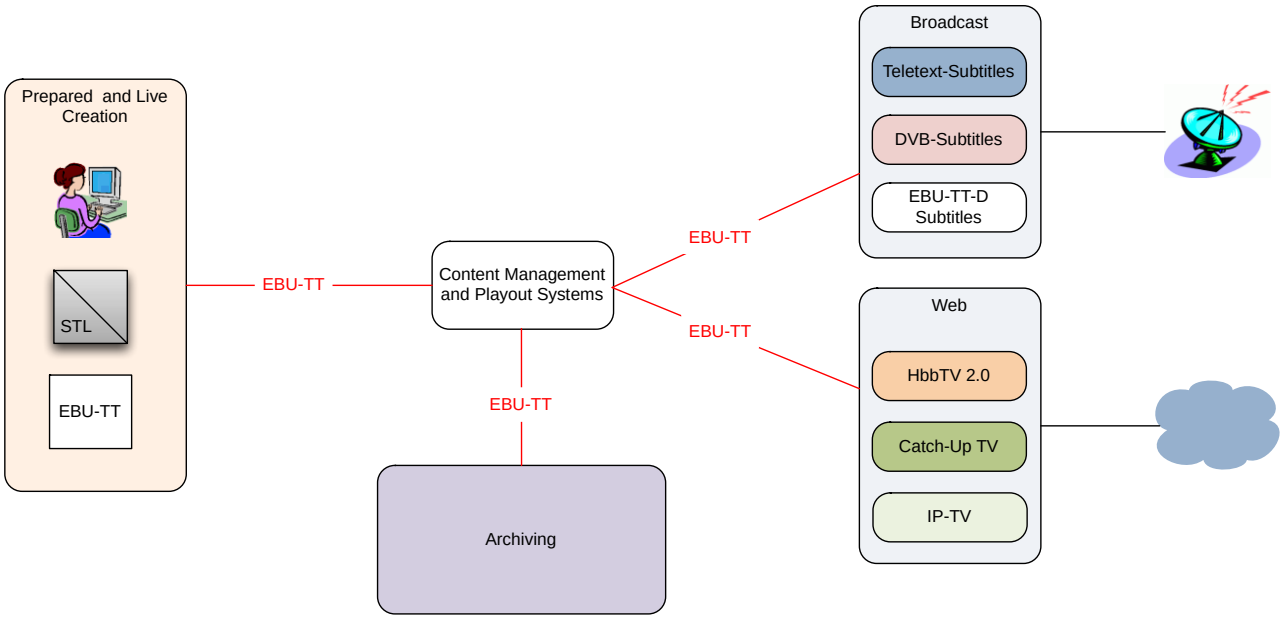

Figure 2: Use of EBU-TT in subtitling workflows

It should be appreciated that an STL file may be used for subtitles intended for Teletext presentation or for subtitles intended for 'open' / burnt-in (or other graphical presentation such as DVB bitmap subtitles). For the first use case, Teletext presentation, an accurate determination of the intended presentation of the subtitles is possible since the rendering of the contained text content is tightly constrained by the Teletext specification. However, when an STL file is used for the second case of 'open' subtitling, the rendering / presentation environment is unknown. For example, no details of the font used for rendering are available within the STL file. Typically when a subtitle inserter renders this type of STL file, the font and other information is provided as part of the device configuration. To accurately convert an 'open' STL file into an EBU-TT file, the conversion algorithm must be supplied with these settings by the 'Document Processing Context'.[13]

In an identical fashion to STL files, EBU-TT files are intended for use in exchange; between systems of different vendors; and between an authoring system and different distribution formats.

We can either envision a single master EBU-TT file that can be transcoded for different distribution platforms, or multiple separate files targeted at each platform.

[13] EBU Tech 3264 files have two forms, Teletext and Open. Only the Teletext format has obtained acceptance as interchangeable; the 'Open' form is too flexible for more than just 'text and timing transfer' with horizontal justification. Vertical placement in the 'open' form is often implemented in a non-standard compliant manner.

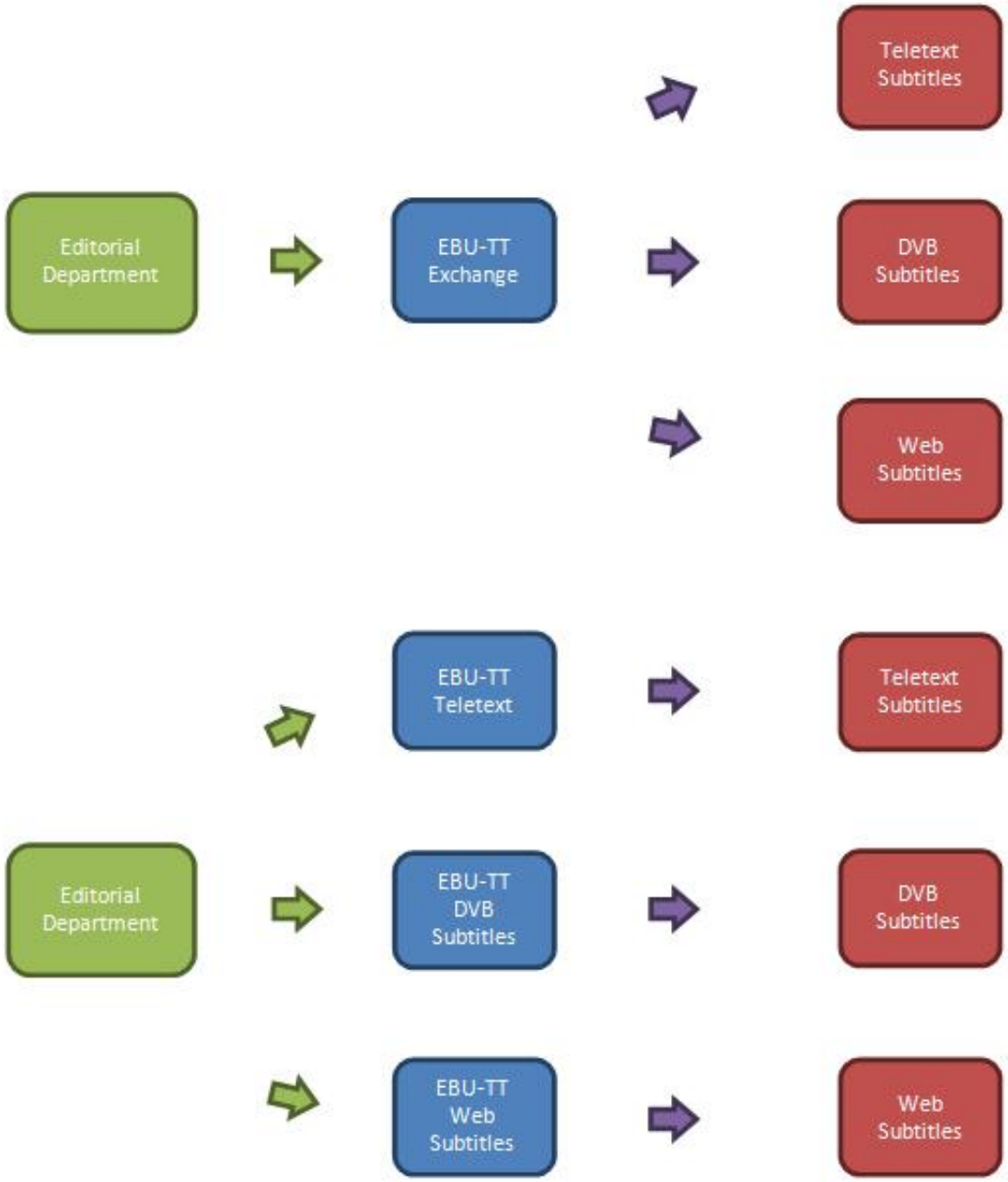

Figure 3: Single and Multiple EBU-TT file usage scenarios

Clearly a single 'master' format (top diagram) is preferable to meet the requirement of universal use. This single master file must be; transformable for all outputs, precise in the mapping of information (e.g. accurate positioning), easy to use and simple to implement.

This document creates a 'conformance point' for mapping STL files into a single 'master' format in compliance with the EBU-TT document structure defined by EBU-TT, Part 1.

1.1 TTML features required for the mapping of STL-to-EBU-TT

The following TTML features shall at least be supported to transform an STL file according to the proposed mapping strategy in this document. The list is meant to be a minimum set of required features and does not preclude a more advanced implementation of EBU-TT (e.g. the support of additional colours or the support of additional TTML features used in EBU Tech 3350).

TTML feature name	Comment
#backgroundColor-inline	The following values shall be supported: • Black ("black") • Red ("red") • Green ("green") • Yellow ("yellow") • Blue ("blue") • Magenta ("magenta") • White ("white") • Cyan ("cyan")
#cellResolution	-
#color	The following values shall be supported: • Black ("black") • Red ("red") • Green ("green") • Yellow ("yellow") • Blue ("blue") • Magenta ("magenta") • White ("white") • Cyan ("cyan")
#content	-
#core	-
#displayAlign	The values "before" and "after" shall be supported.
#dropMode-dropNTSC	-
#dropMode-dropPAL	
#dropMode-nonDrop	-
#fontFamily-generic	-
#fontSize	-
#frameRate	The values "25" and "30" shall be supported.
#frameRateMultiplier	-
#layout	-
#length-cell	-
#length-percentage	-
#length-positive	-
#lineHeight	-
#markerMode-discontinuous	-
#metadata	The tt:metadata element shall be supported. The extension of the tt:metadata elements by private namespaces elements shall be supported.
#structure	-
#styling	-
#styling-inheritance-content	-
#overflow-visible	-
#overflow	-
#showBackground	-
#styling-referential	-
#textAlign-relative	-
#timeBase-smpte	- STL files typically use SMPTE time code cue values
#timeBase-clock	- STL files may also represent recordings of subtitles with 'wall clock' values (see § 1.2.4).
#writingMode-horizontal-lr	-

EBU-TT extends the #tt:metadata feature to map metadata from the GSI Block. Therefore the

following extension shall be supported.

`#ebu-tt-metadata`

The following metadata elements shall be supported:

- ebuttm:appliedProcessing
- ebuttm:authoringTechnique
- ebuttm:referenceClockIdentifier
- ebuttm:conformsToStandard
- ebuttm:localTimeOffset
- ebuttm:relatedMediaDuration
- ebuttm:relatedMediaIdentifier
- ebuttm:relatedObjectIdentifier
- ebuttm:sourceMediaIdentifier
- ebuttm:documentBeginDate
- ebuttm:documentContentType
- ebuttm:documentCreationMode
- ebuttm:documentTransitionStyle
- ebuttm:documentOriginalProgrammeTitle
- ebuttm:documentTranslatedEpisodeTitle
- ebuttm:documentTranslatedProgrammeTitle
- ebuttm:documentTranslatedEpisodeTitle
- ebuttm:documentTranslatorsName
- ebuttm:documentTranslatorsContactDetails
- ebuttm:documentSubtitleListReferenceCode
- ebuttm:documentTotalNumbersOfSubtitles
- ebuttm:documentMaximumNumberOfDisplayableCharacterInAnyRow
- ebuttm:documentStartOfProgramme
- ebuttm:documentCountryOfOrigin
- ebuttm:documentPublisher
- ebuttm:documentEditorsName
- ebuttm:documentEditorsContactDetails
- ebuttm:documentUserDefinedArea
- ebuttm:stlCreationDate
- ebuttm:stlRevisionDate
- ebuttm:stlRevisionNumber
- ebuttm:subtitleZero
- ebuttm:transitionStyle
- ebuttm:broadcastServiceIdentifier

The use of the metadata elements shown above in **bold** is mandatory in this specification.

1.1.1 EBU Tech 3360 (EBU-TT, Part 2) [6] Implementation Summary

A summary of the mapping of STL (EBU Tech 3264) features into an EBU Tech 3350 (Part 1) document in accordance with the strategies described in this document may be found in Annex F.

1.2 Optional features for the mapping of STL-to-EBU-TT

1.2.1 Inclusion of STL files within EBU-TT documents

To support the binary data tunnelling of the original STL file the following feature should be supported:

`#ebu-stl-tunnelling`

1.2.2 Language with inline progression direction right to left

To enable the mapping of STL files in Arabic or Hebrew the following feature should be supported for the right to left inline progression direction:

`#writingMode-horizontal-rl`

1.2.3 Non-Teletext Subtitles

For STL files where the Display Standard Code (DSC) is not 1 or 2, the following features should be supported:

`#fontStyle-italic`

`#textDecoration-under`

1.2.4 Time code source

STL files cannot indicate the source of time code and the implication is that time codes in STL files are synchronisation cues in a SMPTE time code representation, which are used to match against time code values present in an associated video asset. In accordance with the requirements of EBU-TT Part 1, a `ttp:timebase` attribute value shall be specified on the `tt:tt` element. This attribute should normally have the value of “smpte”.

The time code values used in the associated video asset will typically be monotonically increasing values[14], but this is not guaranteed, and can only be determined / signalled by the Document Processing Context. A `ttp:markerMode` attribute value, as defined in EBU-TT Part 1, and required when the `ttp:timebase` attribute has a value of ‘smpte’, should be specified on the `tt:tt` element and should have the value of “discontinuous” to signal to a document processing context that monotonicity of time code values should not be assumed, or in the case where the Document Creation Context clearly signals that monotonicity of time code values in the related media can be assumed, the `ttp:markerMode` attribute value should be set to 'continuous'.

However, the STL file format also supports a cumulative mode of subtitle presentation, which may be used for a specific presentation effect. Cumulative subtitles are also known as “add-on” subtitles, and they allow, for example, the display of a fresh subtitle (or additions to the current on screen text) before the previous displayed text has been erased from the screen (see § 4.5.3). An STL file may also be used in cumulative mode to record subtitles that have been created by a live editing process (e.g. stenographic or voice input). These live recorded cumulative files may record the synchronisation cues as clock based times (i.e. a wall clock value recorded in SMPTE time code representation), but this style of usage of the STL file format can only be signalled by the Document Processing Context. In this case only, when the Document Processing Context is clearly signalled, the `ttp:timebase` attribute value shall be specified on the `tt:tt` element with the value of “clock”. In accordance with the requirements of EBU-TT Part 1, a `ttp:clockMode` attribute value **shall** also be specified on the `tt:tt` element with a value determined by the Document Processing Context. Additionally, an `ebuttm:documentCreationMode` element with a value of “live” may also be used within an `tt:metadata` element of the generated document to indicate that the document content originated from a live recording.

[14] In accordance with any specific drop frame convention!

1.3 Round tripping

This document makes recommendations for conversion from (an) STL file(s) to an EBU-TT file and the strategies used are *not necessarily* bi-directional. The EBU-TT binary tunnelling mechanism may be used to record the **original STL file(s)** within the EBU-TT equivalent.

1.4 Conversion strategies

The STL file format identifies one of four intended display modes for the file content by using the DSC code in the GSI block (see §3.5). The STL format is based on the assumption that a Teletext decoder is used to show the subtitles contained in the Teletext form of STL.[15] This underlying Teletext foundation creates artefacts, e.g. control codes occupy a character cell and are rendered as a space on the display. Teletext subtitles are traditionally displayed using the 'double height' character mode defined by the Teletext standard.

Specifically the Teletext format has a line length limit of 40 characters, which includes the control characters to select colours and to activate background display for normal pages ('Start box' codes). Teletext pages marked as 'sub-title' or newsflash pages within a Teletext service should be displayed as 'boxed text' by a receiver.[16] The control code overhead required for boxed text has resulted in a 'common practise' of using a maximum of 37 (or 36) characters in a subtitle row for text.

The minimal requirement for control codes on output is two consecutive 'Start box' codes at the start of a Teletext subtitle, leading to an absolute maximum number of 38 displayable subtitle characters in a row. (If space exists at the end of the text, two 'End box' codes are required to switch off the background – however the background boxing will 'end' automatically at the end of the line, so for full rows of text no 'End box' codes are needed). In addition the typical 'double height' display of Teletext subtitles requires an additional control code. If the subtitle text is not white on black (the default colour setting), 36 or fewer characters are available as additional control codes are required to select foreground text (and optionally background) colour.

An EBU-TT file may be used in place of an STL file for Teletext subtitling. In such files the limitations of the intended output format (Teletext) must be respected in the generated EBU-TT document. It is recognised that creators of EBU-TT files may desire a relaxation of specific limitations of the STL format, but for other users it is equally important that the regular grid based, mono-spaced nature of the original STL file is maintained, in particular the spacing of the text. The EBU-TT format supports several methods of conveying position information. The conversion strategies described in this document replace the legacy positioning method of STL with percentage based positioning while ensuring that the intent of the content is not distorted.

It is a common practise in STL subtitle files to use space characters from the mono-spaced Teletext font to position subtitle text. However, in document formats that are based on the XML standard, (as is EBU-TT,) whitespace characters are not necessarily treated as 'content' and may be regarded as simply present to make the document easier to read (for humans). A strategy to strictly preserve the visual presentation of STL files in an EBU-TT document is described in §1.4.3.1. Preserving a Teletext style presentation.

The EBU-TT format supports three potential mechanisms for positioning the regions that contain text content over an associated video presentation; the 'cell' measurement unit, percentages and the 'pixel' measurement unit. In accordance with TTML (the foundation for EBU-TT), pixel/cell coordinates and dimensions, and/or percentages may be used to define regions or positions in a "root container region" into which all content is placed. In EBU-TT, the 'root container region' is the active video (see "Definitions of terms"). While all of these positioning mechanisms could be used simultaneously within a single document, this document recommends using a conversion strategy that implements the percentage measurement unit to position the regions used for subtitle

[15] In particular, the use of mono-spaced fonts, regular grid based character layout and a modal control code mechanism for character colours and style. STL subtitle files may also use 'whitespace' for horizontal positioning of subtitles.

[16] See §14.1.3. of EBU Tech 3240 [7] (Main characteristics of a "level.2" teletext system for European broadcasting organisations using the fixed-format principle).

display. Using percentages is a more universal strategy, since even if the absolute dimensions of the associated video are changed (e.g. by scaling) the relative dimensions are still valid.[17] To retain information about the assumed presentation, a `tts:extent` attribute value should be declared in the root element of EBU-TT documents converted from STL (see § 1.4.2).

Subtitles in general, and particularly in Teletext presentation, are rendered for the viewer using specific fonts and row spacing. The spacing of the rows of text in a multiline subtitle presentation is often closer than typical typographic conventions would dictate. Further, the presentation of the 23[18] addressable Teletext row positions used for subtitle display is assumed to be restricted to a 'Subtitle Safe Area'.[19] To preserve the intention of the original subtitle author, the position and size of this 'Subtitle Safe Area' and the contemporary Teletext subtitle row spacing should be taken into account in the conversion strategy.

Use cases can be envisaged where a subsequent transformation of the EBU-TT document is anticipated that will re-position the 'Subtitle Safe Area' occupied by the subtitles (e.g. the viewer will define the subtitle display area dimensions, or the document will be converted back into a Teletext broadcast signal). In such use cases, only the relative positions of the subtitles are required to be stored in the converted EBU-TT document and the 'Subtitle Safe Area' referenced in this document would occupy 100% of the horizontal and vertical active video extent. Files generated for these use cases will not be presented '*correctly*' over video where a margin around the subtitle presentation is assumed to be pre-coded into the subtitle content (i.e. the files will not be 'Subtitle safe').

The foundation of EBU-TT (TTML) allows the definition of font-size by scaling of the 'EM square'. (E.g. 200% means the EM square is doubled in size, 2c means the EM Square has dimensions of 2c by 2c, 28px means scaling the EM Square to 28px). The default size of the EM Square is 1c, although in practice the size of the actual rendered font glyphs depends on the design of the font.

Note: In some fonts certain glyphs may bleed outside their EM squares. These fonts consequently may also result in text rendering that overflows the region boundaries,[20] particularly if a 'minimal sized region strategy' is used. Several strategies are discussed later in this document for the creation and use of regions when converting row based STL file content to EBU-TT.

To support the accurate definition of font sizes and line heights while still using relative dimensions, the cell unit is used. To relate the size of a cell to the active video, the cell resolution of the active video is declared in the EBU-TT document. The definition of cell resolution and use of cell units links the size of a cell, (and consequently the size of font [*if expressed in cell units*[21]*]* and line height), to a relative percentage of the active video.[22]

The default 'Subtitle Safe Area' used in the examples in this document is approximately 90% of the width and height of the active video. Alternatively, the width and height of the 'Subtitle Safe Area' may be supplied by the Document Processing Context or the implementation. The STL subtitle file format has been used as a container file format for subtitles intended for use with video stored in various resolutions/aspect ratios and framerates. The authorial intentions and 'target video'

[17] In practice, rendering systems will convert percentages to pixels, and may round to the nearest pixel (or rendering systems may use sub-pixel accuracy and anti-aliasing).

[18] Only rows 1 to 23 are used for subtitles as the Page Header (row 0) cannot be addressed and Row 24 is not used for subtitles. The last valid 'position' for a 'double height' subtitle is Row 22 as double height characters occupy the current and the subsequent Teletext row.

[19] Described in EBU Recommendation R 95 [8].

[20] EBU-TT (in contrast to TTML), recommends that a processor presents as much active content as possible even if the bounds of the region are exceeded (i.e. acts as if tts:overflow has the value of 'visible'). This may result in text that 'overhangs' the background of a subtitle region.

[21] If tts:fontSize is a percentage value the size of the font may still may be based on the computed cell size if there is no absolute size further up in the style resolution hierarchy, i.e. No ancestor element specifies font size in absolute (non cell based) units (e.g. pixels).

[22] Fractional percentages could be used for font-size and line height definitions but rounding errors could result in inconsistent row spacing. Using cells allows font size and line height to be specified as integers.

characteristics cannot be signalled within the STL file format, but may be supplied by the Document Processing Context. It is further accepted that modern television display equipment does not suffer from a horizontal 'overscan' problem or vertical blanking requirements and it may be considered appropriate to increase the width (and height) of the 'Subtitle Safe Area'.

One approach to achieve vertical positioning of subtitle rows would be to create a region in the EBU-TT document for each separate row (a 'region per row' strategy), and to independently position them on the screen, see Figure 4, below. This approach has a number of limitations; a) it does not utilise the semantics of EBU-TT (and TTML) for text layout and positioning, b) rounding errors when rendering the regions may cause inconsistent spacing between rows, and c) it does not match the requirement of mapping into a universal style of EBU-TT file.

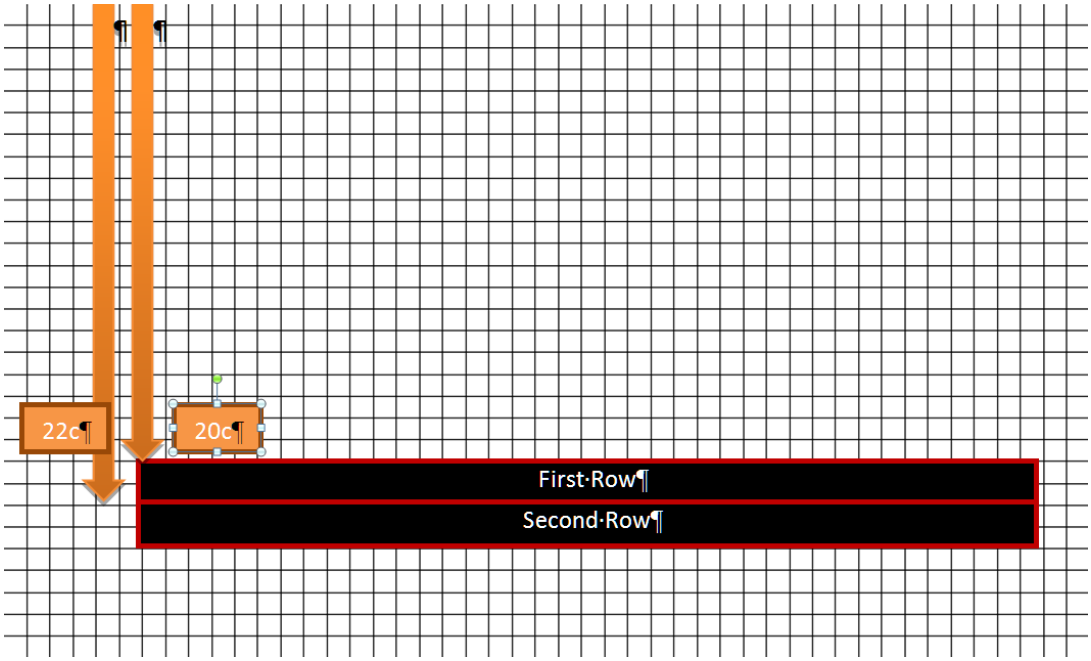

Figure 4: Vertical positioning of subtitle rows using a region for each row

However, in a 'minimal sized region strategy', sequential Teletext subtitle rows could be combined into a single region to resolve the potential issues with inconsistent row spacing and to utilise EBU-TT (and TTML) semantics for text layout. A minimal region strategy may optimise just the height of the generated regions, retaining the full width of the 'Subtitle Safe Area' or may calculate horizontal offsets for regions (*with suitably modified region extents*). (See § 4.5.6.2).

Note: The use of minimal sized regions has implications for overflow and display alignment. Content may render larger than assumed by the conversion process due to font differences (the assumed font height at conversion may differ from the line height metric defined by the font used at presentation). The content may overflow the region in a direction that will depend on the `tts:displayAlign` and `tts:textAlign` properties that are applicable to the region and the text content.

An alternative 'simple region strategy' would be to generalise the subtitle text presentation to using a region that covers the whole 'Subtitle Safe Area' of the screen, and using the display align property, to position the subtitles to either the top or bottom of the screen ('before' / 'after'), as shown in Figure 5.

Figure 5: Vertical positioning of subtitle rows using a single region

By using `tt:br` elements, text can further be moved up/down within the single region (see Figure 6, below). This 'simple region strategy' has two consequences: the background colour of 'empty lines' must be set to 'transparent', to avoid creating empty black bars on the screen; and 'double height' Teletext subtitle positioning must be carefully handled. It is recommended that, if using the 'simple region strategy', two precisely overlapping regions are used, one region for bottom-aligned and the other for top-aligned subtitles since the display align property is fixed by the region definition. This approach covers the two most basic cases. More advanced region use than described here may be implemented by a conversion processor implementation (e.g. different regions for different speakers).

Note: Using multiple regions may result in 'registration errors' between the rendered row positions of a top aligned region and a bottom aligned region.

Using just a single region with the display align property set to 'after' (to cater for the typical bottom placement) could require a large number of `tt:br` elements for any top positioned subtitle and could also result in 'overflow' of the text above the top of the region due to unanticipated text wrapping of long lines.

Note: The use of overlapping regions may cause 'downstream' problems, since some distribution formats do not permit simultaneously active overlapping regions. Subject to determination by the Document Processing Context the 'top' and 'bottom' regions may be defined so they no longer overlap. Subtitles that are intended to be displayed vertically centred on the screen are problematic. Although this is an uncommon requirement, a pre-conversion assessment by the Document Processing Context may be required to determine the intended position of all of the subtitles in the source file in order to find a boundary between the two regions such that every subtitle in the source may be correctly positioned whilst still contained in a single region.

Figure 6: Using `tt:br` elements to move text up/down within a single region

Note: In STL subtitle files the use of carriage returns may be inconsistent between different files when the subtitles are intended for double height display. One OR two carriage return characters may be used in 'double height' Teletext STL files to indicate a single line break between subtitle rows.[23] Correct interpretation of the carriage returns in an STL file is dependent on the Document Processing Context.

Note: Files with complex subtitle positioning (e.g. subtitles positioned near the active speakers) are not considered suitable for conversion using a simple region strategy and may require the use of the minimal region generation strategy with horizontal region offsets and variable region extents to reproduce the intended subtitle presentation. Determination of the suitability of a region strategy should be made by the Document Processing Context.

1.4.1 ttp:cellResolution

A `ttp:cellResolution` attribute value shall be specified on the `tt:tt` element. This element should be defined with a value that takes into account any margin between the 'Subtitle Safe Area' and the edge of the active video. The difference between the row column values as specified by the `ttp:cellResolution` attribute (and thus mapped to the active video) and the implicit 40 x 23 'Subtitle Safe Area' should be calculated such that the `ttp:cellResolution` attribute value effectively defines a cell dimension that is the appropriate size for the positioning and rendering of a 40 x 23 matrix of cell sized characters **within** the 'Subtitle Safe Area'.[24]

For example, to create a 'Subtitle Safe Area' of ~90% of the active width and ~80% height[25] of the picture area, the 40 by 23 matrix of Teletext character cells ('Subtitle Safe Area') can be positioned centrally within a 'notional' 44 by 29 grid (with respect to the `ttp:cellResolution` attribute value).

Note: The `ttp:cellResolution` attribute value does not create any grid or matrix of character positions, it only effectively sets a cell size, because the active video area is

[23] The STL specification (Tech 3264) defines the CR/LF indicator (conveyed by character code 8Ah) as used to initiate the second and subsequent rows of the **subtitle** display. Some STL files (from examination) interpret the CR/LF indicator as indicating the initiation of the subsequent Teletext display row (i.e. half a row of double height text).

[24] The 40 x 23 matrix described here ignores the existence of a Teletext 'service row' (row 0) and the 'Fast Text' row (row 24). The 'page header' and 'Fast Text' rows are suppressed when a Teletext subtitle service page is active.

[25] The 'Subtitle Safe Area' is ~80% of vertical active video height because Row 0 and Row 24 are not part of the 'Subtitle Safe Area', but still occupy (2 rows of) vertical space in a Teletext presentation and therefore limit the vertical extent of the active video that can be used for subtitles. *Assuming that contemporary Teletext presentations used ~87% of the active video for 25 rows (87% / 25 x 23 = 80%).*

sub-divided by the cell resolution. Other suggested values for mapping different sizes of 'Subtitle Safe Area' into `ttp:cellResolution` values, e.g. 16:9 presentation or larger 'Subtitle safe areas', may be found in Annex E.

Note: The actual position of subtitle text (relative to the active video) is determined by the origin and extent attributes of the `tt:region` (and styles e.g. justification) referenced by the container element of the subtitle text.

1.4.2 tts:extent and ebuttm:documentTargetAspectRatio

The `tts:extent` attribute on the `tt:tt` element, and the content of the `ebuttm:documentTargetAspectRatio` element should be set to values for the pixel extent and Display Aspect Ratio of the associated active video when those values are supplied by the Document Processing Context. When the extent and/ or Display Aspect Ratio of the active video is not supplied by the Document Processing Context, the `tts:extent` attribute and `ebuttm:documentTargetAspectRatio` element should be set using the value of the Disk Format Code in the GSI block of the STL file:

Disk Format Code (DFC) from GSI block (see § 3.4)	tts:extent attribute value	ebuttm:documentTargetAspectRatio
"STL25.01"	"704px[26] 576px"	4:3
"STL30.01"	"704px 480px"[27]	4:3

In accordance with the requirements of EBU-TT Part 1, a `tts:extent` attribute value **shall** be specified on the `tt:tt` element if the generated document uses the 'pixel' measurement.

1.4.3 General conversion considerations

In general, the EBU-TT document generated should convey the subtitling intention of the source STL document, but may differ in its exact presentation. The STL source document may be interpreted in a manner that does not exactly preserve positioning and typography - in particular by using proportional fonts. This may have implications for the alignment of multi-row subtitles that rely upon the presentation nature of mono-space fonts.

The subtitle content of the STL file should be mapped to a region within the 'Subtitle Safe Area'. The entire subtitle content for each subtitle is stored within a single `tt:p` element and the rows of the subtitle are 'created' by using the `tt:br` element. Each subtitle is mapped to a single `tt:region`, which, in a 'simple region strategy', will typically fully cover the horizontal and vertical extent of the 'Subtitle Safe Area'.

1.4.3.1. Preserving a Teletext style presentation

It is a common practise in STL subtitle files to use space characters of the mono-spaced Teletext font to position subtitle text. However, in document formats that are based on the XML standard, (as is EBU-TT,) whitespace characters are not necessarily treated as 'content' and may be regarded as simply present to enhance human readability. Although a mechanism exists within the XML standard to identify that whitespace is significant (`xml:space` = "preserve"), some non-compliant xml processors do NOT support this. Further, as XML files are easily editable with simple text editors the risk exists that using these tools would break subtitle positioning if spaces are retained as 'padding' without deliberately placing these spaces inside a container element. In the past (with binary STL) this risk was virtually not present.

To preserve a Teletext style presentation, the following additional conversion strategies may be used as determined by a Document Processing Context:

[26] PAL and NTSC broadcast video standards define the active video image as 704 pixels wide.

[27] The STL standard does not specify that a value of "STL30.01" for the DFC field indicates a specific video **resolution** – it simply defines the timebase used in the STL file (i.e. frame counts in time-code values in TTI blocks may range between 0 and 24, or between 0 and 29 respectively). However all contemporary 30 frames per second video formats had a vertical resolution of 480 active lines (e.g. PAL-M and NTSC).

- a `tt:span` element is generated around the subtitle text.
- a `tt:span` element is generated around the whitespace used to 'position' the text.[28]
- mono spaced fonts may be used for the subtitle text
- leading spaces (or control code spaces) in the subtitle text may be retained.[29]

2. General principles for the conversion of STL subtitle files

An STL file consists of a set of subtitle texts and control information for programme material in one principal language. The file structure comprises a single General Subtitle Information (GSI) block followed by a number of Text and Timing Information (TTI) blocks.

The GSI block carries general information such as display standard, language, programme title in original and local language, etc. In general, the GSI block is converted into header attributes and metadata element content in the EBU-TT document. A TTI block normally includes the information necessary to define one subtitle. Each individual TTI block is converted into EBU-TT document content, i.e. `tt:p` and `tt:span` elements. Exceptionally, additional TTI blocks, called Extension Blocks, may be used.

2.1 Subtitle zero

By convention the first subtitle of an STL file may also contain programme related metadata similar to the GSI block.[30] This first subtitle is typically known as "subtitle-zero". The information it contains can be very important operationally! Typical practise is to include a single subtitle zero, however, multiple 'subtitle zero's might exist and their collective content should be treated as a single unit for processing and conversion.

Although the "subtitle-zero" information may be mapped into EBU-TT subtitle content it is recommended that it be moved into the head of an EBU-TT file as metadata. However, this information cannot automatically be mapped into structured metadata fields, except when the Document Processing Context has knowledge of the labelling and formatting conventions used for "subtitle-zero" content. Where a contradiction exists between metadata within the GSI block of the STL file and any 'subtitle zero' content, the Document Processing Context or conversion process must also determine which value(s) are to be placed in the EBU-TT metadata.

It is recommended that the content of a "subtitle-zero" subtitle is placed into an `ebuttm:subtitleZero` element within the `tt:head` metadata as follows (even if the Document Processing Context has mapped all or part of the subtitle zero content to structured metadata):

```
<tt:tt … >
 <tt:head>
  <tt:metadata>
   <ebuttm:subtitleZero>
    BIG BUG BUNNY
    MUC E889X/01
    VGW001721
   </ebuttm:subtitleZero>
  </tt:metadata>…
 </tt:head>
…
</tt:tt>
```

[28] An xml:space = "preserve" attribute should be used on this element.

[29] Note: For some mono-spaced fonts, the space character may be 'narrower' than non-space characters!

[30] The reasoning is that information in the STL metadata fields does not necessarily pass through or get displayed by all subtitle editing or review equipment, whilst the subtitles do. A subtitle zero is usually not meant to be broadcast.

"Subtitle-zero" subtitle content may also be converted into EBU-TT subtitle content[31] as follows:

```
<tt:tt … >
<tt:head>
  <tt:metadata>
  </tt:metadata>…
  <tt:styling>
    <tt:style…/>
  </tt:styling>
  <tt:layout>
    <tt:region…/>
  </tt:layout>
</tt:head>
<tt:body style="defaultStyle">
  <tt:div>
    <tt:p xml:id="sub0" begin="00:00:00:01" end="00:00:00:02" …>
      <tt:span…>BIG BUG BUNNY</tt:span> <tt:br/>
      <tt:span…>MUC E889X/01</tt:span> <tt:br/>
      <tt:span…>VGW001721</tt:span>
    </tt:p>
    <tt:p xml:id="sub1" begin="10:00:01:02" end="10:00:03:04" …>
      <tt:span…>This is the first subtitle.</tt:span>
    </tt:p>
  </tt:div>
</tt:body>
</tt:tt>
```

2.2 Document Metadata

Metadata information that applies to the whole document should be placed as child elements of a `tt:metadata` element inside the `tt:head` element.[32]

In accordance with the EBU-TT specification (EBU Tech 3350) version 1.2, the value of the EBU-TT Part 1 standard used by this document instance shall be placed in the `ebuttm:conformsToStandard` as a child of a `tt:metadata` element inside the `tt:head` element. A second `ebuttm:conformsToStandard` child element should also be present and should contain the value of the EBU-TT Part 2 standard used as the reference in creation of this EBU-TT document. See table below:

`ebuttm:conformsToStandard` **element value**	**Conformance**	**Note**
"urn:ebu:tt:exchange:2017-05"	Optional	Recommended by Part 1 (v1.2)
"urn:ebu:tt:exchange:stl-mapping:2017-05"	Optional	Recommended this document.

The Software and Version used to create the EBU-TT document should be placed in an `ebuttm:documentOriginatingSystem` elementwithin the `tt:metadata` element inside the

[31] Processors and display renderers of EBU-TT documents may not respect the 'do not display' convention of a subtitle zero that is placed in the body of the document. However, compliant processors and display renderers of EBU-TT documents should display only those subtitles whose start and end times overlap the time code of any given frame of video. If the subtitle zero time code falls outside the programme time code the subtitle should not be displayed to the viewer.

[32] Use of the ebuttm:documentMetadata element shown in earlier versions of this document is now deprecated.

`tt:head` element.

Except in the case of a source STL file that contains a recording of live subtitles (see § 1.2.4), an `ebuttm:documentCreationMode` element with a value of "prepared" may also be included within the `tt:metadata` element inside the `tt:head` element.

The following elements may also be included within the `tt:metadata` element inside the `tt:head` element with values determined by the Document Processing Context.

- ebuttm:authoringTechnique
- ebuttm:referenceClockIdentifier
- ebuttm:localTimeOffset
- ebuttm:relatedMediaDuration
- ebuttm:relatedMediaIdentifier
- ebuttm:relatedObjectIdentifier
- ebuttm:sourceMediaIdentifier
- ebuttm:documentBeginDate
- ebuttm:documentContentType
- ebuttm:documentTransitionStyle
- ebuttm:transitionStyle
- ebuttm:broadcastServiceIdentifier

2.2.1 Metadata to record the applied processing of the STL file

It is recommended that the strategies and parameters used during the conversion process should be recorded in an `ebuttm:appliedProcessing` child element of the `tt:metadata` element inside the `tt:head` element. This `ebuttm:appliedProcessing` element should contain a `process` attribute indicating the process of conversion with a value of "convertFromSTL" and an `appliedDateTime` attribute containing a 'datetime' value to identify when the conversion occurred.[33]

It is further recommended that this `ebuttm:appliedProcessing` element contains an additional child element (described below) to contain further elements that record the parameters used and any Document Processing Context determinations made about the source STL file during conversion. These additional metadata elements are formally defined by the EBU-TT Part M [9] specification.

The use of the `ebuttm:conformsToStandard` and `ebuttm:appliedProcessing` elements and the child elements are illustrated in the example in § 2.3 Binary data tunnelling of the original STL file(s).

2.2.1.1 ebuttm:stlConversion

This element acts as a container element for child elements that record the parameters and/or Document Processing Context determinations relevant to the conversion process used to create this document.

2.2.1.2 ebuttm:stlParameter

An `ebuttm:stlParameter` element MAY be used to record the value of a specific identified parameter or Document Processing Context determination applied to the conversion process used to create this document. Multiple `ebuttm:stlParameter` elements may be used. Each `ebuttm:stlParameter` element shall have a key attribute identifying the contained parameter value. Identifiers specific to STL file conversion are formally defined in the EBU-TT Part M

[33] The ebuttm:appliedProcessing element can contain a (human readable) label or identifier for a specific processing step that has been applied to the (creation of) the EBU-TT document.

specification.[34]

When present, and the key attribute value is an enumerated value as formally defined in the EBU-TT Part M specification, the `ebuttm:stlParameter` element shall contain content in accordance with the following table (see footnote[34]).

Key attribute value	Content values	Note
"regionStrategy"	"simple" |"minimalVertical" |"minimal"	Indicates the region creation strategy used.
"safeAreaOrigin"	ebuttdt:originType	The origin of the 'Subtitle Safe Area' used by the conversion process expressed in ebuttdt:originType with respect to the active video dimension.
"safeAreaExtent"	ebuttdt:extentType	The extent of the 'Subtitle Safe Area' used by the conversion process expressed in ebuttdt:extentType with respect to the active video dimension.
"teletextStyleFont"	"true"|"false"	A value of true indicates that this document has been converted assuming a font that has Teletext characteristics (see § 4.1).
"justificationOverride"	"none" |"left" |"centered" |"right" |"multi-row"	A value of "none" indicates that **all** the subtitles in the source document have been converted and retain their original justification as contained in the source document. Values of "left", "centered", "right" or "multi-row" indicate that the justification of **all** subtitles in the source document has been overridden (all subtitles have been changed to the same indicated justification) in the generated EBU-TT document (see § 4.5.4). Note: The overriding of subtitle justification should not be assumed to be restricted to only subtitles marked with a Justification Code of 00h (see below).
justificationCodeZeroStrategy	"spacePreserve" |"interpreted" |"forced" |"multi-row" |"regionOffset"	The strategies for processing subtitles flagged with a justification code of 00h value (as described in § 4.5.4) can be signalled using these values. Note: The default processing strategy for processing subtitles marked with a Justification Code of 00h is "forced" with justification set as "centered".

[34] Other values for the key attribute may be used by a specific processing implementation but the meaning of the associated element content is undefined by this specification or by EBU-TT Part M.

2.3 Binary data tunnelling of the original STL file(s)

To transport a binary data copy of the input STL document(s)[35] used to generate the EBU-TT document, one or more `ebuttm:binaryData` element(s) within a `tt:metadata` element may be used. The binaryData elements should be in the logical order of source documents. This logical order is identified by the Total Number of Disks (TND) and Disk Sequence Number (DSN) fields within the STL documents when an STL file spans multiple 'disks'.

Note: An STL document represents the contents of a single 'disk'. If an EBU-TT document is constructed from a series of STL files representing individual segments of a longer movie (e.g. the original movie is subtitled in 'reels'), then the logical order of source documents is defined as the order of processing of the STL documents, which is assumed to match the temporal presentation order of the subtitles.

The `tt:metadata` elements containing the binary data may be placed either in the `tt:head` element or in a containing `tt:div` element that should be placed at the end of the `tt:body` element[36] (see examples).

The `fileName` attribute shall be set to the filename(s) of the source STL document. Note that the path (or URL) of the source STL document converted should NOT be included in the `fileName` attribute value (e.g. XYZ.stl, not C:\temp\xyz.stl).

The `creationDate`, `revisionDate` and `revisionNumber` attributes of each `ebuttm:binaryData` element shall be set to the Creation Date (CD), Revision Date (RD) and Revision Number (RN) field values from the GSI block of the corresponding source STL document, decoded as described in §3.14 and §3.15.

It should be appreciated that the binary tunnelling mechanism only allows for the original source file(s) to be encoded within the EBU-TT file. Subsequent editing on the EBU-TT file will create a divergence between the original source file (encoded as base64) and the 'equivalent' EBU-TT file's XML content.

There is no inbuilt mechanism within EBU-TT to record this divergence. E.g. 'EBU tunnelled binary valid at version xxx' may be stored within a private metadata item. Unless the authoring tool used to modify the EBU-TT has also re-issued the STL file, then the recommended practice is to remove any embedded STL files if the substantive XML content is edited.

[35] The EBU Tech 3264 file format was defined when the usual medium for storing and exchanging subtitle files was a 1.44 Mbyte capacity 3.5" floppy disk. In extreme cases (e.g. 'cumulative' subtitling capture) the STL file size might exceed this capacity and the file would span more than one floppy disk.

[36] Placing the binary data at the end of the document reduces the data that needs to be read before accessing subtitle data, when processing the EBU-TT document using an event-based sequential access parser.

```
<tt:tt … >
<tt:head>
  <tt:metadata>
    < ebuttm:conformsToStandard>urn:ebu:tt:exchange:2017-05</ ebuttm:conformsToStandard>
    < ebuttm:conformsToStandard>urn:ebu:tt:exchange:stl-mapping:2017-05</ ebuttm:conformsToStandard>
    <ebuttm:appliedProcessing appliedDateTime=" 2016-07-05T11:07:00" process="convertFromSTL">
       <ebuttm:stlConversion>
            <ebuttm:stlParameter key="regionStrategy">minimal</ebuttm:stlParameter>
            <ebuttm:stlParameter key="safeAreaExtent">91% 85%</ebuttm:stlParameter>
            <ebuttm:stlParameter key="safeAreaOrigin">4.5% 7.5%</ebuttm:stlParameter>
            <ebuttm:stlParameter key="teletextStyleFont">true</ebuttm:stlParameter>
       </ebuttm:stlConversion>
    </ebuttm:appliedProcessing >
    <ebuttm:binaryData textEncoding="BASE64" binaryDataType = "EBU Tech 3264" fileName ="XYZ.STL"
creationDate="2010-01-04" revisionDate="2011-11-10" revisionNumber="7">[37]
TWFuIGlzIGRpc3Rpbmd1aXNoZWQsIG5vdCBvbmx5IGJ5IGhpcyByZWFzb24sIGJ1dCBieSB0aGlz
IHNpbmd1bGFyIHBhc3Npb24gZnJvbSBvdGhlciBhbmltYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg
dGhlIG1pbmQsIHRoYXQgYnkgYSBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpbiB0aGUgY29udGlu…
    </ebuttm:binaryData>…
  </tt:metadata>…
</tt:head>
<tt:body> …
</tt:body>
</tt:tt>
```

Efficient sequential processing of an EBU-TT file containing binary data may be enhanced by the recommended positioning of the binary data at the end of the EBU-TT document as shown below.

```
<tt:tt … >
<tt:head>…
</tt:head>
<tt:body>
  <tt:div…>
    <tt:p…>…
    </tt:p>
  </tt:div>
  <tt:div>
    <tt:metadata>
      <ebuttm:binaryData textEncoding="BASE64" binaryDataType = "EBU Tech 3264" fileName ="XYZ.STL"
creationDate="2010-01-04" revisionDate="2011-11-10" revisionNumber="7">[38]
TWFuIGlzIGRpc3Rpbmd1aXNoZWQsIG5vdCBvbmx5IGJ5IGhpcyByZWFzb24sIGJ1dCBieSB0aGlz
IHNpbmd1bGFyIHBhc3Npb24gZnJvbSBvdGhlciBhbmltYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg
dGhlIG1pbmQsIHRoYXQgYnkgYSBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpbiB0aGUgY29udGlu…
      </ebuttm:binaryData>…
    </tt:metadata>
  </tt:div>…
</tt:body>
</tt:tt>
```

[37], [38] Note: In accordance with WC3 XML 1.0 [10], 'whitespace' in base64 strings is ignored.

3. Mapping and Conversion of STL metadata from the GSI block

Most of the information that is present in the GSI block of the EBU-STL specification is incorporated by EBU-TT as legacy metadata and EBU-TT has adopted the semantics from EBU Tech 3264 for these elements. A `tt:metadata` element inside the `tt:head` element shall be used as the container for the STL metadata information that applies to the whole document.

The General Subtitle Information (GSI) block consists of 1024 bytes. The first 448 bytes are specified by the EBU, and the following 576 bytes may be defined by the user (see the User-defined Area information in Annex A). Part of the information contained in the GSI block relates to the use of the TTI blocks in the document or is calculated from the TTI blocks (e.g. Total number of disks). This internal self-referential information is only relevant to the structure of the STL file and is not a target for conversion into an EBU-TT document. Other information in the GSI block from the translator, about the ownership of the document, the contents of the document or defined by the user, is mapped into the EBU-TT document.

3.1 GSI block conversion notes

Undefined values used in the GSI block should be ignored.[38] The 75 spare bytes in the GSI block are not converted into EBU-TT file content.

All unused bytes in the GSI block are set to 20h. Trailing space characters (20h) should be removed from fields that are converted to EBU-TT element content.

In general the EBU Tech 3264 specification effectively defines that all numeric values are represented using ASCII characters in the GSI block. E.g. '35' is represented 0x30 0x35. The specification does not explicitly state that ASCII characters should be used for all fields, however it does state that the 'text' in the GSI block should be displayable using the DOS type or print command. Most implementations interpret this as using ASCII characters for all numeric values even where not explicitly stated or implied by the number of bytes allocated to the field by the specification and the range of valid values.

However it is accepted that some STL files in circulation incorrectly encode values as binary (e.g. 35 = 0x23) or the numbers represented in ASCII may contain white space in between digits (i.e. 35 = "3 5"). The conversion of STL documents that do not adhere to the specification is not covered, although the conversion strategies described in this document may still be valid following resolution of incorrectly encoded values by a Document Processing Context.

3.2 GSI fields that are not mapped into EBU-TT

The following fields are only internally relevant to the STL file and are not relevant for conversion into an EBU-TT document. These fields are not mapped into EBU-TT:

- Total Number of TTI Blocks (TNB)
- Disk Sequence Number (DSN)
- Total Number of Disks (TND)
- Total Number of Subtitle Groups (TNG)
- Time Code: Status (TCS)
- Maximum Number of Displayable Rows (MNR)
- Time Code: First in-cue (TCF)

An inconsistency between the TNB value and the number of TTI blocks present in an STL file should not prevent conversion. All TTI blocks in the STL file should be converted, regardless of the value of this field.

EBU-TT files do not permit untimed content. Consequently the TCS field is not mapped to EBU-TT.

[38] E.g. the codes 00h or the values 0Ah..0Fh in the TND byte.

If an STL file with 'invalid' time code is converted, all the STL content is deemed to be valid. The Document Processing Context may provide additional relevant metadata in a private namespace.

The Maximum Number of Displayable Rows field is not mapped into EBU-TT. However, the value indicated by this field is relevant for the interpretation of the Vertical Position field of the TTI blocks in the STL file. For Teletext subtitles (see §3.5 Display Standard Code) the number of rows in the presentation is fixed (at 23) and the MNR value is not relevant. The MNR value is encoded as two ASCII numeric characters, e.g. the typical MNR value (for Teletext files) of '23' is encoded as 32h, 33h. In some non-compliant files, the MNR may signal the maximum number of rows that have been used by any single subtitle in the file, e.g. a value of '3' signals that the biggest subtitle has 3 rows. The conversion strategies described in this document may still be valid following the detection, (e.g. an MNR value lower than the VP values in the TTI blocks is likely to be erroneous), and resolution of non-compliant values by an Document Processing Context.

The Time Code: First in-cue field should contain the same decoded TCI time code value as the first TTI block in the STL file that has been used for a genuine subtitle. This field is not mapped into EBU-TT, but might be used to identify the presence of a 'subtitle-zero'.

Note: In practise this value may not be reliable and correct identification of 'subtitle zero' is the responsibility of the Document Processing Context.

3.3 CPN – Code Page Number

The text in the GSI block is assembled using characters selected from one of a restricted range of standard code pages. Note: Other code pages may be used within a given national environment (e.g.: Greek code page 928).

The code page used in the GSI block is identified by the CPN field of the GSI block (the first three bytes of the GSI). To correctly convert an STL file document instance, the CPN field will need to be decoded before any further conversion process of the GSI block commences.

The remaining content of the GSI shall be converted to Unicode for use within EBU-TT documents using the conversion tables defined by the Unicode Consortium.

http://unicode.org/Public/MAPPINGS/VENDORS/MICSFT/PC/CP437.TXT
http://unicode.org/Public/MAPPINGS/VENDORS/MICSFT/PC/CP850.TXT
http://unicode.org/Public/MAPPINGS/VENDORS/MICSFT/PC/CP860.TXT
http://unicode.org/Public/MAPPINGS/VENDORS/MICSFT/PC/CP863.TXT
http://unicode.org/Public/MAPPINGS/VENDORS/MICSFT/PC/CP865.TXT

Code pages used within a given national environment shall also be converted to Unicode using the conversion tables defined by the Unicode Consortium where available.

Note: The code page used for the GSI block has no relationship with the language of the target audience of the subtitles or the character code set used in the TTI blocks of the STL document.

3.4 Disk Format Code (DFC)

The STL format supports television frame-rates of 25 and 30 frames per second. This is indicated by the value of the DFC field, as follows:

Disk Format Code string value 8 characters as bytes[39]	Frames per second
STL25.01	25
STL30.01	30

The generated EBU-TT documented shall have the mandatory `ttp:frameRate` attribute value in

[39], [42] The byte values for these characters are identical in all code pages used for the GSI block.

the `tt:tt` element set to either "25" or "30" respectively. The mandatory `ttp:frameRateMultiplier` attribute value shall be set, to "1 1" or "1000 1001" respectively.[40]

The generated EBU-TT document should also set the `ttp:dropMode` attribute value in the `tt:tt` element as determined by the DFC field as indicated by the following table.

Disk Format Code string value 8 characters as bytes[42]	`ttp:dropMode` attribute value
STL25.01	"nonDrop"
STL30.01	"dropNTSC" | "dropPal"[41]

These values may be replaced by the Document Processing Context, e.g. where private disk format codes have been used. These private codes may indicate framerates, e.g. 23, 24, 29, 50, 59, 60, (or target display resolutions) other than those anticipated by the STL standard and the conversion strategies this document describes. The interpretation of non-compliant Disk Format Codes should be made by a Document Processing Context to determine appropriate `ttp:frameRate` and `ttp:frameRateMultiplier` attribute values. For this type of non-conformant file, the processing and conversion of timing related data as described by the conversion strategies in this document should then be made with consideration to the determined framerate.

3.5 Display Standard Code (DSC)

The STL file format identifies one of four intended display modes[42] for the text content by using the DSC code. The `ebuttm:documentIntendedTargetFormat` attribute value should ideally have a value provided by the Document Processing Context.

Display Standard Code (DSC)	Display standard	Hex representation (DSC field value)
Blank	Undefined	20h
0	Open subtitling	30h
1	Level-1 teletext	31h
2	Level-2 teletext	32h

If the value of the DSC field indicates either Level-1 Teletext or Level-2 Teletext[43] as an intended display mode, then the subtitle content in the STL file may have been created with an underlying assumption of the characteristics of the Teletext system. I.e. a mono-spaced Teletext font and a grid based presentation with dimensions of 40 by 24 cells. STL files with these DSC field values may be candidates for a strict conversion using mono-spaced fonts and optionally by retaining 'positioning' space characters contained within separate `tt:span` elements..

Note: Private Display Standard Codes have also been used. STL documents with non-compliant Display Standard Codes are not covered by the conversion strategies in this document.

3.5.1 Undefined and Open Subtitling

For the other values of the DSC field defined by the STL standard, (Undefined and Open Subtitling), there is no underlying assumption for the presentation of the subtitle content in the STL file. For these STL files, either a mono-spaced font presentation, or a proportional font presentation may have been assumed by the STL file author. For the conversion of this type of STL subtitle file, these rendering / presentation assumptions should be provided by the Document Processing Context since

[40] The STL specification is ambiguous as to the intended target use of files that use a DFC value of "STL30.01". However, contemporary video formats in use at the time of specification (NTSC and PAL-M) used a notional 30 frames per second, both formats having an actual framerate of 29.970029970029… (=30 x 1000/1001).

[41] The dropNTSC value is the default value; however the Document processing Context may determine the use of the "dropPAl" value for subtitle files (originally) intended for use with PAL-M encoded video assets.

[42] It should be understood that the STL subtitle file format has been used as a container file format for subtitles intended for use with video stored in various resolutions/aspect ratios and framerates.

[43] Note: There is no content distinction between Level-1 and Level-2. The DSC codes may indicate the character set used is restricted in level-1, but no guarantee of content restriction is given.

typically, when a hardware 'subtitle inserter' directly renders this type of STL file, the font and other information is provided as part of the device configuration. The text, timing and horizontal justification information in these files can however be converted without this Document Processing Context (although unintended text wrapping may occur due to incorrect assumptions about font size).

Note: Vertical subtitle placement in the 'open' form is sometimes recorded in an Undefined or Open Subtitling STL file in a manner that is atypical, or non-compliant with EBU Tech 3264.

The Maximum Number of Displayable Rows (MNR) value should contain the total number of ***possible*** rows that could be simultaneously displayed.[44] In some Undefined or Open Subtitling STL files this field may be used to signal the number of unique row positions that might be used in the TTI blocks, for example a value of '99' may signal that the subtitle positions used within the file have a granularity of 100 (*0 – 99)* – even though any assumed font size (for readable text) would not permit 100 rows to appear on screen simultaneously without overlap.

Consequently in the EBU Tech 3264 specification, for Undefined or Open Subtitling STL files, the size of the font used for presentation (which dictates the row height) is an unresolved ambiguity. It is recommended that for files with a large MNR value (e.g. '99') the font size (height) should be defined as ~1/15 of the 'Subtitle Safe Area' and a lineHeight of 120% is used to achieve a row height of ~1/12 of the height of the 'Subtitle Safe Area'.[45]

Note: In some non-compliant files, the MNR may signal the maximum number of rows that have been used by any single subtitle in the file, e.g. a value of '3' might signal that the biggest (tallest) subtitle has 3 rows!

A conversion implementation may compare the MNR field value and the values for the vertical position (VP) field in each TTI block to determine what the MNR field value represents in an STL file. A comparison of these values may indicate how the MNR value should be interpreted.[46] The most appropriate strategy for interpretation of non-compliant STL files may require additional information from a Document Processing Context.

3.6 Language Code (LC)

The text in the TTI blocks of an STL file are prepared for a specific language, identified by the Language Code indicated by the LC field in the GSI block. The codes in the LC field correspond to those adopted by the MAC/packet family of systems.[47] These codes should be mapped to an `xml:lang` attribute specified on the `tt:tt` element in accordance with the table in **Annex C** or an `xml:lang` value supplied by the Document Processing Context should be used.

3.7 Character Code Table (CCT) number

The text in the TTI blocks of an STL file are assembled using characters selected from one of five ISO Standard character code tables. The character code table in use is indicated by the CCT number, in accordance with the table below. To correctly convert an STL file document instance, the CCT field will need to be decoded before any further conversion process of the TTI blocks

[44] In effect the MNR is the 'divisor' for the 'Subtitle Safe Area' height, creating a line height for each row.

[45] This is because in general open subtitles are burned into video with proportional fonts rather than Teletext style '100% of line height' fonts. The 'logical' alternative of dividing the MNR value by the 'Subtitle Safe Area' height would result in a text height per row of 1% (*which would be unread-ably small*).

[46] For example if the MNR value is lower than the highest VP values in the file, an appropriate strategy might be to ignore the MNR value and simply use the VP value as an indication of the relative position of the subtitles, with the highest VP value *probably* indicating a subtitle at the bottom of the screen. However, if the file was authored with the intention of avoiding a 'lower third' persistent throughout the media, this strategy might not result in an EBU-TT file that reproduces what was intended. Note: The VP value is the position of the top of the subtitle row(s) so the number of rows in the subtitle and the assumed line height should also be taken into consideration. Alternatively, if the MNR has a value of 99, the VP field values might be interpreted as percentage offsets from the top of the 'Subtitle Safe Area'.

[47] The Language Code field is not necessarily valid in all files; specifically those with a 'subtitle zero' may not have a correct language code.

commences.

Character Code Table (CCT) 2 bytes in CCT field	Language group ISO Standard
00 (30h 30h)	Latin 6937/2-1983/Add.1:1989
01 (30h 31h)	Latin/Cyrillic 8859/5-1988
02 (30h 32h)	Latin/Arabic 8859/6-1987
03 (30h 33h)	Latin/Greek 8859/7-1987
04 (30h 34h)	Latin/Hebrew 8859/8-1988

The content of the TTI blocks should be converted to Unicode for use within EBU-TT documents using the conversion tables defined in **Annex B**.[48] The resulting Unicode strings should be 'normalised' to Unicode Normal Form C.[49]

Note: Private extensions to the STL tables have been made, e.g. to add the Euro symbol. The tables in **Annex B** are not definitive and may be enhanced by a Document Processing Context.

Note: For certain languages and character sequences, Unicode encoded strings may be shorter than their equivalent Teletext encoded strings. This may have implications, e.g. in use cases where EBU-TT documents are targeted for subsequent Teletext format subtitle broadcast, since a Unicode string may expand to a sequence of Teletext characters that exceeds the limitations of the Teletext format (i.e. 40 characters per line).

3.8 Country of Origin (CO)

The CO field should be mapped to an equivalent character string in the `ebuttm:documentCountryOfOrigin` element in accordance with the table in **Annex D** or the value should be supplied by a Document Processing Context.

The `ebuttm:documentCountryOfOrigin` element shall ***not*** be used as a substitute for the `xml:lang` attribute in the `tt:tt` element (see §3.6, Language Code (LC)).

3.9 Time Code: Start-of-Programme (TCP)

If the Time Code Status byte has a value of 31h then the contents of the Time Code: Start-of-Programme field (8 ASCII numeric characters [HHMMSSFF] representing a time code value 00:00:00:00 - 23:59:59:29 decimal) should be converted into a string[50] and placed into an `ebuttm:documentStartOfProgramme` element. See also §1.2.4 Time code source. The Time Code: Start-of-Programme field is sometimes incorrectly used in STL files with a value matching the Time Code: First in-cue field value. The Time Code: Start-of-Programme field should contain a time code value for the **first frame** of the associated video content.[51]

3.10 User-Defined Area (UDA)

The contents of the User-Defined Area field (576 characters) should be converted into a BASE64 encoded string and placed into an `ebuttm:documentUserDefinedArea` element.

---

[48] The accented letters in the Latin-based languages in Teletext are created according to the "floating accent" principle. Column "C0" of the character code table 00 (Latin alphabet) in Annex B contains diacritical marks which are overlaid on another character in the same presentation position. Each single accented character intended for presentation occupies two bytes, and the diacritical mark is sent first (e.g. Ä = C8h 41h, ê = C3h 65h). **This is opposite to the order used in Unicode where combining character(s) follow the base character.**

[49] Unicode Normal Form C is specified in the Unicode® Standard Annex #15 UNICODE NORMALIZATION FORMS (http://unicode.org/reports/tr15/)

[50] Time code fields containing an EBU/SMPTE time-and-control code value in the GSI block of STL files are encoded using 8 ASCII encoded bytes in the sequence HHMMSSFF. E.g. A time code of 10:01:56:27 should be encoded as the hex bytes 31h, 30h, 30h, 31h, 35h, 36h, 32h, 37h.

[51] Video media is typically 'striped' with a time code starting from an arbitrary whole hour value for the first frame of the video programme (e.g. 3:00:00:00 or 10:00:00:00). In video media with a run-in (e.g. video tapes), the preceding frames will have 'earlier' time code values leading up to this first (programme) frame.

Note that trailing spaces may occur in the UDA field and should not be converted.

3.11 General GSI field conversion

The contents of the following 32 character STL document GSI fields should be converted through the code page identified by the CPN field of the GSI block into Unicode and placed into the corresponding element in the EBU-TT document as shown in the table below. Trailing space characters should be removed.

Original Programme Title (OPT)	`ebuttm:documentOriginalProgrammeTitle`
Original Episode Title (OET)	`ebuttm:documentOriginalEpisodeTitle`
Translated Programme Title (TPT)	`ebuttm:documentTranslatedProgrammeTitle`
Translated Episode Title (TET)	`ebuttm:documentTranslatedEpisodeTitle`
Translator's Name (TN)	`ebuttm:documentTranslatorsName`
Translator's Contact Details (TCD)	`ebuttm:documentTranslatorsContactDetails`
Subtitle List Reference Code (SLR)[52]	`ebuttm:documentSubtitleListReferenceCode`
Publisher (PUB)	`ebuttm:documentPublisher`
Editor's Name (EN)	`ebuttm:documentEditorsName`
Editor's Contact Details (ECD)	`ebuttm:documentEditorsContactDetails`

The `ebuttm:documentCreationDate`, `ebuttm:documentRevisionDate` and `ebuttm:documentRevisionNumber` elements are used to hold the creation date, revision date and revision number of the EBU-TT document respectively. The `ebuttm:documentRevisionDate` and `ebuttm:RevisionNumber` elements should be updated on every change to the EBU-TT document.

3.12 Maximum Number of Displayable Characters (MNC)

The contents of the Maximum Number of Displayable Characters field (2 ASCII numeric characters[53] [NN] representing 0-99 decimal) should be converted into a non-negative integer and placed in an `ebuttm:documentMaximumNumberOfDisplayableCharacterInAnyRow` element.[54]

Note: A leading space may occur in the MNC field.

Note: The original value from the STL file may be no longer correct after a TTI block is converted (E.g. the number of characters in a row may be reduced by the collapsing [removal] of control character whitespaces). The Document Processing Context may 'direct' a recalculation of the value of this field and set a new calculated value for the `ebuttm:documentMaximumNumberOfDisplayableCharacterInAnyRow` element after the conversion of all subtitles in the source STL file.

3.13 Total Number of Subtitles (TNS)

The contents of the Total Number of Subtitles field (5 ASCII numeric characters[55] [NNNNN] representing the values 0-99999 decimal) should be converted into a non-negative integer and placed into an `ebuttm:documentTotalNumbersOfSubtitles` element.

Note: Leading spaces may occur.

Note: There may be fewer subtitles in the generated EBU-TT document compared to the

[52] This field ONLY has **16** characters.

[53] 30h to 39h. E.g. the value '40' is encoded as 34h, 30h.

[54] For *Teletext* subtitles this value in the STL file has minimal meaning as the maximum number of characters on a Teletext subtitle row is fixed by the presentation technology (to 40 characters). This element value has greater relevance for Open or Undefined Subtitling where it may indicate a 'target' maximum character count for rows.

[55] 30h to 39h. E.g. the value '275' is encoded as 20h, 20h, 32h, 37h, 35h.

original STL after a conversion (e.g. due to the conversion of subtitle zero into header content). The Total Number of Subtitles is no indication of how many `tt:p` elements may be present in a generated EBU-TT document. The Document Processing Context may direct the processing implementation to set the `ebuttm:documentTotalNumbersOfSubtitles` element value to the number of subtitles in the generated document.

3.14 Creation Date (CD) and Revision Date (RD)

The contents of the creation date and revision date fields (6 ASCII Numeric characters representing YYMMDD)[56] from the STL document may be converted into `xs:date` values and placed into `ebuttm:stlCreationDate` and `ebuttm:stlRevisionDate` elements respectively.[57] These elements shall not be used if original source STL documents are tunnelled as binary data.

Note: Creation dates and revision date fields in STL files are often absent (and sometimes invalid)!

3.15 Revision Number (RN)

The contents of the revision number field (2 ASCII Numeric characters[58] [NN] representing the values 0-99) may be converted into a non-negative integer and placed into an `ebuttm:stlRevisionNumber` element.

Note: A leading space may occur in the RN field in the STL document.

This element shall not be used if original source STL documents are tunnelled as binary data.

```
…
<tt:metadata>
  < ebuttm:conformsToStandard>urn:ebu:tt:exchange:2017-05</ ebuttm:conformsToStandard>
  < ebuttm:conformsToStandard>urn:ebu:tt:exchange:stl-mapping:2017-05</ ebuttm:conformsToStandard>
  >
  <ebuttm:documentCreationDate>2012-12-04</ebuttm:documentCreationDate>
  <ebuttm:documentRevisionDate>2012-12-04</ebuttm:documentRevisionDate>
  <ebuttm:documentRevisionNumber>1</ebuttm:documentRevisionNumber>
  <ebuttm:stlCreationDate>2010-01-04</ebuttm:stlCreationDate>
  <ebuttm:stlRevisionDate>2011-11-10</ebuttm:stlRevisionDate>
  <ebuttm:stlRevisionNumber>7</ebuttm:stlRevisionNumber
</tt:metadata>
…
```

4. Conversion of the TTI Blocks

An EBU-TT document should contain the styling and layout information needed by a subtitling device or processing application to correctly render the contained subtitles. In accordance with EBU Tech 3350, layout and styling information shall be placed in the head of an EBU-TT document and referenced by the subtitles defined by `tt:p` and `tt:span` elements in the body of the EBU-TT

[56] Using the characters 30h to 39h. E.g. the value '961011' representing the YYMMDD date 11th October 1996 is encoded as 39h, 36h, 31h, 30h, 31h, 31h.

[57] Note: Year values between 80 and 99 in the STL file should be interpreted as representing years from the range 1980 to 1999 respectively. Year values less than 80 should be interpreted as representing years from the range 2000 to 2079. EBU Tech 3264 references ISO 8601 [12], which was subsequently revised to remove 2 digit years in ISO 8601:2004. ISO 8601:2004 does not provide an algorithm for calculating the implied century. The above rule may be over-ridden by the Document Processing Context.

[58] 30h to 39h. E.g. the value '1' is encoded as 20h, 31h (*or as 30h, 31h)*, the value '15' is encoded as 31h, 35h.

document.[59] The subtitle text contained within a TTI block is typically converted into a single `tt:p` element (see § 4.4).[60] Carriage returns that occur in the TTI block subtitle text should be converted into one or more `tt:br` elements. Row positions and row spacing are defined by the font metrics and the lineHeight attribute value.

For strict equivalence to the original Teletext presentation in the output EBU-TT document a lineHeight of "1c" and a fontSize of "1c" (assuming a cell resolution value to accommodate a 40 x 23 'Subtitle Safe Area') is recommended. If a 'relaxed' presentation is desired, while still occupying the same vertical space, then a lineHeight of "1.25c" and a fontSize of '0.8c' (assuming the same cell resolution) might be used. This latter recommendation would result in smaller text than a typical Teletext subtitle presentation (where rows of larger text are typically closer than font metrics would normally specify. See § 4.1.1 for more information on fontSize and lineHeight values and their implications on presentation. It should also be recognised that the choice of fontSize and lineHeight values may need to take into consideration any intended final subtitle distribution formats and the limitations of target display devices or rendering processors.

The Text and Timing Information (TTI) blocks in the source STL document contain the subtitle text together with timing and positional data for that subtitle and may also contain additional subtitle data or user-specific data. A subtitle is defined by a set of one or more TTI blocks where each TTI block has the same unique Subtitle Number (SN). Where multiple TTI blocks are used for a subtitle they should be combined into a single subtitle (i.e. a single `tt:p` element) in the EBU-TT output file. All undefined values in the TTI block ignored and are not converted into EBU-TT content.

Within an EBU-TT document, information about the desired presentation of the subtitle text is recorded by embedding references in the text content to style definitions that are declared within the `tt:head` element of the EBU-TT document. The desired position of each subtitle is also recorded by a reference to a defined `tt:region` element. This contrasts with the STL format, where the style information is explicitly signalled by using control codes in each separate subtitle text, and the position of each subtitle is similarly signalled by value in the VP field. Consequently sequential conversion of an STL file to an EBU-TT file requires that the style and region data within the `tt:head` element of the EBU-TT file is updated with new styles or regions (if necessary) as each TTI block is processed.

4.1 Style in tt:head

The `tt:styling` element in the head section of the output EBU-TT document shall contain at least one `tt:style` child element which shall contain an `xml:id` attribute with a unique value, e.g. "defaultStyle".[61] It is recommended that this 'default style' should be a fully defined style, i.e. the style element should contain an explicit attribute value for all inheritable style attributes supported by EBU-TT:

`tts:fontSize`	`tts:fontFamily`
`tts:textAlign`	`tts:lineHeight`
`tts:backgroundColor`	`tts:color`
`tts:fontWeight`	`tts:fontStyle`
`tts:textDecoration`	`tts:wrapOption`

The recommendation of using a fully defined style avoids any confusion over the 'initial values' of the style attributes. An EBU-TT document with fully defined 'base' styles is also immune to changes

[59] The re-use of styles used throughout the EBU-TT document instance is encouraged.

[60] Where 'half row spacing' is used between lines of a subtitle the text may be split across several p elements.

[61] The XML attribute `xml:id` is not only used by the `tt:style` element, but also by the `tt:tt`, `tt:region` and `tt:p` elements and is permitted on other elements. By definition, (see the W3C Specification Extensible Markup Language (XML) 1.0 and XML Schema Part 2: Datatypes), values of `xml:id` attributes must be unique in the entire document. This means that although `tt:style` and `tt:region` are different elements (with a different meaning) it is not valid to give them the same id (e.g. they cannot both have the id: "default").

in the default initial attribute values set by the EBU-TT or TTML standards, changes that may occur following a revision of those standards.

As each subtitle is converted, the relevant style, as encoded by control codes within the TTI blocks, is identified and is either matched to an existing style definition in the header or used to create a new style definition.

Note: Different sequences of control codes within different TTI blocks may encode identical presentation requirements.

Different approaches to referencing styles are used in EBU-TT documents, the recommended approach for converted STL documents is to inherit styles within the body rather than using referential styles within the definitions of the `tt:style` elements. This is illustrated in the following two examples:

```
…<tt:styling>
  <tt:style xml:id="defaultStyle" … />
  <tt:style xml:id="derivedStyle" style="defaultStyle" …/>
…
</tt:styling>
…
<body>
…
  <tt:p style="defaultStyle">
  …
    <tt:span style="derivedStyle">Text in non-default style</tt:span>
  </tt:p>…
```

In the above example the ‘derived style’ definition explicitly contains all the attributes of the referenced ‘default style’. The above approach is not recommended by this document, though it would form valid EBU-TT. The following example illustrates the recommended approach:

```
…<tt:styling>
  <tt:style xml:id="defaultStyle" … />
  <tt:style xml:id="adjustedStyle" [some attributes set here]…/>
…
</tt:styling>
…
<body style="defaultStyle">
…
  <tt:div>
    <tt:p>
      <tt:span>Text in default style</tt:span>
      <tt:span style="adjustedStyle">Text in default style with just adjusted attributes applied</tt:span>
    </tt:p>
  </tt:div>…
```

In the above example, the adjusted style can be applied to a number of different span elements. Each span element may have a differing base style inherited from its own containing element (in this example the div element) and the application of the ‘adjustedStyle’ reference will only change the attribute values that are defined within the adjusted style definition.[62] As TTI blocks are processed, solitary control codes may result in the creation of these additional style definitions that just alter a single style attribute, or a sequence of control codes (such as those typically found

[62] The ‘adjustedStyle’ definition may just change a single attribute, e.g. italics, or text colour.

at the start of subtitle lines) might create additional style definitions that define several style attributes.

The `tts:backgroundColor` attribute of the 'default style' shall be set to a value of transparent since the attribute applies to the entire extent of the container region it is applied to. A non-transparent value for backgroundColor cannot be applied to the region referenced by a subtitle since the background color would fill the entire area of any region referenced by the subtitle, when a subtitle was active. Typically Teletext subtitles are displayed with a black background behind just the rendered characters. Therefore for subtitles that are identified as Teletext subtitles in the source STL document at least one additional style should be defined to set the background colour to black and this style would then be referenced from a `tt:p` element or a `tt:span` element containing subtitle text.

Note: Applying a non-transparent background colour to a `tt:p` element will result in that background colour being applied to the entire width of the region of the row(s) containing the subtitle text content of the `tt:p` element; it will not be constrained to the area just behind the characters. Assigning the background colour to a `tt:p` element (by a style reference) may be useful if the intention is to reproduce a 'stripe' subtitle effect – although the stripe will be restricted to the width of the region.

The initial 'default style' shall be referenced by the `tt:body` element, or by a parent `tt:div` element that encloses all `tt:p` elements that contain subtitle content, thus establishing a common line height and font size for all content.

Note: The line height and font size attribute values have significant implications on the interpretation and conversion of vertical positioning.[63]

[63] See the section regarding conversion of the Vertical Position (VP) field in the TTI block.

```
…<tt:styling>
  <tt:style xml:id="defaultStyle"
    tts:textDecoration="none"
    tts:fontWeight="normal"
    tts:fontStyle="normal"
    tts:backgroundColor="transparent"
    tts:color="white"
    tts:textAlign="center"
    tts:fontFamily=”monospaceSansSerif”[64]
    tts:fontSize=”1c”[65]
    tts:lineHeight=”1c”[66]
    tts:wrapOption=”noWrap”/>[67]
…

  <tt:style xml:id="doubleHeightWhiteOnBlack"
    tts:backgroundColor="black"
    tts:color="white"
    tts:fontSize="2c"
    tts:lineHeight=”2c”[68]
/>
…
</tt:styling>…
```

4.1.1 Font definition in tt:head

The initial style definition (the ‘default style’) should be defined to match the default presentation of an STL file when no additional control codes are present within the TTI block. The default presentation of Teletext characters in Teletext pages is ‘single height’ monospaced white text on the default background (the default background for a Teletext magazine page is black). Teletext characters are rendered starting from the left margin by a European Teletext decoder. No control code is required to switch the background on for Teletext Magazine page rows – this allows a full 40 characters per line. However, subtitle pages should be displayed with ‘Boxing’ as they are presented in ‘mix mode’ over the programme video. In ‘mix mode’ the default background is transparent.

The `tts:fontSize` attribute should be set to a value of ‘1c’ to correctly reflect the baseline size of Teletext characters (in combination with the `tts:lineHeight` attribute – see below), and to allow for positioning (if using `tt:br` elements in a ‘simple region strategy’) at any valid Teletext row location. If the cell resolution is set to an appropriate value, this will result in a baseline font size that accommodates the display of 40 by 23 characters within any chosen ‘subtitle safe area’.

Note: Differences between the original monospaced Teletext character presentation and typical modern proportional spaced EBU-TT font rendering will mean that the rendered text width may differ substantially – depending on the actual browser or player font specified. It may be desirable for implementations to select a font that has the appropriate metrics/aspect ratio.

[64] The `tts:fontFamily` attribute value defined by the ‘default style’ will be dependent upon the conversion strategy used. To preserve a Teletext presentation style, the ‘default style’ should specify a mono-spaced font.

[65] The `tts:fontSize` attribute value for open / undefined subtitles **may** be supplied by the Document Processing Context.

[66] The `tts:lineHeight` value for the open / undefined subtitles **may** be supplied by the Document Processing Context.

[67] In general subtitle files do not use line wrapping. Text wrapping in any region should only occur as a result of poor choice of font size or region extent.

[68] The `tts:lineHeight` value for the open / undefined subtitles **may** be supplied by the Document Processing Context.

However, it is common practise for Teletext subtitles to be displayed in 'double height' by preceding the text to be displayed with a control character. Consequently, it is likely that either a further style will be required, containing a fontSize attribute value of '2c' and a lineHeight attribute value of "2c", or the implementation may choose to define the 'default style' with a fontSize attribute value of '2c' and a lineHeight attribute value of "2c".[69]

Some STL files exist where it appears that double height presentation was intended, but the double height control code is omitted. A Document Processing Context might identify forensic characteristics of these files; E.g. the use of two <crlf> codes between multi-line subtitles (e.g. "Hello<crlf><crlf>World"); or the use of VP values that differ by 2 for single line and dual line subtitles (e.g. VP 22 for single line and VP 20 for two line subtitles). Technically, these files conform to the EBU Tech 3264 specification but unless 'corrected', conversion would result in unintended presentation of the EBU-TT converted document (small text with extra lines between rows).

The `tts:fontFamily` attribute of the 'default style' should be set to a font family determined by the Document Processing Context. If strict equivalence to a Teletext presentation is required then a mono spaced Sans Serif font that has the desired characteristics matching contemporary Teletext fonts, particularly aspect ratio and readability when displayed over video, should be identified.[70]

An 'ideal' Teletext font might be considered to have the following characteristics:

- Monospaced (i.e. the font advance for all glyphs is 1 em)
- Is sans-serif
- Meets the design criteria of the original Teletext fonts, e.g.
  - Maximises distinctions between numerals and letters (e.g. '1' , 'l' and '5' ,'S')
  - Avoids single pixel horizontal strokes that may flicker on interlaced displays.
  - All glyphs fit within the em cell.
  - Every glyph is designed with sufficient blank space around it such that glyphs can be rendered next to each other and on consecutive lines (with a lineHeight of 1c) without touching each other.

An 'ideal' double height Teletext font (e.g. for use when the fontSize attribute is set to '2c') might be considered to have the following additional / changed characteristics:

- Glyphs that fit within a 1 x 0.5 em cell, with a glyph advance of 0.5 em. I.e. the font is specifically designed with asymmetric vertical scaling for 'double height', 'single advance'.
- A 'font advance' for each glyph of 0.5 em – since the design space for a font glyph is square, but the desired font should be 'double height' but 'single advance'.

The logical characteristics of the font used in the generated document may be signalled by including one or more `ebuttm:font` elements as children of a `metadata` element within the head of the document.

[69] Note: The use of fontSize '2c' in the 'default style' has significant implications for subtitle row positioning when using the 'single region strategy' (which uses the tt:br element to move text within a region), since the 'computed line height', and thus the row spacing (of empty rows), will *by default* be twice the prototypical Teletext row height. Half row positioning in this case would require overriding the 2c font size established by the 'default style'.

[70] Sans serif fonts (sometimes called 'gothic' fonts) are recommended for readability and clear display.

```
<tt:tt … >
<tt:head>
  <tt:styling>
    <tt:metadata>
      <ebuttm:font fontFamilyName="monospaceSansSerif" src="urn:ebu:tt:font:teletext"/>
…
    </tt:metadata>…
  </tt:styling>
</tt:head>
<tt:body> …
</tt:body>
</tt:tt>
```

A real world font resource may additionally be specified, allowing the identification of which font was assumed when converting the STL source document:

```
<tt:tt … >
<tt:head>
  <tt:styling>
    <tt:metadata>
      <ebuttm:font fontFamilyName="monospaceSansSerif" src="urn:ebu:tt:font:teletext"/>
      <ebuttm:font fontFamilyName="monospaceSansSerif" src="http://path.to.font.woff"/>
…
    </tt:metadata>…
  </tt:styling>
</tt:head>
<tt:body> …
</tt:body>
</tt:tt>
```

In accordance with EBU-TT Part 1 (EBU Tech 3350) the first entry describes the logical characteristics of the font, and the second entry may be resolved into a real world font.

It is recommended that when a font with Teletext characteristics is used as a constraint during the conversion process, that this is also signalled in any `ebuttm:appliedProcessing` child element by including an `ebuttm:stlParameter` element with an 'key' attribute value of "teletextStyleFont" and an element value of "true" (see § 2.2).

The `tts:lineHeight` attribute of the 'default style' should be set to a value of "1c", if strict equivalence to a Teletext presentation in terms of line (row) spacing is required (assuming a font with Teletext characteristics is selected), since Teletext rendering has a closer line spacing than normal printed text.

4.1.2 Right to Left language subtitles

Note: There is no clear guidance in the STL standard (EBU Tech 3264) regarding the encoding of text for 'right to left' language subtitles. Examination of example STL files for 'right to left' language subtitles indicates that the STL code points for text are stored in logical (i.e. reading) order not presentation order.

It is recommended that if the value of the LC field indicates a language where writing starts from the right of the page, and continues to the left, the 'default style' `tt:style` element should be a fully defined style and the `tts:writingMode` attribute of all regions should have the value of "`rltb`" (right to left top to bottom).

It is anticipated that further guidance on the mapping of right to left language STL files will be included in later versions of this document.

4.2 Layout in tt:head

In accordance with EBU Tech 3350, the head section of an EBU-TT document shall contain one `tt:layout` element with at least one `tt:region` child element. A `tt:region` element's attributes define an area in the active video where subtitle content is to be placed. To position the subtitle over the image, each `tt:p` element references a `tt:region` element. All regions created in an EBU-TT document instance should be fully defined with all attribute values supported by EBU-TT specified:

tts:displayAlign	tts:padding
tts:writingMode	tts:origin
tts:extent	tts:showBackground
tts:overflow	

The recommendation of using a fully defined region avoids any confusion over the 'initial values' of the region attributes. An EBU-TT document with fully defined regions is also immune to changes in the default initial values of the attributes as specified by the EBU-TT or TTML standards, changes that may occur following a revision of those standards.

This example 'defaultRegion' below is defined to cover an assumed 'Subtitle Safe Area'. A region with this definition might be used in a 'simple region strategy', with any vertical (upward) positioning of subtitles achieved by using `tt:br` elements after the text (e.g. Figure 6). The following values would define a ~91% by ~85% 'Subtitle Safe Area':[71]

```
…<tt:layout>
    <tt:region
    xml:id="defaultRegion"
    tts:displayAlign=”after”
    tts:padding=“0c”
    tts:writingMode=”lrtb”
    tts:origin=”4.5% 7.5%”
    tts:extent=“91% 85%”
   tts:showBackground=”whenActive”
   tts:overflow=”visible”/>
  </tt:layout>…
```

The above example could also be defined using 'cell' values for the region origin and extent attributes. If the cell resolution value had been defined by taking a 'Subtitle Safe Area' into account, e.g. using values from Appendix E, for a ~91% by ~85% 'Subtitle Safe Area' of 40 x 23 within a 44 x 27 cell resolution, the region origin would be '2c 2c' and the extent '40c 23c'.

4.3 The default style and region for divs

The `tt:body` element of the output EBU-TT document shall reference the 'defaultStyle' style definition using a `tt:style` attribute and shall contain at least one `tt:div` element that contains all the `tt:p` elements generated by the subtitle content. This `tt:div` element may have an `xml:id` value of 'defaultDiv' and shall inherit the 'defaultStyle' reference from the parent `tt:body` element. Where subtitles are grouped in the STL file with different Subtitle Group Numbers, multiple `tt:div` elements, each inheriting the 'defaultStyle' reference and collectively containing all the `tt:p` elements generated by the subtitle content, may be used to reflect the original grouping.

[71] Other suggested values for mapping the 'Subtitle Safe Area', e.g. for 16:9 presentation, may be found in **Annex E**.

4.3.1 Subtitle Group Number (SGN)

One or more subtitles in an STL file may be grouped together, for example to establish a distinction between subtitles relating to different parts of a single programme. The group is identified by a common Subtitle Group Number (SGN) in the TTI block.[72] The subtitles in a subtitle group should be stored in a continuous ascending Subtitle Number order in the EBU-TT file. Subtitles with matching Subtitle Group Numbers may be placed as child `tt:p` elements within a containing `tt:div` element.[73] A single `tt:div` element should include all the `tt:p` elements created from TTI blocks with a common SGN number. Each `tt:div` element may contain an `xml:id` that holds the Subtitle Group Number converted into a string, for example prefixed by 'SGN' to ensure required uniqueness of the `xml:id` value.

```
… <tt:body style="defaultStyle">
  <tt:div xml:id="SGN0">
    <tt:p …>
    <tt:p …>
    …
  </tt:div>
  <tt:div xml:id="SGN1">
    <tt:p …>
    <tt:p …>
    …
  </tt:div>
</tt:body>…
```

4.3.2 Subtitle Number (SN)

Each TTI block in the STL source subtitle file consists of 128 bytes and a subtitle is defined by a set of one or more TTI blocks. Each subtitle is identified by a unique number stored in the Subtitle Number (SN) field (as a binary 2 byte value) in the one or more TTI blocks used to hold the subtitle data. The SN field is not mapped into EBU-TT.

Individual `tt:p` elements in the EBU-TT document may contain an `xml:id` attribute. These may contain a mapping of the subtitle number from the STL document as determined by the Document Processing Context, in accordance with the uniqueness requirements for `xml:id` values or the subtitle number could be converted into data stored in a private namespace contained within a child `tt:metadata` element, which must be the first child element of the containing `tt:p` element.

4.3.3 Extension Block Number (EBN)

The Extension Block Number field is only internally relevant to the STL file and is not relevant for conversion into an EBU-TT document. This field is not mapped into EBU-TT with one exception: If the Extension Block Number has the value FEh the TTI block contains proprietary User Data. The contents of such TTI blocks may be converted into BASE64 and stored within a binaryData element that is contained within a `tt:metadata` element that is itself a child of a `tt:p` element created for the TTI block(s) (see § 4.4).[74] The binaryDataType attribute of the `ebuttm:binaryData` element should be set to "STL User Data". If more than one TTI block that shares the same SN field value contains proprietary User Data, those blocks' data may be placed into individual `ebuttm:binaryData` elements or concatenated into a single `ebuttm:binaryData` element.

[72] The SGN field is a single byte ranging in value from 00h to FFh.

[73] Note that the time code values of one subtitle group can be different from, or equal to, the time code values of other groups.

[74] This `tt:metadata` element must be the first child element of the containing `tt:p` element.

```
<tt:tt … >
<tt:head>…
</tt:head>
<tt:body>
  <tt:div…>
    <tt:p…>
    <tt:metadata>
      <ebuttm:binaryData textEncoding="BASE64" binaryDataType = "STL User Data" >
TWFuIGlzIGRpc3Rpbmd1aXNoZWQsIG5vdCBvbmx5IGJ5IGhpcyByZWFzb24sIGJ1dCBieSB0aGlz
IHNpbmd1bGFyIHBhc3Npb24gZnJvbSBvdGhlciBhbmltYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg
dGhlIG1pbmQsIHRoYXQgYnkgYSBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpbiB0aGUgY29udGlu…
      </ebuttm:binaryData>
    </tt:metadata>
…
    </tt:p>
  </tt:div>…
</tt:body>
</tt:tt>
```

4.4 TTI Blocks and tt:p Element Generation.

The subtitle text contained within a TTI block is typically converted into a subtitle represented by a single `tt:p` element (for the subtitle) in the generated document. A single subtitle in the source STL file may span multiple TTI blocks. All the TTI blocks for a single subtitle (identified by a common subtitle number (SN value)) are sequentially converted into the content of a single `tt:p` element. TTML permits multiple `tt:p` elements to reference a single region, with line order in the resulting presentation dictated by the 'order' in which the `tt:p` elements occur in the document. To simplify the generated document and for greater control of line spacing, this document recommends that multiple line subtitles (and multi TTI block subtitles) in the source STL file are converted into a single `tt:p` element with each carriage return that occurs in the TTI block subtitle text converted into a `tt:br` element.[75] The positions of each line of rendered text from the single `tt:p` element are then defined solely by the font metrics and the lineHeight attribute values.

4.5 The TTI Block Metadata

Where multiple TTI blocks are used for a subtitle (but NOT for a cumulative subtitle – see § 4.4.3), only the first TTI block of the subtitle carries relevant 'metadata' information in bytes 4-15 (CS, TCI, TCO, VP, JC and CF) of the TTI block.

4.5.1 Time Code In (TCI)

The Time Code In field of the TTI block contains an EBU/SMPTE time-and-control code value encoded as 4 BCD bytes indicating the start time ("in-cue") of a subtitle. This field should always be decoded and the value placed into the begin attribute value of the generated `tt:p` element.

Byte number	Time element	Decimal code range	Hex code range
1	hours	00..23	00h..17h
2	minutes	00..59	00h..3Bh
3	seconds	00..59	00h..3Bh
4	frames	00..24*	00h..18h*

*In the STL30.01 format, the range is 00..29 frames (00h..1Dh).

[75] In some double height Teletext STL files a single 'on screen' carriage return is represented by two carriage returns in the TTI block (see § 4.5.6.3).

4.5.2 Time Code Out (TCO)

The Time Code Out field of the TTI block contains an EBU/SMPTE time-and-control code value encoded as 4 BCD bytes indicating the end time ("out-cue") of a subtitle. This field should be decoded in the same manner as the TCI field (described above) and the value placed into the end attribute value of the generated `tt:p` element. In EBU-TT (derived from TTML) the ending point of a temporal interval is not included in the interval.

Note: Practical implementations of subtitle insertion systems may not detect and process time code values (VITC or LTC in the VBI) quickly enough to insert a subtitle in the same frame as defined by the TCI value. A similar delay may occur in processing the TCO value (out cue). Consequently subtitle presentation may be delayed (or shortened!) by a frame. Additionally a TCI with the same value as the previous TCO may be seen as an overlap in timing and therefore as an error. Implementations may also treat the situation where TCI = previous TCO + 1 as 'back to back subtitles' i.e. no gap between presentations. However, the duration of an STL subtitle is typically calculated as TCO – TCI (with due consideration for time base and drop frame issues).

4.5.3 Cumulative Status (CS)

A value in the range 01h-03h in the Cumulative Status Field of a TTI block, indicates that the text is part of a cumulative subtitle. Cumulative subtitles are also known as "add-on" subtitles, and they allow the display of a changed subtitle before the previous one has been erased from the screen.

The Cumulative Status Field (CS) value shall be interpreted according to the following table:

Position of subtitle in cumulative set	SN (2 byte binary value)	CS	Decoded TCI Example value	Decoded TCO Example value
First	n	01h	10.00.00.00	10.00.15.00
Intermediate[76]	Previous SN + 1	02h	10.00.05.00	10.00.15.00
Last[77]	Previous SN + 1	03h	10.00.10.00	10.00.15.00

Cumulative subtitles should be converted to a series of `tt:span` elements contained within a parent `tt:p` element. The Time Code In and Time Code Out fields of each TTI block that contributes to the same cumulative subtitle set (as determined by the CS field value described above) should be decoded and the value placed into the end attribute value of a generated span for each TTI block as shown in the following example:

```
<tt:body style="defaultStyle">
  <tt:div>
    <tt:p xml:id="sub Cumulative 0 - 1" style="textLeft" region="bottom">
      <tt:span begin="10:00:00:00" end="10:00:15:00" >Cumulative start, </tt:span><tt:br/>
      <tt:span begin="10:00:05:00" end="10:00:15:00" >cumulative intermediate, </tt:span><tt:br/>
      <tt:span begin="10:00:10:00" end="10:00:15:00" >cumulative end</tt:span><tt:br/>
    </tt:p>
  </tt:div>
</tt:body>
```

In cumulative subtitling, where multiple TTI blocks are used, one for each additional "add-on" segment, only the first TTI block of the subtitle carries relevant information for the Time Code Out value, the vertical position, justification code and comment flag. Subtitle text from the subsequent TTI blocks is appended to the end of the current text; a move to a new line for subsequent rows of the subtitle is defined by a carriage-return/line-feed (CR/LF) indicator. The CR/LF indicator, used to initiate the subsequent rows of the subtitle display, is conveyed by character code 8Ah.

[76] Multiple 'intermediate' TTI blocks may exist, each with the SN value incremented from the previous TTI block.

[77] The SN of the last TTI block in a cumulative set should be the SN value of the previous TTI block plus one.

An implementation may, subject to Document Processing Context, combine a cumulative subtitle set of TTI blocks into a single subtitle within a generated EBU-TT document by decoding the Time Code In field value of the first TTI block and the Time Code Out field value of the last TTI block in the cumulative subtitle set decoded and placing the values into the begin and end attribute values of a single generated `tt:p` element. This single generated `tt:p` element should contain the sequentially merged contents of the Text Fields of all the TTI blocks in the cumulative subtitle set, after decoding them using the appropriate character code table (see §4.4.7).

4.5.4 Justification Code (JC)

The JC code controls the horizontal alignment of the displayed subtitle. Four codes can be used within the TTI block.

Justification Code (JC)	Significance
00h	unchanged presentation
01h	left-justified text
02h	centred text
03h	right-justified text

- If the JC field has a value indicating left, right or centred justification, the `tt:p` element created for the subtitle shall reference style definitions that have relative `tts:textAlign` attribute values of 'start', 'end' or 'center' respectively. If an appropriate style does not exist, a style shall be created. All leading spaces and trailing spaces (including 'spaces' created by leading and trailing control characters) should be trimmed.
- If the JC field has a value of 00h (Unchanged presentation), this is often associated with an STL file that uses space characters for alignment of text within the subtitle rows. The relative positions of subtitle text should be determined by the Document Processing Context and may be preserved using one of the strategies listed below:

Note: When processing subtitle files, the processing of Teletext control codes (for colour and boxing etc.) will need careful consideration since they may also be deliberately acting as space characters.[78] In particular, control characters on all the rows of a subtitle should be assessed to determine if a control code on one row is matching a real space character on another row to effect horizontal alignment of the rows or text runs within the rows (e.g. to achieve the effect of multi-row alignment).

The Document Processing Context may determine the intended justification(s) of the text marked as 'unchanged presentation' (JC = '00h') within a subtitle by examination of all the rows of text and the positions of each row that result from leading (and possibly trailing) spaces. However, certain typical Teletext presentation styles, e.g. 'Centre Left' subtitles, are difficult to distinguish from centred subtitles. Correct identification of the intended justification may require an examination of multiple subtitles to determine if there is a common justification for the entire STL file; but the intended justification may often change on a subtitle by subtitle basis. The Document Processing Context should determine subtitle content positioning in the EBU-TT document generated from a source STL file that has a Justification Code of 00h, for example the Document processing Context may do one of the following:[79]

- 'Force' the document to centred text presentation to create an EBU-TT document more suitable for universal use. All leading spaces and trailing spaces (including control characters) should also be trimmed (See §4.5.4.1).[80]
- Preserve the spaces in the original subtitle text and reference a style definition that has a `tts:textAlign` attribute value of 'start' from the containing `tt:p` element (See §4.5.4.2).

[78] In accordance with the Teletext specification, a colour code in Teletext is drawn as a space character on the display.

[79] These strategies should be signalled using an `ebuttm:appliedProcessing` element in the document metadata.

[80] This 'forced' normalisation should be signalled using an `ebuttm:appliedProcessing` element in the document metadata.

Note: The retention of spaces from the original source file may facilitate 'round tripping' of an STL Teletext format file through an EBU-TT based workflow and back into a Teletext distribution form – by preserving the mechanism of positioning text in a Teletext display presentation using spaces.

- Adopt a 'multiRow' strategy that uses the `ebutts:multiRowAlign` attribute defined by EBU-TT part 1 (see EBU Tech 3350 Annex A: Use of `ebutts:multiRowAlign`). The determination of suitable values for the `ebutts:multiRowAlign` attribute based upon an examination of the positions of the text as determined by space characters is complex and outside the scope of this document.
- 'Interpret' the effective justification of each subtitle and assign justification to the subtitle text in a generated document that uses a simple region or minimal vertical region strategy. All leading spaces and trailing spaces (including control characters) should also be trimmed.
- Determine the effective horizontal offset of each of the subtitle texts and generate minimal regions with specific origins and extents to match the predicted text dimensions. This 'regionOffset' strategy assumes that the source STL file is a Teletext format file and that subtitles with multiple lines are justified to a common horizontal margin. Subtitles with multiple lines that have different horizontal offsets for each line would require a region for each line, and would require that each line of text from the source subtitle was contained within a separate `tt:p` element in the generated document. This is NOT recommended.

4.5.4.1 Mapping of Justification Code 00h to centred presentation

```
<tt:tt ttp:cellResolution="44 27"… >
<tt:head>
 <tt:metadata>
 </tt:metadata>…
 <tt:styling>
   <tt:style xml:id="defaultStyle" tts:fontSize="2c" tts:lineHeight="2c" .../>
        <tt:style xml:id="WhiteOnBlack" tts:backgroundColor="black" tts:color="white”/>
        <tt:style xml:id="textCenter" tts:textAlign="center"/>
 </tt:styling>
 <tt:layout>
        <tt:region xml:id="bottom" tts:origin="4.5% 7.5%" tts:extent="91% 85%" tts:displayAlign="after" />
 </tt:layout>
</tt:head>
 <tt:body style="defaultStyle">
   <tt:div>
     <tt:p xml:id="sub JC 0 - 1" style="textCentre" begin="00:00:11:06" end="00:00:13:20" region="bottom">
       <tt:span style="WhiteOnBlack">Alignment with</tt:span><tt:br/>
       <tt:span style="WhiteOnBlack">Spaces</tt:span>
     </tt:p>
     <tt:p xml:id="sub JC 0 - 2" style="textCentre" begin="00:00:14:06" end="00:00:16:20" region="bottom">
       <tt:span style="WhiteOnBlack">Centre Left Alignment</tt:span><tt:br/>
       <tt:span style="WhiteOnBlack">with Spaces</tt:span>
     </tt:p>
   </tt:div>
 </tt:body>
</tt:tt>
```

4.5.4.2 Mapping of Justification Code 00h to Preserve Spaces

```
<tt:tt ttp:cellResolution="44 27"… >
 <tt:head>
 <tt:metadata>
 </tt:metadata>…
 <tt:styling>
  <tt:style tts:textDecoration="none" tts:fontWeight="normal" tts:fontStyle="normal"
tts:backgroundColor="transparent" tts:color="white" tts:textAlign="center" tts:lineHeight="2c"
tts:fontSize="2c" tts:fontFamily="monospaceSansSerif" tts:writingMode=“lrtb” xml:id="defaultStyle" …/>
   <tt:style tts:textAlign="start" xml:id="textLeft"/>
   <tt:style backgroundColor="black" xml:id="withBackground"/>
 </tt:styling>
 <tt:layout
  <tt:region tts:extent="91% 85%" xml:id="bottom" tts:displayAlign="after" tts:origin="4.5% 7.5%"  …/>
 </tt:layout>
 </tt:head>
  <tt:body style="defaultStyle">
   <tt:div>
    <tt:p xml:space=”preserve” xml:id="sub JC 0 - 1" style="textLeft" begin="00:00:11:06" end="00:00:13:20"
region="bottom">
     <tt:span>          Alignment with</tt:span><tt:br/>[81]
```

[81] Note that a mono-spaced font and a line height value of 100% are used in this (and the preceding) example to duplicate

```
   <tt:span>          Spaces</tt:span>
  </tt:p>
  <tt:p xml:id="sub JC 0 - 2" style="textLeft" begin="00:00:14:06" end="00:00:16:20" region="bottom">
   <tt:span xml:space=”preserve”>       </tt:span>
   <tt:span style="withBackground">Centre Left Alignment</tt:span><tt:br/>
   <tt:span xml:space=”preserve”>       </tt:span>
   <tt:span style="withBackground">with Spaces</tt:span>
  </tt:p>
 </tt:div>
</tt:body>
</tt:tt>
```

Note: The example above (with preserved spaces) does not result in a background rendered behind the text for the first subtitle. This may be achieved by placing the text (not the alignment spaces) within `tt:span` elements and setting a style reference on these `tt:span` elements that defines an opaque background colour – as shown for the second subtitle.

Note: The TTML specification defines behaviour for the treatment of whitespace sequences in TTML documents, specifically setting the ‘white-space-collapse’ property to "true" by default. This property specifies the treatment of consecutive white spaces, with the overall effect that some white space characters may be discarded or turned into space characters, and runs of two or more consecutive spaces are replaced by a single space, with space characters immediately preceding linefeeds also discarded.

Authors of EBU-TT documents should be careful to avoid the generation of ‘repeated space character’ sequences within a `tt:p` or `tt:span` element as a result of converting STL positioning that uses spaces, since these ‘whitespaces’ may be removed by a processing application. The `xml:space` attribute with a value set to “preserve” should be used to signal that such ‘whitespaces’ are significant and should be retained. It is therefore recommended that all text to be rendered that includes significant whitespaces is within a containing `tt:p` or `tt:span` element that has (*or inherits)* an `xml:space`=”preserve” attribute.

4.5.5 Comment Flag (CF)

The Comment Flag is used to indicate TTI blocks which contain texts such as translator's comments, instead of subtitle data. If the Comment Flag (CF) has a value of 01h, the text Field of the TTI block contains comments not intended for transmission. It is possible for a comment to span multiple TTI blocks, with each TTI block marked as a comment and all TTI blocks having the same subtitle number.

The text content of any TTI blocks marked as a comment in an STL file may be included in the generated EBU-TT document. If included in the generated EBU-TT document, all of the text content of a TTI block marked as a comment shall be placed into a `ttm:desc` element. The `ttm:desc` element must itself be a child element of a `tt:metadata` element, which must be the first child element of the containing `tt:p` element.

When multiple TTI blocks are used in an STL file to contain a (possibly large) comment for the same subtitle number then these may be sequentially merged into a single `ttm:desc` element or multiple `ttm:desc` elements may be generated, one for each TTI ‘comment’ block.

TTI blocks that collectively form a logical subtitle within an STL file all have the same subtitle

the intended presentation effect of Teletext subtitles.

number. Typically, TTI blocks marked as comments are associated with other TTI blocks that contain subtitle data (i.e. not marked as comments) – all with the same subtitle number. However TTI blocks may exist that are not associated with subtitle data in other TTI blocks (for example a subtitle may be 'commented out' of a file by setting the comment flag in the TTI blocks that comprise the subtitle).[82]

The use of `ttm:desc` elements to record comments from a source STL file is shown in the following example.

```
…
<tt:p xml:id="ID001" begin="10:00:00:00" end="10:00:03:02">
  <tt:metadata>
    <ttm:desc>
This is a comment that appeared in a TTI Block with the same subtitle number (SN) as another TTI block that
contained the subtitle text in the span element below.
    </ttm:desc>
  </tt:metadata>
  <tt:span>This is the associated subtitle text!</<tt:span>
</tt:p>
…
…
<tt:p xml:id="ID002" begin="10:00:05:00" end="10:00:15:00">
  <tt:metadata>
    <ttm:desc>
This is a comment that is independent. It may be subtitle text that is commented out.
    </ttm:desc>
  </tt:metadata>
</tt:p>
…
```

4.5.6 Vertical Position (VP)

This decimal value byte[83] defines the vertical position of the first row of the subtitle. For both Teletext and in-vision subtitles, subsequent rows of the subtitle are created by the presence of the carriage-return/line-feed (CR/LF) indicator in the Text Field (TF). A new row is indicated in the generated `tt:p` element by the inclusion of a single `tt:br` child element at the point at which the new row occurs. A single `tt:br` element will cause the next row of text to be rendered at a row spacing determined by the 'computed line height' value. The computed line height algorithm uses, as a basis, the font size that is defined for the element containing the `tt:br` element.[84]

If expressed in cell units, the percentage font size may be calculated as follows:

Font size * 100 / vertical cell resolution.

If expressed in cell units, the percentage line height may be calculated as follows:

lineHeight * 100 / vertical cell resolution.

[82] These 'independent' comments should have a unique subtitle number field value (SN) in the STL file.

[83] The VP field value ranges from 01h to 017h, encoding the values 1 to 23 decimal.

[84] Note: the tt:br element forces a line break on the 'inline container' that is created by content on a line. The line height that applies to that inline container determines the starting position of the next row.

In STL documents the carriage return (CR) means 'move to the next subtitle row', but in practise, 1 or 2 carriage returns may be used between subtitle lines in double height Teletext subtitling. (I.e. Some implementations use a carriage return as a Teletext row return – thus two are required in double height subtitling, but other implementations use a carriage return as a subtitle line break, so only one is required regardless of the text height.) **The mode of carriage return use in the STL source shall be determined by the Document Processing Context**.

The VP field value always refers to a number of single height Teletext rows. Technically conformant EBU-STL files may exist where the MaxNumberofDisplayableRows and the VP field values are inconsistent (e.g. MaxNumberofDisplayableRows = 23 but with the TTI blocks using values less than VP = 10 or 11, but the intention is that the subtitles are at the bottom of the screen). It is a non-conformant use of the VP field for it to 'count' double height rows, however some STL Files in circulation appear to have TTI blocks containing VP field values that count double height rows. Frequent use of the values of 10 or 11 in the VP fields with no higher values throughout an STL file may point to such an error. The correct identification of such inconsistent files is out of scope of this document but these files might be successfully be converted by doubling the VP value (Strict handling of these files would result in text at the middle of the display).

The interpretation of the Vertical Position field is dependent upon the value of the MNR field in the GSI block of the STL file. For Teletext subtitles (see §3.5 Display Standard Code) the number of rows in the presentation is fixed (at 23) and the MNR value is not relevant. For open or undefined subtitles (see §3.5 Display Standard Code) the MNR value defines a virtual set of rows used for positioning purposes. The MNR value does not imply that any subtitle will contain the specified number of rows and should not be used to determine line height or font size. (E.g. the MNR may have a value of 99 and an individual subtitle TTI block may have a VP [vertical position] of 70, positioning the top of that subtitle 70% down the 'Subtitle Safe Area' of the screen.

In some STL files, multiple carriage returns are intended to create a 'half row' gap between double height subtitle lines.[85] Further, by using the vertical position field, Teletext subtitles may be positioned at single height row boundaries even when displayed as double height text (i.e. a double height subtitle may span rows 1 & 2, or rows 2 & 3 etc.).

The following 'algorithms' describe the interpretation of the vertical position field assuming a region that has a `tts:displayAlign` attribute value of 'after'. Two broad strategies are described, minimal sized regions (where the region size and origin is calculated based upon the predicted position and extent of the subtitle text) and simple regions (where the region covers the entire 'Subtitle Safe Area' and the vertical position of the subtitles is implemented using line breaks).

4.5.6.1 Vertical Position (VP) for 'Minimal Sized Regions'

A conversion implementation should map Teletext subtitle positions into percentage values for region origin and extent definitions. This will require the conversion of the VP field value[86] into a percentage of the active video vertical extent to derive the region vertical origin value – taking into account the offset of any 'Subtitle Safe Area' from the top of the active video area.[87]

('Subtitle Safe Area' vertical size * ((VP – 1) / 23)) + 'Subtitle Safe Area' top vertical offset

The vertical extent of the related region for Teletext subtitles may be similarly calculated using the number of rows in the subtitle instead of the VP value, assuming that the font size and line height have defined values such that each line of rendered text is a single VP row in height (e.g. Single height: `tts:lineHeight` = "1c" `tts:fontSize` = "1c"). For double height text the Number of subtitle rows should be doubled (i.e. it is the number of base Teletext rows that are occupied).

[85] This could be indicated by 3 successive carriage returns or 2 successive carriage returns in the Text Field depending upon the mode of carriage return use!

[86] VP field values range from **1 – 23** for Teletext subtitles.

[87] The 'Subtitle Safe Area' vertical size is calculated as 'Subtitle Safe Area' vertical cells / vertical cell resolution * 100.

('Subtitle Safe Area' vertical size * (Number of subtitle rows / 23))

For 'open' subtitle files, the conversion of the VP field value[88] into a region vertical **origin** value also requires the MNR field value:[89]

('Subtitle Safe Area' vertical size * (VP / MNR)) + 'Subtitle Safe Area' top vertical offset

The vertical extent may be calculated using the defined font size and line height values to derive a 'computed line height' and multiplying this by the number of rows.

These calculations will typically produce fractional percentage values for region co-ordinates and extents, and implementations may choose to adopt a rounding strategy accordingly. It is recommended that any adopted rounding strategy generally increases the defined size of a region.

Alternatively, to 'mirror' the rigid positioning strategy of an original Teletext presentation for an input Teletext subtitle file, a conversion implementation may choose to use cell units for region origin co-ordinates and extents. In this case, the vertical origin of each subtitle's related region should match the value of the VP field minus one (the VP field will contain a value in the range 1-23 decimal corresponding to the Teletext row number of the first (top) subtitle row), plus the addition of the offset required to accommodate the vertical displacement of any 'Subtitle Safe Area'. Although fractional cell values are permitted, if the 'Subtitle Safe Area' is defined using 23 rows and 40 columns with respect to `ttp:cellresolution` (see Annex E), then the resulting cell co-ordinates for region origin and extent will also be simple integer values.

The extent of the related region should match the number of Teletext Rows occupied by content in the subtitle, taking into account single or double height presentation.[90]

In general, the number of rows of content in the subtitle is determined by carriage-return/line-feed (CR/LF) indicators in the Text Field. A single `tt:br` element should be placed in the `tt:p` element for each carriage return in the Text Field. A region definition needs to be created in the EBU-TT document for each unique combination of region origin and region height (as determined by the unique combinations of VP position and number of rows of text in the original subtitles).

[88] VP field values *may* range from **0 – 99** for open subtitles.

[89] The MNR value defines the number of rows that are possible within the area of the screen that may be used by subtitles. **NOTE: This equation assumes that the MNR value has been correctly specified as defined by the STL standard in the input file. See §3.5.1 Undefined and Open Subtitling.**

[90] This similarly assumes that the font size and line height have defined values such that each line of rendered text is a single (*or double*) VP row high (e.g. Single height: tts:lineHeight="1c" tts:fontSize="1c").

```
<tt:tt ttp:cellResolution="44 27" ...>
  <tt:head>
       <tt:style xml:id="defaultStyle" tts:fontSize="1c" tts:lineHeight="1c" .../>
       <tt:style xml:id="WhtOnBlk" tts:backgroundColor="black" tts:color="white" tts:textAlign="center"/>
       <tt:region xml:id="R1" tts:origin="4.5% 70.32%" tts:extent="91% 7.39%" tts:displayAlign="after".../>
...
       </tt:head>
       <tt:body style="defaultStyle">
      <tt:div>
        <tt:p region="R1" begin="00:00:00:00" end="00:00:02:10">
          <tt:span style=" WhtOnBlk">top-line of two on row 18</tt:span>
          <tt:span style=" WhtOnBlk">2nd-line of two on row 19</tt:span>
      </tt:p>
        </tt:div>
       </tt:body>
</tt:tt>
```

The above example illustrates a ‘minimal sized region’ for a two row single height subtitle, using rounded percentage co-ordinates for region origin and extent, with an origin on Teletext row 18, A ‘Subtitle Safe Area’ of 40 x 23 within a cell resolution of 44 x 27 is assumed (~91% x ~85%), giving an offset of 2 cells horizontally and 2 cells vertically (See Annex E for suggested percentage values for vertical and horizontal offsets and for ‘Subtitle Safe Area’ sizes).

‘Subtitle Safe Area’
size: 91% wide by 85% high
offsets: 7.5% from top of active video, 4.5% from side of active video

Origin = (85 * ((18 – 1) / 23)) + 7.5 = (85 * (17 / 23)) + 7.5 = 62.83 + 7.5 = 70.32%

Vertical extent = (85 * (2 / 23)) = 7.39

```
<tt:tt ttp:cellResolution="44 27" ...>
<tt:head><tt:style xml:id="defaultStyle" tts:fontSize="1c " tts:lineHeight="1c" .../>
<tt:style xml:id="WhiteOnBlack" tts:backgroundColor="black" tts:color="white"/>
<tt:style xml:id="textCenter" tts:textAlign="center"/>…
<tt:region xml:id="R1" tts:origin="2c 19c" tts:extent="40c 2c" tts:displayAlign="after".../>
...
</tt:head>
<tt:body style="defaultStyle">
<tt:div>
<tt:p style="textCenter" region="R1" begin="00:00:00:00" end="00:00:02:10">
<tt:span style="WhiteOnBlack">top-line of two single height, row 18</tt:span>
<tt:br/>
<tt:span style="WhiteOnBlack">2nd-line of two single height, row 19</tt:span>
</tt:p>
</tt:div>
</tt:body>
</tt:tt>
```

The above example illustrates a ‘minimal sized region’ (minimal height) for a two row single height subtitle with an origin on Teletext row 18, using cell units for region origin co-ordinates and extent, and assumes a ‘Subtitle Safe Area’ of 40 x 23 within a cell resolution of 44 x 27 (~91% x ~85%), giving an offset of 2 cells horizontally and 2 cells vertically.

```
<tt:tt ttp:cellResolution="44 27" ...>
<tt:head>
<tt:style xml:id="defaultStyle" tts:fontSize="1cc" tts:lineHeight="1c" .../>
<tt:style xml:id="WhiteOnBlack" tts:backgroundColor="black" tts:color="white" tts:fontSize="2c"
tts:lineHeight="2c"/>
<tt:style xml:id="textCenter" tts:textAlign="center"/>
<tt:region xml:id="R1" tts:origin="2c 17c" tts:extent="40c 4c" tts:displayAlign="after".../>
...
</tt:head>
<tt:body style="defaultStyle">
<tt:div>
<tt:p style="textCenter" region=" bottom " begin="00:00:00:00" end="00:00:02:10">
<tt:span style="WhiteOnBlack"> line1 of 2, double height, rows 16/17</tt:span>
<tt:br/>
<tt:span style="WhiteOnBlack">line2 of 2, double height, rows 18/19</tt:span>
</tt:p>
</tt:div>
</tt:body>
</tt:tt>
```

The above example illustrates a 'minimal sized region' (minimal height) for a two row double height subtitle with an origin on row 16, again using cell units for region origin co-ordinates and extent, showing how the region extent is increased (doubled), the region origin is moved up the display area and the font size and line height is modified to generate double height text. A 'Subtitle Safe Area' of 40 x 23 within a cell resolution of 44 x 27 is also assumed, giving an offset of 2 cells horizontally and 2 cells vertically.

4.5.6.2 Horizontal origin and extent for 'Minimal Sized Regions' strategy

The previous two examples illustrate 'minimal sized regions' that use a full row extent. The horizontal extent for the generated regions is the full width of the 'Subtitle Safe Area' and subtitle horizontal position along the row is achieved using justification. An implementation of a minimal sized region strategy (as determined by Document Processing Context) may also optimise the horizontal origin extent of generated regions by calculating the effective horizontal origin of the subtitle text and the rendered length of the subtitle text and by defining the region horizontal origin and extent appropriately.

For an input Teletext subtitle file the rigid positioning strategy of the original Teletext presentation can be used to precisely calculate both the horizontal extent and the horizontal origin of subtitle text as each individual character of an input Teletext subtitle file is intended to be rendered with an identical horizontal extent.

Where a justification code of 00h (unchanged presentation) has been used in this type of STL file, leading spaces and control codes may be 'counted' and used to determine the horizontal offset from the 'Subtitle Safe Area' 'left' boundary.[91]

For other justification codes, the implementation should use a calculation based upon the number of characters in the subtitle text and the maximum possible row length of 40 characters to determine a horizontal region offset - again taking into consideration the length of the longest line in a multi-row subtitle.[92] The computed value of the horizontal origin will depend upon the subtitle text length and the original intended justification.

The extent of the region necessary to contain the subtitle text can be similarly calculated by

[91] In Teletext, characters are transmitted in visual layout order. Rendering is 'dumb' - proceeding from left to right.

[92] The presence of (invisible) control codes at the start of the line to control presentation (e.g. text colour, background and double height) should also be taken into account.

counting characters– taking into consideration the length of the longest line in a multi-row subtitle. Alternatively, the region extent can be allowed to 'run to' the 'Subtitle Safe Area' 'right' boundary.[93]

Justification Code (JC)	Significance	Region Origin
00h	unchanged presentation	SSA 'left' boundary + counted leading 'spaces'
01h	left-justified text	SSA 'left' boundary
02h	centred text	SSA 'left' boundary + ((40 – subtitle length) / 2)
03h	right-justified text	SSA 'left' boundary + (40 – subtitle length)

For Undefined or Open Subtitling STL files the rigid positioning strategy of the original Teletext presentation does not apply (unless a monospace font with suitable metrics is used in the conversion strategy). However, the previously described strategies can still be used to determine the horizontal extent of the generated region (and thus to determine an appropriate origin) but implementations should take into consideration the potential differences in rendering lengths as determined by a conversion processing implementation and when the text is rendered by a presentation processor.[94] Typical practise, for existing systems that process Open Subtitling STL files, is to assume that leading spaces each occupy the same horizontal extent as a Teletext character (i.e. 1/40th of 'Subtitle Safe Area'). Subtitle text characters (and the internal spaces with the texts) are generally rendered using the selected font metrics.

The calculation of region horizontal extent should always be rounded such that the computed region width is greater or equal to the size required, to minimise the possibilities for text overflow. This is especially important when the use of proportional fonts is anticipated for rendering the subtitle text, E.g. Undefined or Open Subtitling STL files.

4.5.6.3 Vertical Position (VP) for 'Simple Region' strategy

The calculation of the number of line breaks required to correctly position the subtitle text in the vertical direction is dependent upon the assumed height of the subtitles in the source STL file. Teletext subtitles may be displayed as single height or double height (the latter is more common) and this is achieved by using control codes within the TTI blocks of the STL file. The determination of the nature of a source STL file (double or single height) is possible by examination of the TTI block content and the presence of double height control codes (or may be signalled by the Document Processing Context). Although it is technically possible for both single and double height to be mixed within a single STL source file, in practise all the subtitles in an STL file will be either single height or all double height.

4.5.6.3.1 Single Height Teletext Subtitles for 'Simple Region' strategy

For single height Teletext subtitles (DSC has a value of 01h or 02h), the VP field will contain a value in the range 1-23 decimal corresponding to the Teletext row number of the first (top) subtitle row.[95] This vertical position value should be used to adjust the content of the `tt:p` element for single height subtitles as follows:[96]

- The number of rows of content in the subtitle is determined (as created by carriage-return/line-feed (CR/LF) indicators in the Text Field). A single `tt:br` element is placed in the `tt:p` element for each carriage return in the Text Field. Leading carriage returns (not an expected usage) may be handled by incrementing the vertical position value.

[93] The text in the generated document should be enclosed within a tt:span element to control the extent of background colour, otherwise the regions may render with background colour that extends to the right boundary of the safe area.

[94] These differences arise because there is no mechanism **to guarantee** a strong correlation between font metrics in the processing context and in the presentation context.

[95] 23 single height start positions are possible.

[96] Assuming a default style with a font size of 1c.

- The vertical position value is subtracted from the maximum number of possible Teletext rows for subtitle presentation (23) to calculate the offset of the first row from the bottom of the presentation area. For subtitles positioned near the top of the screen (i.e. a low VP value) it may be convenient to use an additional region (co-located with the default region) but with a `tts:displayAlign` attribute value of 'before'. To use such a region, the logic in the following sections should be inverted. If two overlapping regions are used, one with displayAlign 'after', the other displayAlign 'before', it is logically possible to have text selected into overlapping areas of the screen. The regions should be transparent to allow the text drawn to each region to be seen.
- The number of rows of content in the subtitle is subtracted from the offset plus one to determine how many additional 'padding' rows need to be added to the bottom of the subtitle to correctly position the top row.
- The additional empty rows are added to the end of the `tt:p` element as `tt:br` elements.

```
<tt:tt ttp:cellResolution="44 27" ...>
  <tt:head>
...
       <tt:style xml:id="defaultStyle" tts:fontSize="1c" tts:lineHeight="1c" .../>
       <tt:style xml:id="WhiteOnBlack" tts:backgroundColor="black" tts:color="white"/>
       <tt:style xml:id="textCenter" tts:textAlign="center"/>…
       <tt:region xml:id="bottom" tts:origin="4.5% 7.5%" tts:extent="91% 85%" tts:displayAlign="after".../>
...
       </tt:head>
       <tt:body style="defaultStyle">
      <tt:div>
        <tt:p style="textCenter" region=" bottom " begin="00:00:00:00" end="00:00:02:10">
          <tt:span style="WhiteOnBlack">top-line of two on row 18</tt:span>
          <tt:br/>
          <tt:span style="WhiteOnBlack">2nd-line of two on row 19</tt:span>
          <tt:br/>
          <tt:br/>
          <tt:br/>
          <tt:br/>
      </tt:p>
        </tt:div>
       </tt:body>
</tt:tt>
```

4.5.6.3.2 Double Height Teletext Subtitles for 'Simple Region' strategy

For double height Teletext subtitles (DSC has a value of 01h or 02h), the text in the Text Field will be preceded by a double height control code. This control code may be present only at the start of the text in the Text Field or at the start of each logical row (i.e. after each carriage return).[97]

The double height Teletext control code causes text display over both the selected Teletext row and the row beneath. The VP field will contain a value in the range 1-22[98] decimal corresponding to the Teletext row number of the first (top) subtitle row and should be used to adjust the content of the `tt:p` element for double height subtitles as follows:

- The number of Teletext rows occupied by the subtitle is determined (as created by carriage-

[97] Typically double height mode is assumed to apply to the entire subtitle.

[98] 22 double height start positions are possible; double height rows can be positioned at single row height intervals but a double height row at row position 23 is not possible.

return/line-feed (CR/LF) indicators in the Text Field). In STL documents the carriage return (CR) means 'move to the next subtitle row', but in practise, 1 or 2 carriage returns may be used between subtitle lines in double height Teletext STL files. (I.e. Some implementations use a carriage return as a Teletext row return; thus two are required in double height subtitling, but other implementations use a carriage return as a subtitle line break, so only one is required regardless of the text height.)

Note: The mode of carriage return use in the STL source must be determined by the Document Processing Context also if each row of text in the STL Text Field occupies two Teletext rows in double height mode.

- A single `tt:br` element is placed in the `tt:p` element for each 'logical line break' in the Text Field. These `tt:br` elements are 'double height line breaks' due to their location in line with content that is double height.[99]
- The vertical position value is subtracted from the maximum number of possible Teletext rows for subtitle presentation (**23**) to calculate the offset of the top row from the bottom of the presentation area.
- The number of Teletext rows of content in the subtitle is subtracted from the offset plus one to determine how many additional 'padding' rows need to be added to the bottom of the subtitle to correctly position the top row.
- The additional empty rows are added to the end of the `tt:p` element as `tt:br` elements. These `tt:br` elements are 'single height line breaks' because they inherit the single height font size from their parent `tt:p` element.

Note: If the height of the computed fontSize of the p element is 2c and the lineHeight is 2c, then these line breaks will also be double height (w.r.t. to the base Teletext rows) and it is recommended that the number of tt:br elements is calculated accordingly. 'Half row' positioning will not be possible.

- The `tt:span` elements containing the text in the `tt:p` element should reference a style that selects a double height font-size.

[99] In accordance with XSL:FO layout concepts.

```
<tt:tt ttp:cellResolution="44 27" ...>
  <tt:head>...
        <tt:style xml:id="defaultStyle" tts:fontSize="1c" tts:lineHeight="1c" .../>
        <tt:style xml:id="WhiteOnBlack" tts:backgroundColor="black" tts:color="white" tts:fontSize="2c"
tts:lineHeight="2c"/>
        <tt:style xml:id="textCenter" tts:textAlign="center"/>
        <tt:region xml:id="bottom" tts:origin="4.5% 7.5%" tts:extent="91% 85%" tts:displayAlign="after".../>...
        </tt:head>
  <tt:body style="defaultStyle">
     <tt:div>
     <tt:p style="textCenter" region=" bottom " begin="00:00:00:00" end="00:00:02:10">
          <tt:span style="WhiteOnBlack">top-line of two on rows 16 & 17</tt:span>
          <tt:br/>
          <tt:span style="WhiteOnBlack">2nd-line of two on rows 18 & 19</tt:span>
          <tt:br/>
          <tt:br/>
          <tt:br/>
          <tt:br/>
     </tt:p>
     </tt:div>
  </tt:body>
</tt:tt>
```

4.5.6.3.3 Open or Undefined Subtitles for 'Simple Region' strategy

When the DSC field in the GSI block has a value of 00h, the vertical position field will contain a value in the range from 0...to the maximum number of rows indicated in the MNR field.[100] This vertical position represents the number of row positions from the top of the screen to the top of the subtitle. Typically the VP field is used as a percentage, where maximum rows (MNR) is set to 99 (0 – 99).

This value may be used to identify the closest equivalent Teletext row as follows:

Vertical row value = Round down (VP * 22 / Maximum Number of Rows)

The process described above (Vertical Position (VP) for double height Teletext subtitles) is then used to position the top line of the subtitle with the following notes:[101]

- Every Carriage return in the Text Field is interpreted as a logical line break.
- Each line of text in the Text Field is interpreted as double height.

Alternatively, font-size, line height, region and `tt:br` use may be determined by the Document Processing Context.

4.5.7 Text Field (TF)

The Text Field of a TTI block contains all, or some of, the text and control characters to convey a subtitle. It has a fixed length of 112 bytes and may contain any valid character code selected from the character code table specified by the value of the CCT field of the GSI block. Regardless of which character code table is in use, the following conventions are used in the TTI block:

- The CR/LF indicator, used to initiate the second and subsequent rows of the subtitle, is conveyed by character code 8Ah.

[100] The maximum number of rows cannot be greater than 99 decimal [63h]).

[101] This assumes that the intended presentation is close to the typical double height Teletext subtitle.

- The Text Field of the last TTI block of a subtitle will be terminated with code 8Fh.[102]
- Unused space in the Text Field will also be set to 8Fh.

The Text Field[103] shall be converted to one or more `tt:span` elements within a single `tt:p` element as follows:

- All text content will be contained within `tt:span` elements. These spans shall NOT be nested.
- Double Height text shall be placed in a `tt:span` element that references a double height style definition.
- Each logical row of text in the Text Field that follows an existing row in a subtitle is preceded by a single `tt:br` child element. For double height text, the single tt:br element will generate a 'double height line break' because it is located 'in line' with content that is double height.
- Leading space characters (in each row) of the Text Field (including spaces 'created' by control codes) shall not be converted into content of the `tt:span` element unless the Document Processing Context dictates the preservation of a Teletext presentation.
- Control codes for colours, italics and underline shall result in the closing of the current `tt:span` element and the start of a new `tt:span` element within the `tt:p` element. The new `tt:span` element shall reference a style definition that has the appropriate style attributes set to implement the intent of the control characters. If an appropriate style definition does not exist a style shall be created.
- For Teletext subtitles, the `tt:span` elements should reference a style definition that has a non-transparent background colour.

The Text Field of a TTI block will contain control codes from the Teletext character set. These codes are 'officially' split between codes used for 'open' (In-vision) subtitles and closed Teletext subtitles.

4.5.7.1 Processing Control Codes for Teletext STL files

The following table identifies the handling of specific control codes. The default style for Teletext subtitles is set by the Teletext Specification to be white text in a Boxed display mode. In Boxed display mode, the background must be turned on before the text by using explicit duplicated Start Box control codes.

In Teletext subtitles two 'Start box' command codes are sent to switch on the background. The background colour is switched on at the start of the second 'Start box' command. This has the visual effect of a black 'space' before the characters of the subtitle text. A similar effect occurs for the 'End box' command where the background colour is switched off after the reception of the command. This has the visual effect of a black 'space' after the characters of the subtitle text.

Note: Only a single 'End box' command is required.

These black spaces could be reproduced in an EBU-TT document that uses mono-spaced fonts by preceding and post-fixing the text with a single space character (and using an `xml:space` attribute with a value of "preserve") or by using an appropriate value for an `ebutts:linePadding` attribute in the style definition referenced by the content. The style attribute `ebutts:linePadding` is defined by EBU-TT-D. This attribute is used to apply padding (or inset space) on the start and end edges of rendered line areas.

Implementations should, but may not, implement EBU-TT extension features (such as `ebutts:linePadding`) that are not part of TTML 1.0; these are designed so that the default

[102] This character is not converted into the EBU-TT output

[103] Or Text Fields if multiple TTI blocks comprise a subtitle.

TTML 1.0 behaviour in the absence of those features is still acceptable, but inferior).

The semantics of consecutive control codes that have a "Generate span" action are accumulated and generate only a single span element that reference the appropriate style.

Control Code	Meaning	TTML Mapping	rgba	Action
00	Alpha black	'black'	#00000000	If a span is open this will be closed. A new span is generated referencing a style with the appropriate background and foreground colour.
01	Alpha red	'red'	#FF0000FF	" "
02	Alpha green	'lime'	#00FF00FF	" "
03	Alpha yellow	'yellow'	#FFFF00FF	" "
04	Alpha blue	'blue'	#0000FFFF	" "
05	Alpha magenta	'magenta'	#FF00FFFF	" "
06	Alpha cyan	'cyan'	#00FFFFFF	" "
07	Alpha white	'white'	#FFFFFFFF	" "
08 - 09				Not used[104]

Note: There is one inconsistency between rgba colour mappings for named colours when converting from the Teletext colour names to the pre-defined TTML colour names. The Teletext colour 'Alpha green' and the TTML colour 'green' are not equivalent. The Teletext colour 'Alpha green' is rgba #00FF00FF, whereas in contrast the TTML colour 'green' is rgba #008000FF.

Consequently, Teletext colour 'Alpha green' should be mapped to the TTML colour 'lime' (which is defined as rgba #00FF00FF).

[104] This code is technically valid in an STL TTI block, but would not be expected to be used.

Control Code	Meaning	Action
0A	End box	If a span is open this will be closed.
0B	Start box	If a span is open this will be closed. A new span is generated referencing a style with the appropriate background and foreground colour.[105]
0C	Normal height	See Vertical Position (VP) for Single Height Teletext Subtitles
0D	Double height	See Vertical Position (VP) for Double Height Teletext Subtitles
0E – 1B		Not used[106]
1C	Black background	If a span is open this will be closed. A new span is generated referencing a style with a black background and appropriate foreground colour.
1D	New background	In combination with the semantics of preceding and following control codes a new span is generated referencing a style with the appropriate background and foreground colour.[107]
1E	Hold mosaic	Not used
1F	Release mosaic	Not used

A typical Teletext row in a TTI block may have the following control codes:

<Start box><Start box>This is the Text<End box><End box>

Blue text on a yellow background would typically use the following control code sequence:

<Alpha yellow><New background><Alpha blue><Start box><Start box>Blue on yellow<End box><End box>

A red word within a white on black Teletext row would typically use the following control code sequence:

<Start box><Start box>A<Alpha red>red<Alpha white>word<End box><End box>

Note: The Alpha red and Alpha white control codes act as spaces between the words.

[105] The default background state for Teletext subtitles is a transparent background until a 'Start box' code is encountered. The Teletext specification requires that all subtitles are boxed.

[106] This code is technically valid in an STL TTI block, but would not be expected to be used.

[107] In practice this control code will typically appear in a sequence of 3 control codes, the first control code (00 – 0F) will set a colour, the second control code (1D) will switch this colour to the background, and the final control code will set a new foreground colour.

```
<tt:tt ...>
  <tt:head>...
    <tt:styling>
      <tt:style xml:id="WhiteOnBlack" tts:color="white" tts:backgroundColor="black" .../>
      <tt:style xml:id="RedOnBlack" tts:color="red" tts:backgroundColor="black" ..../>
      <tt:style xml:id="BlueOnYellow" tts:color="blue" tts:backgroundColor="yellow"…/>
    </tt:styling>....
  <tt:body style="defaultStyle">...
    <tt:p ...>
      <tt:span style="WhiteOnBlack">This the Text</tt:span>
    </tt:p>
    <tt:p ...>
      <tt:span style="BlueOnYellow">Blue on yellow</tt:span>
    </tt:p>
    <tt:p ...>
      <tt:span style="WhiteOnBlack">A </tt:span>
      <tt:span style="RedOnBlack">red</tt:span>
      <tt:span style="WhiteOnBlack"> word</tt:span>
    </tt:p>...
  </tt:body>
</tt:tt>
```

4.5.7.2 Processing Control Codes for Open (In-vision) STL files

The following table identifies the handling of specific control codes. STL files intended for open subtitles are typically interpreted by external systems that render the contained text. The default style for the subtitles is effectively set by a Document Processing Context rather than by the Teletext specification. Colour mappings into TTML are identical to § 4.4.7.1.

Control Code	Meaning	Action
00	Alpha black[108]	If a span is open this will be closed. A new span is generated referencing an appropriate style
01	Alpha red	“ “
02	Alpha green	“ “
03	Alpha yellow	“ “
04	Alpha blue	“ “
05	Alpha magenta	“ “
06	Alpha cyan	“ “
07	Alpha white	“ “
80h	Italics ON	“ “
81h	Italics OFF	“ “
82h	Underline ON	“ “
83h	Underline OFF	“ “
84h	Boxing ON[109]	If a span is open this will be closed. A new span is generated referencing an appropriate style with a non-transparent background.
85h	Boxing OFF	If a span is open this will be closed. A new span is generated referencing an appropriate style with a transparent background.

[108] The control codes for colours are officially intended for Teletext only, but may be encountered in Open (in-vision) STL subtitle files.

[109] The default background state for Open subtitles is undefined by the STL specification, but is typically interpreted as follows (transparent background until a ‘Start box’ code is encountered).

5. References

[1] **EBU Tech 3264** Specification of the EBU Subtitling data exchange format. http://tech.ebu.ch/publications/tech3264

[2] **EBU Tech 3350** **EBU-TT, Part 1;** Subtitling format definition. http://tech.ebu.ch/publications/tech3350

[3] **SMPTE ST 2016-1:2009** "SMPTE Standard - Format for Active Format Description and Bar Data"

[4] **EBU Tech 3241** Main characteristics of a teletext system for European broadcasting organisations using the variable-format principle (1982).

[5] **EBU Tech 3370** EBU-TT, Part 3 Live Subtitling applications http://tech.ebu.ch/publications/tech3370

[6] **EBU Tech 3360** EBU-TT, Part 2 Mapping EBU-STL (Tech 3264) to EBU-TT subtitle files http://tech.ebu.ch/publications/tech3360

[7] **EBU Tech 3240** Main characteristics of a "level.2" teletext system for European broadcasting organisations using the fixed-format principle (1982).

[8] **EBU R 95** Recommendation on Safe areas for 16:9 television production. http://tech.ebu.ch/publications/r095

[9] **EBU Tech 3390** EBU-TT, Part M Metadata definitions http://tech.ebu.ch/publications/tech3390

[10] **XML 1.0** Tim Bray, et al. Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Rec. 26 November 2008. http://www.w3.org/TR/2008/REC-xml-20081126/

[11] **ISO 8601** Data elements and interchange formats; Information interchange; Representation of dates and times

6. Bibliography

EBU R 133 Recommendation on transport of subtitles inside and outside MXF files http://tech.ebu.ch/publications/r133

EBU Tech 3380 EBU-TT, Part D Subtitling Distribution Format https://tech.ebu.ch/publications/tech3380

ISO 3166 Codes for the representation of names of countries and their subdivisions. https://www.iso.org/obp/ui/#search

Open Stand Open Stand Principles https://open-stand.org/about-us/principles/

RFC 3066 H Alvestrand, ed. RFC 3066: Tags for the Identification of Languages 1995. http://www.ietf.org/rfc/rfc3066.txt

SMIL 2.1 Synchronized Multimedia Integration Language (SMIL 2.1), W3C Rec. http://www.w3.org/TR/2005/REC-SMIL2-20051213/

SMPTE-12M-1:2008 "SMPTE Standard for Television -- Time and Control Code"

SMPTE ST 2052-1:2010 "SMPTE Standard for Television -- Timed Text Format (SMPTE-TT)

TTML 1.0 Timed Text Markup Language (TTML) 1.0 (Second Edition), W3C Rec. http://www.w3.org/TR/2013/REC ttml1 20130924/

UAX9 Mark Davis. Unicode Standard Annex #9. Unicode Bidirectional Algorithm. http://unicode.org/reports/tr9/

UAX15 Mark Davis. Unicode Standard Annex #15. Unicode Normalization Forms. http://unicode.org/reports/tr15/

XML Schema Part 2 Paul Biron and Ashok Malhotra, XML Schema Part 2: Datatypes, W3C Rec. 28 October 2004. http://www.w3.org/TR/xmlschema-2/

XML Path James Clark, Steve DeRose, XML Path Language (XPATH) Version 1.0 W3C Recommendation 16 November 1999 (Status updated October 2016). https://www.w3.org/TR/xpath/

XPATH FUNCTIONS 3.0 XPath and XQuery Functions and Operators 3.0 http://www.w3.org/TR/2014/REC-xpath-functions-30-20140408/

Annex A: Mapping of the STL GSI Block into EBU-TT

STL Information	Mnemonic	EBU-TT Element/Attribute	EBU-TT Location
Code Page Number	CPN	_	_
Disk Format Code	DFC	_	_
Display Standard Code	DSC	_	_
Character Code Table Number	CCT	_	_
Language Code	LC	@xml:lang	tt
Original Programme Title	OPT	ebuttm:documentOriginalProgrammeTitle	/tt/head/metadata/ebuttm:documentMetadata
Original Episode Title	OET	ebuttm:documentTranslatedEpisodeTitle	/tt/head/metadata/ebuttm:documentMetadata
Translated Programme Title	TPT	ebuttm:documentTranslatedProgrammeTitle	/tt/head/metadata/ebuttm:documentMetadata
Translated Episode Title	TET	ebuttm:documentTranslatedEpisodeTitle	/tt/head/metadata/ebuttm:documentMetadata
Translator's Name	TN	ebuttm:documentTranslatorsName	/tt/head/metadata/ebuttm:documentMetadata
Translator's Contact Details	TCD	ebuttm:documentTranslatorsContactDetails	/tt/head/metadata/ebuttm:documentMetadata
Subtitle List Reference Code	SLR	ebuttm:documentSubtitleListReferenceCode	/tt/head/metadata/ebuttm:documentMetadata
Creation Date	CD	ebuttm:stlCreationDate	/tt/head/metadata/ebuttm:documentMetadata
Revision Date	RD	ebuttm:stlRevisionDate	/tt/head/metadata/ebuttm:documentMetadata
Revision Number	RN	ebuttm:stlRevisionNumber	/tt/head/metadata/ebuttm:documentMetadata
Total Number of Text Timing Information (TTI) blocks	TNB		_
Total Numbers of Subtitles	TNS	ebuttm:documentTotalNumbersOfSubtitles	/tt/head/metadata/ebuttm:documentMetadata
Total Numbers of Subtitle Groups	TNG		_
Maximum Number of Displayable Character in any text Row	MNC	ebuttm:documentMaximumNumberOfDisplayableCharacterInAnyRow	/tt/head/metadata/ebuttm:documentMetadata
Maximum Number of Displayable Rows	MNR	_	_
Time Code: Status	TCS	_	_
Time Code: Start-of-Programme	TCP	ebuttm:documentStartOfProgramme	/tt/head/metadata/ebuttm:documentMetadata
Time Code: First In-Cue	TCF	_	_
Total Numbers of Disks	TND		_
Disk Sequence Number	DSN	_	_
Country of Origin	CO	ebuttm:documentCountryOfOrigin	/tt/head/metadata/ebuttm:documentMetadata
Publisher	PUB	ebuttm:documentPublisher	/tt/head/metadata/ebuttm:documentMetadata
Editor's Name	EN	ebuttm:documentEditorsName	/tt/head/metadata/ebuttm:documentMetadata
Editor's Contact Details	ECD	ebuttm:documentEditorsContactDetails	/tt/head/metadata/ebuttm:documentMetadata
User-Defined Area	UDA	ebuttm:documentUserDefinedArea	/tt/head/metadata/ebuttm:documentMetadata

Annex B: Conversion tables for the STL TTI Blocks to Unicode

Character code table 00 - Latin alphabet (6937/2-1983/Add.1:1989) to Unicode Mapping.

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			[SPC] 0020	0 0030	@ 0040	P 0050	` 0060	p 0070			[NBSP] 00A0	° 00B0		— 2015	Ω 2126	к 0138
01			! 0021	1 0031	A 0041	Q 0051	a 0061	q 0071			¡ 00A1	± 00B1	̀ 0300	¹ 00B9	Æ 00C6	æ 00E6
02			" 0022	2 0032	B 0042	R 0052	b 0062	r 0072			¢ 00A2	² 00B2	́ 0301	® 00AE	Đ 00D0	đ 0111
03			# 0023	3 0033	C 0043	S 0053	c 0063	s 0073			£ 00A3	³ 00B3	̂ 0302	© 00A9	ª 00AA	ð 00F0
04			¤ 00A4	4 0034	D 0044	T 0054	d 0064	t 0074			$ 0024	× 00D7	̃ 0303	™ 2122	Ħ 0126	ħ 0127
05			% 0025	5 0035	E 0045	U 0055	e 0065	u 0075			¥ 00A5	µ 00B5	̄ 0304	♪ 266A		ı 0131
06			& 0026	6 0036	F 0046	V 0056	f 0066	v 0076				¶ 00B6	̆ 0306	¬ 00AC	Ĳ 0132	ĳ 0133
07			' 0027	7 0037	G 0047	W 0057	g 0067	w 0077			§ 00A7	· 00B7	̇ 0307	¦ 00A6	Ŀ 013F	ŀ 0140
08			(0028	8 0038	H 0048	X 0058	h 0068	x 0078				÷ 00F7	̈ 0308		Ł 0141	ł 0142
09			) 0029	9 0039	I 0049	Y 0059	i 0069	y 0079			‘ 2018	’ 2019			Ø 00D8	ø 00F8
0A			* 002A	: 003A	J 004A	Z 005A	j 006A	z 007A			“ 201C	” 201D	̊ 030A		Œ 0152	œ 0153
0B			+ 002B	; 003B	K 004B	[005B	k 006B	{ 007B			« 00AB	» 00BB	̧ 0327		º 00BA	ß 00DF
0C			, 002C	< 003C	L 004C	\ 005C	l 006C	| 007C			← 2190	¼ 00BC	̲ 0332	⅛ 215B	Þ 00DE	þ 00FE
0D			- 002D	= 003D	M 004D	] 005D	m 006D	} 007D			↑ 2191	½ 00BD	̋ 030B	⅜ 215C	Ŧ 0166	ŧ 0167
0E			. 002E	> 003E	N 004E	^ 005E	n 006E	~ 007E			→ 2192	¾ 00BE	̛ 0328	⅝ 215D	Ŋ 014A	ŋ 014B
0F			/ 002F	? 003F	O 004F	_ 005F	o 006F				↓ 2193	¿ 00BF	̌ 030C	⅞ 215E	’n 0149	- 00AD

Character code table 01 - Latin/Cyrillic alphabet (from ISO 8859/5-1988) to Unicode Mapping.

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			[SPC] 0020	0 0030	@ 0040	P 0050		p 0070			[NBSP] 00A0	А 0410	Р 0420	а 0430	р 0440	№ 2116
01			! 0021	1 0031	A 0041	Q 0051	a 0061	q 0071			Ё 0401	Б 0411	В 0421	б 0431	с 0441	ё 0451
02			" 0022	2 0032	B 0042	R 0052	b 0062	r 0072			Ђ 0402	В 0412	Т 0422	в 0432	т 0442	ђ 0452
03			# 0023	3 0033	C 0043	S 0053	c 0063	s 0073			Ѓ 0403	Г 0413	У 0423	г 0433	у 0443	ѓ 0453
04			$ 0024	4 0034	D 0044	T 0054	d 0064	t 0074			Є 0404	Д 0414	Ф 0424	д 0434	ф 0444	є 0454
05			% 0025	5 0035	E 0045	U 0055	e 0065	u 0075			S [ґ] 0405 [0490]	Е 0415	Х 0425	е 0435	х 0445	s [ґ] 0454 [0491]
06			& 0026	6 0036	F 0046	V 0056	f 0066	v 0076			І 0406	Ж 0416	Ц 0426	ж 0436	ц 0446	і 0456
07			' 0027	7 0037	G 0047	W 0057	g 0067	w 0077			Ї 0407	З 0417	Ч 0427	з 0437	ч 0447	ї 0457
08			(0028	8 0038	H 0048	X 0058	h 0068	x 0078			Ј 0408	И 0418	Ш 0428	и 0438	ш 0448	ј 0458
09			) 0029	9 0039	I 0049	Y 0059	i 0069	y 0079			Љ 0409	Й 0419	Щ 0429	й 0439	щ 0449	љ 0459
0A			* 002A	: 003A	J 004A	Z 005A	j 006A	z 007A			Њ 040A	К 041A	Ъ 042A	к 043A	ъ 044A	њ 045A
0B			+ 002B	; 003B	K 004B	[005B	k 006B	{ 007B			Ћ 040B	Л 041B	Ы 042B	л 043B	ы 044B	ћ 045B
0C			, 002C	< 003C	L 004C	\ 005C	l 006C	| 007C			Ќ 040C	М 041C	Ь 042C	м 043C	ь 044C	ќ 045C
0D			- 002D	= 003D	M 004D	] 005D	m 006D	} 007D			 00AD	Н 041D	Э 042D	н 043D	э 044D	§ 00A7
0E			. 002E	> 003E	N 004E	^ 005E	n 006E	~ 007E			Ў 040E	О 041E	Ю 042E	о 043E	ю 044E	ў 045E
0F			/ 002F	? 003F	4F 004F	_ 005F	o 006F				Џ 040F	П 041F	Я 042F	п 043F	я 044F	џ 045F

Note: For the Ruthenian language, the characters in code positions Ah/5h and Fh/5h change!

Character code table 02 - Latin/Arabic alphabet (from ISO 8859/6-1987) to Unicode Mapping.

	00	10	20	30		40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			[SPC] 0020	0 0030	٠ 0660	@ 0040	P 0050		p 0070			[NBSP] 00A0			ـ 0630	ـ 0640	ِ 0650
01			! 0021	1 0031	١ 0661	A 0041	Q 0051	a 0061	q 0071					ء 0621	ر 0631	ف 0641	ّ 0651
02			" 0022	2 0032	٢ 0662	B 0042	R 0052	b 0062	r 0072					آ 0622	ز 0632	ق 0642	ْ 0652
03			# 0023	3 0033	٣ 0663	C 0043	S 0053	c 0063	s 0073					أ 0623	س 0633	ك 0643	
04			$ 0024	4 0034	٤ 0664	D 0044	T 0054	d 0064	t 0074			¤ 00A4		ؤ 0624	ش 0634	ل 0644	
05			% 0025	5 0035	٥ 0665	E 0045	U 0055	e 0065	u 0075					إ 0625	ص 0635	م 0645	
06			& 0026	6 0036	٦ 0666	F 0046	V 0056	f 0066	v 0076					ئ 0626	ض 0636	ن 0646	
07			' 0027	7 0037	٧ 0667	G 0047	W 0057	g 0067	w 0077					ا 0627	ط 0637	ه 0647	
08			(0028	8 0038	٨ 0668	H 0048	X 0058	h 0068	x 0078					ب 0628	ظ 0638	و 0648	
09			) 0029	9 0039	٩ 0669	I 0049	Y 0059	i 0069	y 0079					ة 0629	ع 0639	ى 0649	
0A			* 002A	: 003A		J 004A	Z 005A	j 006A	z 007A					ت 062A	غ 063A	ي 064A	
0B			+ 002B	; 003B		K 004B	[005B	k 006B	{ 007B				؛ 061B	ث 062B		ً 064B	
0C			, 002C	< 003C		L 004C	\ 005C	l 006C	| 007C			، 060C		ج 062C		ٌ 064C	
0D			- 002D	= 003D		M 004D	] 005D	m 006D	} 007D			- 00AD		ح 062D		ٍ 064D	
0E			. 002E	> 003E		N 004E	^ 005E	n 006E	~ 007E					خ 062E		ّ 064E	
0F			/ 002F	? 003F		4F 004F	_ 005F	o 006F					؟ 061F	د 062F		ْ 064F	

Character code table 03 - Latin/Greek alphabet from ISO 8859/7-1987 to Unicode Mapping.

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			[SPC] 0020	0 0030	@ 0040	P 0050	` 0060	p 0070			[NBSP] 00A0	° 00B0	ΐ 0390	Π 03A0	ΰ 03B0	π 03C0
01			! 0021	1 0031	A 0041	Q 0051	a 0061	q 0071			‘ 2018	± 00B1	Α 0391	Ρ 03A1	α 03B1	ρ 03C1
02			" 0022	2 0032	B 0042	R 0052	b 0062	r 0072			’ 2019	² 00B2	Β 0392		β 03B2	ς 03C2
03			# 0023	3 0033	C 0043	S 0053	c 0063	s 0073			£ 00A3	³ 00B3	Γ 0393	Σ 03A3	γ 03B3	σ 03C3
04			$ 0024	4 0034	D 0044	T 0054	d 0064	t 0074				΄ 0384	Δ 0394	Τ 03A4	δ 03B4	τ 03C4
05			% 0025	5 0035	E 0045	U 0055	e 0065	u 0075				΅ 0385	Ε 0395	Υ 03A5	ε 03B5	υ 03C5
06			& 0026	6 0036	F 0046	V 0056	f 0066	v 0076			¦ 00A6	Ά 0386	Ζ 0396	Φ 03A6	ζ 03B6	φ 03C6
07			' 0027	7 0037	G 0047	W 0057	g 0067	w 0077			§ 00A7	· 00B7	Η 0397	Χ 03A7	η 03B7	χ 03C7
08			(0028	8 0038	H 0048	X 0058	h 0068	x 0078			¨ 00A8	Έ 0388	Θ 0398	Ψ 03A8	θ 03B8	ψ 03C8
09			) 0029	9 0039	I 0049	Y 0059	i 0069	y 0079			© 00A9	Ή 0389	Ι 0399	Ω 03A9	ι 03B9	ω 03C9
0A			* 002A	: 003A	J 004A	Z 005A	j 006A	z 007A				Ί 038A	Κ 039A	Ϊ 03AA	κ 03BA	ϊ 03CA
0B			+ 002B	; 003B	K 004B	[005B	k 006B	{ 007B			« 00AB	» 00BB	Λ 039B	Ϋ 03AB	λ 03BB	ϋ 03CB
0C			, 002C	< 003C	L 004C	\ 005C	l 006C	| 007C			¬ 00AC	Ό 038C	Μ 039C	ά 03AC	μ 03BC	ό 03CC
0D			- 002D	= 003D	M 004D	] 005D	m 006D	} 007D			 00AD	½ 00BD	Ν 039D	έ 03AD	ν 03BD	ύ 03CD
0E			. 002E	> 003E	N 004E	^ 005E	n 006E	~ 007E				Ύ 038E	Ξ 039E	ή 03AE	ξ 03BE	ώ 03CE
0F			/ 002F	? 003F	4F 004F	_ 005F	o 006F				― 2015	Ώ 038F	Ο 039F	ί 03AF	ο 03BF	

Character code table 04 - Latin/Hebrew alphabet from ISO 8859/8-1988 to Unicode Mapping.

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			[SPC] 0020	0 0030	@ 0040	P 0050		p 0070			[NBSP] 00A0	° 00B0			א 05D0	נ 05E0
01			! 0021	1 0031	A 0041	Q 0051	a 0061	q 0071				± 00B1			ב 05D1	ס 05E1
02			" 0022	2 0032	B 0042	R 0052	b 0062	r 0072			¢ 00A2	² 00B2			ג 05D2	ע 05E2
03			# 0023	3 0033	C 0043	S 0053	c 0063	s 0073			£ 00A3	³ 00B3			ד 05D3	ף 05E3
04			$ 0024	4 0034	D 0044	T 0054	d 0064	t 0074			¤ 00A4	´ 00B4			ה 05D4	פ 05E4
05			% 0025	5 0035	E 0045	U 0055	e 0065	u 0075			¥ 00A5	µ 00B5			ו 05D5	ץ 05E5
06			& 0026	6 0036	F 0046	V 0056	f 0066	v 0076			¦ 00A6	¶ 00B6			ז 05D6	צ 05E6
07			' 0027	7 0037	G 0047	W 0057	g 0067	w 0077			§ 00A7	· 00B7			ח 05D7	ק 05E7
08			(0028	8 0038	H 0048	X 0058	h 0068	x 0078			¨ 00A8	¸ 00B8			ט 05D8	ר 05E8
09			) 0029	9 0039	I 0049	Y 0059	i 0069	y 0079			© 00A9	¹ 00B9			י 05D9	ש 05E9
0A			* 002A	: 003A	J 004A	Z 005A	j 006A	z 007A			× 00D7	÷ 00F7			ך 05DA	ת 05EA
0B			+ 002B	; 003B	K 004B	[005B	k 006B	{ 007B			« 00AB	» 00BB			כ 05DB	
0C			, 002C	< 003C	L 004C	\ 005C	l 006C	| 007C			¬ 00AC	¼ 00BC			ל 05DC	
0D			- 002D	= 003D	M 004D	] 005D	m 006D	} 007D			- 00AD	½ 00BD			ם 05DD	
0E			. 002E	> 003E	N 004E	^ 005E	n 006E	~ 007E			® 00AE	¾ 00BE			מ 05DE	
0F			/ 002F	? 003F	4F 004F	_ 005F	o 006F				¯ 00AF			‗ 2017	ן 05DF	

Annex C: Language Code mapping to xml:lang attribute value

The language names indicated here are not necessarily the official names of languages used in specific countries or territories. The information given here does not imply, on the part of the European Broadcasting Union, any opinion regarding the political status of these languages or associated countries or territories.

C.1 European languages written in Latin-based alphabets

Code	Language	xml:lang
00	Unknown/not applicable	und
01	Albanian	sq
02	Breton	br
03	Catalan	ca
04	Croatian	hr
05	Welsh (Cymraeg)	cy
06	Czech	cs
07	Danish	da
08	German	de
09	English	en
0A	Spanish (Castilian)	es
0B	Esperanto	eo
0C	Estonian	et
0D	Basque	eu

Code	Language	xml:lang
0E	Faroese	fo
0F	French	fr
10	Frisian	fy
11	Irish	ga
12	Gaelic (Scottish Gaelic)	gd
13	Galician (Gallegan)	gl
14	Icelandic	is
15	Italian	it
16	Lappish (Sami)	se
17	Latin	la
18	Latvian	lv
19	Luxembourgian (Luxembourgish)	lb
1A	Lithuanian	lt
1B	Hungarian	hu

Code	Language	xml:lang
1C	Maltese	mt
1D	Dutch	nl
1E	Norwegian	no
1F	Occitan	oc
20	Polish	pl
21	Portugese	pt
22	Romanian	ro
23	Romansh	rm
24	Serbian	sr

Code	Language	xml:lang
25	Slovak	sk
26	Slovenian	sl
27	Finnish	fi
28	Swedish	sv
29	Turkish	tr
2A	Flemish	*vls
2B	Wallon	wa
2C – 2E		
2F – 3F	Reserved for National assignment	

C.2 Other languages

Code	Language	xml:lang
7F	Amharic	am
7E	Arabic	ar
7D	Armenian	hy
7C	Assamese	as
7B	Azerbaijani	az
7A	Bambora	bm
79	Bielorussian	be

Code	Language	xml:lang
78	Bengali	bn
77	Bulgarian	bg
76	Burmese	my
75	Chinese	zh
74	Churash	cv
73	Dari	*fa-AF
72	Fulani	*ff

Code	Language	xml:lang
71	Georgian	ka
70	Greek	el
6F	Gujurati	gu
6E	Gurani	gn
6D	Hausa	ha
6C	Hebrew	he
6B	Hindi	hi
6A	Indonesian	id
69	Japanese	ja
68	Kannada	kn
67	Kazakh	kk
66	Khmer	km
65	Korean	ko
64	Laotian	lo
63	Macedonian	mk
62	Malagasay	mg
61	Malaysian	ms
60	Moldavian	mo
5F	Marathi	mr

Code	Language	xml:lang
5E	Ndebele	*nd
5D	Nepali	ne
5C	Oriya	or
5B	Papamiento	pap
5A	Persian	fa-IR
59	Punjabi	pa
58	Pushtu	ps
57	Quechua	qu
56	Russian	ru
55	Ruthenian	*rue
54	Serbo-croat	*hr
53	Shona	sn
52	Sinhalese	si
51	Somali	so
50	Sranan Tongo	srn
4F	Swahili	sw
4E	Tadzhik	tg
4D	*Tamil*	*ta*
4C	*Tatar*	*tt*

Code	Language	xml:lang
4B	*Telugu*	*te*
4A	*Thai*	*th*
49	*Ukrainian*	*uk*
48	*Urdu*	*ur*
47	*Uzbek*	*uz*
46	*Vietnamese*	*vi*
45	*Zulu*	*zu*
44 - 40		

Note: ***The languages identified by codes marked with an asterisk may require further clarification by the Document Processing Context.***

Annex D: Country of Origin Code mapping to ebuttm:documentCountryOfOrigin element value

The three-letter (Alpha-3) codes given in the table below are those used in the Country of Origin (CO) field of the GSI block. The 2 character codes and 4 character codes given are from the current ISO Standard 3166-3. The country names indicated here are not necessarily the official names of the countries or territories. The information given here does not imply, on the part of the European Broadcasting Union, any opinion regarding the political status of these countries or territories. **Country code mappings in bold indicate codes for formerly used names of countries.**

CO Code	Country Name	2 Letter Code
ABW	Aruba	AW
AFG	Afghanistan	AF
AGO	Angola	AO
AIA	Anguilla	AI
ALB	Albania	AL
AND	Andorra	AD
ANT	**Netherlands Antilles**	**ANHH**
ARE	United Arab Emirates	AE
ARG	Argentina	AR
ARM	Armenia	AM
ATA	Antarctica	AQ
ATF	French Southern Territories	TF
ATG	Antigua and Barbuda	AG
ATN	**Dronning Maud Land**	**NQAQ**
AUS	Australia	AU
AUT	Austria	AT

CO Code	Country Name	2 Letter Code
BDI	Burundi	BI
BEL	Belgium	BE
BEN	Benin	BJ
BFA	Burkina Faso	BF
BGD	Bangladesh	BD
BGR	Bulgaria	BG
BHR	Bahrain	BH
BHS	Bahamas	BS
BLZ	Belize	BZ
BMU	Bermuda	BM
BOL	Bolivia, Plurinational State of	BO
BRA	Brazil	BR
BRB	Barbados	BB
BRN	Brunei Darussalam	BN
BTN	Bhutan	BT
BUR	**Burma**	**BUMM**

CO Code	Country Name	2 Letter Code
BVT	Bouvet Island	BV
BWA	Botswana	BW
BYS	Byelorussian SSR (Name changed to Belarus)	BY
CAF	Central African Republic	CF
CAN	Canada	CA
CCK	Cocos (Keeling) Islands	CC
CHE	Switzerland	CH
CHL	Chile	CL
CHN	China	CN
CIV	Cote d'Ivoire	CI
CMR	Cameroon	CM
COG	Congo	CG
COK	Cook Islands	CK
COL	Colombia	CO
COM	Comoros	KM
CPV	Cape Verde	CV
CRI	Costa Rica	CR
CSK	**Czechoslovakia**	**CSHH**
CTE	Canton and Enderbury Islands (Merged into Kiribati)	CT
CUB	Cuba	CU
CXR	Christmas Island	CX
CYM	Cayman Islands	KY
CYP	Cyprus	CY
DDR	**German Democratic Republic**	**DDDE**
DEU	Germany	DE
DHM	Cambodia, Kingdom of (was Khmer Republic / Kampuchea, Democratic)	KH
DJI	Djibouti	DJ
DMA	Dominica	DM
DNK	Denmark	DK
DOM	Dominican Republic	DO
DZA	Algeria	DZ
ECU	Ecuador	EC
EGY	Egypt	EG
ESH	Western Sahara	EH
ESP	Spain	ES
EST	Estonia	EE
FIN	Finland	FI
FJI	Fiji	FJ
FLK	Falkland Islands (Malvinas)	FK
FRA	France	FR
FRO	Faroe Islands	FO
FSM	Micronesia, Federated States of	FM
GAB	Gabon	GA
GBR	United Kingdom	GB

CO Code	Country Name	2 Letter Code
GHA	Ghana	GH
GIB	Gibraltar	GI
GIN	Guinea	GN
GLP	Guadeloupe	GP
GMB	Gambia	GM
GNB	Guinea-Bissau	GW
GNQ	Equatorial Guinea	GQ
GRC	Greece	GR
GRD	Grenada	GD
GRL	Greenland	GL
GTM	Guatemala	GT
GUF	French Guiana	GF
GUM	Guam	GU
GUY	Guyana	GY
HKG	Hong Kong	HK
HMD	Heard Island and McDonald Islands	HM
HND	Honduras	HN
HTI	Haiti	HT
HUN	Hungary	HU
HVO	Upper Volta (Name changed to Burkina Faso)	BF
IDN	Indonesia	ID
IND	India	IN
IOT	British Indian Ocean Territory	IO
IRL	Ireland	IE
IRN	Iran, Islamic Republic of	IR
IRQ	Iraq	IQ
ISL	Iceland	IS
ISR	Israel	IL
ITA	Italy	IT
JAM	Jamaica	JM
JOR	Jordan	JO
JPN	Japan	JP
JTN	**Johnston Island**	**JTUM**
KEN	Kenya	KE
KIR	Kiribati	KI
KNA	Saint Kitts and Nevis	KN
KOR	Korea, Republic of	KR
KWT	Kuwait	KW
LAO	Lao People's Democratic Republic	LA
LBN	Lebanon	LB
LBR	Liberia	LR
LBY	Libya	LY
LCA	Saint Lucia	LC
LIE	Liechtenstein	LI

CO Code	Country Name	2 Letter Code
LKA	Sri Lanka	LK
LSO	Lesotho	LS
LUX	Luxembourg	LU
MAC	Macao	MO
MAR	Morocco	MA
MCO	Monaco	MC
MDG	Madagascar	MG
MDV	Maldives	MV
MEX	Mexico	MX
MHL	Marshall Islands	MH
MID	US Minor Outlying Islands (Midway Islands)	UM
MLI	Mali	ML
MLT	Malta	MT
MNG	Mongolia	MN
MNP	Northern Mariana Islands	MP
MOZ	Mozambique	MZ
MRT	Mauritania	MR
MSR	Montserrat	MS
MTQ	Martinique	MQ
MUS	Mauritius	MU
MWI	Malawi	MW
MYS	Malaysia	MY
NAM	Namibia	NA

CO Code	Country Name	2 Letter Code
NCL	New Caledonia	NC
NER	Niger	NE
NFK	Norfolk Island	NF
NGA	Nigeria	NG
NIC	Nicaragua	NI
NIU	Niue	NU
NLD	Netherlands	NL
NOR	Norway	NO
NPL	Nepal	NP
NRU	Nauru	NR
NTZ	**Neutral Zone**	**NTHH**
NZL	New Zealand	NZ
OMN	Oman	OM
PAK	Pakistan	PK
PAN	Panama	PA
PCI	**Pacific Islands, Trust Territory of the**	**PCHH**
PCN	Pitcairn	PN
PER	Peru	PE
PHL	Philippines	PH
PLW	Palau	PW
PNG	Papua New Guinea	PG
POL	Poland	PL
PRI	Puerto Rico	PR

CO Code	Country Name	2 Letter Code
PRK	Korea, Democratic People's Republic of	KP
PRT	Portugal	PT
PRY	Paraguay	PY
PUS	**U.S. Miscellaneous Pacific Islands**	**PUUM**
PYF	French Polynesia	PF
QAT	Qatar	QA
REU	Réunion	RE
ROU	Romania	RO
RWA	Rwanda	RW
SAU	Saudi Arabia	SA
SDN	Sudan	SD
SEN	Senegal	SN
SGP	Singapore	SG
SHN	Saint Helena, Ascension and Tristan da Cunha	SH
SJM	Svalbard and Jan Mayen	SJ
SLB	Solomon Islands	SB
SLE	Sierra Leone	SL
SLV	El Salvador	SV
SMR	San Marino	SM
SOM	Somalia	SO
SPM	Saint Pierre and Miquelon	PM
STP	Sao Tome and Principe	ST

CO Code	Country Name	2 Letter Code
SUN	**USSR**	**SUHH**
SUR	Suriname	SR
SWE	Sweden	SE
SWZ	Swaziland	SZ
SYC	Seychelles	SC
SYR	Syrian Arab Republic	SY
TCA	Turks and Caicos Islands	TC
TCD	Chad	TD
TGO	Togo	TG
THA	Thailand	TH
TKL	Tokelau	TK
TON	Tonga	TO
TMP	**East Timor**	**TPTL**
TTO	Trinidad and Tobago	TT
TUN	Tunisia	TN
TUR	Turkey	TR
TUV	Tuvalu	TV
TWN	Taiwan, Province of China	TW
TZA	Tanzania, United Republic of	TZ
UGA	Uganda	UG
UKR	Ukraine	UA
UMI	United States Minor Outlying Islands	UM

CO Code	Country Name	2 Letter Code
URY	Uruguay	UY
USA	United States	US
VAT	Holy See (Vatican City State)	VA
VCT	Saint Vincent and the Grenadines	VC
VEN	Venezuela, Bolivarian Republic of	VE
VGB	Virgin Islands, British	VG
VIR	Virgin Islands, U.S.	VI
VNM	Viet Nam	VN
VUT	Vanuatu	VU
WAK	United States Minor Outlying Islands (Wake Island)	UM
WLF	Wallis and Futuna	WF
WSM	Samoa	WS
YEM	Yemen	YE
YMD	Yemen, Democratic	YE
YUG	**Yugoslavia**	**YUCS**
ZAF	South Africa	ZA
ZAR	Zaire (Name change to Congo, the Democratic Republic)	CD
ZMB	Zambia	ZM
ZWE	Zimbabwe	ZW

Annex E: Relationship between 40 x 23 'Subtitle Safe Area' and ttp:cellResolution

Horizontal ttp:cellResolution value	~ Subtitle Safe Area %	Horizontal Offset of 40 x 23 'Subtitle Safe Area' for Centring within Active Video	
40	100	0 cells	0%
41	98	0.5 cells	1%
42	95	1 cells	2.5%
43	93	1.5 cells	3.5%
44	91	2 cells	4.5%
45	89	2.5 cells	5.5%
46	87	3 cells	6.5%
47	85	3.5 cells	7.5%
48	83	4 cells	8.5%
49	82	4.5 cells	9%
50	80	5 cells	10%
51	78	5.5 cells	11%
52	77	6 cells	11.5%
53	75	6.5 cells	12.5%
54	74	7 cells	13%
55	73	7.5 cells	13.5%
56	71	8 cells	14.5%
57	70	8.5 cells	15%
58	69	9 cells	15.5%
59	68	9.5 cells	16%
60	67	10 cells	16.5%
61	66	10.5 cells	17%
62	65	11 cells	17.5%
63	64	11.5 cells	18%
64	63	12 cells	18.5%

Horizontal ttp:cellResolution value	~ Subtitle Safe Area %	Horizontal Offset of 40 x 23 'Subtitle Safe Area' for Centring within Active Video	
65	62	12.5 cells	19%
66	61	13 cells	19.5%
67	60	13.5 cells	20%

Vertical ttp:cellResolution value	~ Subtitle Safe Area %	Vertical Offset of 40 x 23 Safe Area for Centring within Active Video	
23	100	0 cells	0%
24	96	0.5 cells	2%
25	92	1 cells	4%
26	88	1.5 cells	6%
27	85	2 cells	7.5%
28	82	2.5 cells	9%
29	79	3 cells	10.5%
30	77	3.5 cells	11.5%
31	74	4 cells	13%
32	72	4.5 cells	14%
33	70	5 cells	15%
34	68	5.5 cells	16%
35	66	6 cells	17%
36	64	6.5 cells	18%
37	62	7 cells	19%
38	61	7.5 cells	19.5%
39	59	8 cells	20.5%

Target Video Format	Subtitle Safe Area	Subtitle Safe Area extent	Safe Area origin (cells)	Region origin & (extent) %
Standard for 4:3 delivery	80% W 79% H	40 x 23 within 50 x 29	5, 3	10, 10.5, (80, 79)
Standard for 4:3 delivery	85% W 85% H	40 x 23 within 47 x 27	3.5, 2	7.5, 7.5 (85, 85)
Standard for 4:3 delivery	91% W 92% H	40 x 23 within 44 x 25	2, 1	4.5, 4 (91, 92)
Widescreen centred (4:3 80%) for 16:9	65% W 92% H	40 x 23 within 62 x 25	11, 1	17.5, 4 (65, 92)
Widescreen centred (4:3 90%) for 16:9	70% W 92% H	40 x 23 within 57 x 25	8.5, 1	15, 4 (70, 92)
Widescreen full width for 16:9	80% W 79% H	40 x 23 within 50 x 29	5, 3	10, 10.5 (80, 79)
Widescreen full width for 16:9	85% W 85% H	40 x 23 within 47 x 27	3.5, 2	7.5, 7.5 (85, 85)
Widescreen full width for 16:9	91% W 92% H	40 x 23 within 44 x 25	2, 1	4.5, 4 (91, 92)
Widescreen 14:9 protect for 16:9	70% W 92% H	40 x 23 within 57 x 25	8.5, 1	15, 4 (70, 92)

Increasing the vertical size of the ‘Subtitle Safe Area’ above 90% of active video height is not recommended since subtitles that have been deliberately positioned to avoid on screen Subtitle (e.g. ‘lower thirds’) might subsequently be displayed over them.

Annex F: EBU Tech 3360 (Part 2) additional requirements added to EBU Tech 3350 (Part 1)

Documents generated following the recommendations and mandatory behaviour defined by EBU Tech 3360 should also conform to the requirements of EBU Tech 3350. There are a number of differences that will be evident between Part 1 documents created as a result of conversion from original STL files and documents created by other mechanisms (e.g. direct generation by a subtitle creation workstation, or documents created by recording 'off air' subtitle data). These additional required differences are detailed in the following table:

Difference	EBU Tech 3350 (Part 1)	EBU Tech 3360 (Part 2)
ttp:timebase	"smpte"| "media" | "clock"	"smpte" | "clock"[110]
ttp:cellResolution	Optional	Mandatory
tts:color	Implementation dependent	"white" (recommended default value)
tts:displayAlign	"before" (initial value)	"after" (recommended default value)
tts:fontFamily	"default" (initial value)	"monospacedSansSerif" (recommended default value)
tts:lineHeight	"normal" (initial value)	Context dependent (expressed in cell units)
tts:padding	Permitted on style element and region element	Recommended value for region element only (set to 0)
tts:writingMode	"lrtb" | "rltb" | "tbrl" | "tblr" | "lr" | "rl" | "tb"	Only "lrtb" | "rltb" values referenced. STL files do not cater for vertical writing modes.
tts:showBackground	Initial value "always"	Recommended value "whenActive"

[110]STL files may also represent recordings of subtitles with 'wall clock' values used in the STL file. In this case only, when the Document Processing Context clearly signals this usage, the ttp:timebase attribute value shall be specified on the tt:tt element with the value of "clock".

Annex G: Conversion Summary

STL to EBU-TT at a glance

This table provides an overview of STL components and their conversion to EBU-TT. This section is an INFORMATIVE introduction to conversion techniques. It is not a complete guide; please consult the relevant sections for technical details and in-depth discussion of conversion strategies.

Glossary of terms:

Mapped	A value in STL that can be placed without reformatting in a directly corresponding EBU-TT element.
Converted	A value in STL that can be placed in a directly corresponding EBU-TT element after reformatting.
Recalculated	A value in STL that must be recalculated *from the resulting EBU-TT document content* before insertion into a corresponding element in the resulting EBU-TT document.
Replaced	A value that has no direct corresponding element in EBU-TT but that has a functional equivalent in EBU-TT.

GSI block values

In STL, the GSI contains information necessary for the use of TTI blocks, calculated data derived from the TTI blocks (e.g. how many there are), metadata about the file and user-defined information. In EBU-TT, some of these data can be mapped directly to EBU-TT-defined metadata (ebuttm) elements in tt/head/metadata; others need to be converted to a different format but are otherwise unchanged.

Values that do not have corresponding `ebuttm` elements may influence decoding of the STL but do not need to be explicitly included within the EBU-TT. If they are still required in the EBU-TT file, use your own metadata elements within the generic metadata container tt/head/metadata/.

Undefined values used in the GSI block should be ignored.

Trailing spaces should be removed.

Numeric values not represented by ASCII characters or containing space (e.g. '1 000') do not adhere to the STL specification and should be resolved prior to conversion.

STL Field	Conversion Summary	Notes	Section Reference
Code Page Number (CPN)	Should be decoded before further processing.	This value is not mapped in EBU-TT, but it is necessary to decode it before any further processing of the GSI block can take place. To decode it, use the conversion tables defined by the Unicode Consortium. The code page used for the GSI block has no relationship with the language of the subtitles or the character code set used in the TTI blocks. The EBU-TT document should use UTF-8 encoding.	3.3
Disk Format Code (DFC)	Replaced by two attributes of the <tt> element: ttp:frameRate and ttp:frameRateMultiplier.	These attributes are mandatory when converting from STL.	3.4
Display Standard Code (DSC)	Replaced by a value taken from the EBU's Classification Scheme (CS) and placed in /tt/head/metadata/ ebuttm:documentIntendedTargetFormat	No direct mapping exists between DSC values and EBU CS names. DSC values of 1 or 2 may indicate that the conversion should assume Teletext display features (e.g. monospaced font). Undefined or 0 DSC (open subtitles) do not necessarily indicate assumptions about the presentation, but additional processing on all subtitles in the original file may reveal the original presentation assumptions. Conversion without such processing is still possible, although unintended text wrapping and positioning may occur.	3.5
Language Code (LC)	Converted and placed in the xml:lang attribute of <tt:tt>.	Codes should be converted using the table in Annex C.	3.6 Annex C
Character Code Table Number (CCT)	Should be decoded before further processing.	The preferred encoding for subtitle text (characters in TTI blocks) in EBU-TT is Unicode. The CCT should therefore be decoded before conversion takes place.	3.7 Annex B
Original Programme Title (OPT)	Mapped to /tt/head/metadata/ ebuttm:documentOriginalProgrammeTitle	Values should be converted into Unicode. See Code Page Number above.	3.3
Original Episode Title (OET)	Mapped to /tt/head/metadata/ ebuttm:documentOriginalEpisodeTitle	Trailing space characters should be removed.	3.11
Translated Programme Title (TPT)	Mapped to /tt/head/metadata/ ebuttm:documentTranslatedProgrammeTitle		
Translated Episode Title (TET)	Mapped to /tt/head/metadata/ ebuttm:documentTranslatedEpisodeTitle		

STL Field	Conversion Summary	Notes	Section Reference
Translator's Name (TN)	Mapped to /tt/head/metadata/ ebuttm:documentTranslatorsName		
Translator's Contact Details (TCD)	Mapped to /tt/head/metadata/ ebuttm:documentTranslatorsContactDetails		
Subtitle List Reference Code (SLR)	Mapped to /tt/head/metadata/ ebuttm:documentSubtitleListReferenceCode		
Creation Date (CD)	Converted and placed in /tt/head/metadata/ ebuttm:stlCreationDate	Values should be converted from 6 ASCII Numeric characters (YYMMDD) into xs:date (2002-09-24).	3.14
Revision Date (RD)	Converted and placed in /tt/head/metadata/ ebuttm:stlRevisionDate	These elements shall not be used if original source STL documents are tunnelled as binary data.	
Revision Number (RN)	Mapped to /tt/head/metadata/ ebuttm:stlRevisionNumber	The value may be converted into a non-negative integer. This element shall not be used if original source STL documents are tunnelled as binary data.	3.15
Total Number of TTI blocks (TNB)	Not relevant for conversion.		
Total Numbers of Subtitles (TNS)	Recalculated and placed in /tt/head/metadata/ ebuttm:documentTotalNumbersOfSubtitles	The value after conversion may be different from the value in the STL (for example if subtitle zero has been removed because its information has been moved to <tt:head>). The new value should be used.	3.13
Total Numbers of Subtitle Groups (TNG)	Not relevant for conversion.		
Maximum Number of Displayable Characters in any text Row (MNC)	Recalculated and placed in /tt/head/metadata/ ebuttm:documentMaximumNumberOf DisplayableCharacterInAnyRow	The value after conversion may be different from the value in the STL (for example if control characters and trailing spaces have been removed). The new value should be used.	3.12
Maximum Number of Displayable Rows (MNR)	Not relevant for conversion.		
Time Code: Status (TCS)	Not relevant for conversion.	Although this value is not mapped to any EBU-TT elements, it has significance for mapping Time Code: Start-of-Programme (see below).	

STL Field	Conversion Summary	Notes	Section Reference
Time Code: Start-of-Programme (TCP)	Converted and placed in /tt/head/metadata/ ebuttm:documentStartOfProgramme	If the Time Code Status byte has a value of 31h then the contents of the Time Code: Start-of-Programme field should be converted into a string.	3.9 1.2.4
Time Code: First In-Cue (TCF)	Not relevant for conversion.		
Total Numbers of Disks (TND)	Not relevant for conversion.	If the STL is split across multiple files (one per "disk") then each file in sequence should be converted into a single EBU-TT document; if the STLs are being embedded then they should each be embedded in a separate ebuttm:binaryData element, with the elements placed in the document in ascending order of Disk Sequence Number.	2.3 (footnote 38)
Disk Sequence Number (DSN)	Not relevant for conversion.		
Country of Origin (CO)	Converted and placed in /tt/head/metadata/ ebuttm:documentCountryOfOrigin	3-letter Country codes should be converted to codes from ISO 3166-3. The ebuttm:documentCountryOfOrigin element shall NOT be used as a substitute for the xml:lang attribute in the tt:tt element (see Language Code).	3.8 3.6 Annex D
Publisher (PUB)	Mapped to /tt/head/metadata/ ebuttm:documentPublisher	Values should be converted into Unicode. See Code Page Number above.	
Editor's Name (EN)	Mapped to /tt/head/metadata/ ebuttm:documentEditorsName	Trailing space characters should be removed.	
Editor's Contact Details (ECD)	Mapped to /tt/head/metadata/ ebuttm:documentEditorsContactDetails		
User-Defined Area (UDA)	Converted and placed in /tt/head/metadata/ ebuttm:documentUserDefinedArea	The value should be converted into a BASE64 encoded string. Trailing spaces should not be converted.	

TTI block values

In STL, Text and Timing Information (TTI) blocks contain the subtitle text together with timing and positional data for that subtitle and may also contain additional subtitle data or user-specific data. A subtitle is defined by **a set of one or more** TTI blocks where each TTI block has the same unique Subtitle Number (SN).

In EBU-TT, each subtitle is represented by a single <tt:p> element. Multiple TTI blocks that are used to create a single subtitle should be

combined in EBU-TT into a single <tt:p> element.

All undefined values in the TTI block are ignored and are not converted into EBU-TT content.

In EBU-TT, subtitle text style and position are set by referencing <tt:style> and <tt:region> elements defined within the <tt:head> element. This contrasts with the STL format, where style and position are explicitly set using control codes in each separate subtitle. Consequently, converting from STL to EBU-TT requires that each TTI block is processed sequentially and new style and region definitions are added as necessary. These added definitions should then be referenced by the <tt:p> or <tt:span> elements that make each subtitle.

Carriage returns that occur in the TTI block subtitle text should be converted to <tt:br/> elements.

Row positions and row spacing are defined by the displayAlign attribute value, the applicable region's origin and extent values, font metrics and the lineHeight attribute value, as well as any automatic or explicit (<tt:br/> element) line breaks.

STL	EBU-TT	Example	Details
Subtitle Group Number (SGN)	In EBU-TT, a <tt:div> is used to group subtitles. Subtitles with matching Subtitle Group Numbers may be placed as child tt:p elements within a containing tt:div element. A single tt:div element should include all the tt:p elements created from TTI blocks with a common SGN number. Each tt:div element may contain an xml:id that holds the Subtitle Group Number converted into a string, for example prefixed by 'SGN' to ensure the uniqueness of the xml:id value required by the XML standard.	<tt:body> <tt:div xml:id="SGN0"> <tt:p xml:id="Subtitle0">...</tt:p> <tt:p xml:id="Subtitle1">...</tt:p> </tt:div> <tt:div xml:id="SGN1"> <tt:p xml:id="Subtitle2">...</tt:p> <tt:p xml:id="Subtitle3">...</tt:p> </tt:div> </tt:body>	4.3.1
Subtitle Number (SN)	The SN is not directly mapped to an EBU-TT element but may be preserved by: • Assigning the value to the xml:id attribute of the <tt:p> element that corresponds to each TTI block, for example prefixed by 'SN' to ensure the uniqueness of the xml:id value required by the XML standard.	<tt:body> <tt:div> <tt:p xml:id="SN0">...</tt:p> <tt:p xml:id="SN1">...</tt:p> </tt:div> </tt:body>	4.3.2

STL	EBU-TT	Example	Details
	• Converting the SNs into data stored in a private namespace contained within a child <tt:metadata> element, which must be the first child element of the containing <tt:p> element.	<tt:body> <tt:div> <tt:p> <tt:metadata> <my:SN>0</my:SN> </tt:metadata> Subtitle text </tt:p> </tt:div> </tt:body>	
Extension Block Number (EBN)	This field is not mapped to EBU-TT with one exception: • If the Extension Block Number has the value FEh the TTI block contains proprietary User Data. This data may be stored in a private namespace contained within <tt:metadata>, which must be the first child element of the containing <tt:p>.	<tt:body> <tt:div> <tt:p> <tt:metadata> <my:EBData>Some data </my:EBData> </tt:metadata> Subtitle text </tt:p> </tt:div> </tt:body>	4.3.3
Cumulative Status (CS)	Cumulative subtitles should be converted into a series of <tt:span> elements within <tt:p>. The Time Code In and Time Code Out values of each TTI block that contributes to the same cumulative subtitle set should be decoded and placed into the 'begin' and 'end' attributes of each <tt:span>. Begin and end attributes should be omitted from the span elements' parent p.	<tt:p> <tt:span begin="10:00:00" end="10:00:10"> First word</tt:span> <tt:span begin="10:00:05" end="10:00:10">Second word</tt:span> </tt:p>	4.4.3
Time Code In (TCI)	The value should be converted and placed in the begin attribute of the <tt:p> element (or <tt:span> in the case of cumulative subtitles – see Cumulative Status above).	<tt:p begin="10:00:00" end="10:00:10"> First subtitle</tt:p>	4.4.1

STL	EBU-TT	Example	Details
Time Code Out (TCO)	The value should be converted and placed in the end attribute of <tt:p> (or <tt:span> in the case of cumulative subtitles – see Cumulative Status above). Note that TCO values in STL are *inclusive* of the last frame. In EBU-TT these values are *exclusive*. For example, a TCO of 10:10:10:20 in STL should be converted to an end value of 10:10:10:21 in EBU-TT.	<tt:p> <tt:span begin="10:00:11" end="10:00:20"> Second subtitle</tt:span> </tt:p>	4.4.2
Vertical Positioning (VP)	EBU-TT offers more flexibility than STL and relies on font size, line height and regions for positioning. Consequently, a number of conversion strategies can be employed: • Create a region for each unique combination of VP position and number of rows of subtitle text. • Create a single region and use <tt:br/> elements to position the subtitle within the region (variations of this technique exist for single and double-height lines). • For undefined or open subtitles, the VP is typically a percentage that can be translated into a line number. Conversion then proceeds as above.	This is an example of the first strategy: <tt:tt ttp:cellResolution="44 27" ...> <tt:head> <tt:region xml:id="Region1" tts:origin="4.5% 70.32%" tts:extent="91% 7.39%" tts:displayAlign="after" .../> ... </tt:head> <tt:body> <tt:div> <tt:p region="Region1" ...> Subtitle text</tt:p> </tt:div> </tt:body> </tt:tt>	4.4.6

STL	EBU-TT	Example	Details
Justification Code (JC)	For values 01h, 02h and 03h (left, centred and right justified respectively), the <tt:p> element created for the subtitle should reference a style definition that has tts:textAlign attribute values of ‘start’, 'center' or ‘end’. All leading spaces and trailing spaces should be trimmed. For value 00h (unchanged presentation), one of these strategies is recommended: • Remove all control characters and trailing spaces and centre the subtitles (tts:textAlign="center") to create EBU-TT documents more suitable for universal use. • If space and control characters have been used for alignment (to left-align top and bottom rows), their processing will need careful consideration. It may be necessary to examine all subtitle rows to determine the intention of the author and the appropriate handling.	JC value of 02h converted to centre aligned subtitles: <tt:head> <tt:styling> <tt:style xml:id="Style1" tts:textAlign="center" ... /> </tt:styling> ... </tt:head> <tt:body> <tt:div> <tt:p style="Style1" ... > Subtitle text</tt:p> </tt:div> </tt:body>	4.4.4
Comment Flag (CF)	If the Comment Flag has a value of 01h, the content of the Text Field is mapped to a <ttm:desc> element within <tt:metadata>, which must be the first child element of the containing <tt:p>. If multiple TTI blocks with the same Subtitle Number contain comment flags, their text fields should be merged sequentially into a single <ttm:dec> element.	<tt:p ...> <tt:metadata> <ttm:desc>Content of text field marked as comment in TTI</ttm:desc> </tt:metadata> </tt:p>	4.4.5

STL	EBU-TT	Example	Details
Text Field (TF)	Converted to one or more <tt:span> elements within a single <tt:p>. Line breaks converted to <tt:br/>. Leading and trailing space characters should be removed, unless they are used for positioning (see Vertical Positioning). Control characters used for styling text will be converted to style definitions referenced by <tt:span> elements that must not be nested. If control characters were also used for positioning, they should be handled carefully. See Vertical Positioning.	<tt:tt ...> <tt:head> ... <tt:styling> <tt:style xml:id="WhiteOnBlack" tts:color="white" tts:backgroundColor="black" .../> </tt:styling> .... </tt:head> <tt:body> ... <tt:p ...> <tt:span style="WhiteOnBlack"> This the text <tt:br/> on two lines. </tt:span> </tt:p> ... </tt:body> </tt:tt>	4.4.7