OVERVIEW

PRESENTATION

FILE FORMAT

APPENDICES

Subtitle Guidelines

Version 1.2.3

June 2024

OVERVIEW

1 Introduction

- The BBC Academy has produced an online guide to subtitling. If you are new to subtitling, please start there.

Subtitles are primarily intended to serve viewers with loss of hearing, but they are used by a wide range of people: around 10% of broadcast viewers use subtitles regularly, increasing to 35% for some online content. The majority of these viewers are not hard of hearing.

This document describes 'closed' subtitles only, also known as 'closed captions'. Typically delivered as a separate file, closed subtitles can be switched off by the user and are not 'burnt in' to the image.

There are many formats in circulation for subtitle files. In general, the BBC accepts EBU-TT part 1 with STL embedded for broadcast, and EBU-TT-D for online only content. For a full description of the delivery requirements, see the File format section.

The Subtitle Guidelines describe best practice for authoring subtitles and provide instructions for making subtitle files for the BBC. This document brings together documents previously published by Ofcom and the BBC and is intended to serve as the basis for all subtitle work across the BBC: prepared and live, online and broadcast, internal and supplied.

Who should read this?

Anyone providing or handling subtitles for the BBC:

- authors of subtitle (respeakers, stenographers, editors);
- producers and distributors of content;
- developers of software tools for authoring, validating, converting and presenting subtitles;
- anyone involved in controlling subtitle quality and compliance.

In addition, if you have an interest in accessibility you will find a lot of useful information here.

What prior knowledge is expected?

The editorial guidelines in the Presentation section are written in plain English, requiring only general familiarity with subtitles. In contrast, to follow the technical instructions in the File format section you will need good working knowledge of XML and CSS. It is recommended that you also familiarise yourself with Timed Text Markup Language and SMPTE timecodes.

What should I read for...

- **An overview of subtitles:** read this introduction and the first few sections of Presentation, Timing, Identifying speakers and EBU-TT and EBU-TT-D Documents in detail. Scanning through the examples will also give you a good understanding of how subtitles are made.
- **Editing and styling subtitles:** read the Presentation section for text, format and timing guidelines.
- **Making subtitle files for online-only content:** if your software does not support EBU-TT-D you will need to create an XML file yourself. Assuming you are familiar with XML and CSS, start with Introduction to the TTML document structure and Example EBU-TT-D document. Then follow the

quick EBU-TT-D how-to.

Further assistance

Assistance with these guidelines and specific technical questions can be emailed to subtitle-guidelines@bbc.co.uk. For help with requirements for specific subtitle documents contact the commissioning editor.

1.1 Document conventions

The following symbols are used throughout this document.

Examples indicate the appearance of a subtitle. When illustrating bad or unrecommended practice, the example has a strike-though, like this: ~~counter-example~~ . Note that the subtitle style used here is only an approximation. It should not be used as a reference for real-world files or processors.

Most of this document applies to both online and broadcast subtitles. When there are differences between subtitles intended for either platform, this is indicated with one of these flags:
ONLINE - applies only to subtitles for online use (not for broadcast).
BROADCAST - applies to broadcast-only subtitles (not online).
When no broadcast or online flag is indicated, the text applies to all subtitles.

Subtitles must conform to one of two specifications: EBU-TT-D (subtitles intended for online distribution only) or EBU-TT version 1.0 (for broadcast and online). Sections that only apply to one of the specifications are indicated by one of these flags: **EBU-TT-D** or **EBU-TT 1.0**.

Specific actual values are indicated with double quotes, like this: `"2"` . These values must be used without the quotes. Descriptions of values are given in brackets: `[a number between 1 and 3]` . When several values are possible, they are separated by a pipe: `"1" | "2" | "3"` .

Text intended to guide developers in how to meet editorial guidelines is placed in sections like this within the Presentation section.

> Example sections are inset and styled with a side border.
>
> ```
> <tt:tt ...>
> <-- Code examples use explicit namespace prefixes
>     for the avoidance of doubt -->
> ```

1.2 Navigation

Since this is a longish sort of a document, we've added in some features to help navigation:

- When the window is wide enough, the table of contents appears on the left-hand side instead of the top.

- The table of contents by default just shows the top level headings - headings with a chevron to the right of them, can be expanded by clicking the chevron.

- If you want a direct link to a given section, you can click on the link icon to the right-hand side of the heading.

- Clicking on a heading in the main part of the document will make sure the heading is visible in the table of contents.

1.3 Document status

This version covers editorial and technical contribution and presentation guidelines, including resources to assist developers in meeting these guidelines. Future versions will build on these guidelines or describe changes, or address issues raised. We intend to release small updates often.

1.3.1 Changes since September 2016

Amongst many smaller tweaks, the following changes accumulated so far since version 1, released in September 2016, are notable:

- Minor clarifications to presentation guidelines in response to comments received, for example:

    - the wording about use of reaction shots to gain time.
    - the word rate for live subtitles has been adjusted to 160-180wpm from 130-150wpm.
    - the use of numbers.
    - capitalisation in speech.
- Technical details moved to the end, in the File Format section, including specification references and BBC-specific requirements.
- Added details about delivery, including multiple STL files and online exclusives.
- Added a section describing the details of EBU-TT and EBU-TT-D documents with a downloadable example document and further links to examples provided by IRT.
- Added links from the presentation sections to the technical implementation details.
- Added links from the technical implementation details to the presentation requirements they support.
- Added anchor links by headings for ease of reference.
- Made table of contents expandable, set to include top level details only on load.
- Accessibility improvements.
- Added details on positioning, including mapping of Teletext positions to percentage positions in EBU-TT-D/IMSC.
- Added more details about authoring and presentation font family, font size and line height, size customisation options and the use of Reith Sans font.
- Updated the references.
- Added requirement for compatibility of EBU-TT-D with IMSC; added

technical details of `itts:fillLineGap` and `ittp:activeArea`.

- Improved formatting of examples, code blocks and requirements.
- Made page layout more responsive to work better with smaller and larger screens.
- Added downloadable examples of an EBU-TT document and the result following conversion to EBU-TT-D.
- Improved accessibility and table of contents.
- Removed the outdated requirement to adjust the font size and line height when using the Reith Sans font.
- Updated workflow diagram in Appendix 5 to reflect improvements made over time.
- Added size and position guidance for 9:16 aspect ratio (vertical) video as distinct from 16:9, 4:3 or 1:1 aspect ratio video.
- Restricted duration of subtitle zero to a maximum of 2 frames.

Thank you to everyone who has helped to review this version. You know who you are!

1.4 How to contribute

Queries and comments may be raised at any time on the subtitle guidelines github project by those with *sufficient project access levels*. Readers who do not have access to the project should email subtitle-guidelines@bbc.co.uk.

When raising new issues please summarise in a short line the issue in the Title field and include enough information in the Description field, as well as the selected text, to allow the team to identify the relevant part(s) of the document.

PRESENTATION

Good subtitling is an art that requires negotiating conflicting requirements. On the whole, you should aim for subtitles that are faithful to the audio. However, you will need to balance this against considerations such as the action on the screen, speed of speech or editing and visual content.

> For example, if you subtitle a scene where a character is speaking rapidly, these are some of the decisions you may have to make:
>
> - Can viewers read the subtitles at the rate of speech?
> - Should you edit out some words to allow more time?
> - Can subtitles carry over to the next scene so they ‘catch up’ with the speaker?
> - Should you use cumulative subtitles to convey the rhythm of speech (for example, if rapping)?
> - If there are shot changes within the sequence, should the subtitles be synchronised with those?
> - Should you use one, two or three lines of subtitles?
> - Should you change the position of the subtitle to avoid obscuring important visual information or to indicate the speaker?

Clearly, it is not possible (or advisable) to provide a set of hard rules that cover all situations. Instead, this document provides some guidelines and practical advice. Their implementation will depend on the content, the genre and on the subtitler’s expertise.

2 Editing text

2.1 Prefer verbatim

If there is time for verbatim speech, do not edit unnecessarily. Your aim should be to give the viewer as much access to the soundtrack as you possibly can within the constraints of time, space, shot changes, and on-screen visuals, etc. You should never deprive the viewer of words/sounds when there is time to include them and where there is no conflict with the visual information.

However, if you have a very "busy" scene, full of action and disconnected conversations, it might be confusing if you subtitle fragments of speech here and there, rather than allowing the viewer to watch what is going on.

Don't automatically edit out words like "but", "so" or "too". They may be short but they are often essential for expressing meaning.

Similarly, conversational phrases like "you know", "well", "actually" often add flavour to the text.

2.2 Don’t simplify

It is not necessary to simplify or translate for deaf or hard-of-hearing viewers. This is not only condescending, it is also frustrating for lip-readers.

2.3 Retain speaker’s first and last words

If the speaker is in shot, try to retain the start and end of their speech, as these are most obvious to lip-readers who will feel cheated if these words are removed.

2.4 Edit evenly

Do not take the easy way out by simply removing an entire sentence. Sometimes this will be appropriate, but normally you should aim to edit out a bit of every sentence.

2.5 Keep names

Avoid editing out names when they are used to address people. They are often easy targets, but can be essential for following the plot.

2.6 Preserve the style

Your editing should be faithful to the speaker's style of speech, taking into account register, nationality, era, etc. This will affect your choice of vocabulary. For instance:

- register: mother vs mum; deceased vs dead; intercourse vs sex;

- nationality: mom vs mum; trousers vs pants;

- era: wireless vs radio; hackney cab vs taxi.

Similarly, make sure if you edit by using contractions that they are appropriate to the context and register. In a formal context, where a speaker would not use contractions, you should not use them either.

Regional styles must also be considered: e.g. it will not always be appropriate to edit "I've got a cat" to "I've a cat"; and "I used to go there" cannot necessarily be edited to "I'd go there."

2.7 Consider the previous subtitle

Having edited one subtitle, bear your edit in mind when creating the next subtitle. The edit can affect the content as well as the structure of anything that follows.

2.8 Keep the form of the verb

Avoid editing by changing the form of a verb. This sometimes works, but more often than not the change of tense produces a nonsense sentence. Also, if you do edit the tense, you have to make it consistent throughout the rest of the text.

2.9 Keep words that can be easily lip-read

Sometimes speakers can be clearly lip-read - particularly in close-ups. Do not edit out words that can be clearly lip-read. This makes the viewer feel cheated. If editing is unavoidable, then try to edit by using words that have similar lip-movements. Also, keep as close as possible to the original word order.

2.10 Subtitle illegible text

If the onscreen graphics are not easily legible because of the streamed image size or quality, the subtitles must include any text contained within those graphics which provide contextual information. This must include the speaker's identity, what they do and any organisations they represent. Other displayed information affected by legibility problems that must be included in the subtitle includes; phone numbers, email addresses, postal addresses, website URLs, or other contact information.

If the information contained within the graphics is off-topic from what is being spoken, then the information should not be replicated in the subtitle.

2.11 Strong language

Do not edit out strong language unless it is absolutely impossible to edit elsewhere in the sentence - deaf or hard-of-hearing viewers find this extremely irritating and condescending.

If the BBC has decided to edit any strong language, then your subtitles must reflect this in the following ways.

2.11.1 Bleeped words

If the offending word is bleeped, put the word BLEEP in the appropriate place in the subtitle - in caps, in a contrasting colour and without an exclamation mark.

BLEEP

If only the middle section of a word is bleeped, do not change colour mid-word:

f-BLEEP-ing

2.11.2 Dubbed words

If the word is dubbed with a euphemistic replacement - e.g. frigging - put this in. If the word is non-standard but spellable put this in, too:

frerlking

If the word is dubbed with an unrecognisable sequence of noises, leave them out.

2.11.3 Muted words

If the sound is dipped for a portion of the word, put up the sounds that you can hear and three dots for the dipped bit:

Keep your f...ing nose out of it!.

Never use more than three dots.

If the word is mouthed, use a label:

So (MOUTHS) f...ing what?

3 Line breaks

3.1 Line length

In Teletext, which is used to display subtitles on some broadcast platforms, line length is limited to 37 fixed-width (monospaced) characters, since at least 3 of the 40 available bytes are used for control codes. Other platforms use proportional fonts, making it impossible to determine the width of the line based on the number of characters alone. In this case, lines are constrained by the width of the region in which they are displayed. Guidelines for both platforms are summarised in the table below.

If targeting both online and broadcast platforms you must apply both constraints, i.e. ensure that the number of characters within a region does not exceed 37.

Platform	Max length	Notes

Platform	Max length	Notes
BROADCAST	37 characters, reduced if coloured text is used	Teletext constraint
ONLINE	68% of the width of a landscape 16:9 video; 90% of the width of a 4:3 video; 90% of the width of a square 1:1 video; 90% of the width of a vertical 9:16 video.	The number of characters that generate this width is determined by the font used, the given font size (see fonts) and the width of the characters in the particular piece of text (for example, 'lilly' takes up less width than 'mummy' even though both contain the same number of characters). As a guide, the equivalent to 37 characters in a 75% width region of a 16:9 (landscape) video is 25 characters in a 90% width region of a 9:16 (vertical) video. Using a proportionally spaced font, it may be possible to fit a few more characters in, especially narrow ones like 'i' or 'l' but if there a lot of wide characters like 'w' or 'm' then even 25 characters might not fit.

In EBU-TT-based implementations, line length is determined by the following attributes:

- `tt:region` (`tts:extent` attribute)
- `ttp:cellResolution`
- `tts:fontFamily`
- `tts:fontSize`
- `ebutts:linePadding`

3.2 Subtitles should contain single sentences

Each subtitle should comprise a single complete sentence. Depending on the speed of speech, there are exceptions to this general recommendation (see live subtitling, short and long sentences below)

3.3 Number of lines

For landscape or square (16:9, 4:3 or 1:1) video, a maximum subtitle length of two lines is recommended.

For vertical (9:16) video, a maximum subtitle length of three lines is recommended.

Extra lines may be used if you are confident that no important picture information will be obscured.

When deciding between one long line or two short ones, consider line breaks, number of words, pace of speech and the image.

> A `tt:region` sized to fit 3 lines at a recommended computed value of `tts:lineHeight` of 8% of the height of the root container region would have a minimum `tts:extent` height of 24%.

3.4 Break at natural points

Subtitles and lines should be broken at logical points. The ideal line-break will be at a piece of punctuation like a full stop, comma or dash. If the break has to be elsewhere in the sentence, avoid splitting the following parts of speech:

- article and noun (e.g. the + table; a + book)
- preposition and following phrase (e.g. on + the table; in + a way; about + his life)
- conjunction and following phrase/clause (e.g. and + those books; but + I went there)
- pronoun and verb (e.g. he + is; they + will come; it + comes)
- parts of a complex verb (e.g. have + eaten; will + have + been + doing)

However, since the dictates of space within a subtitle are more severe than between subtitles, line breaks may also take place after a verb. For example:

We are aiming to get
a better television service.

Line endings that break up a closely integrated phrase should be avoided where possible.

~~We are aiming to get a~~
~~better television service.~~

Line breaks within a word are especially disruptive to the reading process and should be avoided. Ideal formatting should therefore compromise between linguistic and geometric considerations but with priority given to linguistic considerations.

> Manual line breaks within `<tt:p>` and `<tt:span>` elements are specified using `<tt:br/>`. Automatic line breaks occur between adjacent active `<tt:p>` elements.

3.5 Breaks in justified subtitles

BROADCAST Left, right and centre justification can be useful to identify speaker position, especially in cases where there are more than three speakers on screen. In such cases, line breaks should be inserted at linguistically coherent points, taking eye-movement into careful consideration. For example:

We all hope
you are feeling much better.

This is left justified. The eye has least distance to travel from ‘hope’ to ‘you’.

We all hope you are
feeling much better.

This is centre justified. The eye now has least distance to travel from ‘are’ to ‘feeling’.

Problems occur with justification when a short sentence or phrase is followed by a longer one.

Oh.
He didn’t tell me you would be here.

In this case, there is a risk that the bottom line of the subtitle is read first.

Oh.
He didn’t tell me you would be here.

This could result in only half of the subtitle being read.

Allowances would therefore have to be made by breaking the line at a linguistically non-coherent point:

Oh. He didn’t tell me
you would be here.

Oh. He didn’t tell me you would be
here.

Left, centre and right justification can be specified using `tts:textAlign`; additional alignment options are available using `ebutts:multiRowAlign`.

3.6 Consider the image

When making a choice between one long line or two short lines, you should consider the background picture. In general, ‘long and thin’ subtitles are less disruptive of picture content than are ‘short and fat’ subtitles, but this is not always the case. Also take into account the number of words, line breaks etc.

3.7 Consider speaker positioning

BROADCAST In dialogue sequences it is often helpful to use horizontal displacement in order to distinguish between different speakers. ‘Short and fat' subtitles permit greater latitude for this technique.

3.8 Short sentences

Short sentences may be combined into a single subtitle if the available reading time is limited. However, you should also consider the image and the action on screen. For example, consecutive subtitles may reflect better the pace of speech.

3.9 Long sentences

In most cases verbatim subtitles are preferred to edited subtitles (see this research by BBC R&D) so avoid breaking long sentences into two shorter sentences. Instead, allow a single long sentence to extend over more than one subtitle. Sentences should be segmented at natural linguistic breaks such that each subtitle forms an integrated linguistic unit. Thus, segmentation at clause boundaries is to be preferred. For example:

When I jumped on the bus

I saw the man who had taken
the basket from the old lady.

Segmentation at major phrase boundaries can also be accepted as follows:

On two minor occasions
immediately following the war,

small numbers of people
were seen crossing the border.

There is considerable evidence from the psycho-linguistic literature that normal reading is organised into word groups corresponding to syntactic clauses and phrases, and that linguistically coherent segmentation of text can significantly improve readability.

Random segmentation must certainly be avoided:

~~On two minor occasions~~
~~immediately following the war, small~~

~~numbers of people, etc.~~

In the examples given above, no markers are used to indicate that segmentation is taking place. It is also acceptable to use sequences of dots (three at the end of a to-be-continued subtitle, and two at the beginning of a continuation) to mark the fact that a segmentation is taking place, especially in legacy subtitle files.

> Because line breaks require considering all of the above, they are better inserted manually. Implementers should avoid automatic line breaking. See the `tts:wrapOption` XML attribute.

3.10 Prioritise editing and timing over line breaks

Good line-breaks are extremely important because they make the process of reading and understanding far easier. However, it is not always possible to produce good line-breaks as well as well-edited text and good timing. Where these constraints are mutually exclusive, then well-edited text and timing are more important than line-breaks.

4 Timing

The recommended subtitle speed is 160-180 words-per-minute (WPM) or 0.33 to 0.375 second per word. However, viewers tend to prefer verbatim subtitles, so the rate may be adjusted to match the pace of the programme. Most subtitle authoring tools calculate the WPM and can be configured to give a warning when the word rate exceeds a certain WPM threshhold. You can also calculate the WPM manually (see box).

> To calculate the word-per-minute (WPM) speed of a subtitle in an EBU-TT document, divide the number of words in a subtitle (`<tt:p>` element) by its duration. The duration value can be calculated from the `begin` and `end` attributes. In the example fragment below, the first subtitle has a word rate of 2 words per second or 120 WPM (0.5s per word). The second subtitle is cumulative: the word 'three' appears on its own for 3

seconds, then 'four!' is added and both are displayed for another 2 seconds, giving 5 seconds for 'three' and 2 seconds for 'four!'. Note that end times in EBU-TT are exclusive.

```
<p xml:id="subtitle1" region="bottomRegion" style="paragraphStyle"
    begin="00:00:02" end="00:00:04">
      <span style="spanStyle">one, two...</span>
</p>
<p>
  <span style="spanStyle" begin="00:01:30" end="00:01:35">three...</span>
  <span style="spanStyle" begin="00:01:33" end="00:01:35">Four!</span>
</p>
```

4.1 Target minimum timing

Based on the recommended rate of 160-180 words per minute, you should aim to leave a subtitle on screen for a minimum period of around 0.3 seconds per word (e.g. 1.2 seconds for a 4-word subtitle). However, timings are ultimately an editorial decision that depends on other considerations, such as the speed of speech, text editing and shot synchronisation. When assessing the amount of time that a subtitle needs to remain on the screen, think about much more than the number of words on the screen; this would be an unacceptably crude approach.

4.2 When to give less time

Do not dip below the target timing unless there is no other way of getting round a problem. Circumstances which could mean giving less reading time are:

4.2.1 Shot changes

Give less time if the target timing would involve clipping a shot, or crossing into an unrelated, "empty" [containing no speech] shot. However, always consider the alternative of merging with another subtitle.

4.2.2 Lip reading

Give less time to avoid editing out words that can be lip-read, but only in very specific circumstances: i.e. when a word or phrase can be read very clearly even by non-lip-readers, and if it would look ridiculous to take out or change the word.

4.2.3 Catchwords

Avoid editing out catchwords if a phrase would become unrecognisable if edited.

4.2.4 Retaining humour

Give less time if a joke would be destroyed by adhering to the standard timing, but only if there is no other way around the problem, such as merging or crossing a shot.

4.2.5 Critical information

In a news item or factual content, the main aim is to convey the "what, when, who, how, why". If an item is already particularly concise, it may be impossible to edit it into subtitles at standard timings without losing a crucial element of the original.

4.2.6 Very technical items

These may be similarly hard to edit. For instance, a detailed explanation of an economic or scientific story may prove almost impossible to edit without depriving the viewer of vital information. In these situations a subtitler should be prepared to vary the timing to convey the full meaning of the original.

4.3 When to give extra time

Try to allow extra reading time for your subtitles in the following circumstances:

4.3.1 Unfamiliar words

Try to give more generous timings whenever you consider that viewers might find a word or phrase extremely hard to read without more time.

4.3.2 Several speakers

Aim to give more time when there are several speakers in one subtitle.

4.3.3 Labels

Allow an extra second for labels where possible, but only if appropriate.

4.3.4 Visuals and graphics

When there is a lot happening in the picture, e.g. a football match or a map, allow viewers enough time both to read the subtitle and to take in the visuals.

4.3.5 Placed subtitles

If, for example, two speakers are placed in the same subtitle, and the person on the right speaks first, the eye has more work to do, so try to allow more time.

4.3.6 Long figures

Give viewers more time to read long figures (e.g. 12,353).

4.3.7 Shot changes

Aim for longer timing if your subtitle crosses one shot or more, as viewers will need longer to read it.

4.3.8 Slow speech

Slower timings should be used to keep in sync with slow speech.

4.4 Use consistent timing

It is also very important to keep your timings consistent. For instance, if you have given 3:12 for one subtitle, you must not then give 4:12 to subsequent subtitles of similar length - unless there is a very good reason: e.g. slow speaker/on-screen action.

4.5 Gaps

If there is a pause between two pieces of speech, you may leave a gap between the subtitles - but this must be a minimum of one second, preferably a second and a half. Anything shorter than this produces a very jerky effect. Try to not squeeze gaps in if the time can be used for text.

5 Synchronisation

5.1 Match subtitle to speech onset

Impaired viewers make use of visual cues from the faces of television speakers. Therefore subtitle appearance should coincide with speech onset. Subtitle disappearance should coincide roughly with the end of the corresponding speech segment, since subtitles remaining too long on the screen are likely to be re-read by the viewer.

When two or more people are speaking, it is particularly important to keep in sync. Subtitles for new speakers must, as far as possible, come up as the new speaker starts to speak. Whether this is possible will depend on the action on screen and rate of speech.

The same rules of synchronisation should apply with off-camera speakers and even with off-screen narrators, since viewers with a certain amount of residual hearing make use of auditory cues to direct their attention to the subtitle area.

5.2 Match subtitle to pace of speaking

The subtitles should match the pace of speaking as closely as possible. Ideally, when the speaker is in shot, your subtitles should not anticipate speech by more than 1.5 seconds or hang up on the screen for more than 1.5 seconds after speech has stopped.

However, if the speaker is very easy to lip-read, slipping out of sync even by a second may spoil any dramatic effect and make the subtitles harder to follow. The subtitle should not be on the screen after the speaker has disappeared.

Note that some decoders might override the end timing of a subtitle so that it stays on screen until the next one appears. This is a non-compliant behaviour that the subtitle author and broadcaster have no control over.

> Decoders need to match the begin and end timing specified in documents as closely as possible to maintain the careful synchronisation we expect from subtitle authors. In particular, see Annex E of EBU-TT-D regarding quantisation of timing for example if the video can only be presented at a low frame rate, such as in poor network conditions.

5.3 Display subtitles when lips are moving

A subtitle (or an explanatory label) should always be on the screen if someone's lips are moving. If a speaker speaks very slowly, then the subtitles will have to be slow, too - even if this means breaking the timing conventions. If a speaker speaks very fast, you have to edit as much as is necessary in order to meet the timing requirements (see timing).

5.4 Keep lag behind speech to a minimum

Your aim is to minimise lag between speech and the appearance of the subtitle. But sometimes, in order to meet other requirements (e.g. matching shots), you will find it difficult to avoid slipping slightly out of sync. In this case, subtitles should never appear more than 2 seconds after the words were spoken. This should be avoided by editing the previous subtitles.

It is permissible to slip out of sync when you have a sequence of subtitles for a single speaker, providing the subtitles are back in sync by the end of the sequence.

If the speech belongs to an out-of-shot speaker or is voice-over commentary, then it's not so essential for the subtitles to keep in sync.

5.5 Do not pre-empt an effect

Do not bring in any dramatic subtitles too early. For example, if there is a loud bang at the end of, say, a two-second shot, do not anticipate it by starting the label at the beginning of the shot. Wait until the bang actually happens, even if this means a fast timing.

5.6 Keep speakers separate

Do not simultaneously caption different speakers if they are not speaking at the same time.

6 Matching shots

6.1 Match subtitles to shot

It is likely to be less tiring for the viewer if shot changes and subtitle changes occur at the same time. Many subtitles therefore start on the first frame of the shot and end on the last frame.

6.2 Maintain a minimum gap when mismatched

If you have to let a subtitle hang over a shot change, do not remove it too soon after the cut. The duration of the overhang will depend on the content.

6.3 Avoid straddling shot changes

Avoid creating subtitles that straddle a shot change (i.e. a subtitle that starts in the middle of shot one and ends in the middle of shot two). To do this, you may need to split a sentence at an appropriate point, or delay the start of a new sentence to coincide with the shot change.

> Authoring tools may use automated shot detection to avoid this scenario.

6.4 Merge subtitles for short shots

If one shot is too fast for a subtitle, then you can merge the speech for two shots – provided your subtitle then ends at the second shot change.

Bear in mind, however, that it will not always be appropriate to merge the speech from two shots: e.g. if it means that you are thereby "giving the game away" in some way. For example, if someone sneezes on a very short shot, it is more effective to leave the "Atchoo!" on its own with a fast timing (or to merge it with what comes afterwards) than to anticipate it by merging with the previous subtitle.

6.5 End subtitle with speech

Where possible, avoid extending a subtitle into the next shot when the speaker has stopped speaking, particularly if this is a dramatic reaction shot.

6.6 End subtitle with scene

Never carry a subtitle over into the next shot if this means crossing into another scene or if it is obvious that the speaker is no longer around (e.g. if they have left the room).

6.7 Wait for scene change to subtitle speaker

Some film techniques introduce the soundtrack for the next scene before the scene change has occurred. If possible, the subtitler should wait for the scene change before displaying the subtitle. If this is not possible, the subtitle should be clearly labelled to explain the technique.

JOHN: And what have we here?

---

7 Identifying speakers

Several techniques can be used to assist the viewer in identifying speakers. The BBC's preferred techniques are colour and single quotes, but other techniques exist in legacy subtitle files and subtitles repurposed from non-UK sources. Re-use of existing files with legacy techniques is acceptable, but unless specifically requested, new content should not use legacy techniques.

The available techniques include:

- Colour: This is the preferred method that should be used in most cases.
- Single quotes: Used to indicate an out-of-vision speaker, such as someone speaking via telephone, or to distinguish between in- and out-of-vision voices when both are spoken by the same character (or by the narrator) and therefore using the same colour (e.g. a narrator who is sometimes in-vision).
- Arrows: Used to indicate the direction of out-of-vision sounds when the origin of the sound is not apparent. (infrequently used)
- Label: Can be used to resolve ambiguity as to who is speaking.
- Horizontal positioning: This is a legacy technique for identifying in-vision speakers, but it is still used for indicating off-screen speech. It is also used with Vertical positioning to avoid obscuring important information.
- Dashes: This is a legacy technique. Must only be used with colour when unavoidable.

7.1 Use colours

Use colours to distinguish speakers from each other (see Colours). This is the preferred method for identifying speakers.

Where the speech for two or more speakers of different colours is combined in one subtitle, their speech runs on: i.e. you don't start a new line for each new speaker.

Did you see Jane? I thought she went home.

However, if two or more WHITE text speakers are interacting, you have to start a new line for each new speaker, preceded by a dash.

By convention, the narrator is indicated by a yellow colour.

Colour is implemented using `tts:color` and `tt:span`.

7.2 Use horizontal positioning

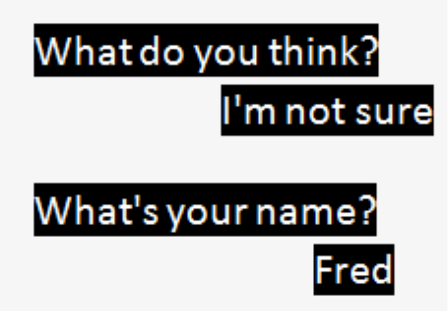
This is a legacy technique that is no longer used in new content for identifying in-vision speakers (it may be present in files created before it was deprecated). Use colour instead.

Horizontal positioning is used in combination with arrows to indicate out-of-vision voices.

BROADCAST Where colours cannot be used you can distinguish between speakers with placing.

Put each piece of speech on a separate line or lines and place it underneath the relevant speaker. You may have to edit more to ensure that the lines are short enough to look placed.

Try to make sure that pieces of speech placed right and left are "joined at the hip" if possible, so that the eye does not have to leap from one side of the screen to the other.



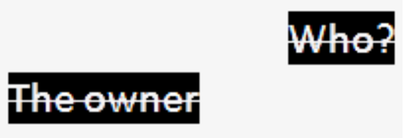
What do you think?

I'm not sure

What's your name?

Fred

Not:



~~Who?~~

~~The owner~~

When characters move about while speaking, the caption should be positioned at the discretion of the subtitler to identify the position of the speaker as clearly as possible.

Horizontal positioning is determined by these EBU-TT attributes:

- tts:direction (TTML1 specification link)
- tts:textAlign
- ebutts:multiRowAlign
- tts:unicodeBidi (TTML1 specification link)
- tts:writingMode

7.3 Use dashes

This is a legacy technique that is no longer used for new content (but may be present in files created before it was deprecated or sourced from outside the UK). Use colour to indicate a change of speaker.

If colour cannot be used (or if colour is being used but two consecutive speakers are both assigned the same colour), put each piece of speech on a separate line and insert a white dash (not a hyphen) before each piece of speech, thereby clearly distinguishing different speakers' lines. If possible, align the dashes so that they are proud of the text, although not all formats support this well.

– Found anything?
– If this is the next new weapon,
we're in big trouble.

The longest line should be centred on the screen, with the shorter line/lines left-aligned with it (not centred). If one of the lines is long, inevitably all the text will be towards the left of the screen, but generally the aim is to keep the lines in the centre of the screen.

Note that dashes only work as a clear indication of speakers when each speaker is in a separate consecutive shot.

7.4 Use single quotes for voice-over

If you need to distinguish between an in-vision speaker and a voice-over speaker, use single quotes for the voice-over, but only when there is likely to be confusion without them (single quotes are not normally necessary for a narrator, for example). Confusion is most likely to arise when the in-vision speaker and the voice-over speaker are the same person.

Put a single quote-mark at the beginning of each new subtitle (or segment, in live), but do not close the single quotes at the end of each subtitle/ segment - only close them when the person has finished speaking, as is the case with paragraphs in a book.

'I've lived in the Lake District since I was a boy.

'I never want to leave this area.
I've been very happy here.

'I love the fresh air and the beautiful scenery.'

If more than one speaker in the same subtitle is a voice-over, just put single quotes at the beginning and end of the subtitle.

'What do you think about it? I'm not sure.'

The single quotes will be in the same colour as the adjoining text.

7.5 Use single quotes for out-of-vision speaker

When two white text speakers are having a telephone conversation, you will need to distinguish the speakers. Using single quotes placed around the speech of the out-of-vision speaker is the recommended approach. They should be used throughout the conversation, whenever one of the speakers is out of vision.

Hello. Victor Meldrew speaking.
'Hello, Mr Meldrew. I'm calling about your car.'

Single quotes are not necessary in telephone conversations if the out-of-vision speaker has a colour.

7.6 Use double quotes for mechanical speech and for quoting

Double quotes "..." can suggest mechanically reproduced speech, e.g. radio, loudspeakers etc., or a quotation from a person or book. Start the quote with a capital letter:

He said, "You're so tall".

7.7 Use arrows for off-screen voices

Generally, colours should be used to identify speakers. However, when an out-of-shot speaker needs to be distinguished from an in-shot speaker of the same colour, or when the source of off-screen/off-camera speech is not obvious from the visible context, insert a ‘greater than’ (>) or ‘less than’ (<) symbols to indicate the off-camera speaker.

If the out-of-shot speaker is on the left or right, type a left or right arrow (< or >) next to their speech and place the speech to the appropriate side. Left arrows go immediately before the speech, followed by one space; right arrows immediately after the speech, preceded by one space.

Do come in.
Are you sure? >

When are you leaving?
< I was thinking of going

at around 8 o'clock in the evening.

When I find out where he is,
you'll be the first to know. >

NOT:

~~When I find out where he is, >~~
~~you'll be the first to know.~~

If possible, make the arrow clearly visible by keeping it clear of any other lines of text, i.e. the text following the arrow and the text in any lines below it are aligned. However, not all formats support hanging indent well.

> Non-breaking spaces can be inserted to simulate the indent behaviour reasonably closely.

< When I find out where he is,
  you'll be the first to know

The arrows are always typed in white regardless of the text colour of the speaker.

If an off-screen speaker is neither to the right nor the left, but straight ahead, do not use an arrow.

ONLINE Arrow characters (← and →) can be used instead of < and > for online-only subtitles.

7.8 Use labels for off-screen voices

If you are unable to use any other technique, use a label to identify a speaker, but only if it is unclear who was speaking or when more than four characters are speaking, requiring a shared colour. Type the name of the speaker in white caps (regardless of the colour of the speaker's text), immediately before the relevant speech.

If there is time, place the speech on the line below the label, so that the label is as separate as possible from the speech. If this is not possible, put the label on the same line as the speech, centred in the usual way.

JAMES:
What are you doing with that hammer?

JAMES: What are you doing?

If you do not know the name of the speaker, indicate the gender or age of the speaker if this is necessary for the viewer's understanding:

MAN: I was brought up in a close-knit family.

When two or more people are speaking simultaneously, do the following, regardless of their colours:

Two people:

BOTH: Keep quiet! (all white text)

Three or more:

ALL: Hello! (all white text)

TOGETHER: Yes! No! (different colours with a white label)

7.9 Use metadata to identify speakers

The subtitle file formats used by the BBC allow non-presentation metadata that can be used to include information about the speaker of a subtitle. Including this information is useful for searching, identifying speakers and other purposes.

Speakers can be identified using the `ttm:agent` attribute defined in the `tt/head/metadata` element and referenced by a `tt:span` element. This should be used wherever possible in **EBU-TT 1.0** documents and may be removed from **EBU-TT-D** documents prior to distribution, if the data is not needed by the presentation processor.

8 Colours

8.1 Use white on black

Most subtitles are typed in white text on a black background to ensure optimum legibility.

See Stress for the single case where colour may be used for emphasis.

Colours are implemented using `tts:color` and `tts:backgroundColor` applied to a `tt:span`.

Where two words are in different colours the space must be placed within one of the `tt:span` elements, usually as the first character of the second span.

8.2 Avoid coloured background

Background colours are no longer used. Use labels to identify non-human speakers:

ROBOT: Hello, sir

Use left-aligned sound labels for alerts:

BUZZER

8.3 Speaker colours

A limited range of colours can be used to distinguish speakers from each other. In order of priority:

Colour	RGB hex	Notes

Colour	RGB hex	Notes
White	`#FFFFFF`	
Yellow	`#FFFF00`	
Cyan	`#00FFFF`	
Green	`#00FF00`	In CSS, EBU-TT and TTML this is named colour `lime` .

All of the above colours must appear on a black background to ensure maximum legibility.

8.4 Apply speaker colour consistently

Once a speaker has a colour, they should keep that colour. Avoid using the same colour for more than one speaker - it can cause a lot of confusion for the viewer.

The exception to this would be content with a lot of shifting main characters like EastEnders, where it is permissible to have two characters per colour, providing they do not appear together. If the amount of placing needed would mean editing very heavily, you can use green as a "floater": that is, it can be used for more than one minor character, again providing they never appear together.

8.5 Multiple speakers in white

White can be used for any number of speakers. If two or more white speakers appear in the same scene, you have to use one of a number of devices to indicate who says what - see Identifying Speakers.

9 Typography

9.1 Fonts

Subtitle fonts are determined by the platform, the delivery mechanism and the client as detailed below. Since fonts have different character widths, the final pixel width of a line of subtitles cannot be accurately determined when authoring. See also Line Breaks.

To minimise the risk of unwanted line wrapping, use a wide font such as Reith Sans, Verdana or Tiresias when authoring the subtitles. Presentation processors usually use a narrower font (e.g. Arial) so the rendered line will likely fit within the authored area. Note that platforms may use different reference fonts when resolving the generic font family name specified in the subtitle file. For example, the HbbTV standard maps both `default` and `proportionalSansSerif` to Tiresias, whereas IMSC maps `proportionalSansSerif` only to any font with substantially the same dimensions for rendered text as Arial. See also Conformance with IMSC 1.0.1 Text Profile.

Platform	Delivery	Description
BROADCAST	DVB	The subtitle encoder creates bitmap images for each subtitle using the Tiresias Screenfont font
BROADCAST	Teletext	The set top box or television determines the font - this is most commonly used on the Sky platform
ONLINE	IP (XML)	The client determines the font using information from within the subtitle data (e.g. 'SansSerif'). Generally it is better to use system font for readability (e.g. Helvetica for iOS and Roboto for Android). Use of non-platform fonts can adversely impact clarity of presented text.

For implementation details, see tts:fontFamily.

9.2 Size

The final displayed size of closed captions text is determined by multiple factors: the instructions in the subtitle file, the processor and the set of installed fonts available to it, the device screen size and resolution and (on some devices) also user-defined preferences.

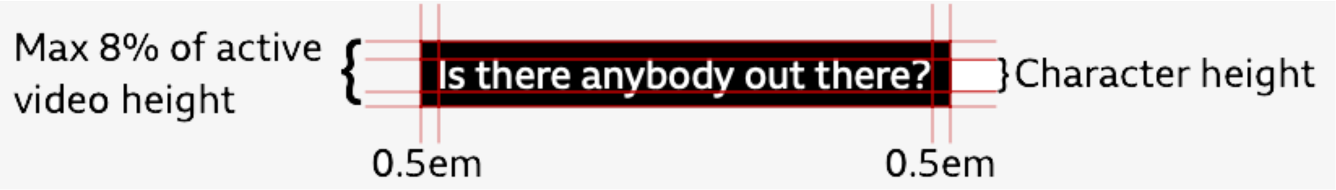
While it is not possible (or advisable) to pre-determine the final subtitle size, adhering to the below guidelines will ensure that subtitles are legible at a typical distance from the device and that lines do not reflow or overflow for the vast majority of users. In particular, the final size should never be larger than the authored size so that the subtitler can ensure that important parts of the of the video are not obscured.

9.2.1 Authoring font size

Font size should be set to fit within a line height of 8% of the active video height for 16:9, 4:3 and 1:1 aspect ratio videos.

Font size should be set to fit within a line height of 4.5% of the active video height for 9:16 aspect ratio videos.

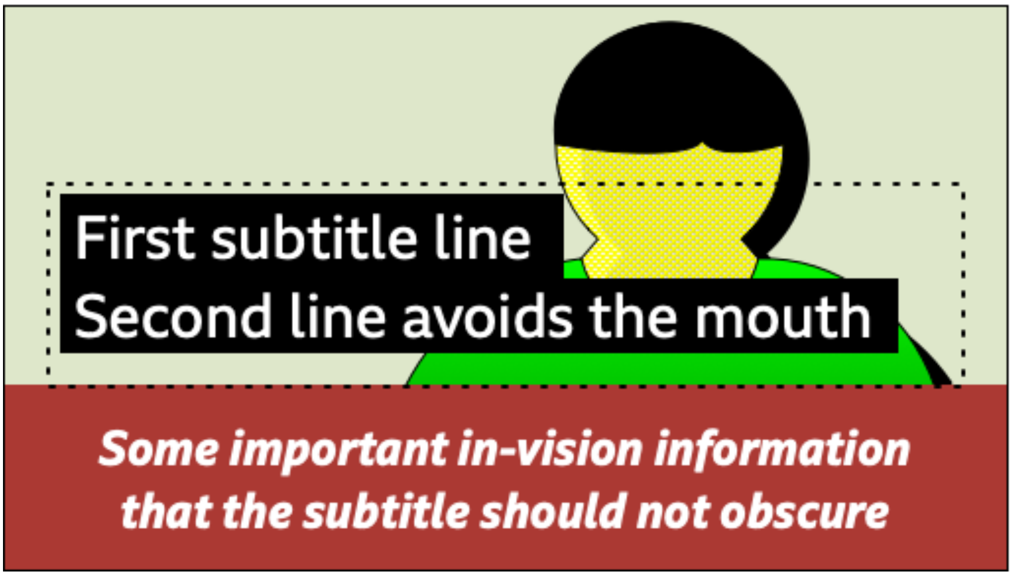
This font height is the largest size needed for presentation and is an *authoring* requirement.



Use a wide font such as Reith Sans when authoring subtitles (see Fonts and `tts:fontFamily`). If that is not the font used to present it, then the alternative is likely to be a narrower font, so if you author in a wide font you can be reasonably confident that lines will not reflow.

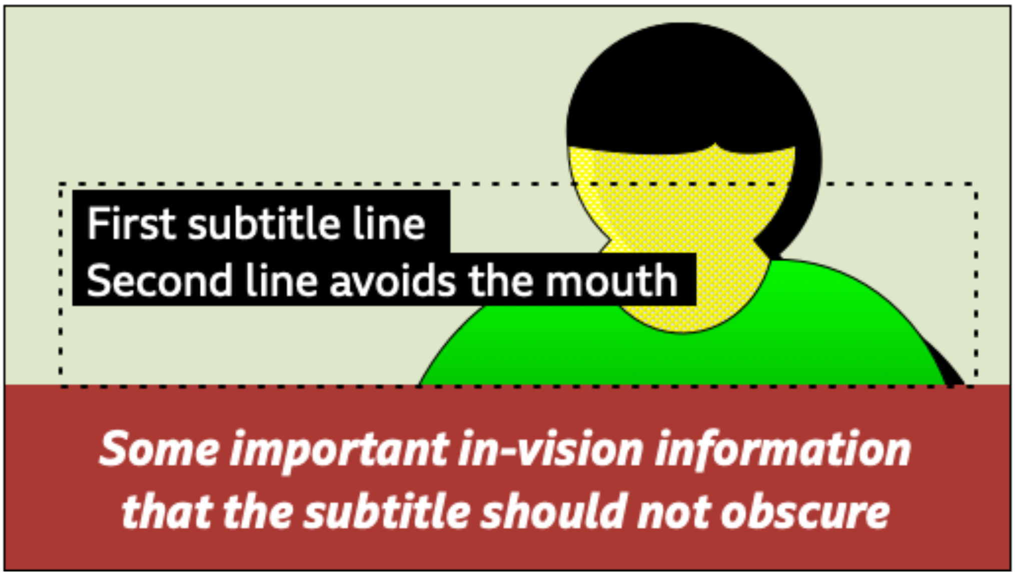
No changes need to be made to other styling attributes to accommodate processors potentially using a smaller font, however care needs to be taken when positioning subtitles in case a smaller font is used, as the following examples show:

Authored font size, correct positioning:


First subtitle line
Second line avoids the mouth
Some important in-vision information
that the subtitle should not obscure

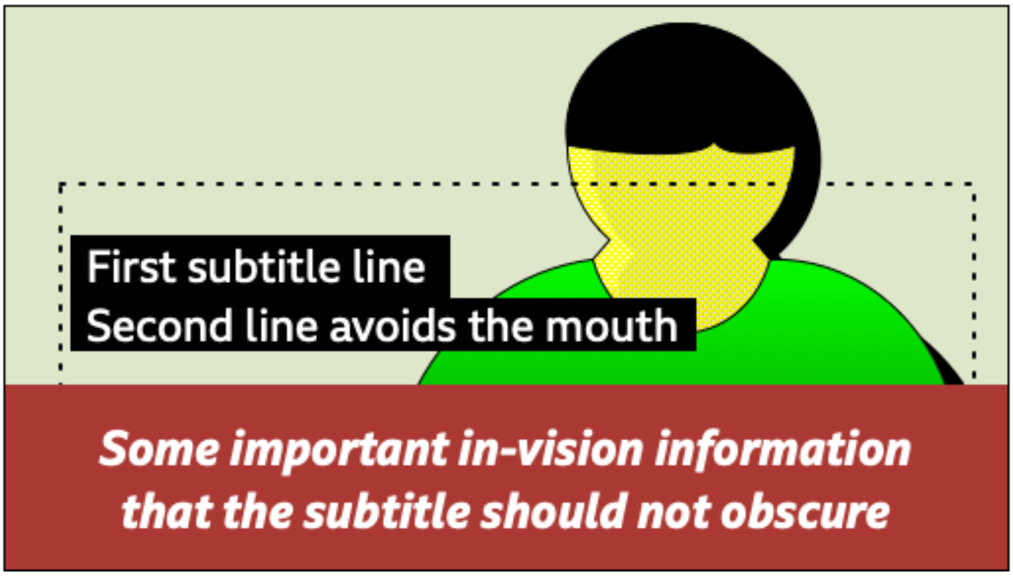
The processor displays the larger font size, as authored. The region (not displayed) is indicated with a dotted line.

Reduced font size, wrong positioning:


First subtitle line
Second line avoids the mouth
Some important in-vision information
that the subtitle should not obscure

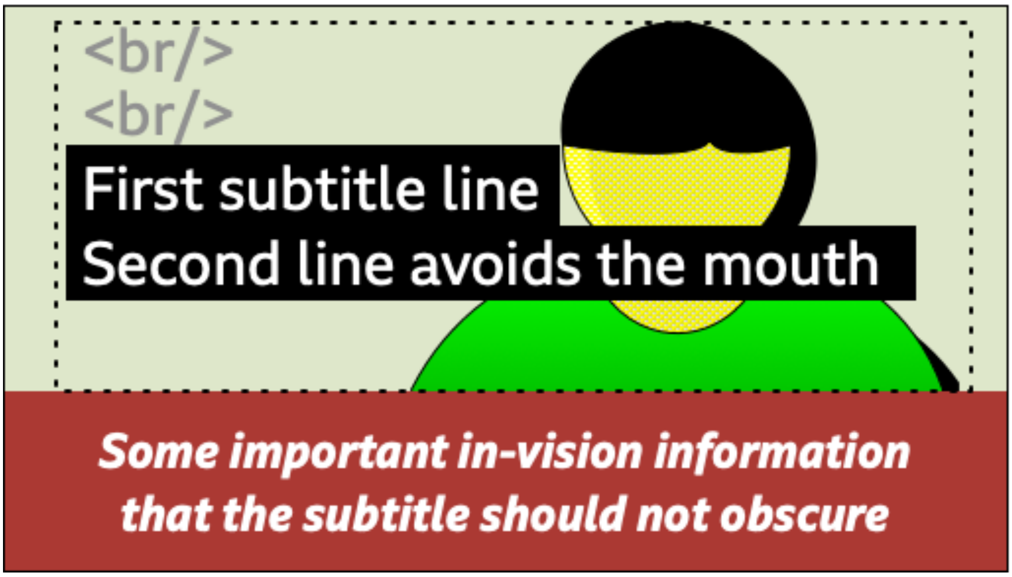
The region's `tts:displayAlign` is set to "before" so with a smaller font size the text moves up and the second line obscures the mouth.

Reduced font size, correct positioning:


First subtitle line
Second line avoids the mouth
Some important in-vision information
that the subtitle should not obscure

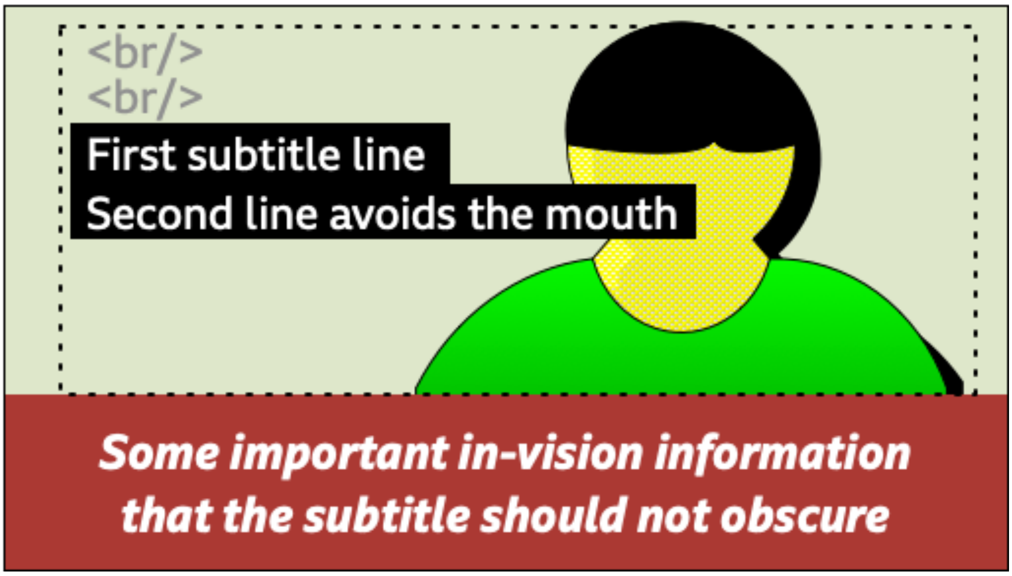
To avoid this, set the region's `tts:displayAlign` property to "center" or "after".

Authored font size, large region:


<br/>
<br/>
First subtitle line
Second line avoids the mouth
Some important in-vision information
that the subtitle should not obscure

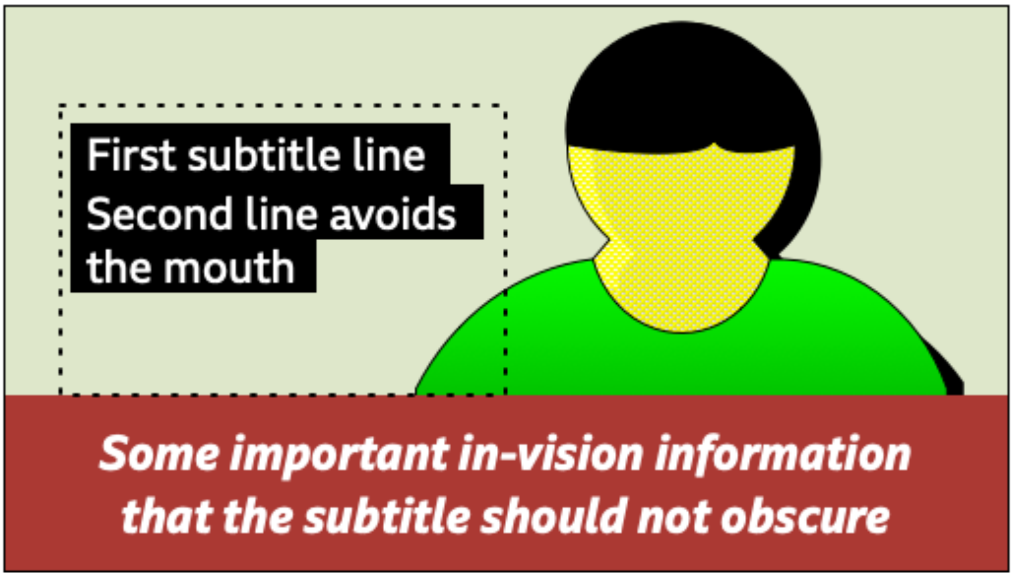
Line breaks were used to position the subtitles lower within the region.

Reduced font size, large region:


<br/>
<br/>
First subtitle line
Second line avoids the mouth
Some important in-vision information
that the subtitle should not obscure

The line breaks are resized with the rest of the text.

Reduced font size, defined region:


First subtitle line
Second line avoids
the mouth
Some important in-vision information
that the subtitle should not obscure

Better to define the region so that it does not cover the face and avoid white space.

The font size is determined by tts:fontSize in combination with ttp:cellResolution.

The 8% value originates in Teletext, where it is approximately the height of a double height line, with the Teletext rendering area covering around 90% of the video area height, as is typical when they are rendered within a safe area which accommodates overscan. We currently broadcast teletext subtitles on our digital satellite platforms: in principle, by connecting a television to a set top box using for example a SCART connector it would still be possible to display subtitles in this way, though in practice few if any new products are available that support such a mechanism. The double height line was used instead of a single height line for readability when commonly used television sizes were much smaller than today's median sizes (see BBC R&D White Paper 287 (PDF) for relevant research on this).

The 4.5% value results from adjusting the 8% value as applied to a 16:9 aspect ratio video for a 9:16 vertical video, i.e. by multiplying it by 9/16, so that subtitle text is approximately the same size on a landscape video as on a portrait/vertical video; since text size is proportional to the video height, 8% of a vertical video is much larger than 8% of a landscape video, when each is rotated to fill the same 16:9 ratio display.

9.2.2 Presentation font size

Depending on device size, viewing distance, screen resolution etc., a *processor* (such as a player) may choose to reduce (but not to increase) the authored font size so that the final presentation font size is smaller than the authored font size. For example, on a very large TV the subtitles may appear too large when displayed at the original authored size, so the processor can apply a scaling factor, or a multiplier of less than 1, to the value of `tts:fontSize`.

For most screen sizes, the preferred font size is between 0.6 and 0.8 times the required authoring font size. For small mobile phones (e.g. 4" diagonal screen size) the presentation size should be the unmodified authored font size (i.e. a multiplier of 1).

Along with reading distance, the physical height of the video when displayed on the device's screen is the most direct determinant of font size as a proportion of video height. In practice, however, a processor may not know the actual physical height and may have to rely on other data, for example pixel size and resolution (which may not be reliable indicators of physical size). The examples below illustrate devices and their recommended multipliers. For devices that support configurable sizes, a recommended range is shown. When the processor cannot determine the screen size, it should use the unmodified authored size to mitigate the risk of illegibly small text (i.e. default to a multiplier of 1).

Device type	Example device	Screen height (landscape)	Recommended multiplier	Recommended range
4" (10cm) phone	iPhone SE	50mm	x1	x0.67-x1
4.7" (12cm) phone	iPhone 6	59mm	x1	x0.67-x1
5.5" (14cm) phone	Samsung S7	68mm	x0.67	x0.5-x1
7" (17.8cm) tablet	Amazon Fire	87mm	x0.8	x0.6-x1
9.7" (24.5cm) tablet	iPad	148mm	x0.67	x0.6-x1
Laptop and desktop computers	16:9 monitor	187mm-300mm	x0.6	x0.5-x1
TVs (32"-42")	16:9 or 21:9 display	398mm-523mm	x0.67	x0.5-x1
Unknown device	Unknown	Unknown	x1	x0.67-x1

The same multipliers apply regardless of the aspect ratio of the video. See Authoring font size above for the size adjustment needed for vertical (9:16) video.

In the absence of other information, a default size of 0.5° subtended at the eye may be used to derive the default line height and calculate the multiplier, however this may be too small for some devices.

> When scaling down the font size, the processor should respect all other styling attributes. Subtitle text should be scaled by applying the multiplier to the values of `tts:fontSize` and `ebutts:linePadding` . In EBU-TT-D line height is specified as a percentage of font size, so its computed value scales proportionally without having to modify the value of `tts:lineHeight` .

9.2.3 Additional adjustments for Reith Sans font

EBU-TT-D This section previously contained guidance for adjusting the line height and font size when presenting using the Reith Sans font. That guidance no longer applies due to adjustments made in versions of the Reith Sans font released in or after January 2021, so the contents of this section have been removed.

9.2.4 Background size

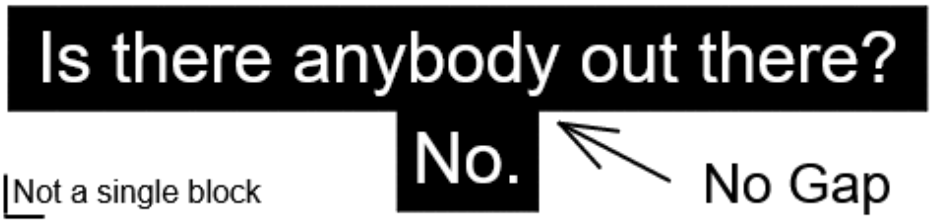
The width of the background is calculated per line, rather than being the largest rectangle that can fit all the displayed lines in.

> To achieve this, wrap the text in a `tt:span` and apply a `tts:backgroundColor` style to the `tt:span` by referencing a `<tt:style>` element that sets that style attribute.

The height of the background should be the height of the line; there should be no gap between background areas of successive lines.

> Reference a `<tt:style>` element that sets `itts:fillLineGap="true"` from the `tt:p` parent, or its `tt:div` ancestor, to instruct the processor to fill gaps between adjacent line background areas.

On both sides of every line, the background colour should extend by the width of 0.5 em.



> In EBU-TT-D, the background of lines is extended using `ebutts:linePadding`. Note, however, that the size of line padding is expressed in cell units, requiring additional calculation. For this purpose, `1em` can be assumed to equal font size. See example in `ebutts:linePadding`. If scaling the presentation font size, a processor

should reduce the value of `ebutts:linePadding` by the same factor.

9.3 Supported characters

9.3.1 Broadcast

If the subtitles are intended for broadcast, a limited set of characters must be used.

Use alphanumeric and English punctuation characters:

A-Z a-z 0-9 !) (, . ? : -

The following characters can be used:

> < & @ # % + * = / £ $ ¢ ¥ © ® ¼ ½ ¾ ¾ ™

Do not use accents.

Additional characters are supported but not normally used (see Appendix 1)

9.3.2 Characters permitted online

In addition to the characters above, the following characters are allowed if the subtitles are intended for online use only.

ONLINE € ♫ (replaces # to indicate music) ← → (arrows can replace < and >).

9.3.3 Encoding characters

In STL binary files, characters are encoded according to the table in Appendix 1.

Subtitles delivered as XML (EBU-TT or EBU-TT-D) require that characters with special significance in XML are escaped:

Character	Escaped	Example
<	&lt;	<span style="spanStyle">3 &lt; 5</span>
>	&gt;	<span style="spanStyle">5 &gt; 3</span>
&	&amp;	<span style="spanStyle">Trotter &amp; Sons</span>

Quote marks within subtitle content don't have to be escaped. This is valid:

```
<span style="spanStyle">"Hello"</span>
```

Note, however, that curly quotes are not included in the list of allowed characters (some word processors transform straight quotes to curly ones automatically).

You may not be able directly to key in some of the other allowed characters. In this case you can use the Unicode code. For example, use &#9835; for the character ♫, like this:

```
<span style="spanStyle">&#9835; Happy birthday to you</span>
```

This will be displayed as:

♫ Happy birthday to you

You can view a list of Unicode codes on the Unicode website.

10 Positioning

The subtitles should overlay the video image, and may be placed within any black bars present within the video at the top or bottom.

ONLINE For 16:9 video in landscape mode, subtitles should not be placed outside the central 90% vertically and the central 75% horizontally.

ONLINE For 9:16 video in portrait or vertical mode, this is reversed: subtitles should not be placed outside the central 75% vertically and the central 90% horizontally.

ONLINE Regions can be extended horizontally to allow extra space for line padding.

ONLINE For online subtitles, the subtitle rendering area (root container in EBU-TT-D) should exactly overlap the video player area unless controls or other overlays are visible, in which case the system should take steps to avoid the subtitles being obscured by the overlays. These could include:

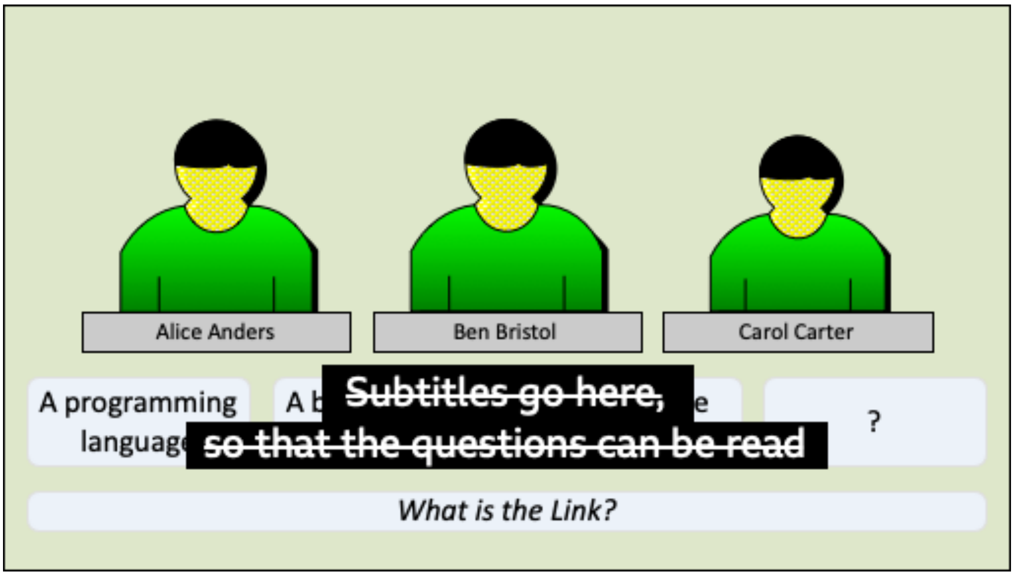
- Scaling the root container to avoid overlap
- Detecting and resolving screen area clashes by moving subtitles around
- Pausing the presentation while the overlays are visible.

10.1 Vertical positioning

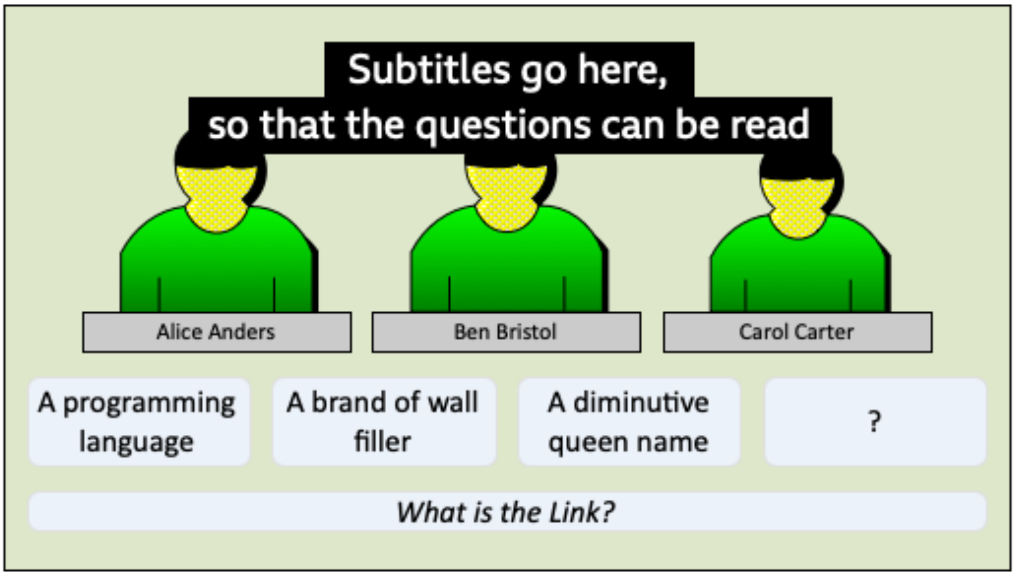
The normally accepted position for subtitles is towards the bottom of the screen (Teletext lines 20 and 22. Line 18 is used if three subtitle lines are required). In obeying this convention it is most important to avoid obscuring ‘on-screen’ captions, any part of a speaker’s mouth or any other important activity. Certain special programme types carry a lot of information in the lower part of the screen (e.g. snooker, where most of the activity tends to centre around the black ball) and in such cases top screen positioning will be a more acceptable standard.

In vertical (9:16) videos, it is common to position subtitles a little higher up, though still generally in the lower third of the screen. This is because faces are generally in the top half of the screen, and positioning the subtitles at the bottom makes it harder for the viewer to read the text and see the person speaking.

Generally, vertical displacement should be used to avoid obscuring important information (such as captions) while horizontal displacement should be reserved for indicating speakers (see Identifying Speakers).



✗ This is bad, placing subtitles here would cover the names.



✓ This is better, in this instance, the subtitles should go here so that the names can be read.

In some cases vertical displacement is not sufficient to avoid obscuring important information, for example when placing the captions above a graphic would cover a face. In such cases, horizontal positioning may be used.

Vertical positioning is controlled mainly by the `tt:region` element, which is defined using `tts:extent` and `tts:origin`. However, other attributes can also affect positioning within the region. See `tts:displayAlign` for more details.

10.2 Under image positioning

Some platforms (e.g. online media player) support the display of subtitles under the image. If the media player is embedded in the page the layout should change to accommodate the subtitle display.

When subtitles are displayed under the image area, vertical displacement will be ignored by the device and only horizontal positioning will be used (e.g. to identify speakers).

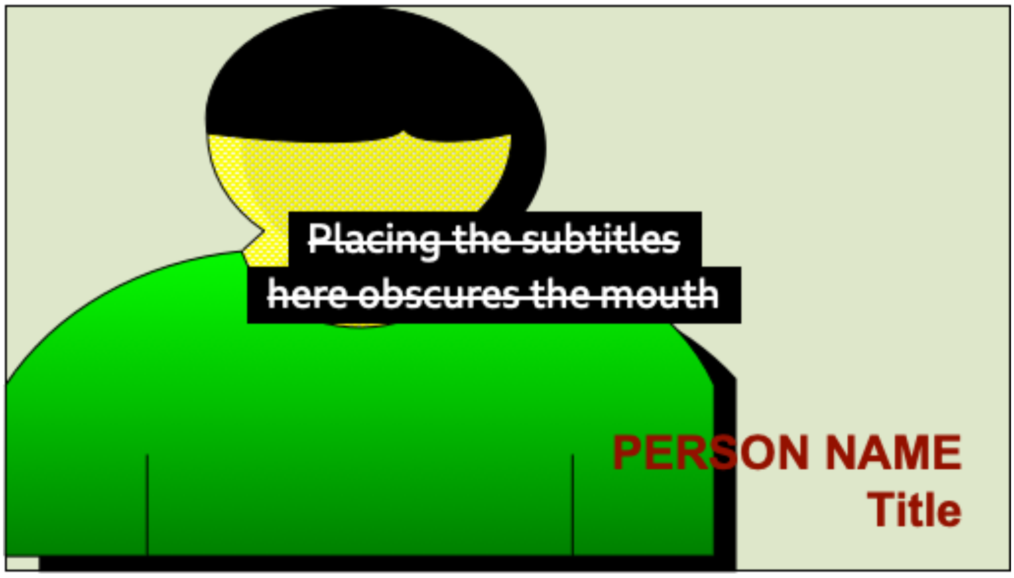
10.3 Horizontal positioning

Prepared subtitles are normally centre-aligned within a subtitle region that is horizontally centred relative to the video. Live subtitles (cued blocks and cumulative) are normally left-aligned.

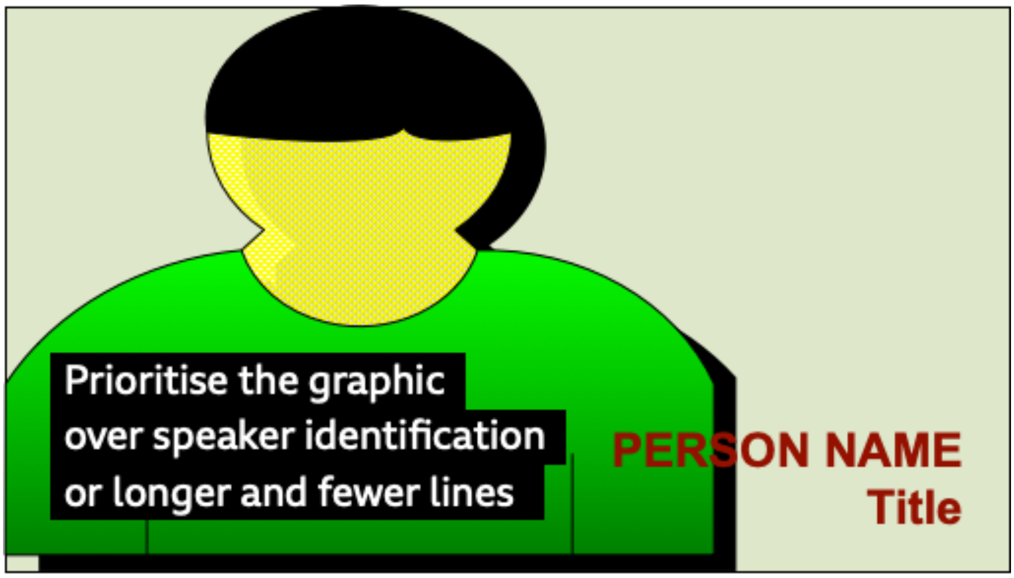
Other horizontal positioning may be used to:

- Avoid obscuring important information (such as captions and mouths) when vertical positioning is not sufficient (see below).
- Indicate the direction of off-screen sounds. See arrows for off-screen voices.
- Identifying in-vision speakers (legacy technique). See Identifying speakers with horizontal positioning.

In some cases vertical positioning is not sufficient to avoid obscuring important information, for example when placing the captions above a graphic would cover a face. In such cases, prioritise the important information over speaker identification, using horizontal positioning if appropriate.



✗ This is bad, placing the subtitles here would obscure the speaker's mouth.



✓ This is better, prioritise the graphic over the speaker's identification, or longer and fewer lines.

Horizontal positioning is controlled by the `tt:region` element, whose size and position are defined using `tts:extent` and `tts:origin`. Within the region, horizontal alignment of lines is achieved using `tts:textAlign` and `ebutts:multiRowAlign`.

11 Intonation and emotion

11.1 Sarcasm

To indicate a sarcastic statement, use an exclamation mark in brackets (without a space in between):

Charming(!)

To indicate a sarcastic question, use a question mark in brackets:

You're not going to work today, are you(?)

11.2 Stress

Use caps to indicate when a word is stressed. Do not overuse this device - text sprinkled with caps can be hard to read. However, do not underestimate how useful the occasional indication of stress can be for conveying meaning:

It's the BOOK I want, not the paper.

I know that, but WHEN will you be finished?

The word "I" is a special case. If you have to emphasise it in a sentence, make it a different colour from the surrounding text. However, this is rare and should be used sparingly and only when there is no other way to emphasise the word.

She knows. Elaine wrote to her. But
if I write, she'll know it's urgent.

Use caps also to indicate when words are shouted or screamed:

HELP ME!

However, avoid large chunks of text in caps as they can be hard to read.

11.2.1 Italics

ONLINE Subtitles for online exclusives can use italics for emphasis instead of caps (this is an experimental option and should not be included for general use). If this approach is adopted italics should be used in most instances, with caps reserved for heavier emphasis (e.g. shouting).

Note that there is currently little research to indicate the effectiveness of italics for emphasis in subtitles.

> Italics can be specified by using `tts:fontStyle="italic"` on a style referenced by a `tt:span`.

11.3 Whisper

To indicate whispered speech, a label is most effective.

WHISPERS:
Don't let him near you.

However, when time is short, place brackets around the whispered speech:

(Don't let him near you.)

If the whispered speech continues over more than one subtitle, brackets can start to look very messy, so a label in the first subtitle is preferable.

Brackets can also be used to indicate an aside, which may or may not be whispered.

11.4 Incredulous question

Indicate questions asked in an incredulous tone by means of a question mark followed by an exclamation mark (no space):

You mean you're going to marry him?!

12 Accents

This section deals with accents in speech and dialects. For accented characters see Typography.

12.1 Indicate accent only when required

Do not indicate accent as a matter of course, but only where it is relevant for the viewer's understanding. This is rarely the case in serious/straight news reports, but may well be relevant in lighter factual items. For example, you would only indicate the nationality of a foreign scientist being interviewed on Horizon or the Ten O'Clock News if it were relevant to the subject matter and the viewer could not pick the information up from any other source, e.g. from their actual words or any accompanying graphics. However, in a drama or comedy where a character's accent is crucial to the plot or enjoyment, the subtitles must establish the accent when we first see the character and continue to reflect it from then on.

12.2 Indicate accent sparingly

When it is necessary to indicate accent, bear in mind that, although the subtitler's aim should always be to reproduce the soundtrack as faithfully as possible, a phonetic representation of a speaker's foreign or regional accent or dialect is likely to slow up the reading process and may ridicule the speaker. Aim to give the viewer a flavour of the accent or dialect by spelling a few words phonetically and by including any unusual vocabulary or sentence construction that can be easily read. For a Cockney speaker, for instance, it would be appropriate to include quite a few "caffs", "missus" and "ain'ts", but not to replace every single dropped "h" and "g" with an apostrophe.

12.3 Incorrect grammar

You should not correct any incorrect grammar that forms an essential part of dialect, e.g. the Cockney "you was".

A foreign speaker may make grammatical mistakes that do not render the sense incomprehensible but make the subtitle difficult to read in the given time. In this case, you should either give the subtitle more time or change the text as necessary:

I and my wife is being marrying four years since and are having four childs, yes

This could be changed to:

I and my wife have been married four years and have four childs, yes

12.4 Use label

The speech text alone may not always be enough to establish the origin of an overseas/regional speaker. In that case, and if it is necessary for the viewer's understanding of the context of the content, use a label to make the accent clear:

AMERICAN ACCENT:
All the evidence points to a plot.

---

13 Difficult speech

13.1 Edit lightly

Remember that what might make sense when it is heard might make little or no sense when it is read. So, if you think the viewer will have difficulty following the text, you should make it read clearly. This does not mean that you should always sub-edit incoherent speech into beautiful prose. You should aim to tamper with the original as little as possible - just give it the odd tweak to make it intelligible. (Also see Accents)

13.2 Consider the dramatic effect

The above is more applicable to factual content, e.g. News and documentaries. Do not tidy up incoherent speech in drama when the incoherence is the desired effect.

13.3 Use labels for incoherent speech

If a piece of speech is impossible to make out, you will have to put up a label saying why:

(SLURRED): But I love you!

Avoid subjective labels such as "UNINTELLIGIBLE" or "INCOMPREHENSIBLE" or "HE BABBLES INCOHERENTLY".

13.4 Use labels for inaudible speech

Speech can be inaudible for different reasons. The subtitler should put up a label explaining the cause.

APPLAUSE DROWNS SPEECH

TRAIN DROWNS HIS WORDS

MUSIC DROWNS SPEECH

HE MOUTHS

13.5 Explain pauses in speech

Long speechless pauses can sometimes lead the viewer to wonder whether the subtitles have failed. It can help in such cases to insert explanatory text such as:

INTRODUCTORY MUSIC

LONG PAUSE

ROMANTIC MUSIC

13.6 Break up subtitles slow speech

If a speaker speaks very slowly or falteringly, break your subtitles more often to avoid having slow subtitles on the screen. However, do not break a sentence up so much that it becomes difficult to follow.

13.7 Indicate stammer

If a speaker stammers, give some indication (but not too much) by using hyphens between repeated sounds. This is more likely to be needed in drama than factual content. Letters to show a stammer should follow the case of the first letter of the word.

I'm g-g-going home

W-W-What are you doing?

14 Hesitation and interruption

14.1 Indicate hesitation only if important

If a speaker hesitates, do not edit out the "ums" and "ers" if they are important for characterisation or plot. However, if the hesitation is merely incidental and the "ums" actually slow up the reading process, then edit them out. (This is most likely to be the case in factual content, and too many "ums" can make the speaker appear ridiculous.)

14.2 Within a single subtitle

When the hesitation or interruption is to be shown within a single subtitle, follow these rules:

14.2.1 Pause within a sentence

To indicate a pause within a sentence, insert three dots at the point of pausing, then continue the sentence immediately after the dots, without leaving a space.

Everything that matters...is a mystery

You may need to show a pause between two sentences within one subtitle. For example, where a phone call is taking place and we can only witness one side of it, there may not be time to split the sentences into separate subtitles to show that someone we can't see or hear is responding. In this case, you should put two dots immediately before the second sentence.

How are you? ..Oh, I'm glad to hear that.

A very effective technique is to use cumulative subtitles, where the first part appears before the second, and both remain on screen until the next subtitle. Use this method only when the content justifies it; standard prepared subtitles should be displayed in blocks.

14.2.2 Unfinished sentence

If the speaker simply trails off without completing a sentence, put three dots at the end of their speech. If they then start a new sentence, no continuation dots are necessary.

Hello, Mr... Oh, sorry! I've forgotten your name

14.2.3 Unfinished question/exclamation

If the unfinished sentence is a question or exclamation, put three dots (not two) before the question mark or exclamation mark.

What do you think you're...?!

14.2.4 Interruption

If a speaker is interrupted by another speaker or event, put three dots at the end of the incomplete speech.

14.3 Across subtitles

When the hesitation or interruption occurs in the middle of a sentence that is split across two subtitles, do the following:

14.3.1 Indicate time lapse with dots

Where there is no time-lapse between the two subtitles, put three dots at the end of the first subtitle but no dots in the second one.

I think...
I would like to leave now.

Where there is a time-lapse between the two subtitles, put three dots at the end of the first subtitle and two dots at the beginning of the second, so that it is clear that it is a continuation.

I'd like...

..a piece of chocolate cake

Remember that dots are only used to indicate a pause or an unfinished sentence. You do not need to use dots every time you split a sentence across two or more subtitles.

15 Humour

In humorous sequences, it is important to retain as much of the humour as possible. This will affect the editing process as well as when to leave the screen clear.

15.1 Separate punchlines

Try wherever possible to keep punchlines separate from the preceding text.

15.2 Reactions

Where possible, allow viewers to see actions and facial expressions which are part of the humour by leaving the screen clear or by editing. Try not to leave a subtitle on screen when the next shot contains no speech and shows the character's reaction, as this distracts from the reaction and spoils the punchline.

15.3 Keep catchphrases

Never edit characters' catchphrases.

16 Music and songs

EBU-TT 1.0 documents should set `ttm:role="music"` on the relevant `tt:p` or `tt:span` element to indicate that the contents represent music.

16.1 Label source music

All music that is part of the action, or significant to the plot, must be indicated in some way. If it is part of the action, e.g. somebody playing an instrument/a record playing/music on a jukebox or radio, then write the label in upper case:

SHE WHISTLES A JOLLY TUNE

POP MUSIC ON RADIO

MILITARY BAND PLAYS SWEDISH NATIONAL ANTHEM

16.2 Describe incidental music

If the music is "incidental music" (i.e. not part of the action) and well known or identifiable in some way, the label begins "MUSIC:" followed by the name of the music (music titles should be fully researched). "MUSIC" is in caps (to indicate a label), but the words following it are in upper and lower case, as these labels are often fairly long and a large amount of text in upper case is hard to read.

MUSIC: "The Dance Of The Sugar Plum Fairy"
by Tchaikovsky

MUSIC: "God Save The Queen"

MUSIC: A waltz by Victor Herbert

MUSIC: The Swedish National Anthem

(The Swedish National Anthem does not have quotation marks around it as it is not the official title of the music.)

16.3 Combine source and incidental music

Sometimes a combination of these two styles will be appropriate:

HE HUMS "God Save The Queen"

SHE WHISTLES "The Dance Of The Sugar Plum Fairy"
by Tchaikovsky

16.4 Label mood music only when required

If the music is "incidental music" but is an unknown piece, written purely to add atmosphere or dramatic effect, do not label it. However, if the music is not part of the action but is crucial for the viewer’s understanding of the plot, a sound-effect label should be used:

EERIE MUSIC

16.5 Indicate song lyrics with

Song lyrics are almost always subtitled - whether they are part of the action or not. Every song subtitle starts with a white hash mark (#) and the final song subtitle has a hash mark at the start and the end:

These foolish things remind me of you

There are two exceptions:

- In cases where you consider the visual information on the screen to be more important than the song lyrics, leave the screen free of subtitles.

- Where snippets of a song are interspersed with any kind of speech, and it would be confusing to subtitle both the lyrics and the speech, it is better to put up a music label and to leave the lyrics unsubtitled.

ONLINE Instead of # the symbol, ♬ may be used.

16.6 Avoid editing lyrics

Song lyrics should generally be verbatim, particularly in the case of well-known songs (such as God Save The Queen), which should never be edited. This means that the timing of song lyric subtitles will not always follow the conventional timings for speech subtitles, and the subtitles may sometimes be considerably faster.

If, however, you are subtitling an unknown song, specially written for the content and containing lyrics that are essential to the plot or humour of the piece, there are a number of options:

- edit the lyrics to give viewers more time to read them

- combine song-lines wherever possible

- do a mixture of both - edit and combine song-lines.

NB: If you do have to edit, make sure that you leave any rhymes intact.

16.7 Synchronise with audio

Song lyric subtitles should be kept closely in sync with the soundtrack. For instance, if it takes 15 seconds to sing one line of a hymn, your subtitle should be on the screen for 15 seconds.

Song subtitles should also reflect as closely as possible the rhythm and pace of a performance, particularly when this is the focus of the editorial proposition. This will mean that the subtitles could be much faster or slower than the conventional timings.

There will be times where the focus of the content will be on the lyrics of the song rather than on its rhythm - for example, a humorous song like Ernie by Benny Hill. In such cases, give the reader time to read the lyrics by combining song-lines wherever possible. If the song is unknown, you could also edit the lyrics, but famous songs like Ernie must not be edited.

Where shots are not timed to song-lines, you should either take the subtitle to the end of the shot (if it's only a few frames away) or end the subtitle before the end of the shot (if it's 12 frames or more away).

16.8 Centre lyrics subtitles

All song-lines should be centred on the screen.

> This can be achieved by referencing a `tt:region` that is positioned centrally (horizontally), and a style with `tts:textAlign="center"` and `ebutts:multiRowAlign` either unspecified or set to `"auto"` .

16.9 Punctuation

It is generally simpler to keep punctuation in songs to a minimum, with punctuation only within lines (when it is grammatically necessary) and not at the end of lines (except for question marks). You should, though, avoid full stops in the middle of otherwise unpunctuated lines. For example,

Turn to wisdom. Turn to joy
There’s no wisdom to destroy

Could be changed to:

# Turn to wisdom, turn to joy
There’s no wisdom to destroy

In formal songs, however, e.g. opera and hymns, where it could be easier to determine the correct punctuation, it is more appropriate to punctuate throughout.

The last song subtitle should end with a full stop, unless the song continues in the background.

If the subtitles for a song don't start from its first line, show this by using two continuation dots at the beginning:

# ..Now I need a place to hide away
# Oh, I believe in yesterday. #

Similarly, if the song subtitles do not finish at the end of the song, put three dots at the end of the line to show that the song continues in the background or is interrupted:

# I hear words I never heard in the Bible... #

---

17 Sound effects

EBU-TT 1.0 Sound effects should be labelled as such using an appropriate `ttm:role` (TTML1 link), for example by adding the attribute `ttm:role="sound"` to the `tt:p` element.

17.1 Subtitle effects only when necessary

As well as dialogue, all editorially significant sound effects must be subtitled. This does not mean that every single creak and gurgle must be covered - only those which are crucial for the viewer's understanding of the events on screen, or which may be needed to convey flavour or atmosphere, or enable them to progress in gameplay, as well as those which are not obvious from the action. A dog barking in one scene could be entirely trivial; in another it could be a vital clue to the story-line. Similarly, if a man is clearly sobbing or laughing, or if an audience is clearly clapping, do not label.

Do not put up a sound-effect label for something that can be subtitled. For instance, if you can hear what John is saying, JOHN SHOUTS ORDERS would not be necessary.

17.2 Describe sounds, not actions

Sound-effect labels are not stage directions. They describe sounds, not actions:

GUNFIRE

not:

~~THEY SHOOT EACH OTHER~~

17.3 Format

A sound effect should be typed in white caps. It should sit on a separate line and be placed to the left of the screen - unless the sound source is obviously to the right, in which case place to the right.

> There is no style attribute that enforces all caps; the text needs to be capitalised within the subtitle document.

17.4 Subject + verb

Sound-effect labels should be as brief as possible and should have the following structure: subject + active, finite verb:

FLOORBOARDS CREAK

JOHN SHOUTS ORDERS

Not:

~~CREAKING OF FLOORBOARDS~~

Or

~~FLOORBOARDS CREAKING~~

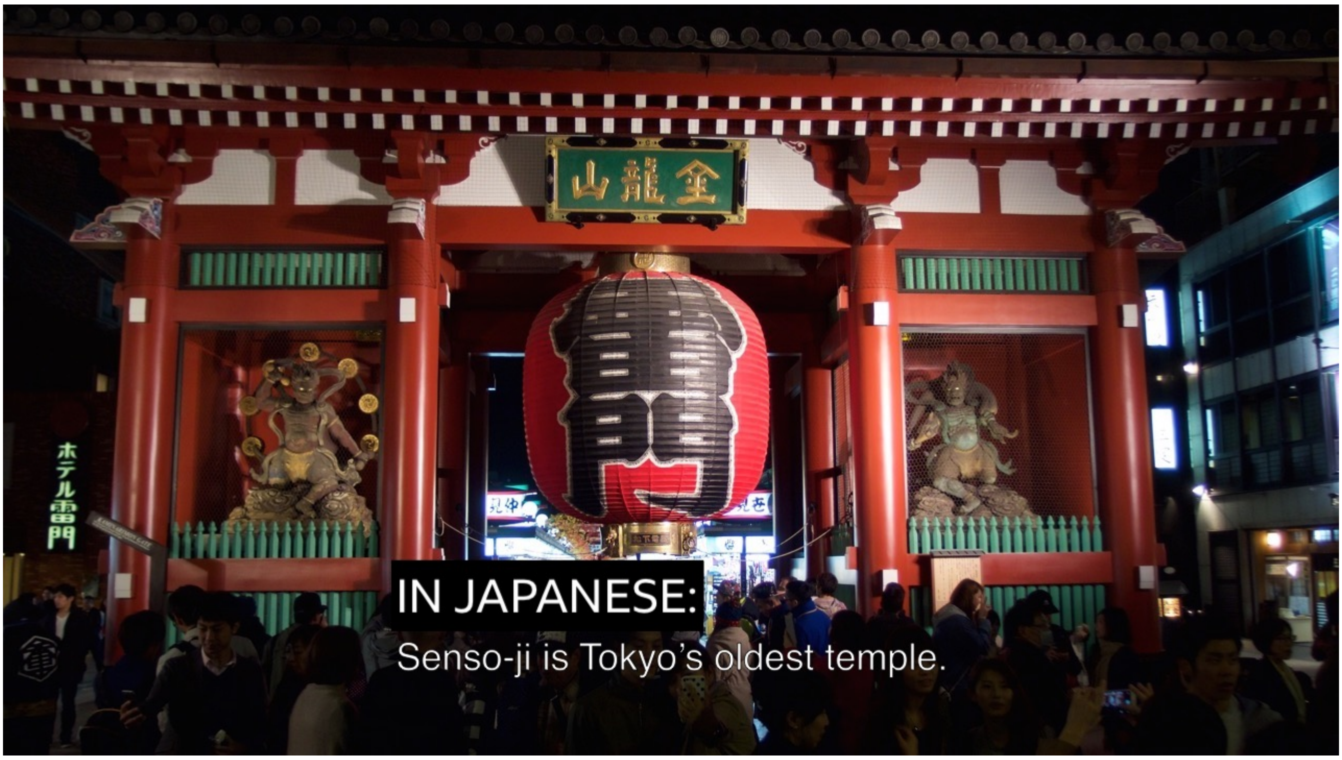
Or

~~ORDERS ARE SHOUTED BY JOHN~~

> There is no obvious value for `ttm:role` for such labels. The closest fit is probably `"description"` .

17.5 In-vision translations

If a speaker speaks in a foreign language and in-vision translation subtitles are given, use a label to indicate the language that is being spoken. This should be in white caps, ranged left above the in-vision subtitle, followed by a colon. Time the label to coincide with the timing of the first one or two in-vision subtitles. Bring it in and out with shot-changes if appropriate.



IN JAPANESE:

Senso-ji is Tokyo’s oldest temple.

If there are a lot of in-vision subtitles, all in the same language, you only need one label at the beginning - not every time the language is spoken.

If the language spoken is difficult to identify, you can use a label saying TRANSLATION:, but only if it is not important to know which language is being spoken. If it is important to know the language, and you think the hearing viewer would be able to detect a language change, then you must find an appropriate label.

17.6 Animal noises

The way in which subtitlers convey animal noises depends on the content style. In factual wildlife, for instance, lions would be labelled:

LIONS ROAR

However, in an animation or a game, it may be more appropriate to convey animal noises phonetically. For instance, "LIONS ROAR" would become something like:

Rrrarrgghhh!

18 Numbers

18.1 Spelling out

In general, the numeral form should be used. However, you can spell out numbers when this is editorially justified as detailed below.

The numbers 1-10 are often better spelled out:

I'll see you in three days

~~I'll see you in 3 days~~

But use the numeral with units:

It takes 1kJ of energy to lift someone.

~~It takes one kJ of energy to lift someone.~~

Emphatic numbers are always spelled out:

She gave me hundreds of reasons

~~She gave me 100s of reasons~~

Spell out any number that begins a sentence:

Three days from now.

~~3 days from now.~~

If there is more than one number in a sentence or list, it may be more appropriate to display them as numerals instead of words:

On her 21st birthday party, 54 guests turned up

Consistency is important, so avoid

~~the score was three - 1~~

Numerals over 4 digits must include appropriately placed commas:

There are 1,500 cats here.

For sports, competitions, games or quizzes, always use numerals to display points, scores or timings.

18.2 Dates

For displaying the day of the month, use the appropriate numeral followed by lowercase "th", "st" or "nd":

April 2nd.

18.3 Money

18.3.1 Sterling

Use the numerals plus the £ sign for all monetary amounts except where the amount is less than £1.00:

We paid £50.

For amounts less than £1.00 the word "pence" should be used after the numeral:

58 pence.

If the word "pound" is used in sentence without referring to a specific amount, then the word must be used, not the symbol.

18.3.2 Other currencies

See the list of supported characters for currency symbols you can use for broadcast and online.

BROADCAST Spell out other currencies, including Euro (the Euro symbol is not supported in Teletext).

ONLINE Use the correct Unicode symbol for the currency, e.g. the Euro symbol €.

All subtitle documents should be encoded in UTF-8, however the actual set of code points usable in an **EBU-TT 1.0** document intended for broadcast presentation is currently restricted to the Teletext character set.
No such restriction exists for **EBU-TT-D** documents intended for online-only presentation, however care should be taken that there is a reasonable expectation that the presentation device will have a font installed that contains glyphs for all the code points used.

18.4 Time

Indicate the time of the day using numerals in a manner which reflects the spoken language:

The time now is 4:30

The alarm went off at 4 o’clock

18.5 Measurement

Never use symbols for units of measurement.

Abbreviations can be used to fit text in a line, but if the unit of measurement is the subject do not abbreviate.

19 Cumulative subtitles

A cumulative subtitle consists of two or three parts - usually complete sentences. Each part will appear on screen at a different time, in sync with its speaker, but all parts will have an identical out-cue.

19.1 Use only when necessary

Cumulatives should only be used when there is a good reason to delay part of the subtitle (e.g. dramatic impact/song rhythm) and no other way of doing it - i.e. there is insufficient time available to split the subtitle completely.

This is most likely to happen in an interchange between speakers, where the first speaker talks much faster than the second. Delaying the speech of the second person by using a cumulative means that the first subtitle will still be on screen long enough to be read, while at the same time the speech is kept in sync.

19.2 Common scenarios

Cumulatives are particularly useful in the following situations:

- For jokes - to keep punch lines separate
- In quizzes - to separate questions and answers

- In songs - e.g. for backing singers. They are particularly effective when one line starts before the previous one finishes
- To delay dramatic responses (However, if a response is not expected, a cumulative can give the game away)
- When an exclamation/sound effect label occurs just before a shot-change, and would otherwise need to be merged with the preceding subtitle
- To distinguish between two or more white speakers in the same shot

19.3 Timing

Make sure there is sufficient time to read each segment of a cumulative, especially the final one. Consider leaving the final part on screen for a slightly longer time to allow the viewer to scan the line again.

If you use cumulatives in children's content, observe children's timings.

> Further detail on how to specify cumulatives is described in `tt:p` and `tt:span` . Where possible, each individual word that forms part of a cumulative subtitle should be included in the subtitle document exactly once, with appropriate timing specified by putting groups of words that appear with the same timing within a `tt:span` with `begin` and `end` attributes. This allows the plain text of the subtitle transcript to be extracted more easily since there is no need to de-duplicate words.
>
> There is an alternative approach in which multiple `tt:p` elements are each timed to follow on from each other, with the first words being a repeat of the words in the previous `tt:p` and additional words appended. This approach creates the same visual effect but should be avoided.

19.4 Avoid cumulative where shots change

Be wary of timing the appearance of the second/third line of a cumulative to coincide with a shot-change, as this may cause the viewer to reread the first line.

19.5 Avoid obscuring important information

Remember that using a cumulative will often mean that more of the picture is covered. Don’t use cumulatives if they will cover mouths, or other important visuals

19.6 Stick to three lines

Stick to a maximum of three lines unless you are subtitling a fast quiz like University Challenge where it is preferable to show the whole question in one subtitle and where you will not be obscuring any interesting visuals

20 Children’s subtitling

The following guidelines are recommended for the subtitling of programmes targeted at children below the age of 11 years (ITC).

20.1 Editing

There should be a match between the voice and subtitles as far as possible.

A strategy should be developed where words are omitted rather than changed to reduce the length of sentences.

For example,

Can you think why they do this?

Why do they do this?

Can you think of anything you could do with all the heat produced in the incinerator?

What could you do with the heat from the incinerator?

Difficult words should also be omitted rather than changed. For example:

First thing we're going to do is make his big, ugly, bad-tempered head.

First we're going to make his big, ugly head.

All she had was her beloved rat collection.

She only had her beloved rat collection.

Where possible the grammatical structure should be simplified while maintaining the word order.

You can see how metal is recycled if we follow the aluminium.

See how metal is recycled by following the aluminium.

We need energy so our bodies can grow and stay warm.

We need energy to grow and stay warm.

Difficult and complex words in an unfamiliar context should remain on screen for as long as possible. Few other words should be used. For example:

Nurse, we'll test the reflexes again.

Nurse, we'll test the reflexes.

Air is displaced as water is poured into the bottle.

The water in the bottle displaces the air.

Care should be taken that simplifying does not change the meaning, particularly when meaning is conveyed by the intonation of words.

Often, the aim of schools programmes is to introduce new vocabulary and to familiarize pupils with complex terminology. When subtitling schools programmes, introduce complex vocabulary in very simple sentences and keep it on screen for as long as possible.

20.2 Preferred timing

In general, subtitles for children should follow the speed of speech. However, there may be occasions when matching the speed of speech will lead to subtitle rate that is not appropriate for the age group. The producer/ assistant producer should seek advice on the appropriate subtitle timing for a programme.

20.3 Avoid variable timing

There will be occasions when you will feel the need to go faster or slower than the standard timings - the same guidelines apply here as with adult timings (see Timing). You should however avoid inconsistent timings e.g. a two-line subtitle of 6 seconds immediately followed by a two-line subtitle of 8 seconds, assuming equivalent scores for visual context and complexity of subject matter.

20.4 Allow more time for visuals

More time should be given when there are visuals that are important for following the plot, or when there is particularly difficult language.

20.5 Syntax and Vocabulary

Do not simplify sentences, unless the sentence construction is very difficult or sloppy.

Avoid splitting sentences across subtitles. Unless this is unavoidable, keep to complete clauses.

Vocabulary should not be simplified.

There should be no extra spaces inserted before punctuation.

21 Live subtitling (BBC-ASP, OFCOM-IQLS, OFCOM-GSS)

21.1 General

The subtitler should have a direct pre-broadcast-encoding feed from the broadcaster, so they can hear the output a few seconds earlier than if relying on the broadcast service.

Maintain a regular subtitle output with no long gaps (unless it is obvious from the picture that there is no commentary) even if this means subtitling the picture or providing background information rather than subtitling the commentary.

Aim for continuity in subtitles by following through a train of thought where possible, rather than sampling the commentary at intervals.

Do not subtitle over existing video captions where avoidable (in news, this is often unavoidable, in which case a speaker's name can be included in the subtitle if available).

21.2 Preparation

Find out specialist vocabulary, and specific editorial guidelines for the genre (e.g. sport). Familiarise yourself with Prepared segments that have been subtitled and their place in the running order, but be prepared for the order to change.

When available to the subtitler, pre-recorded segments should be subtitled prior to broadcast (not live) and cued out at the appropriate moment.

When cueing prepared texts for scripted parts of the programme:

- Try to cue the texts of pre-recorded segments so that they closely match the spoken words in terms of start time.
- Do not cue texts out rapidly to catch up if you get left behind - skip some

and continue from the correct place.

- Try to include speakers' names if available where in-vision captions have been obliterated.

21.3 Editing

Subtitles should use upper and lower case as appropriate.

Standard spelling and punctuation should be used at all times, even on the fastest programmes.

Produce complete sentences even for short comments because this makes the result look less staccato and hurried.

Strong or inappropriate language must not appear on screen in error.

For news programmes, current affairs programmes and most other genres, subtitles should be verbatim, up to a subtitling speed of around 160-180wpm. Above that speed, some editing would be expected.

For some genres, such as in-play sporting action, the subtitling may be edited more heavily so as to convey vital commentary information while allowing better access to the visuals. (BBC-SPG)

21.4 Corrections

Any serious or misleading errors in real-time subtitling should be corrected clearly and promptly. The correction should be preceded by two dashes:

The minster’s shrew is unchanged -- view.

However be aware that too many on-air corrections, or corrections that are not sufficiently prompt, can actually make the subtitles harder for a viewer to follow.

Ultimately the subtitler may have to decide whether to make a correction or omit some speech in order to catch up. Sometimes this can be done without detracting from the integrity of the subtitling, but this is not always the case. Do not correct minor errors where the reader can reasonably be expected to deduce the intended meaning (e.g. typos and misspellings).

If necessary, an apology should be made at the end of the programme. If possible, repeat the subtitle with the error corrected.

21.5 Formatting

Live subtitles should appear word by word, from left to right, to allow maximum reading time. Live subtitles are justified left (not centred).

> Live subtitles should be placed in an appropriately sized `tt:region` with a preset `tts:origin` x coordinate (for left to right text; for right to left text ensure the right edge is preset).
>
> A style with `tts:textAlign` set to `"start"` (always works) or `"left"` for left to right text only or `"right"` for right to left text only should be used.
>
> `ebutts:multiRowAlign` should be avoided (i.e. left unset, or set to `"auto"`) since it can result in lines being moved horizontally whenever a new word appears.

Two-lines of scrolling text should be used.

For live subtitling, use a reduced set of formatting techniques. Focus on colour and vertical positioning.

- A change of speaker should always be indicated by a change of colour.
- Scrolling subtitles, while usually appearing at the bottom of the screen, should be raised as appropriate in order to avoid any vital action, visual information, name labels, etc.

  > Subtitle vertical position can be set by referencing a `tt:region` with appropriate `tts:origin`, `tts:extent` and `tts:displayAlign` attributes.

An alternative strategy is to insert `<tt:br/>` elements as necessary; for example if `tts:displayAlign="after"` then every `<tt:br/>` element appended after a subtitle will raise that subtitle by the height of the line. Although using line breaks for positioning is discouraged for prepared subtitles (see Authoring font size), this technique saves time when live subtitling. Note that if the region height is exceeded by entering too many line breaks, lines can 'fall off' the top, and be clipped.

If a subtitle needs to be moved while it is visible and inserting `<tt:br/>` elements is not possible then the `<tt:p>` should be ended and a new `<tt:p>` begun that references a differently positioned region. That new `<tt:p>` can contain the same words and style references.

FILE FORMAT

22 Files

The format for prepared subtitles depends on the delivery route and platform. In general, subtitles for programmes scheduled for linear broadcast, including iPlayer-first, are delivered to Playout and to File Based Delivery as STL and EBU-TT Part 1 files. Online-only content not scheduled for linear broadcast is delivered as EBU-TT-D files, typically for uploading into a BBC content management system. There are some exceptions to this, so if in doubt ask your commissioning editor about the correct delivery route and files formats.

Platform	Format	Extension	Specification	Notes
Broadcast and online	EBU-STL	`.stl`	https://tech.ebu.ch/publications/tech3264	Required for linear broadcast legacy systems.
	EBU-TT	`.xml`	https://tech.ebu.ch/docs/tech/tech3350v1-0.pdf (to be replaced by v1.1)	With the STL embedded. See

Platform	Format	Extension	Specification	Notes
				below.
ONLINE	EBU-TT-D	`.ebuttd.xml`	https://tech.ebu.ch/publications/tech3380	

Note that the above standards support a larger set of characters than is allowed by the BBC. For linear playout, all characters for presentation must be in the set in Appendix 1.

23 STL file

23.1 File name

The file name must follow this pattern: [UID with slash removed].stl

For example:

UID	File name
DRIB511W/02	DRIB511W02.stl

23.2 General subtitle information (GSI) block

Subtitles must conform to the EBU specification TECH 3264-E. However, the BBC requires certain values in particular elements of the General Subtitle Information Block. See the table below.

GSI block data	Short	Value	Notes	Example
Code Page Number	CPN	"850"	Required	
Disk Format Code	DFC	"STL25.01"	Required	
Display Standard Code	DSC	"1"	Required	
Character Code Table	CCT	"00"	Required	
Language Code	LC	"09"	Required	
Original Programme Title	OPT	[string]	Required	Snow White

GSI block data	Short	Value	Notes	Example
Original Episode Title	OET	[A tape number]	Required if a tape number exists.	HDS147457
Translated Programme Title	TPT	[string]	Required if translated	
Translated Episode Title	TET	[string]	Optional	Series 1, Episode 1
Translator's Name	TN	[Up to 32 characters]	Optional	Jane Doe
Translator's Contact Details	TCD	[Up to 32 characters]	Optional	
Subtitle List Reference Code	SLR	[On-air UID]	**BROADCAST** Required for Prepared linear	ABC D123W/02
Creation Date	CD	[date in format YYMMDD]	Required	150125
Revision Date	RD	[date in format YYMMDD]	Required	150128
Revision Number	RN	[0 – 99]	Required	1
Total Number of TTI Blocks	TNB	[0 – 99999]	Required. Must accurately reflect the number of blocks in the file.	767
Total Number of Subtitles	TNS	[0 – 99999]	Required. Must accurately reflect the number of subtitles in the file.	767
Total Number of Subtitle Groups	TNG	"1"	Required. Fixed at 1.	1
Maximum Number of Displayable Characters in any text row	MNC	[0 – 99]	Required	37
Maximum Number of Displayable Rows	MNR	"11"	Required	
Time Code: Status	TCS	"1"	Required	
Time Code: Start-of-Programme	TCP	[time in format HHMMSSFF]	Required	10000000, 20000000

GSI block data	Short	Value	Notes	Example
Time Code: First in-cue	TCF	[time in format HHMMSSFF]	Required. The timecode of the first in-cue in the subtitle list.	
Total Number of Disks	TND	[Number of files]	Required. Almost always 1. For very long programmes where the subtitles must be split into multiple files, contact the commissioning editor.	1
Disk Sequence Number	DSN	[The file number of this file]	Required. Always 1 when there is one STL file in the sequence. For very long programmes where the subtitles must be split into multiple files, contact the commissioning editor.	1
Country of Origin	CO	[3-letter country code]	Required	GBR
Publisher	PUB	[Up to 32 characters]	Required	Company name
Editor's Name	EN	[Up to 32 characters]	Required	John Doe
Editor's Contact Details	ECD	[Up to 32 characters]	Optional	
Spare bytes	SB	[Empty]	Optional	
User-Defined Area	UDA	[Up to 576 characters]	Not used.	

23.3 Timecode

The Time Code Out (TCO) values in STL files are *inclusive* of the last frame; in other words the subtitle shall be visible on the frame indicated in the TCO value but not on subsequent frames. This differs from the end time expressions in EBU-TT and TTML, which are *exclusive*.

For example, in an STL file a subtitle with a TCO of `10:10:10:20` would map in an EBU-TT document to an `end` attribute value of `10:10:10:21`.

23.4 Subtitle zero

It is common practice to place metadata (programme ID, name etc.) in a subtitle at the beginning of the file. This first subtitle is typically known as 'subtitle zero' and is used for example to check that the correct subtitles have been loaded during pre-roll. A 'subtitle zero' is not intended to be broadcast, and this is achieved by setting the in-cue and out-cue times for this subtitle earlier than the first timecode value that occurs in the corresponding media (for example, setting subtitle zero to display between 00:00:00 and 00:00:01 when the programme starts at 10:00:00).

Subtitles that begin (TCI) at timecode `00:00:00:00` in documents that have a start of programme timecode (TCP) other than `00000000` SHALL end (TCO) no later than `00:00:00:01`, in other words they must have a duration no longer than 2 frames. They SHOULD have a duration of 1 frame.

Subtitle Zero is optional but common in legacy STL files. When an STL file is embedded in an EBU-TT document, the subtitle zero must be handled as detailed below:

File	Notes
EBU-TT v1.0	Subtitle zero MAY be included in the body of the document. If the subtitle zero is included in the embedded STL file and is included in the body of the EBU-TT document then they SHALL be identical. If the subtitle zero is included in the body of the EBU-TT document then it SHALL have an end attribute no later than `00:00:00:02`. If subtitle zero is not included in the embedded STL file then the EBU-TT file SHALL NOT contain a subtitle zero.
EBU-TT v1.1	Subtitle zero MAY be included in the body of the document. If a subtitle zero is included in the embedded STL file then its content SHALL be copied into `ebuttm:subtitleZero` element. If the subtitle zero is included in the embedded STL file and is included in the body of the EBU-TT document then they SHALL be identical. See ebuttm:documentMetadata. If the subtitle zero is included in the body of the EBU-TT document then it

File	Notes
	SHALL have an end attribute no later than `00:00:00:02`. If a subtitle zero is not included in the embedded STL file then the element `ebuttm:subtitleZero` SHALL NOT be included.

24 EBU-TT file

EBU-TT is the BBC's strategic file format for capturing subtitles and associated metadata. The BBC needs to continue to operate systems that use older formats such as Teletext: in cases where those legacy systems impose constraints, those constraints are incorporated into these guidelines. In the future, as legacy systems are phased out, the constrained requirements will be relaxed. Where we have control over the distribution and presentation chain those constraints are already removed; for example the requirements for EBU-TT-D delivery for online distribution allow greater flexibility in how to achieve the presentation requirements.

Teletext and STL constraints

Teletext is still used on some platforms to carry and/or display subtitles; the BBC expects EBU-TT files that preserve some aspects of this technology (or that have been converted from STL files). For example, Teletext uses a fixed grid of 40x24 cells that (for BBC use) must be preserved in EBU-TT files authored for linear broadcast (`ttp:cellResolution="40 24"`), even though EBU-TT does not require use of this specific grid. Subtitles authored for non-linear platforms are already free of these constraints. For example, EBU-TT-D files for online distribution can use the default cell resolution of 32x15 (see EBU-TT-D cell resolution).

When present, the STL file(s) must be embedded in an EBU-TT document. See below for further details.

Embedded STL files may be omitted if the subtitles are created live and then captured.

Avoid pixel units

Although EBU-TT allows pixel length units, the BBC requires that only percent or cell units are used. Pixel length values are sometimes misunderstood in the context of video resolutions. It is less confusing to avoid use of pixel units when authoring resolution-independent content. It is also simpler to transform EBU-TT Part 1 into EBU-TT-D if pixel units are not used, since no calculations need to be made relating pixel values to the `tts:extent` attribute of the `tt:tt` element.

EBU-TT Part 1 Versions

The BBC currently uses version 1.0 of EBU-TT, but intends to move to version 1.1. Significant changes were made to the metadata structure between the versions, with some elements moved from the BBC to the EBU namespace. Both versions are given here but only v1.0 specifications are stable. Delivery of v1.1 files must be approved in advance and the specification confirmed.

24.1 File name

The file name has this format:

```
[ebuttm:documentIdentifier]-preRecorded.xml
```

See the rules for constructing ebuttm:documentIdentifier below.

24.2 Character encoding

The file must be UTF-8 encoded.

The file must not begin with a byte order mark (BOM).

24.3 tt:tt attributes

The following table lists standard EBU-TT elements and their required values.

Attribute	Value	Notes	Example

Attribute	Value	Notes	Example
xml:space		Optional	preserve
`ttp:timeBase`	"smpte"	Required	
`ttp:framerate`		Required. Must match the frame rate of the associated video.	25
`ttp:frameRateMultiplier`		Required if `ttp:timeBase="smpte"` .	1 1
`ttp:markerMode`	"discontinuous"	Required.	
`ttp:dropMode`		Required when `ttp:timebase="smpte"` .	nonDrop
`ttp:cellResolution`	"40 24"	Required. This value is used to preserve Teletext single line height, where the assumption is that a Teletext font is readable with a line height equal to 100% of the font size, for both single and double height lines i.e. `tts:fontSize="1c 1c"` or `tts:fontSize="1c 2c"` and `tts:lineHeight="100%"` . It is also possible to define or configure in Teletext-based implementations that `tts:lineHeight="normal"` shall be interpreted as 100% in the context of a document originally authored to Teletext constraints. *This approach is likely to change when we are no longer authoring to Teletext constraints.*	
`xml:lang`		Required	en-GB

24.4 ebuttm:documentMetadata elements (v1.0)

The below table lists the required document metadata values for BBC subtitle documents based on EBU-TT Part 1 v1.0, which is the current actively used format.

Element	Value	Notes	Example
`ebuttm:documentEbuttVersion`	`"v1.0"`	Required	

Element	Value	Notes	Example
ebuttm:documentIdentifier	See below.	Required if not live	ABCD123W02-1
ebuttm:documentOriginatingSystem	[Software and version]	Required	TTProducer 1.7.0.0
ebuttm:documentCopyright	"BBC"	Required	
ebuttm:documentReadingSpeed	[Calculate per document]	Required	176
ebuttm:documentTargetAspectRatio	"4:3" ("16:9" allowed for online use only)	Required	
ebuttm:documentIntendedTargetFormat	Required if also targeting broadcast applications.	Required	WSTTeletextSubtitles
ebuttm:documentOriginalProgrammeTitle	[string]	Required	Snow White
ebuttm:documentOriginalEpisodeTitle	[string]	Required	Series 1, Episode 1
ebuttm:documentSubtitleListReferenceCode	[UID]	Required	ABC D123W/02
ebuttm:documentCreationDate	[date in format YYYY-MM-DD]	Required	2015-01-20
ebuttm:documentRevisionDate	[date in format YYYY-MM-DD]	Required if a revision	2015-01-20
ebuttm:documentRevisionNumber	[integer]	Required if a revision	1
ebuttm:documentTotalNumberOfSubtitles	[Calculated per document]	Required	767
ebuttm:documentMaximumNumberOfDisplayableCharacterInAnyRow	[integer]		37
ebuttm:documentStartOfProgramme	"10:00:00:00" | "20:00:00:00"	Required. Value must match the timecode	

Element	Value	Notes	Example
		of the start of the programme content.	
`ebuttm:documentCountryOfOrigin`	`"GBR"`	Required	
`ebuttm:documentPublisher`	[string]	Required	`Company name`
`ebuttm:documentEditorsName`	[string]	Required	`John Doe`

24.5 Document identifier

The document identifier is obtained by reading the string from the embedded STL's GSI "Reference Code" field (On Air UID) and then deleting any spaces and `"/"` character. This string is appended with a hyphen and the value of the Revision Number field in the STL's GSI block.

24.6 ebuttm:documentMetadata elements (EBU-TT Part M)

BBC specifications based on version 1.2 of EBU-TT Part 1 and on the EBU-TT Part M Metadata specification are still in development. Information in this section is therefore subject to change.

The table below lists the required document metadata values for BBC subtitle documents based on the EBU-TT Part M Metadata specification, which is not yet in active use by the BBC.

Element	Value	Notes	Example
`ebuttm:conformsToStandard`	`"urn:ebu:tt:exchange:2017-05"`	Required	
`ebuttm:documentIdentifier`	[OnAir UID]"-"[subtitle file version]	Required if not live	`ABCD123W02-1`
`ebuttm:documentOriginatingSystem`	[Software and version]	Required	`TTProducer 1.7.0.0`

Element	Value	Notes	Example
ebuttm:documentCopyright	"BBC"	Deprecated. Instead, use a ttm:copyright element in the <tt:head>.	
ebuttm:documentReadingSpeed	[Calculated per document]	Required	176
ebuttm:documentTargetAspectRatio	"4:3" ("16:9" allowed for online use only)	Required	
ebuttm:documentTargetActiveFormatDescriptor	[one of the AFD codes specified in SMPTE ST 2016-1:2009 Table 1]		
ebuttm:documentIntendedTargetBarData	[Bar Data from SMPTE ST 2016-1:2009 Table 3. Note additional attributes may be required. See the EBU-TT specification]	Optional	
ebuttm:documentIntendedTargetFormat	"Enhanced Teletext Level 1" | "DVBBitmapSubtitles" | "EBU-TT-D"	All three are required, each in its own ebuttm:documentIntendedT	

Element	Value	Notes	Example
		`argetFormat` element. The URI of the classification scheme should be specified in the `link` attribute with the term ID. For example, `https://www.ebu.ch/metadata/cs/EBU-TTSubtitleTargetFormatCodeCS.xml#1.11` for EBU-TT-D.	
`ebuttm:documentCreationMode`	`"live" | "prepared"`	Required	
`ebuttm:documentContentType`	`"hardOfHearingSubtitles"`	Required	
`ebuttm:sourceMediaIdentifier`	[OnAir UID][version #]-[sub file version]	Required	`ABCD123W02-1`
`ebuttm:relatedMediaIdentifier`	[string]		
`ebuttm:relatedObjectIdentifier`		Optional	
`ebuttm:appliedProcessing`		Optional	

Element	Value	Notes	Example
`ebuttm:relatedMediaDuration`		Optional	
`ebuttm:documentBeginDate`	[Date in YYYY-MM-DD format]	Required for live captured subtitles. The corresponding date of creation of the earliest begin time expression (i.e. the begin time expression that is the first coordinate in the document time line).	
`ebuttm:localTimeOffset`	[Timezone in ISO 8601 when `ttp:timebase="clock"` AND `ttp:clockmode="local"`]	Required for live captured subtitles.	`Z, +01:00`
`ebuttm:referenceClockIdentifier`		Optional. Allows the reference clock source to be identified. Permitted only when `ttp:time`	

Element	Value	Notes	Example
		`Base="clock"` AND `ttp:clockMode="local"` OR when `ttp:timeBase="smpte"`.	
`ebuttm:broadcastServiceIdentifier`	[The value of `<id type="service_id">` for the service]	Optional. The list of all services is at https://api.live.bbc.co.uk/pips/api/v1/service/ (API access required). You may need to request the service identifier list prior to delivery.	`BBC1, CBeebies`
`ebuttm:documentTransitionStyle`	[Empty element. Only the attributes `inUnit` or `outUnit` are specified].	Optional	
The following elements support the information that is present in the GSI block of the STL file. If more than one STL source file is used to generate an EBU-TT document, the GSI metadata cannot be mapped into ebuttm:documentMetadata unless the value of a GSI field			

Element	Value	Notes	Example
is the same across all STL documents.			
`ebuttm:documentOriginalProgrammeTitle`	[Original programme title]	Required	`Snow White`
`ebuttm:documentOriginalEpisodeTitle`		Use `bbctt:otherId` (see below)	
`ebuttm:documentTranslatedProgrammeTitle`		Required if translated	
`ebuttm:documentTranslatedEpisodeTitle`		Optional	`Series 1, Episode 1`
`ebuttm:documentTranslatorsName`	[Up to 32 characters]	Optional	`Jane Doe`
`ebuttm:documentTranslatorsContactDetails`	[Up to 32 characters]	Optional	
`ebuttm:documentSubtitleListReferenceCode`	[On-air UID]	**BROADCAST** Required for Prepared linear	`ABC D123W/02`
`ebuttm:documentCreationDate`	[Date in format YYYY-MM-DD]	Required	`2012-06-30`
`ebuttm:documentRevisionDate`	[Date in format YYYY-MM-DD]	Required if a revision	`2015-01-28`
`ebuttm:documentRevisionNumber`	[0 – 99]	Required if a revision	`1`
`ebuttm:documentTotalNumberOfSubtitles`	[Non-negative integer]	Required	`767`
`ebuttm:documentMaximumNumberOfDisplayableCharacterInAnyRow`	[0 – 37]	Required	`58`
`ebuttm:documentStartOfProgramme`	[HH:MM:SS:FF]	Required	`10:00:00:00`, `20:00:00:00`

Element	Value	Notes	Example
`ebuttm:documentCountryOfOrigin`	[3-letter country code]	Required	`GBR`
`ebuttm:documentPublisher`	[Up to 32 characters]	Required	`Company name`
`ebuttm:documentEditorsName`	[Up to 32 characters]	Required	`John Doe`
`ebuttm:documentEditorsContactDetails`	[Up to 32 characters]	Optional	
`ebuttm:documentUserDefinedArea`	[Up to 576 characters]	Not used	
`ebuttm:stlCreationDate`	[Date in format YYYY-MM-DD]	Optional. If the STL file is embedded using `ebuttm:binaryData`, do not use this element. Instead, use the `creationDate` attribute of `ebuttm:binaryDataElement`.	
`ebuttm:stlRevisionDate`	[Date in format YYYY-MM-DD]	Optional. If the STL file is embedded, use the `revisionDate` attribute of `ebuttm:b`	

Element	Value	Notes	Example
		inaryDataElement.	
ebuttm:stlRevisionNumber	[Integer]	Optional. If the STL file is embedded, use the revisionNumber attribute of ebuttm:binaryDataElement.	
ebuttm:subtitleZero	If the subtitle zero is present, copy the content of subtitle zero from the STL	Optional	

24.7 Extended BBC metadata (v1.0)

This section lists the required extended BBC metadata values for BBC subtitle documents based on EBU-TT Part 1 v1.0, which is the current actively used format.

In addition to the standard EBU-TT elements listed above, the BBC requires the below metadata elements within a <bbctt:metadata> element. The <bbctt:metadata> element is the last child of <tt:metadata>. See Appendix 2 for a sample XML and Appendix 3 for the XSD.

In the following tables, prefixes are used as shortcuts for the following namespaces:

Prefix	Namespace	Notes
bbctt:	http://www.bbc.co.uk/ns/bbctt	The BBC TTML metadata namespace

bbctt:schemaVersion	
Cardinality	1..1
Parent	`bbctt:metadata`
Description	The BBC metadata scheme used. Currently v1.0.
Value	`"v1.0"`
Example	`<bbctt:schemaVersion>v1.0</bbctt:schemaVersion>`

bbctt:timedTextType	
Cardinality	1..1
Parent	`bbctt:metadata`
Description	Indicates whether subtitles were live or prepared. If live subtitles are modified following broadcast, this value must be changed to preRecorded.
Value	`"preRecorded" | "audioDescription" | "recordedLive" | "editedLive"`
Example	`<bbctt:timedTextType>preRecorded</bbctt:timedTextType>`

bbctt:timecodeType	
Cardinality	1..1
Parent	`bbctt:metadata`
Description	Indicates whether timecode uses "programme" time for pre-recorded subtitles or "timeOfDay" UTC time for live authored subtitles.
Value	`"programme" | "timeOfDay"`
Example	`<bbctt:timecodeType>programme</bbctt:timecodeType>`

bbctt:programmeId	
Cardinality	0..1
Parent	`bbctt:metadata`
Description	Required if not live.
Value	[On-air UID]
Example	`<bbctt:programmeId>DRIB511W/02</bbctt:programmeId>`

bbctt:otherId type="tapeNumber"	
Cardinality	0..1. Required if not live.
Parent	`bbctt:metadata`

bbctt:otherId type="tapeNumber"	
Description	Use tape number for programmes that have a material reference. Use Mat ID for programmes delivered as file.
Value	[String]
Example	`<bbctt:otherId 147457</bbctt:otherId>`

bbctt:houseStyle owner=""	
Cardinality	0..*
Parent	`bbctt:metadata`
Description	Required if live.
Value	
Example	

bbctt:recordedLiveService	
Cardinality	0..*.. Required for a live recording if intended for broadcast. **BROADCAST**
Parent	`bbctt:metadata`
Description	Required for subtitles created live only.
Value	The value of `<id type="service_id">` for the service. The list of all services is at https://api.live.bbc.co.uk/pips/api/v1/service/. You may need to apply for API access or request the service identifier prior to delivery.
Example	

bbctt:div	
Cardinality	0..*
Parent	`bbctt:metadata`
Description	Generic container of type "shotChange" or "Script"
Value	`<systemInfo>`, `<chapter>`, `<item>` or `<event>` elements

Example	`<bbctt:div type="shotChange">`
	`<bbctt:systemInfo>Quantum Video Indexer`
	`v5.0</bbctt:systemInfo> <bbctt:event begin="09:59:30:00"`
	`id="sc1" /> </bbctt:div>`

bbctt:systemInfo	
Cardinality	1..1
Parent	`bbctt:div`

bbctt:systemInfo	
Description	The system that produced the sibling elements.
Value	Single instance of `bbctt:systemInfo` and multiple instances of `bbctt:event`
Example	`<bbctt:systemInfo>Quantum Video Indexer v5.0</bbctt:systemInfo>`

bbctt:event			
Cardinality	0..*		
Parent	`bbctt:div`		
Description	A single event, e.g. a shot change in a `bbctt:div` of `type="shotChange"`		
Attributes	**Attribute**	**Required?**	**Type**
	`begin`	Yes	`ebuttdt:timingType`
	`end`	AD fades only	`ebuttdt:timingType`
	`endlevel`	AD fades only	Integer
	`xml:id`	No	NCName
	`pan`	AD fades only	Integer
	`type`	No	NCName
Value	This is an empty element. Information is represented as element attributes.		
Example	`<bbctt:event begin="01:23:45:25" id="sc1"/>`		

bbctt:chapter id=""	
Cardinality	0..*
Parent	`bbctt:div`
Description	Used to divide content into semantic chapters.
Value	One or more `bbctt:item` elements
Example	

bbctt:item			
Cardinality	In `bbctt:div` : 0..* | In `bbctt:chapter` : 1..*		
Parent	`bbctt:div | bbctt:chapter`		
Description	Generic container for the programme script elements.		
Attributes	**Attribute**	**Required?**	**Type**
	`xml:id`	Yes	string
	`begin`	No	`ebuttdt:timingType`
	`end`	No	`ebuttdt:timingType`

bbctt:item	
Value	`<bbctt:p>`, `<bbctt:itemid>`, `<bbctt:title>` or `<bbctt:associatedFile>` elements.
Example	`<bbctt:item xml:id="it1"> <bbctt:p> <bbctt:span ttm:role="x-direction">Snow White</bbctt:span> </bbctt:p> <bbctt:p> <bbctt:span ttm:role="x-direction">(CONT’D) </bbctt:span> </bbctt:p> </bbctt:item>`

bbctt:itemId	
Cardinality	0..*
Parent	`bbctt:item`
Description	Used to link an item with an external system
Value	`bbctt:itemId`
Example	

bbctt:title	
Cardinality	0..1
Parent	`bbctt:item`
Description	Used to link an item with an external system
Value	[String]
Example	

bbctt:associatedFile	
Cardinality	0..1
Parent	`bbctt:item`
Description	Used to link an item with an external system
Value	`bbctt:associatedFile`
Example	

bbctt:p	
Cardinality	1..*
Parent	`bbctt:item`
Description	A single script element (paragraph)
Value	Single `bbctt:span` element
Example	`<bbctt:p><bbctt:span ttm:role="x-direction">Snow White</bbctt:span></bbctt:p>`

bbctt:span	
Cardinality	1..1
Parent	`bbctt:p`
Description	A single line of script
Value	[Dialogue or direction text]
Example	`<bbctt:span ttm:role="dialog" ttm:agent="sp9">Snow white, wake up!</bbctt:span>`

24.8 Extended BBC metadata (EBU-TT Part M)

BBC specifications for version 1.2 of EBU-TT Part 1 and EBU-TT Part M are still in development and are not yet in active use. Information in this section is therefore subject to change. This section lists the required extended BBC metadata values for BBC subtitle documents based on EBU-TT Part M.

Some metadata that the BBC requires in version 1.0 of EBU-TT Part 1 were incorporated into version 1.1 and then transferred into EBU-TT Part M, which is incorporated by reference into EBU-TT Part 1 v1.2, meaning that BBC-specific elements (in the `bbctt` namespace) can be replaced by elements in the standard EBU-TT namespace (`ebuttm`). The following table summarises the changes:

v1.0	**Value**	**EBU-TT Part 1 v1.1 or EBU-TT Part M**	**Value**
`bbctt:timedTextType`	"preRecorded"	`ebuttm:documentCreationMode`	"prepared"
	"audioDescription"	`ebuttm:documentContentType`	"audioDescriptionScript"
	"recordedLive"	`ebuttm:documentCreationMode`	"live"
	"editedLive"	`ebuttm:documentCreationMode`	"prepared"
`bbctt:timecodeType`	"programme"	`ttp:timeBase`	"smpte"
	"timeOfDay"	**Replaced by BOTH attributes below:**	
		`ttp:timeBase`	"clock"
		`ttp:clockMode`	"utc"
`bbctt:programmeId`		`ebuttm:sourceMediaIdentifier`	

v1.0	Value	EBU-TT Part 1 v1.1 or EBU-TT Part M	Value
`bbctt:otherId`		`ebuttm:relatedObjectIdentifier`	
`bbctt:recordedLiveService`		`ebuttm:broadcastServiceIdentifier`	

These are the BBC metadata required for EBU-TT v1.1 or later.

bbctt:schemaVersion	
Cardinality	1..1
Parent	`bbctt:metadata`
Description	The BBC metadata scheme used. Currently v1.0.
Value	[TBC for v1.1]
Example	`<bbctt:schemaVersion>v1.0</bbctt:schemaVersion>`

bbctt:timecodeType	
Cardinality	1..1
Parent	`bbctt:metadata`
Description	Indicates whether timecode uses programme (pre-recorded) or UTC time (live)
Value	`"programme" | "timeOfDay"`
Example	`<bbctt:timecodeType>programme</bbctt:timecodeType>`

bbctt:div	
Cardinality	0..*
Parent	`bbctt:metadata`
Description	Generic container of type "shotChange" or "Script"
Value	`<systemInfo>`, `<chapter>`, `<item>` or `<event>` elements

Example	`<bbctt:div type="shotChange">`
	`<bbctt:systemInfo>Quantum Video Indexer v5.0</bbctt:systemInfo>`
	`<bbctt:event begin="09:59:30:00" id="sc1" />`
	`</bbctt:div>`

bbctt:systemInfo

bbctt:systemInfo	
Cardinality	1..1
Parent	bbctt:div
Description	The system that produced the sibling elements.
Value	[Single instance of bbctt:systemInfo and multiple instances of bbctt:event
Example	`<bbctt:systemInfo>Quantum Video Indexer v5.0</bbctt:systemInfo>`

<table>
<tr><th colspan="4">bbctt:event</th></tr>
<tr><td><b>Cardinality</b></td><td colspan="3">0..*</td></tr>
<tr><td><b>Parent</b></td><td colspan="3">bbctt:div</td></tr>
<tr><td><b>Description</b></td><td colspan="3">A single event, e.g. a shot change in a bbctt:div of type "shotChange"</td></tr>
<tr><td rowspan="7"><b>Attributes</b></td><td><b>Attribute</b></td><td><b>Required?</b></td><td><b>Type</b></td></tr>
<tr><td>begin</td><td>Yes</td><td>ebuttdt:timingType</td></tr>
<tr><td>end</td><td>AD fades only</td><td>ebuttdt:timingType</td></tr>
<tr><td>endlevel</td><td>AD fades only</td><td>Integer</td></tr>
<tr><td>xml:id</td><td>No</td><td>NCName</td></tr>
<tr><td>pan</td><td>AD fades only</td><td>Integer</td></tr>
<tr><td>type</td><td>No</td><td>NCName</td></tr>
<tr><td><b>Value</b></td><td colspan="3">This is an empty element. Information is represented as element attributes</td></tr>
<tr><td><b>Example</b></td><td colspan="3">&lt;bbctt:event begin="01:23:45:25" id="sc1"/&gt;</td></tr>
</table>

bbctt:chapter id=""	
Cardinality	0..*
Parent	bbctt:div
Description	Used to divide content into semantic chapters.
Value	One or more bbctt:item elements.
Example	

<table>
<tr><th colspan="4">bbctt:item</th></tr>
<tr><td><b>Cardinality</b></td><td colspan="3">In bbctt:div : 0..* | In bbctt:chapter : 1..*</td></tr>
<tr><td><b>Parent</b></td><td colspan="3">bbctt:div | bbctt:chapter</td></tr>
<tr><td><b>Description</b></td><td colspan="3">Generic container for the programme script elements.</td></tr>
<tr><td rowspan="2"><b>Attributes</b></td><td><b>Attribute</b></td><td><b>Required?</b></td><td><b>Type</b></td></tr>
<tr><td>xml:id</td><td>Yes</td><td>string</td></tr>
</table>

bbctt:item			
	begin	No	ebuttdt:timingType
	end	No	ebuttdt:timingType
Value	<bbctt:p>, <bbctt:itemid>, <bbctt:title>, <bbctt:associatedFile>		

Example	`<bbctt:item xml:id="it1">`
	`<bbctt:p><bbctt:span ttm:role="x-direction">Snow White</bbctt:s`
	`<bbctt:p><bbctt:span ttm:role="x-direction">(CONT’D)</bbctt:spa`
	`</bbctt:item>`		

bbctt:itemId	
Cardinality	0..*
Parent	bbctt:item
Description	Used to link an item with an external system
Value	bbctt:itemId
Example	

bbctt:title	
Cardinality	0..1
Parent	bbctt:item
Description	Used to link an item with an external system
Value	[String]
Example	

bbctt:associatedFile	
Cardinality	0..1
Parent	bbctt:item
Description	Used to link an item with an external system
Value	bbctt:associatedFile
Example	

bbctt:p	
Cardinality	1..*
Parent	bbctt:item
Description	A single script element (paragraph)

bbctt:p	
Value	[Single `bbctt:span` element]

Example	`<bbctt:p>`
	`    <bbctt:span ttm:role="x-direction">Snow White</bbctt:span>`
	`</bbctt:p>`

bbctt:span	
Cardinality	1..1
Parent	bbctt:p
Description	A single line of script
Value	[Dialogue or direction text]

Example	`<bbctt:span`
	`    ttm:role="dialog"`
	`    ttm:agent="sp9">Snow white, wake up!</bbctt:span>`

24.9 Embedded STL

The STL file(s), if present, must be embedded within the EBU-TT file, within the element `ebuttm:binaryData`:

ebuttm:binaryData	
Cardinality	0..*
Parent	`tt:metadata`
Description	Transitional requirement
Value	[The complete STL file, BASE64 encoded. Type: EBU Tech 3264]

Example	`<ebuttm:binaryData`
	`    textEncoding="BASE64" binaryDataType="EBU Tech 3264"`
	`    fileName="DRIB511W02.STL">ODUwU1RMMjUuMDExMD….</ebuttm:bina`

25 EBU-TT-D file

ONLINE The file must conform to both EBU-TT-D 1.0.1 and IMSC 1.0.1 Text Profile standards. Subtitles must be relative to a programme begin time of 00:00:00.000. The timebase must be set to 'media'.

25.1 Conformance with IMSC 1.0.1 Text Profile

To allow the file to be played on as many devices as possible, the EBU-TT-D must also conform to the IMSC 1.0.1 Text Profile, a closely related profile of TTML. In general, a valid EBU-TT-D document that conforms to these Guidelines will also conform to IMSC 1.0.1, provided that:

- It uses UTF-8 encoding
- No more than 4 regions are active *at the same time* (any number of regions can be defined in the document, but no more than four can be used simultaneously).
- The metadata element `ebuttm:conformsToStandard` is included with the value that corresponds to IMSC 1.0.1 (as well as the value that corresponds to EBU-TT-D 1.0.1):

```
<ebuttm:conformsToStandard>http://www.w3.org/ns/ttml/profile/ims
<ebuttm:conformsToStandard>urn:ebu:tt:distribution:2018-04</ebut
```

IMSC 1.0.1 also imposes complexity contraints, however these are not likely to be exceeded if you follow these guidelines.

25.2 File name

ONLINE For scheduled programmes (with an On Air UID), the file must be named [UID with slash removed].ebuttd.xml. Contact the commissioning editor for guidance on file names for for non-scheduled content (where no UID exists).

Note that embedded STL files should not be included within EBU-TT-D documents.

25.3 Character encoding

The file must be UTF-8 encoded.

The file must not begin with a byte order mark (BOM).

25.4 Mapping Teletext positions to percentage positions

This section applies when creating subtitles for landscape 16:9 aspect ratio videos.

There is no standard specification stating where the Teletext rendering area is located over video. Teletext was created when televisions had a 4:3 aspect ratio, so it is reasonable to assume that the Teletext rendering area was intended also to be 4:3. However most implementations avoid the edges of the screen, because televisions typically overscanned, which meant that the edges were not visible to the viewer.

As a rough basis then, positioning a 4:3 area in the central 90% vertically is a good starting point to keep the subtitles within the "safe area". Such an area within a 16:9 aspect ratio video would have a horizontal width of 67.5% of that root container region.

However, whereas Teletext was designed to be displayed using a monospaced font, modern systems typically use a proportionally spaced font. For double height text, as was traditionally used for subtitles, the Teletext approach was simply to create glyphs twice as tall, without modifying the width. Adopting the same approach today with proportionally spaced fonts is likely to result in text that is unpleasant to look at and hard to read: it would have to be a highly condensed variant of the font.

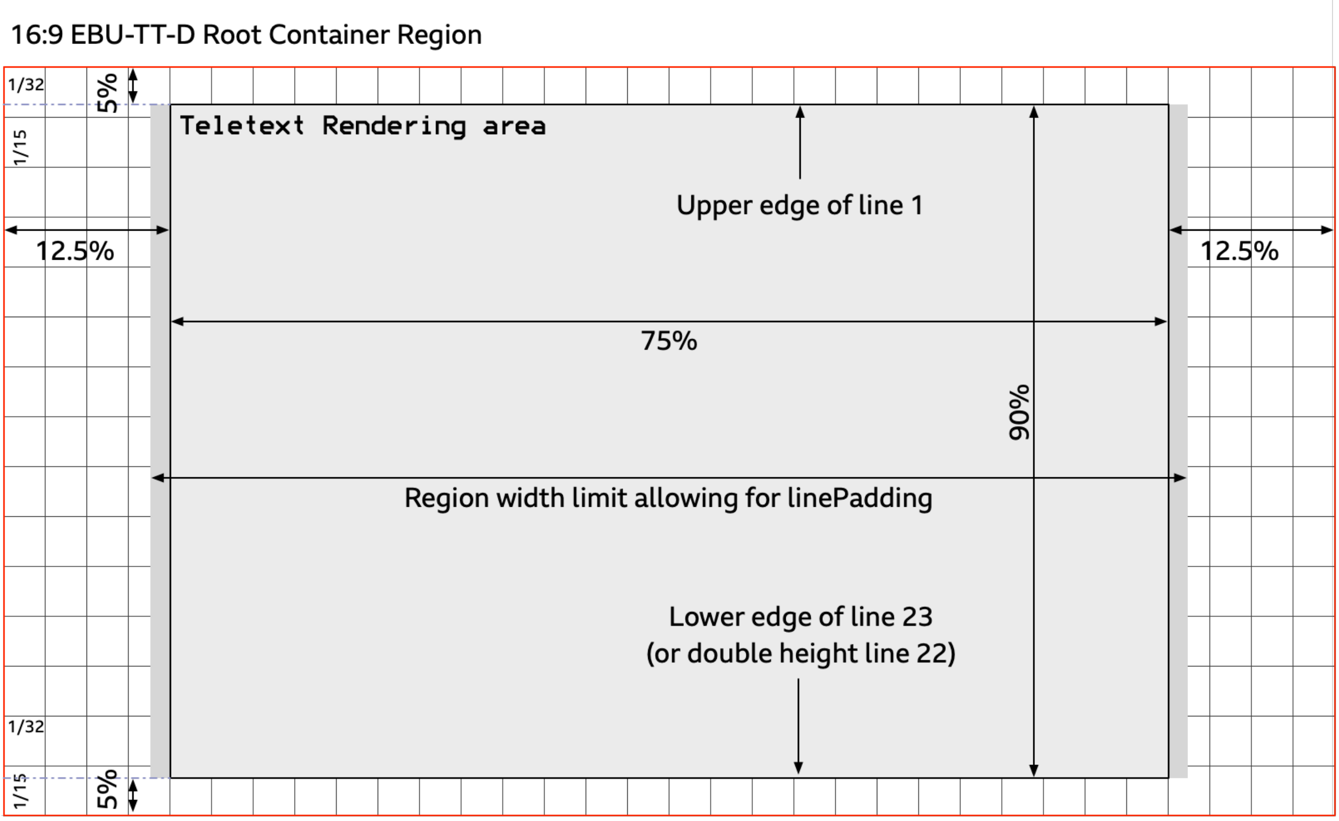
Instead, we need to find a balance between font size and line width that presents the text at a readable size while minimising the chance of unwanted line breaks in case the rendered text does not fit within the allocated space. Therefore the width available is extended to 75% of the 16:9 video area, with any additional space needed to accommodate line padding added on either side.

25.4.1 Teletext grid positioning requirements

Teletext specifies lines in the range 0-23, however no subtitles may be placed on line 0. Double height lines in Teletext occupy the line on which they are specified and the following line. Therefore there are 23 addressable single height lines and 22 addressable double height lines.

The top edge of text specified on line 1 must be positioned at 5% from the top of the root container region. The bottom edge of text specified on line 23 (single height) or line 23 (double height) must be positioned at 95% from the top of the root container region.

The following diagram illustrates this in visual form. Note that the underlying grid is virtual and that elements don't necessarily align to it. See `ttp:cellResolution`.


16:9 EBU-TT-D Root Container Region
1/32
5%
1/15
Teletext Rendering area
Upper edge of line 1
12.5%
12.5%
75%
90%
Region width limit allowing for linePadding
Lower edge of line 23
(or double height line 22)
1/32
1/15
5%

Teletext subtitle lines begin with at least 3 or 4 spacing control characters, which set the box colour and the text colour if not white. Lines do not need to *end* with control characters, but may do so if the text does not run to the end of the line. Therefore there are a maximum of 37 characters per line, occupying positions 3-39 inclusive (zero-indexed).

The left edge of a character at position 3 must be positioned no less than 12.5% from the left of the root container region. The right edge of a character at position 39 must be positioned no more than 87.5% from the left of the root container region.

25.4.2 Alignment of groups of lines

Since Teletext does not signal any authorial intent behind the positioning of text, implementations may need to make inferences about how to align groups of more than one adjacent line that are visible at the same time. The algorithms for making those inferences may include content-based preferences and analysis of sequences of subtitles as they change over time, for example.

Groups of lines may each be considered as being aligned horizontally to their centre position if they are within a character of that position. For example, if three adjacent lines have respective centre points at positions 20.5, 21, and 21.5, a heuristic may consider them all to be centered about position 21.

Lines that are aligned to a left or right position should have the same position for their first or last character respectively.

In some cases it may be that more than one possible alignment exists. For example, those same three adjacent lines could all have the same left position. If there is any other indication of authorial intent available, then that should be honoured where possible. For example, within a sequence of left-aligned subtitles, a single centered subtitle looks strange, and is unlikely to have been intended. Conversely, within a sequence of centered subtitles, a single left-aligned subtitle looks odd. When subtitles are cumulative, centered subtitles should not be preferred, because the change in position when words are added to a line makes the text difficult to read.

Such adjacent lines should be placed within the same `tt:p` element and separated by `tt:br` elements. The style applied to the `tt:p` element should include a `tts:textAlign` attribute set to the value corresponding to their alignment edge (or centre).

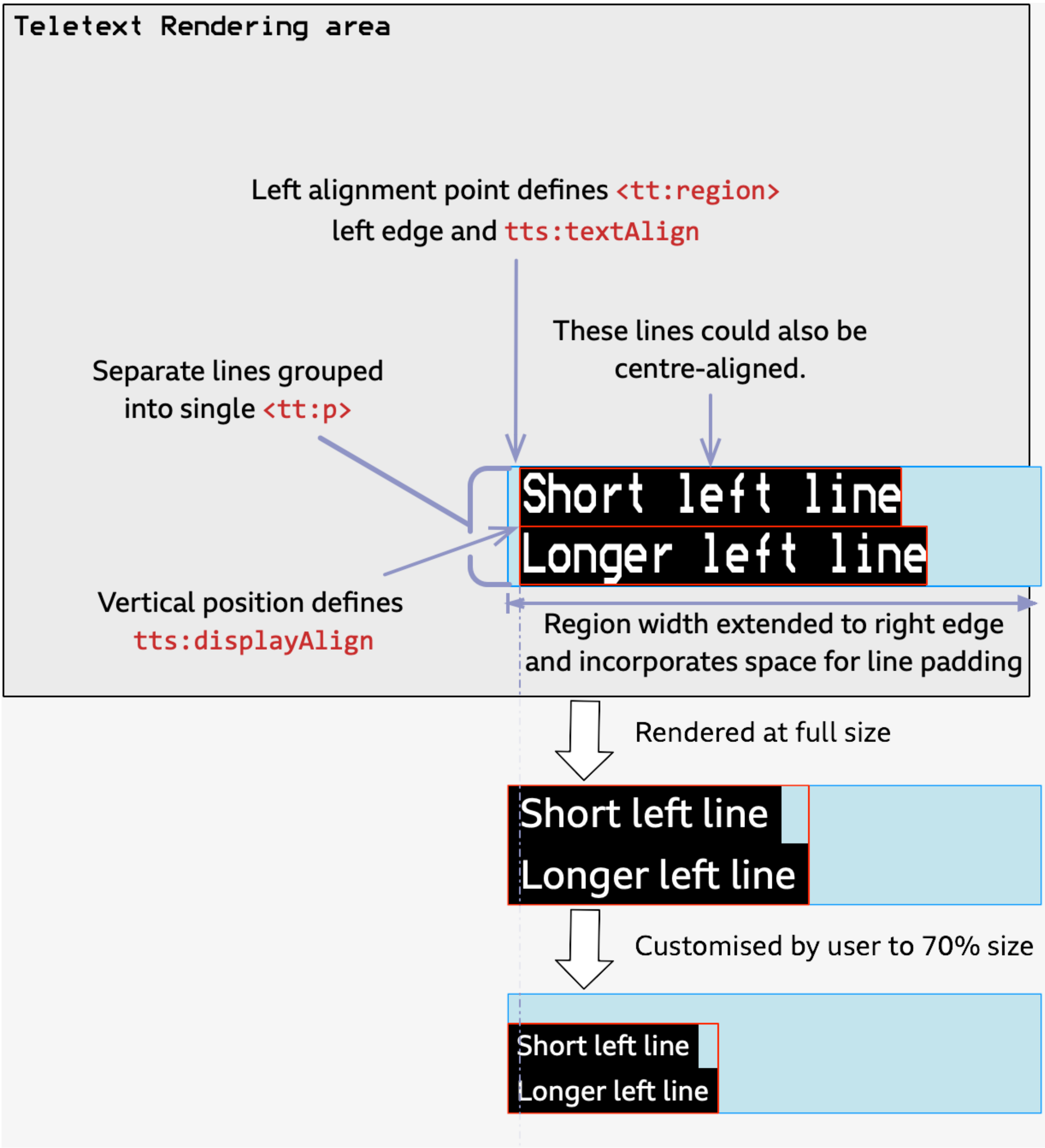
Those `tt:p` elements should be placed within regions positioned to apply the equivalent horizontal and vertical areas and whose `tts:displayAlign` attribute is set to an appropriate value depending on the position in the root container region. For example, regions at the top should be `"before"` edge aligned, and those at the bottom should be `"after"` edge aligned. Regions in the central vertical area may be `"center"` aligned.

Whereas Teletext is a monospaced system, typically the resulting subtitles will be presented using a proportionally spaced font. When the text is rendered, it may occupy more width than the original Teletext. To allow for this, where all the text in a `tt:region` has the same value of `tts:textAlign`, the left and right edges of the region should be extended such that the text is in the same position, up to the limits specified above. This reduces the chance of unexpected line breaks.

Why is this important?

Applying these rules should allow any text size customisation to retain appropriate relative positioning of each line, without introducing gaps between lines, for example.

Illustrative diagram:


Teletext Rendering area
Left alignment point defines <tt:region> left edge and tts:textAlign
Separate lines grouped into single <tt:p>
These lines could also be centre-aligned.
Short left line
Longer left line
Vertical position defines tts:displayAlign
Region width extended to right edge and incorporates space for line padding
Rendered at full size
Short left line
Longer left line
Customised by user to 70% size
Short left line
Longer left line

26 Timecode

BROADCAST Prepared subtitles for linear programmes must use the SMPTE timebase with a start of programme aligned to the source media. This is usually (but not always) 10:00:00:00. See the BBC's Technical requirements for delivery as AS-11 DPP files.

ONLINE Prepared subtitles for online exclusives must be relative to a programme begin time of 00:00:00.000 .

> EBU-TT (Part 1 v1.0) files captured from live created subtitles must set `bbctt:timecodeType` to `"timeOfDay"` . Time expressions must be in UTC. [EBU-TT 1.1] files should use `ttp:timeBase="clock"` and `ttp:clockMode="utc"` to indicate this information.
> For implementation details, see `ttp:timeBase` .

27 EBU-TT and EBU-TT-D Documents in detail

This section contains detailed instruction for developers of subtitle authoring tools that output EBU-TT or EBU-TT-D documents, and for processors of those files. It is structured around the key TTML elements and attributes: see the example document below and click on elements and attributes to go to their respective section.

This is intended to be a developer-friendly view of the specifications, but not to replace them. However where BBC-specific constraints exist they are described, in relation to the subtitle guidelines that they support. The specifications remain authoritative and they should be consulted alongside this document:

- TTML 1
- EBU-TT 1.0
- EBU-TT-D 1.0.1

Because closed subtitles are processed from file, it is possible for a presentation processor (e.g. a set-top box or a browser) to override the instructions in the subtitles file. Generally, the processor should respect the author's intentions. However, where requirements exist that are specific for the authoring or processing of subtitle documents, they are listed separately under the relevant XML element.

Note that in the spirit of an iterative process, there may be further releases making improvements to the developer guidance.

In particular, the focus here is on EBU-TT-D creation for online only subtitle delivery; where there is commonality with EBU-TT Part 1 delivery for archive and downstream conversion to a distribution format this is described; however we do not expect that all existing EBU-TT Part 1 delivery requirements are captured here.

All feedback is welcome.

27.1 Introduction to the TTML document structure

TTML is a markup language based on XML, using structural elements like in HTML - `head`, `body`, `div`, `p` and `span` in the TTML namespace (shown in this document with the `tt:` prefix), with styling semantics taken from XSL-FO and timing semantics taken from SMIL. EBU-TT and EBU-TT-D are subsets of TTML with a couple of extensions. Styling and layout are applicative, in other words styling and positional information are defined and identified, and content specifies the styles and positioning by referencing those identified style and regions.

The top level `<tt:tt>` element carries parameters needed for presenting the content.

The `<tt:head>` element carries styling, layout and document level metadata.

The `<tt:body>` element carries the timed content that is to be presented, in a `<tt:div>`, `<tt:p>` and `<tt:span>` / `<tt:br/>` hierarchy. Content elements can be timed using `begin` and `end` attributes.

The following example illustrates this structure.

27.2 Example EBU-TT-D document

This example can also be downloaded here.

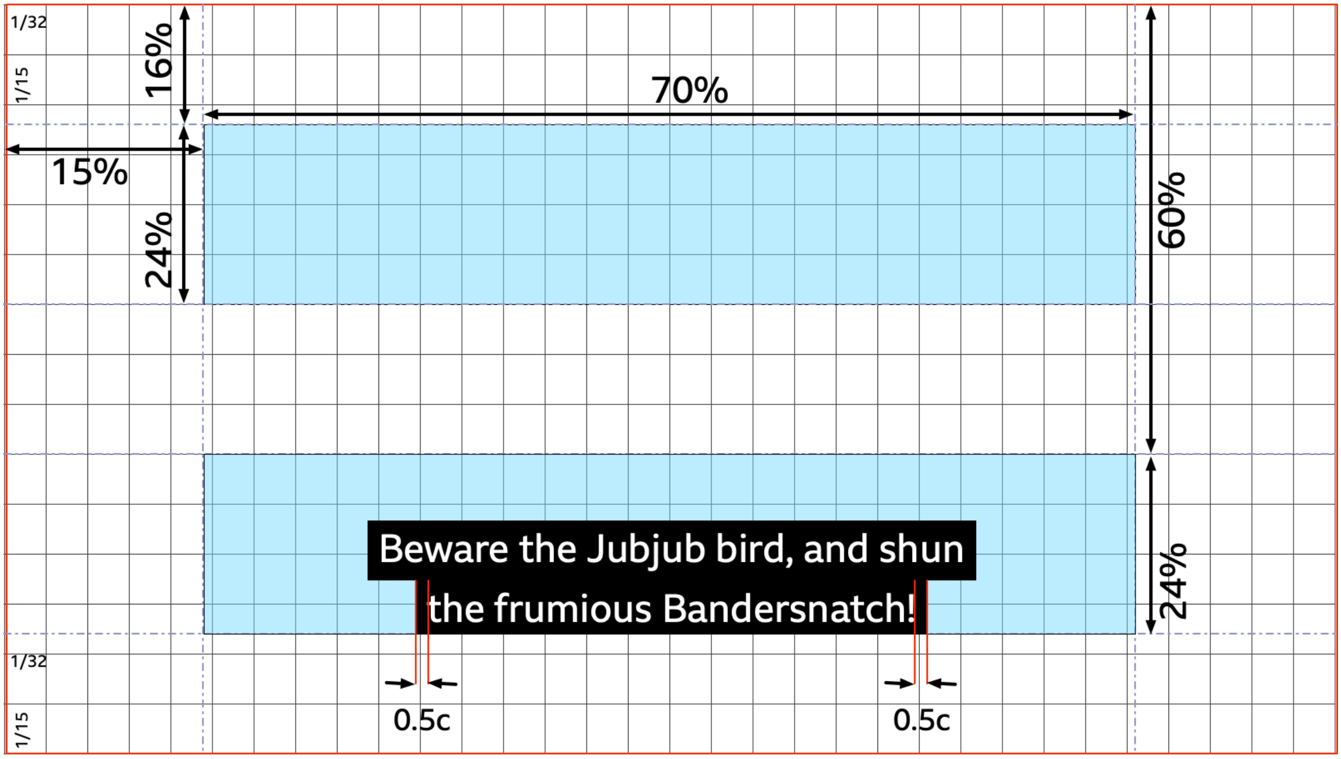
```
<?xml version="1.0" encoding="UTF-8"?>
<tt xmlns="http://www.w3.org/ns/ttml"
  xmlns:ttp="http://www.w3.org/ns/ttml#parameter"
  xmlns:tts="http://www.w3.org/ns/ttml#styling"
  xmlns:ebutts="urn:ebu:tt:style"
```

```
xmlns:ebuttm="urn:ebu:tt:urn:ebu:tt:metadata"
ttp:timeBase="media"
ttp:cellResolution="32 15"
xml:lang="en" >
<head>
 <metadata>
   <ebuttm:documentMetadata>
     <ebuttm:conformsToStandard>urn:ebu:tt:distribution:2018-04</ebuttm:confo
     <ebuttm:conformsToStandard>http://www.w3.org/ns/ttml/profile/imsc1/text<
   </ebuttm:documentMetadata>
 </metadata>
   <!--
     The styling element defines the styles that will be applied to <p> and <
     EBU-TT uses referenced styles only - inline styles are not supported.
   -->
  <styling>
    <style xml:id="paragraphStyle"
      tts:fontFamily="ReithSans, Arial, Roboto, proportionalSansSerif, defaul
      tts:fontSize="100%"
      tts:lineHeight="120%"
      tts:textAlign="center"
      tts:wrapOption="noWrap"
      ebutts:multiRowAlign="center"
      ebutts:linePadding="0.5c" />
    <style xml:id="spanStyle"
      tts:color="#FFFFFF"
      tts:backgroundColor="#000000" />
    <style xml:id="yellowStyle"
      tts:color="#FFFF00"
      tts:backgroundColor="#000000" />
  </styling>
  <!--
    The layout element defines the regions where subtitle text is displayed.
    Here, a top and a bottom regions are defined, with a clearance of 2 lines
    text from the top and bottom.
    With a cell resolution of 32 by 15, a font height of 100% (of cell height
    6.66% (100/15). A line height of 120% of the font size equals 8% of the h
    the active video (1.2 x 6.66). Each region accommodates 3 lines of text:
    3 x 8% = 24% which sets the region's height.
    The width of the regions is set at 71.25% to take into account any potent
    cut of 16:9 video on 4:3 displays. The amount of text that can fit within
    is restricted by its size and also by the required application of 1c of l
    padding (2 x 0.5c). This width has been calculated also to accommodate th
    maximum 38 characters that can be practically put on a Teletext line at t
    size, where the font is not unusually wide.
  -->
  <layout>
    <region xml:id="topRegion"
```

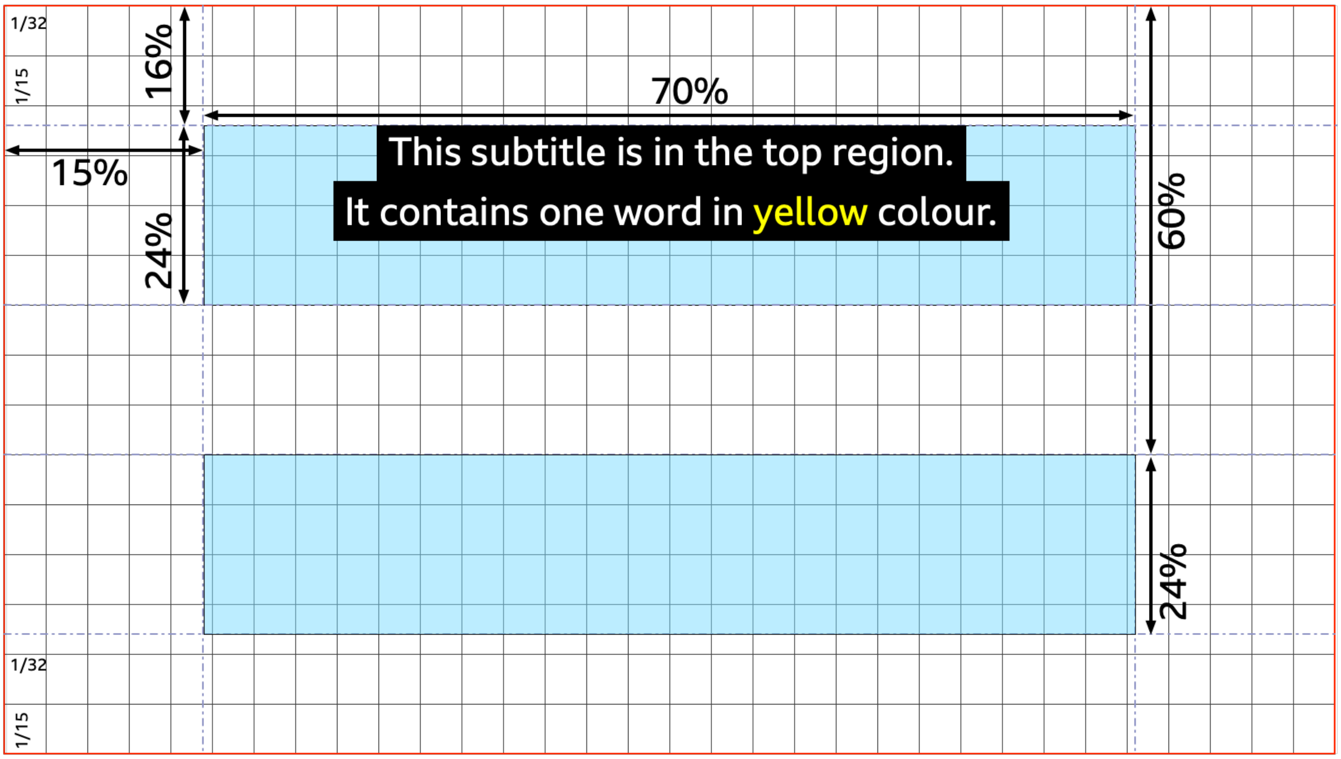
```
         tts:origin="14.375% 16%"
         tts:extent="71.25% 24%"
         tts:displayAlign="before"
         tts:writingMode="lrtb"
         tts:overflow="visible" />
       <region xml:id="bottomRegion"
         tts:origin="14.375% 60%"
         tts:extent="71.25% 24%"
         tts:displayAlign="after"
         tts:writingMode="lrtb"
         tts:overflow="visible" />
     </layout>
  </head>
  <body>
   <!--
     The intended use of DIVs is to hold semantic information, for example secti
     within a programme. DIVs are not intended to be used for presentation, alth
     style applied to them would cascade to descendent elements.
   -->
   <div>
     <!--
       A paragraph holds a single subtitle of one or more lines, with a
       time range and region allocation.
     -->
     <p xml:id="subtitle1" region="bottomRegion" style="paragraphStyle"
       begin="00:00:10.000" end="00:00:20.000">
       <!--
         A span is used to apply style to the text, by reference.
       -->
       <span style="spanStyle">Beware the Jubjub bird, and shun
       <br/>
       The frumious Bandersnatch!</span>
     </p>
     <p xml:id="subtitle2" region="topRegion" style="paragraphStyle"
       begin="00:00:30.000" end="00:00:31.000">
       <!--
         Nesting <span> elements is not allowed in EBU-TT-D.
         Avoid white space characters (e.g. space, linebreak, tab, carriage retu
         between <span> elements as these may render as gaps
         (see backgroundColor).
         The space between words in adjacent spans should be inserted at the
         end of the first <span>, or more usually, at the beginning of the
         second <span>.
       -->
       <span style="spanStyle">This subtitle is in the top region.<br/>
       it contains one word in</span><span style="yellowStyle"> yellow</span><sp
       style="spanStyle"> colour.</span>
     </p>
   </div>
  </body>
 </tt>
```

This illustration shows how the document above is interpreted (only the subtitle text and the black background will be displayed). Note that the underlying grid is virtual and that elements don't necessarily align to it. See `ttp:cellResolution` .

Displayed between 00:00:10 and 00:00:20



Displayed between 00:00:30 and 00:00:31


1/32
1/15
16%
70%
15%
24%
60%
This subtitle is in the top region.
It contains one word in yellow colour.
24%
1/32
1/15

27.3 Namespaces

In the following tables, prefixes are used as shortcuts for the following namespaces:

Prefix	Namespace	Notes
`tt:`	`http://www.w3.org/ns/ttml`	The main TTML namespace
`ttp:`	`http://www.w3.org/ns/ttml#parameter`	The TTML parameter namespace
`tts:`	`http://www.w3.org/ns/ttml#styling`	The TTML styling namespace - for style attributes
`ttm:`	`http://www.w3.org/ns/ttml#metadata`	The TTML metadata namespace
`ebutts:`	`urn:ebu:tt:style`	The EBU-TT and EBU-TT-D style extension namespace
`ebuttm:`	`urn:ebu:tt:metadata`	The EBU-TT and EBU-TT-D metadata extension namespace
`ittp:`	`http://www.w3.org/ns/ttml/profile/imsc1#parameter`	The IMSC Parameter namespace
`itts:`	`http://www.w3.org/ns/ttml/profile/imsc1#styling`	The IMSC Styling namespace

Note that although the examples in this document explicitly include the `tt:` prefix, there is no requirement that real world documents do so. For example a common approach is to declare the *default* XML namespace prefix to be the main TTML namespace, and then omit the relevant prefixes.

> For example:
>
> ```
> <tt xmlns="http://www.w3.org/ns/ttml" ... >
> ```
>
> is equivalent to:
>
> ```
> <tt:tt xmlns:tt="http://www.w3.org/ns/ttml" ... >
> ```
>
> which is in turn equivalent to:
>
> ```
> <someotherprefix:tt xmlns:someotherprefix="http://www.w3.org/ns/t
> ```

27.4 Parameter Attributes

27.4.1 ttp:timeBase

BBC-specific requirements apply.

Description

Defines the time coordinate system for all time expressions.

- If the timebase is `"smpte"`, subtitle begin and end time expressions are interpreted in the SMPTE 12M-1-2008 system: hh:mm:ss:ff (hour:minute:second:frame). If this timebase is used, `ttp:markerMode`, `ttp:dropMode`, `ttp:frameRate` and `ttp:frameRateMultiplier` attributes must be specified on the `tt` element.
- If the timebase is `"media"`, begin and end times denote a coordinate on the time-line of a media object. This can be either:
  - Full-Clock-value: hh:mm:ss followed by an optional fraction with a leading period, e.g. 02:30:03, 01:00:10.25
  - Timecount-value: value followed by an optional fraction and a symbol for the time metric, e.g. 3.2h (3 hours and 12 minutes). Allowed time metrics are h, m, s, ms (millisecond)

EBU-TT-D ttp:timeBase must be set to "media" and only a Full-Clock-value time expressions are allowed.

Cardinality	1..1
BBC requirement	EBU-TT 1.0 ttp:timeBase must be set to "smpte" .
Values	"media" | "smpte" EBU-TT-D Only "media" is allowed. EBU-TT 1.0 Only "smpte" is allowed.
Default value	
Example	
Reference	Section 4.12 of EBU-TT

Document Requirements

Shall requirement:

EBU-TT-D For EBU-TT-D output, set ttp:timeBase to "media" and use full clock time expressions on begin and end attributes.

Example:

```
<!--
EBU-TT-D must use "media" timebase and
Full Clock format time expressions.
-->
<tt:tt ttp:timeBase="media" ... />
...
<!--
Begin and end times in Full clock,
optional fraction with leading period
-->
<tt:p begin="01:00:10.25" end="01:00:11" ... >
<tt:span style="spanStyle">Subtitle text.</tt:span>
</tt:p>
<tt:p begin="01:00:12.345" end="01:00:23.456" ... >
<tt:span style="spanStyle">More Subtitle text.</tt:span>
</tt:p>
```

```
...
```

Shall requirement:

EBU-TT 1.0 For EBU-TT output, set `ttp:timeBase="smpte"` , also set `ttp:dropMode` , `ttp:markerMode` , `ttp:frameRate` and `ttp:frameRateMultiplier`

> Example:
>
> ```
> <!--
> If SMPTE timebase is used, these elements
> are also required:
> ttp:frameRate - used to
> interpret SMPTE time expressions
> ttp:frameRateMultiplier - applied to
> compute the effective frame rate.
> If the frame rate is a whole number of
> frames per second then the value
> of frameRateMultiplier is "1 1"
> ttp:markerMode -  value must be
> "discontinuous".
> See specification for details.
> ttp:dropMode - specifies constraints on
> the interpretation and use of
> frame counts associated with SMPTE timebase.
> When the calculation of the framerate from
> the ttp:frameRate and  ttp:frameRateMultiplier
> results in an integer then the value is "nonDrop".
> See TTML
> -->
> <tt:tt
> ttp:timeBase="smpte" ttp:frameRate="24"
> ttp:frameRateMultiplier="1 1" ttp:markerMode="discontinuous"
> ttp:dropMode ="nonDrop"... />
> ...
> <!-- Begin and end times in hh:mm:ss:ff SMPTE format -->
> ```

```
<tt:p begin="01:31:59:07" end="01:32:04:22" ... >
    <tt:span style="spanStyle">Subtitle text.</tt:span>
</tt:p>
...
```

Processor requirements

Shall requirement:

Attempt to display subtitles as close as possible to their respective begin and end times, regardless of the actual displayed frame rate. See Annex E of EBU-TT-D specification.

27.4.2 ttp:cellResolution

BBC-specific requirements apply.

Description

Expresses a virtual 2 dimensional grid of cells. The first value defines the number of columns and the second value defines the number of rows. The cell height ('c' unit) is used as the basis for computing font size and therefore indirectly line height. For example, the default value `"32 15"` creates a cell with height 6.66% (=100/15) and width 3.125% (=100/32) of the root container region's height and width. The root container region is defined as the active video area in EBU-TT but implementation defined in EBU-TT-D.

Font size percentages are relative to the parent element's font size, or if none is set, the cell height. For example a font size of 100% set on an element with no ancestor that sets font size would be computed as 1/15 (=6.66%) of the root container region height; a line height of 120% applied to that would be 120% of the font size, i.e. 1.2 * 1/15 = 8%.

EBU-TT 1.0 If the ‘cell’ measurement unit is used (e.g. as part of a `tts:fontSize` attribute value) then the `ttp:cellResolution` attribute must be specified.

Cardinality	0..1
BBC requirement	This attribute is required (cardinality: 1..1).
Values	Two integers separated by a space.
Default value	EBU-TT-D "32 15" EBU-TT 1.0 "40 24"
Example	XML | Image
Reference	https://www.w3.org/TR/ttml1/#parameter-attribute-cellResolution
Presentation	Cell resolution is used for setting the font size and therefore the line length. It is also used to set the line padding of the background colour. Cell units may also be used in the definition of regions that control vertical and horizontal positioning.

Document Requirements

Shall requirement:

EBU-TT 1.0 Set the cell resolution explicitly even if using the default value.

> Example:
>
> ```
> <tt:tt ttp:cellResolution="32 15" ... >
> ```

Processor Requirements

Shall requirement:

For 16:9, 4:3 and 1:1 aspect ratio videos, the computed font size must fit within a line height of between 7% and 9% of the active video height.

For 9:16 aspect ratio videos, the computed font size must fit within a line height of between 4% and 5% of the active video height.

See Section 10.2 Font size

27.4.3 ittp:activeArea

Description

Describes the area within the root container region that contains subtitles, i.e. the area that needs to be minimally visible to the viewer. This area typically fully contains all of the referenced regions within the Document Instance.

The active area may be used by a player to ensure that the subtitles remain in the visible area of the screen if the player area is not the same shape as the video image, for example.

Cardinality	0..1
BBC requirements	This attribute is optional (cardinality: 0..1) and should be present.
Values	Four percentages separated by spaces, representing respectively left, top, width and height.
Default value	`"0% 0% 100% 100%"`
Reference	https://www.w3.org/TR/ttml-imsc1.0.1/#ittp-activeArea
Presentation	There are no specific presentation requirements for active area, however players should ensure that all of the active area is visible within the player window.

Document Requirements

Should requirement:

EBU-TT-D Specify the active area.

```
<tt:tt ... ittp:activeArea="12% 5% 76% 90%" ... >
```

Processor Requirements

Should requirement:

The visible area of the video must include the specified active area occupied by subtitles. In normal conditions the registration or positional alignment of the subtitle rendering area against the video image must not be modified to achieve this. The exception would be if the subtitle rendering area is adjusted temporarily for example to accommodate controls.

27.5 Style Attributes

27.5.1 tts:fontFamily

Description

Sets a generic or a named font family. This attribute can contain a prioritised list of font names, which are typically processed in order until a match is found, thus allowing predictable fallbacks to be used. This list may be evaluated on a per glyph basis to deal with the case where most glyphs are present in a font but later fonts include specific required glyphs omitted from earlier fonts, for example.

Cardinality	0..1
BBC requirement	Set the font family to `ReithSans, Arial, Roboto, proportionalSansSerif, default` for all content in the document. This can be done efficiently for example by referencing a style that includes a `tts:fontFamily` specification from the `body` element, or by ensuring that every `style` specifies a `tts:fontFamily` itself or, for EBU-TT, references another `style` that does.
Values	`"default" | "monospace" | "sansSerif" | "serif" | "monospaceSansSerif" | "monospaceSerif" | "proportionalSansSerif" | "proportionalSerif" | [named font]`
Default value	`"default"`

Reference	See informative discussion of font usage in section 2.7 of EBU-TT part 1. The font family data type is defined in https://www.w3.org/TR/ttml1/#style-attribute-fontFamily
Presentation	Used to specify the subtitle font. The choice of font also determines the line height and may also affect the supported characters. Because fonts have different widths, changing the font may also alter the width of each line.

Document Requirements

Shall requirement:

Set to Reith Sans, fall back to device-specific and then generic proportional sans-serif fonts so that the end device uses its default font (e.g. Roboto in Android).

> Example:
>
> ```
> <tt:style xml:id="s0"
>    tts:fontFamily="ReithSans, Arial, Roboto, proportionalSansS
> ```

Processor Requirements

Shall requirement:

Map a generic font family name to the best appropriate matching font on the device.

Shall requirement:

Use downloadable fonts if available.

Shall requirement:

Fall back to the system defined sans serif font if a downloadable font is not available. Prefer proportional fonts if there is a choice.

27.5.2 tts:fontSize

BBC-specific requirements apply.

Description

EBU-TT 1.0 Sets the font size using percent, pixel or cell values. Double values can be used to set height and width separately, known as anamorphic font sizing - this scales the font by different amounts horizontally and vertically.

EBU-TT-D Sets the font size using a percentage of cell height value (see cell resolution). A single value only can be used.

Percentage values are relative to the parent element's font size, or the cell size when the parent element (and every ancestor) has no specified font size.

Cardinality	0..1
BBC requirement	The font size shall be explicitly set, without relying on the default initial value. The computed value of font size must be appropriate to result in the correct size relative to the active video. This can be achieved by setting a value of `ttp:cellResolution` and referencing a `style` that includes a `tts:fontSize` specification from the `body` element, or by ensuring that the `style` specifies a `tts:fontSize` itself or references another `style` that does. **Note** that applying `tts:fontSize` attributes to more than one element in the same hierarchy, e.g. both a `div` and its parent `p` results in the percentages being *multiplied* together, not overridding each other.
Values	**EBU-TT 1.0** one or two positive decimals followed by "%", "px" or "c". If a single value is specified, then this length applies equally to horizontal and vertical scaling; if two values are specified, then the first expresses the horizontal scaling and the second expresses vertical scaling. If "c" is used then ttp:cellResolution must be specified. If "px" is used, then tts:extent must be specified.

	EBU-TT-D one percentage value (of cell height). "c" and "px" are not allowed.
Default value	EBU-TT 1.0 "1c 2c" EBU-TT-D "100%"
Example	EBU-TT-D font size at 80%: XML | Image
Reference	EBU-TT 1.0 data type: Section 4.5 of the specification EBU-TT-D: Section 4.7 of the specification
Presentation	Used to set the font size.

Document Requirements

Should requirement:

For BBC subtitles for 16:9, 4:3 or 1:1 aspect ratio videos, set the font size to be approximately 1/15th (6.667%) of the height of the root container, for example by setting `ttp:cellResolution` to `"32 15"` and `tts:fontSize` to `"100%"` .

For BBC subtitles for 9:16 aspect ratio videos, set the font size to be approximately 3.75% of the height of the root container, for example by setting `ttp:cellResolution` to `"32 15"` and `tts:fontSize` to `"56.25%"` . An approximation can be made by setting `ttp:cellResolution` to `"32 27"` and `tts:fontSize` to `"100%"` .

> Example:
>
> ```
> <tt:tt
>    [namespace, parameter, style attributes etc.]
>    ttp:cellResolution="32 15">
> ...
>    <tt:style
>       xml:id="defaultSpanStyle"
>       [other style attributes]
>       tts:fontSize="100%" />
> ```

Processor Requirements

Shall requirement:

Calculate percentage values relative to the parent element's font size, if specified, or the cell size otherwise.

> Example:
>
> ```
> <tt:tt
>    [namespace, other parameter, style attributes etc.]
>    ttp:cellResolution="32 15">
> ...
>    <tt:style
>       xml:id="bigStyle"
>       tts:fontSize="150%"/>
>    <tt:style
>       xml:id="smallStyle"
>       tts:fontSize="50%"/>
> ...
>    <tt:div style="bigStyle">
>       <tt:p>Big text</>
>       <!--
>       The text on the above line will
>       render at 150% of the cell height
>       -->
>       <tt:p style="smallStyle">Small text</tt:p>
>       <!--
>       The text on the above line will render at
>       50% of 150% (i.e. 75%) of the cell height
>       -->
>    </tt:div>
> ```

Should requirement:

Apply a scaling factor based on the device's physical screen size (see Presentation font size).

> Example:

> For a 32" TV and an authored font size corresponding to the authored font size guideline, apply a scaling factor of 0.67 so that the computed font size is as though the font size was specified at 67% of cell height.

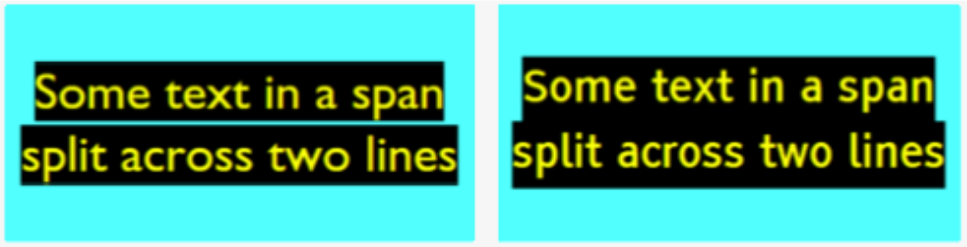
27.5.3 tts:lineHeight

Description

Sets inter-baseline separation between line areas.

Note that there is no uniform implementation of the value "normal" by CSS-based rendering processors. Additionally, different browsers render different line heights for the same font and size. This contributes to a known issue where a gap appears between lines of text. See also `itts:fillLineGap`.

The example below illustrates this: different fonts of the same size were used, with a line height set to "normal". Processors should render as on the right example, without a gap between the lines:



Some text in a span
split across two lines

Some text in a span
split across two lines

Cardinality	0..1
Values	`"normal" | [Percent]`
Default value	"normal"
Example	Line height set at 125%: XML | Image
Reference	https://www.w3.org/TR/ttml1/#style-attribute-lineHeight
Presentation	Line height sets the distance between baselines of successive lines of text. The number of lines that fit within a region is therefore affected by line height: subtitles may occupy up to 3 lines.

Document Requirements

Should requirement:

Set the line height explicitly on a style applied to a `<p>` element using percentage values, evaluated relative to the computed font size.

> Example:
>
> ```
> <tt:style xml:id="paragraphStyle" tts:lineHeight="120%" ... />
> ...
> <tt:p xml:id="p123" style="paragraphStyle">...</tt:p>
> ```

Should requirement:

Set `tts:lineHeight="120%"` to accommodate commonly used web fonts.

Processor Requirements

Shall requirement:

Calculate the line height for a line area using the font's ascender, descender and lineGap attributes, including leading if available.

27.5.4 tts:textAlign

Description

Alignment of inline areas in a containing block. The alignment values `"start"` and `"end"` depend on the value of the writing mode, which in turn depends on the Unicode bidi mode and the style attributes `tts:unicodeBidi` , `tts:direction` and `tts:writingMode` applied to the element.

See also `ebutts:multiRowAlign` , which provides extra alignment options.

Cardinality	0..1
Values	`"left" | "center" | "right" | "start" | "end"`
Default value	**EBU-TT 1.0** `"center"` **EBU-TT-D** `"start"` [TTML] `"start"`
Example	Text align end: XML | Image
Reference	https://tech.ebu.ch/publications/tech3350 - see Appendix A for the effects of different combinations with tts:multiRowAlign
Presentation	With `tt:region` and `ebutts:multiRowAlign` , used for horizontal positioning of subtitles for speaker identification and to centre song lyrics (within a sequence of left- or right-aligned subtitles). This property is also used to control breaks in justified subtitles.

Document Requirements

Shall requirement:

Set this explicitly even if using defaults. **EBU-TT 1.0**

> Example:
>
> ```
> <tt:style xml:id="paragraphStyle" tts:textAlign="center"/>
> ...
> <tt:p xml:id="p123" style="paragraphStyle">...</tt:p>
> ```

Processor Requirements

Should requirement if only supporting left to right scripts; **Shall** requirement if support for any non-Latin or non-left-to-right text is required:

Support Unicode characters and the Unicode bidirectional algorithm (UAX9).

Shall requirement:

Calculate text alignment correctly based on the value of `tts:textAlign`, the Unicode bidirectional algorithm, and all defined values of `tts:unicodeBidi` and `tts:direction` and `tts:writingMode`.

> Example:
>
> ```
> <tt:region
>    xml:id="topRegion"
>    [origin, extent, other attributes]
>    tts:writingMode="lrtb" />
> <tt:style
>    xml:id="startStyle"
>    [other style attributes]
>    tts:textAlign="start"/>
> <tt:style
>    xml:id="rtlStyle"
>    tts:unicodeBidi="bidiOverride"
>    tts:direction="rtl"/>
> ...
> <!--
> The lines below will be left aligned
> (start = left here)
> -->
> <tt:p region="topRegion" style="leftStyle">
> Little birds are playing<br/>
> Bagpipes on the shore,<br/>
> <!--
> The line below will display
> ".erons stsiruot eht erehw"
> and will be right aligned
> (start = right for rtl)
> ```

```
-->
  <tt:span style="rtlStyle">
    where the tourists snore.
  </tt:span>
</tt:p>
```

Shall requirement:

Calculate text alignment relative to the region after taking into account any start or end padding.

Shall requirement:

Align the line areas generated by a `<tt:p>` element after applying any line padding; for example, if there is `0.5c` of line padding applied to each line area and `1c` of start and end padding on the region, then the first glyph of a left aligned line area will be `1.5c` to the right of the region origin's x coordinate.

27.5.5 tts:wrapOption

EBU-TT-D only.

Description

In EBU-TT-D only, defines whether automatic line wrapping applies within an element. If the value is `"wrap"`, automated line-breaking occurs if the line overflows the region. If the value is `"noWrap"`, no automated line-breaking occurs and overflow is treated in accordance with the value of `tts:overflow` attribute of the corresponding region.

Note that if `tts:wrapOption` is set to `"noWrap"`, the region that corresponds to the affected content should have the attribute tts:overflow set to "visible" so that any overflowing text remains visible.

Although the default value is "wrap", it is better to have the subtitler, rather than the software, control line breaks by inserting <tt:br/>. Subtitlers and authoring software are expected to manage the width of text on each line so that the text does not overflow.

There is no constraint on adding manual breaks regardless of the value of tts:wrapOption.

Cardinality	0..1
Values	"wrap" | "noWrap"
Default value	"wrap"
Example	Text overflows with tts:wrapOption="noWrap": XML | Image
Reference	TTML | EBU-TT-D
Presentation	Because good line breaks and handling of long sentences are essential to quality subtitles, it is expected that the subtitler will enter those manually and automatic wrapping will be disabled.

Document Requirements

Should requirement:

Disable automatic line wrapping so that the editor creates line break manually.

> Examples:
>
> - Set tts:wrapOption="noWrap";
> - separate lines of content by putting into separate <tt:p> elements or by inserting <tt:br/> elements

Shall requirement:

EBU-TT 1.0 For EBU-TT 1.0, do not include this attribute.

Should requirement:

If `tts:wrapOption` is set to `"noWrap"`, set the attribute `tts:overflow` to `"visible"` on the region that corresponds to the affected content.

Should requirement:

When deriving break points, use the UAX14 line breaking algorithm.

Processor Requirements

Should requirement:

Use the UAX14 line breaking algorithm.

Note that when the document has `tts:wrapOption="noWrap"` the line breaking algorithm will not apply.

Shall requirement:

If the text overflows its region, attempt to display the overflow (even if ugly) so that viewers who depend on subtitles don't miss important information.

Shall requirement:

EBU-TT 1.0 For EBU-TT processing, this attribute should be ignored and manual line breaks used instead. See page 33 of the specification

27.5.6 ebutts:multiRowAlign

Description

Defines how multiple ‘rows’ of inline areas are aligned relative to each other within a containing block area.

This attribute acts as a ‘modifier’ to the action defined by the `tts:textAlign` attribute value, whether that value is explicitly or implicitly specified. This attribute effectively creates additional alignment points for multiple rows of text, thus *it has no effect if only a single row of text is present*.

`ebutts:multiRowAlign` modifies the behaviour of `tts:textAlign` so that, rather than each line generated by the `tt:p` being aligned relative to the region, each line in the group can be left/centre/right aligned relative to the longest line and the group of lines is then aligned according to `tts:textAlign`. See the references for more detail on this.

Cardinality	0..1
Values	**EBU-TT 1.0** "start" | "center" | "end" | "auto"
Default value	"auto"
Example	Combination of tts:textAlign="start" and ebutts:multiRowAlign="end": XML | Image
Reference	**EBU-TT 1.0** https://tech.ebu.ch/publications/tech3350 **EBU-TT-D** Annex C in https://tech.ebu.ch/publications/tech3380
Presentation	No editorial requirement exists for using multiRowAlign in these guidelines however it is permitted to use it if the need arises.

Processor Requirements

Shall requirement:

If the ebutts:multiRowAlign attribute as specified on a tt:p element has the same value as tts:textAlign or is set to "auto", each generated line area in the tt:p shall be aligned according to the computed value of tts:textAlign.

> Example:
>
> ```
> <tt:style xml:id="paragraphStyle"
> tts:textAlign="center"
> ebutts:multiRowAlign="center"
> tts:lineHeight="120%"/>
> ...
> <tt:p xml:id="subtitle1" region="top"
> begin="00:00:30.000" end="00:00:31.000"
> style="paragraphStyle">
> These two lines <tt:br/>
> Will be centred.
> </tt:p>
> ```

Shall requirement:

The behaviour of this attribute in combination with tts:textAlign is as defined in Annex C in https://tech.ebu.ch/publications/tech3380.

> Example:
>
> ```
> <tt:style xml:id="startEnd"
> tts:textAlign="start"
> ebutts:multiRowAlign="end"/>
> ...
> <tt:p xml:id="subtitle1" region="regionTop"
> style="startEnd" begin="00:00:00"
> end="00:00:03">
>   Longer line left-aligned in the region.
>   <tt:br/>
>   shorter right-aligned with "region.".
> </tt:p>
> ```

27.5.7 ebutts:linePadding

EBU-TT-D only.

BBC-specific requirements apply.

Description

In EBU-TT-D only, adds padding on the start and end edges of each rendered line. Background color applies to the line area including the padding.

Application of padding affects the layout of text, for example by reducing the maximum width available in which to render text on a single line (see line length and region definition). Note this attribute is different from `tts:padding`, which applies space to a region (and in TTML2, to other content elements). Must be applied to `tt:p` only.

Cardinality	0..1
BBC requirements	All content must have a computed value for this style that is the equivalent to half a character on each side (see Document Requirements below). This can be achieved for example by referencing a style that includes an `ebutts:linePadding` specification from the `tt:body` element, or by ensuring that every `style` attribute applied to a `tt:p` element specifies an `ebutts:linePadding` value itself or references another `tt:style` element that does.
Values	Non-negative decimal appended by "c".
Default value	`"0c"`
Example	Subtitle with line padding: XML | Image
Reference	See Annex D of the EBU-TT part 1 specification for a detailed description of how the attribute can be used.
Presentation	The primary use of line padding is to add an extra area of background colour to both sides of a subtitle line, as described in typography. Line padding also affects the length of lines since it adds to the space taken up by text within a region.

Document Requirements

Shall requirement:

EBU-TT-D For EBU-TT-D, set line padding to approximately half a character width. This should be calculated from the aspect ratio, the grid and the font size. For the purposes of the calculation, 1em can be assumed to be equal to the font size.

> The following example calculation uses non-recommended values for illustration purposes only.
>
> Assuming an aspect ratio of 16:9, a cellResolution of "32 15" and a font size of 80%:
>
> - font height = 5.33% of video height (80% x 100% / 15)
> - font width (also 1em), expressed as a fraction of the width of the root container region: 2.99% (5.33% x 9 / 16)
> - 0.5em = 1.495% (2.99 / 2)
> - Expressed in cells: 0.47 (32 * 1.495 / 100)
>
> ```
> ebutts:linePadding="0.47c"
> ```

Processor Requirements

Should requirement:

If no line padding exists and there is sufficient space available, add 0.5c of padding on the sides of each line. Note that the recommended behaviour for `tts:overflow` is that it is `"visible"` and that `tts:wrapOption` is `"noWrap"`.

Shall requirement:

When laying out line areas inset the line areas by twice the value of `ebutts:linePadding` from the start and end edges of the region, after having applied any `tts:padding` values.

Should requirement:

If scaling down the font size, also reduce the line padding.

If scaling the line padding, it may be reduced by up to the same percentage as the relative reduction in font size (i.e., if multiplying the font size by 50%, the line padding may be multiplied by a value in the range 50%-100%).

27.5.8 tts:color

BBC-specific requirements apply.

Description

The foreground (text) color of an area.

Cardinality	0..1
BBC requirements	The text colour must be explicitly set to a value that is one of the values listed below (see Document Requirements).
Values	**EBU-TT-D** Hex notated RGB color triple (e.g. `"#000000"`) or a hex notated RGBA color tuple (e.g. `"#000000FF"`). **EBU-TT 1.0** permits both RGB triple and RGBA tuple values as well as named colours.
Default value	Undefined (see below)
Example	
Reference	EBU-TT-D colour datatype: Section 4.2 in https://tech.ebu.ch/publications/tech3380
Presentation	The primary use of colour is to identify speakers. Only a limited set of speaker colours is allowed. Most subtitles are in white text on black.

Document Requirements

Shall requirement:

Set the default font colour to white `"#FFFFFF"`

> Example:
>
> ```
> <tt:style
>    xml:id="defaultParagraphStyle"
>    tts:color="#FFFFFF"
>    tts:textAlign="center"
>    ebutts:multiRowAlign="center"
>    tts:lineHeight="120%"/>
> <tt:style
>    xml:id="defaultSpanStyle"
>    tts:backgroundColor="#000000"/>
> <tt:style
>    xml:id="yellowSpan"
>    tts:color="#FFFF00" />
> ...
> <tt:p
>    xml:id="subtitle3"
>    begin="00:00:30.000" end="00:00:31.000"
>    style="defaultParagraphStyle">
>    <tt:span
>       style="defaultSpanStyle yellowSpan">
>       This subtitle is in yellow that overrides
>       the white in the defaultParagraph style.
>    </tt:span>
> </tt:p>
> ```

Shall requirement:

The attribute can have one of these values only (see Speaker Colours):

- "#FFFFFF" (white),
- "#FFFF00" (yellow),
- "#00FFFF" (cyan),
- "#00FF00" (green)

Processor Requirements

Shall requirement:

Apply the specified colour to text.

27.5.9 tts:backgroundColor

BBC-specific requirements apply.

Description

Background colour of an inline area generated by a <tt:span> element. This attribute can also be applied to block elements and other colours are supported, but BBC subtitles use black background applied to <tt:span> elements only.

Note that the TTML tts:opacity attribute is not supported by EBU-TT and EBU-TT-D but alpha values may be included on RGB colours.

Cardinality	0..1
BBC requirements	The background colour must be explicitly set on all text content in the document to a value equivalent to solid black. This can be done by wrapping all text in a span element that references a style that includes a tts:backgroundColor specification.
Values	**EBU-TT-D** Hex notated RGB color triple (e.g. "#000000") or a hex notated RGBA color tuple (e.g. "#000000FF"). **EBU-TT 1.0** permits both RGB triple and RGBA tuple values as well as named colours.

Default value	`"transparent"`
Example	EBU-TT-D with background colours applied to both `<tt:span>` and `<tt:p>`: XML | Image
Reference	
Presentation	All subtitles display on a black background.

Document Requirements

Shall requirement:

Set background colour to solid black (do not allow opacity).

> Example:
>
> ```
> tts:backgroundColor="#000000"
> ```

Shall requirement:

Apply background to `<tt:span>` elements only

> Example:
>
> ```
> <tt:style
>    xml:id="spanStyle"
>    [other style attributes]
>    tts:backgroundColor="#000000" />
> ...
> <tt:p>
>    <tt:span style="spanStyle">
>       Beware the Jubjub bird, and shun
>       <tt:br/>
>       The frumious Bandersnatch!
>     </tt:span>
> </tt:p>
> ```

Shall requirement:

Avoid white space between adjacent <tt:span> elements. White space that is not styled with a background colour will appear in browsers as gaps in the background.

Required white space between words must be included inside a <tt:span> element, usually immediately before a word, at the beginning of the element contents.

> Example of what *not* to do:
>
> If the styles applied to the <span> define a background colour, the end of line character [EOL] between the <tt:span>s is unstyled:
>
> ```
> <tt:p>[EOL]
>   <tt:span style="White">Hey!</tt:span>[EOL]
>   <tt:span style="Yellow">What?</tt:span>[EOL]
> <tt:p>[EOL]
> ```
>
> This will render as:
>
> Hey!   What?

> Example of what to do instead:
>
> ```
> <tt:style
>    xml:id="spanStyle1"
>    [other style attributes]
>    tts:backgroundColor="#000000" />
> <tt:style
>    xml:id="spanStyle2"
>    [other style attributes]
>    tts:backgroundColor="#000000" />
>
> ...
> <tt:p>
>    <tt:span style="spanStyle1">
>    Beware the Jubjub bird <tt:br/></tt:span><tt:span
> ```

```
    style="spanStyle2">and shun the
    frumious Bandersnatch!</tt:span>
</tt:p>
```

Processor Requirements

Shall requirement:

Draw the background area behind each generated line area in the specified colour.

Shall requirement:

Make the height of the background equal to the font's computed line height so that no gap exists between lines. See `tt:span` .

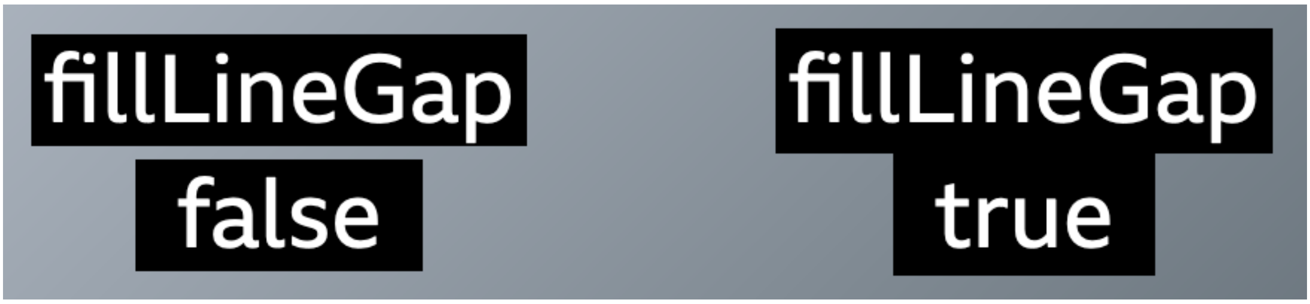
27.5.10 itts:fillLineGap

BBC-specific requirements apply.

Description

Specifies whether any gap between the background areas of adjacent lines should be filled or left unfilled.

> The example below illustrates this: the same font and line height are specified, with `itts:fillLineGap` set to `false` on the left, and `true` on the right.



Cardinality	0..1
BBC requirements	There must be no gaps between background areas of adjacent lines within the same subtitle, in a single region. If two separate subtitles are visible at the same time, for example a sound effect and dialogue, it is permissible to have a gap between their background areas.
Values	EBU-TT-D `true	false`
Default value	`"false"`
Reference	https://www.w3.org/TR/ttml-imsc1.0.1/#itts-fillLineGap
Presentation	Subtitles must not have gaps between the backgrounds of adjacent lines within a paragraph, regardless of whether those lines are explicitly specified using line breaks or `<br/>` elements or generated due to line wraps during layout. See also `tts:lineHeight` above.

Document Requirements

Shall requirement:

EBU-TT-D `itts:fillLineGap` must be set to `true`.

Example:

```
<tt:style xml:id="pStyle" itts:fillLineGap="true" ... >
...
<tt:p style="pStyle" ...>
```

Shall requirement:

EBU-TT-D Apply `itts:fillLineGap` to `<tt:p>` elements only.

> Example:
>
> ```
> <style
>    xml:id="pStyle"
>    [other style attributes]
>    itts:fillLineGap="true" />
> <style
>    xml:id="spanStyle"
>    [other style attributes]
>    tts:backgroundColor="#000000" />...
> <tt:p style="pStyle">
>    <tt:span style="spanStyle">
>       Beware the Jubjub bird, and shun<tt:br/>
>       The frumious Bandersnatch!
>     </tt:span>
> </tt:p>
> ```

Processor Requirements

Shall requirement:

Make the height of the background area of each line extend so that there is no gap between adjacent line background areas, for every line area generated by a `<tt:p>` element whose computed value of `itts:fillLineGap` is `true`.

27.6 Regions

27.6.1 tt:region

Description

Defines an area in which subtitle content is to be placed. `tt:div` and `tt:p` elements may reference a region.

For a 16:9 aspect ratio video, setting the width of a region to 71.25%, with zero padding, should be sufficient to carry all 38 possible characters across a Teletext line and add 0.5c line padding. A region of such a size should be centred horizontally (i.e. have an origin x coordinate of 14.375%) to allow for it to be displayed in its entirety even if a centre cut out is used to display the central 4:3 area of a 16:9 root container region.

For a 9:16 aspect ratio video, where it is assumed that the subtitles have not been authored with Teletext line length constraints, and that 4:3 centre cut out considerations do not apply, the width of the region can be extended to 90% to avoid requiring too many lines, while also using a safe proportion of the video width.

Cardinality	1..*
Default value	If no `tt:layout` element (and therefore no `tt:region` elements) is defined in a document, the *default region* applies, which is the entire size of the rendering area; Default values of the region style attributes also apply. This is extremely unlikely to be a desirable effect. Note that legacy (non-EBU-TT) flavours of TTML may exist that omit the `tt:layout` element: at the time when those documents were created, the default region semantic may have been different. Processors may therefore apply more sensible defaults in this scenario, for example locating subtitles in a predefined position such as bottom and centre.
Reference	https://www.w3.org/TR/ttml1/#layout-vocabulary-region
Presentation	Regions are primarily used to control vertical positioning and horizontal positioning. They also restrict the maximum width of lines and the maximum number of subtitle lines that can be displayed within the region.

Example:

```
<tt:region xml:id="r0"
   tts:displayAlign="after"
   tts:extent="68.5% 7.826%"
   tts:origin="12% 87.174%"
   tts:overflow="visible"/>
```

Document Requirements

Shall requirement:

Documents must not contain overlapping regions that are active at the same time (where a region is active if any content that is flowed into it is active).

Shall requirement:

For 16:9 aspect ratio video, the region's origin x coordinate must be greater than or equal to 12.5% of the root container region. This allows for a 4:3 centre cut of a 16:9 active video.

For 4:3, 1:1 or 9:16 aspect ratio video, the region's origin x coordinate must be greater than or equal to 9.5% of the root container region.

Shall requirement:

For 16:9 aspect ratio video, the sum of the region's origin x coordinate and extent width must be less than or equal to 87.5% of the root container region. This allows for a 4:3 centre cut of a 16:9 active video.

For 4:3, 1:1 or 9:16 aspect ratio video, the sum of the region's origin x coordinate and extent width must be less than or equal to 90.5% of the root container region.

Shall requirement:

The number of regions active at any one time must not exceed 4 (IMSC 1 requirement).

Processor Requirements

Should requirement:

Support at least eight regions that are active at the same time.

Shall requirement:

Support at least four regions that are active at the same time.

Should requirement:

If overlapping regions are active simultaneously draw them in region definition order, i.e. the order of regions in the `layout` element.

Note that this is not permitted in EBU-TT and EBU-TT-D documents.

27.6.2 tts:origin

Description

The x and y coordinates of the top left corner of a region with respect to the root container region, which is the active video for EBU-TT 1.0, and some implementation dependent rendering plane for EBU-TT-D, but generally expected to match the displayed video. Presentation implementations are expected to map these to device pixels for optimum display of text.

Example: with `tts:origin="20% 80%"` the top left corner of the region is 20% of the root container region width from the left and 80% of the root container region height from the top.

Cardinality	1..1
Values	**EBU-TT-D** 2 percentage values separated by a space **EBU-TT 1.0** Two length values (`"%"` | `"px"` | `"c"`) separated by a space, i.e. two TTML Length datatype values, except that the "em" unit is not allowed.
Default value	`"auto"` being equivalent to `"100% 100%"`
Example	EBU-TT-D: Any of the examples here
Reference	TTML
Presentation	Determines the position of a region, which is used for vertical positioning and horizontal positioning.

Document Requirements

Shall requirement:

The sum of the value for the x-coordinate of the region and the value for the width of the region (specified by `tts:extent`) must be less than or equal to 100%.

Shall requirement:

The sum of the value for the y-coordinate of the region and the value for the height of the region (specified by `tts:extent`) must be less or equal to 100%.

27.6.3 tts:extent

Description

This attribute can be specified on either `tt:region` or `tt:tt` elements. It sets the width and height of the region area, being *either* the root container region, when specified on the `tt:tt` element *or* a defined region within that, when specified on a `tt:region` element.

Note that where pixel coordinates are used they are *logical* coordinates in the TTML space only and do not need to match actual encoded video or device pixels.

EBU-TT-D Only percentage values are allowed.

EBU-TT-D `tts:extent` is only permitted on `region` elements.

EBU-TT 1.0 Only length expressions in pixels are allowed on `tts:extent` when specified on the `tt` element.

EBU-TT 1.0 If pixel length expressions are used anywhere in a document then `tts:extent` must be present on the `tt` element.

EBU-TT 1.0 Percentage and pixel values are allowed.

Cardinality	1..1
Values	EBU-TT-D 2 percentage values separated by a space EBU-TT 1.0 Two length values (`"%"` | `"px"` | `"c"`) separated by a space, i.e. two TTML Length datatype values, except that the "em" unit is not allowed.
Default value	`"100% 100%"` when applied to a `region` There is no default when applied to the `tt` element.
Example	EBU-TT-D: Any of the examples here
Presentation	A region's `extent` determines the length of subtitle lines within the region and its maximum number of lines. With `displayAlign`, it also controls the vertical positioning of subtitles. For example, in the default writing mode (left to right, top to bottom), the displayAlign value "after" would result in the subtitles aligned to to the bottom of the region defined by `extent`.

Document Requirements

Shall requirement:

The sum of the value for the x-coordinate of the region and the value for the width of the region (specified by tts:extent) must be less than or equal to 100%.

Shall requirement:

The sum of the value for the y-coordinate of the region and the value for the height of the region (specified by tts:extent) must be less or equal to 100%.

Shall requirement:

EBU-TT 1.0 The `tts:extent` must be present on the `tt:tt` element if any length unit in the document is expressed in pixels.

> Example:
>
> ```
> <tt:tt tts:extent="400px 300px">
> ```

Shall requirement:

EBU-TT-D `tts:extent` must not be present on any element other than `tt:region`

Shall requirement:

EBU-TT-D EBU-TT-D document must not express the value of `tts:extent` in pixel units.

> Example:
>
> ```
> <tt:region xml:id="r0" tts:extent="68.5% 7.826%" .../>
> ```

Processor Requirements

Should requirement:

Clip any region that extends beyond the root container region (the rectangle corresponding to an origin of 0% 0% with an extent 100% 100%) to the area that intersects with the root container region.

27.6.4 tts:displayAlign

Description

Alignment in the block progression direction. When block progression direction is top-to-bottom, "before" would result in "top" alignment and "after" would result in "bottom" alignment.

Cardinality	0..1
Values	`"before" | "center" | "after"`
Default value	`"after"`
Example	Display align center: XML | Image
Reference	TTML. Note that in EBU-TT v1 the default value was changed to `"after"` and that this was reverted to the TTML1 default of `"before"`. Therefore it is unwise to rely upon the default; to avoid ambiguity the desired value should always be specified.
Presentation	In combination with other attributes, controls vertical positioning within a region.

Document Requirements

Shall requirement:

A `tts:displayAlign` attribute shall be present on every `region` element.

```
<tt:region xml:id="r0" tts:displayAlign="after" ... />
```

Processor Requirements

Shall requirement:

The active lines within the region are aligned in the block progression direction to the before edge of the region (for `"before"`, usually the top for top to bottom left to right), the middle (for `"center"`) or the after edge (for `"after"`, usually the bottom for top to bottom left to right).

Shall requirement:

EBU-TT 1.0 In an EBU-TT Part 1 v1.0 document if no `tts:displayAlign` attribute is present the default of `"after"` shall be applied.

Shall requirement:

In an EBU-TT-D or other TTML document (e.g. EBU-TT Part 1 v1.1 etc) or if the document type is undetermined then if no `tts:displayAlign` attribute is present the TTML default of `"before"` shall be applied.

27.6.5 tts:writingMode

Description

Defines the directions for stacking block and inline areas within a region area. Applies to region elements only. This attributes interacts with `tts:direction` and `tts:unicodeBidi`.

- `"lrtb"` : "Left to Right Top to Bottom"
- `"rltb"` : "Right to Left Top to Bottom"
- `"tbrl"` : "Top to Bottom Right to Left"

- `"tblr"` : "Top to Bottom Left to Right"
- `"lr"` : "Left to Right Top to Bottom"
- `"rl"` : "Right to Left Top to Bottom"
- `"tb"` : "Top to Bottom Right to Left"

Cardinality	0..1
Values	`"lrtb" | "rltb" | "tbrl" | "tblr" | "lr" | "rl" | "tb"`
Default value	`"lrtb"`
Example	`rltb` : XML | Image
Reference	https://www.w3.org/TR/ttml1/#style-attribute-writingMode
Reference	With other style attributes, controls horizontal positioning.

Document Requirements

Should requirement:

Specify `tts:writingMode` on a region.

> Example:
>
> ```
> <tt:style xml:id="paragraphStyle"
> tts:direction="rtl" tts:unicodeBidi="bidiOverride"/>
> ...
> <tt:region xml:id="r1" tts:writingMode="rltb"
> tts:origin="15% 16%" tts:extent="70% 24%"/>
> ...
> <tt:p region="r1" style="paragraphStyle">
> <!-- This line will display ".uoy evol I", right aligned. -->
> I love you.
> </tt:p>
> ```

Processor Requirements

May requirement (where support for Latin scripts or left-to-right-top-to-bottom scripts only is required):

Support writingMode semantics.

Shall requirement (where support for any non left-to-right-top-to-bottom script is required):

Support writingMode semantics.

27.6.6 tts:overflow

EBU-TT-D only.

BBC-specific requirements apply.

Description

Defines whether a region area is clipped if the content of the region overflows the specified extent of the region. If the author intends to avoid truncated content the `tts:overflow` attribute should always be specified and be set to `"visible"` . Note that setting the value to `"visible"` does not guarantee that content that overflows the region will be presented, for example if it overflows the active video region ("root container"). See also `tts:wrapOption` .

Cardinality	0..1
BBC requirements	This attribute is required (cardinality: 1..1). The value must be set to `"visible"` .
Values	`"visible" | "hidden"`
Default value	`"hidden"`

Document Requirements

Shall requirement:

Set overflow to `"visible"` so that subtitles are visible even if they overflow.

```
<region xml:id="r0" tts:overflow="visible">
```

27.7 Content Elements

27.7.1 tt:div

Description

A logical container of subtitle text. Intended to hold semantic information, for example sections within a programme. `<div>`s may be placed in regions, which apply to the div and all its descendants. A `<div>` may have style references, which are inherited by all of its descendants except where a descendant overrides it with a different style. Begin and end times are not permitted on `<div>`s: this is a constraint in EBU-TT and EBU-TT-D rather than in TTML.

Where `<div>`s are used for semantic information, it may be specified as metadata, using attributes such as `ttm:role`, `xml:lang` etc and/or a `metadata` element.

Cardinality	1..*
Example	See code sample
Reference	TTML specification (note that EBU-TT documents do not allow temporal attributes for `tt:div`)
Presentation	Inheritable styles applied to a `tt:div` cascade to descendant elements (`tt:p` and `tt:span`).

Document Requirements

Shall requirement:

A tt:div must contain at least one tt:p element.

> Example:
>
> ```
> <tt:p xml:id="subtitle1" region="top"
> begin="00:00:30.000" end="00:00:31.000"
> style="paragraphStyle">
>   <tt:span style="spanStyle">
>   This subtitle is in the top region.
>   </tt:span>
> </tt:p>
> ```

27.7.2 tt:p

Description

Represents a logical paragraph. When reference is made to "a subtitle" it is most closely analogous to a tt:p element in general.

Any subtitle text in a <tt:p> must be within a <tt:span> element so that the background color is correctly applied.

Multiple line subtitles should be placed within a single tt:p element so that any processor that permits customisation of the size of text can scale and position all the lines together as a group rather than each line separately; when lines are in separate <tt:p> elements this can lead to gaps or alignment errors between lines.

Timing may be applied to a tt:p element using the begin and end attributes, or to each span inside the element, but in EBU-TT-D such timing must not be present in both. Cumulative subtitles, for example where words are appended at different times, should be represented by timed `<tt:span>`s within a `<tt:p>`; this approach is preferred to a set of differently timed `<tt:p>` elements each being the same as the previous but with the new word or phrase appended, because it is simpler to extract the plain text version when this approach is used.

Every `<tt:p>` is required by EBU-TT and EBU-TT-D to have an `xml:id` attribute.

Where `<tt:p>`s are used for semantic information, it may be specified as metadata, using attributes such as `ttm:role`, `xml:lang` etc and/or a `metadata` element.

Cardinality	1..*
Example	See code sample
Reference	TTML specification
Presentation	Most of the time, i.e. when not using cumulative subtitles, use the attributes `begin` and `end` on this element to control the timing and synchronisation of a block of subtitles. Note that you must not specify a background color on this element - see typography.

Document Requirements

Shall requirement:

Subtitle text (character content) must not be outside a `<tt:span>` element.

> Example:
>
> ```
> <tt:p xml:id="subtitle1" region="top" begin="00:00:30.000"
> end="00:00:31.000" style="paragraphStyle">
>   <tt:span style="spanStyle">
>   This subtitle is in the top region
>   </tt:span>
> </tt:p>
> ```

Shall requirement:

Each `tt:p` element must have an `xml:id` attribute value that is unique in the document.

> Example:
>
> ```
> <tt:p xml:id="s2874" region="top" begin="00:00:30.000"
> end="00:00:31.000" style="paragraphStyle">
>   <tt:span style="spanStyle">
>   This subtitle is in the top region
>   </tt:span>
> </tt:p>
> ```

Processor Requirements

Shall requirement:

Do not infer subtitle sequence from `xml:id`.

27.7.3 tt:span

BBC-specific requirements apply.

Description

Used to apply style information to the enclosed textual content. This style information is added to or overwrites style information from the currently active context.

Background colour must be applied to this element (rather than `<tt:p>` or `<tt:div>` so that the background is applied to the text area).

For cumulative subtitles, set begin and end time on parts of a subtitle using `<tt:span>` (see example).

EBU-TT 1.0 May include nested `<tt:span>`.

EBU-TT-D Must not include nested `<tt:span>`.

Cardinality	0..*
BBC requirements	This element is required (cardinality: 1..*). All text must be enclosed in a span that references a style to set the background colour.
Example	Background applied to `<tt:span>` (without the required line padding and with a gap between lines that should be removed by processors): XML | Image
Presentation	Use `<tt:span>` to apply colour to the text (see speaker identification and colours) and to set the background colour (see typography). For cumulative subtitles only, set `begin` and `end` on this element instead of `<tt:p>`.

Document Requirements

Shall requirement:

All subtitle text must be wrapped in a `<tt:span>` with a black background style applied.

> Example:
>
> ```
> <tt:style xml:id="spanStyle" tts:wrapOption="noWrap"
> ebutts:linePadding="0.5c"
> tts:fontFamily="proportionalSansSerif"
> tts:fontSize="100%" tts:backgroundColor="#000000" />
> ...
> <tt:p>
>   <tt:span style="spanStyle" begin="00:01:30" end="00:01:35">
>   This subtitle is displayed for 5 seconds.
>   </tt:span>
>   <tt:span style="spanStyle" begin="00.01.33" end="00:01:35">
>   This one is added after 3 and remains on screen for 2.
>   </tt:span>
> ```

```
</tt:p>
```

Processor Requirements

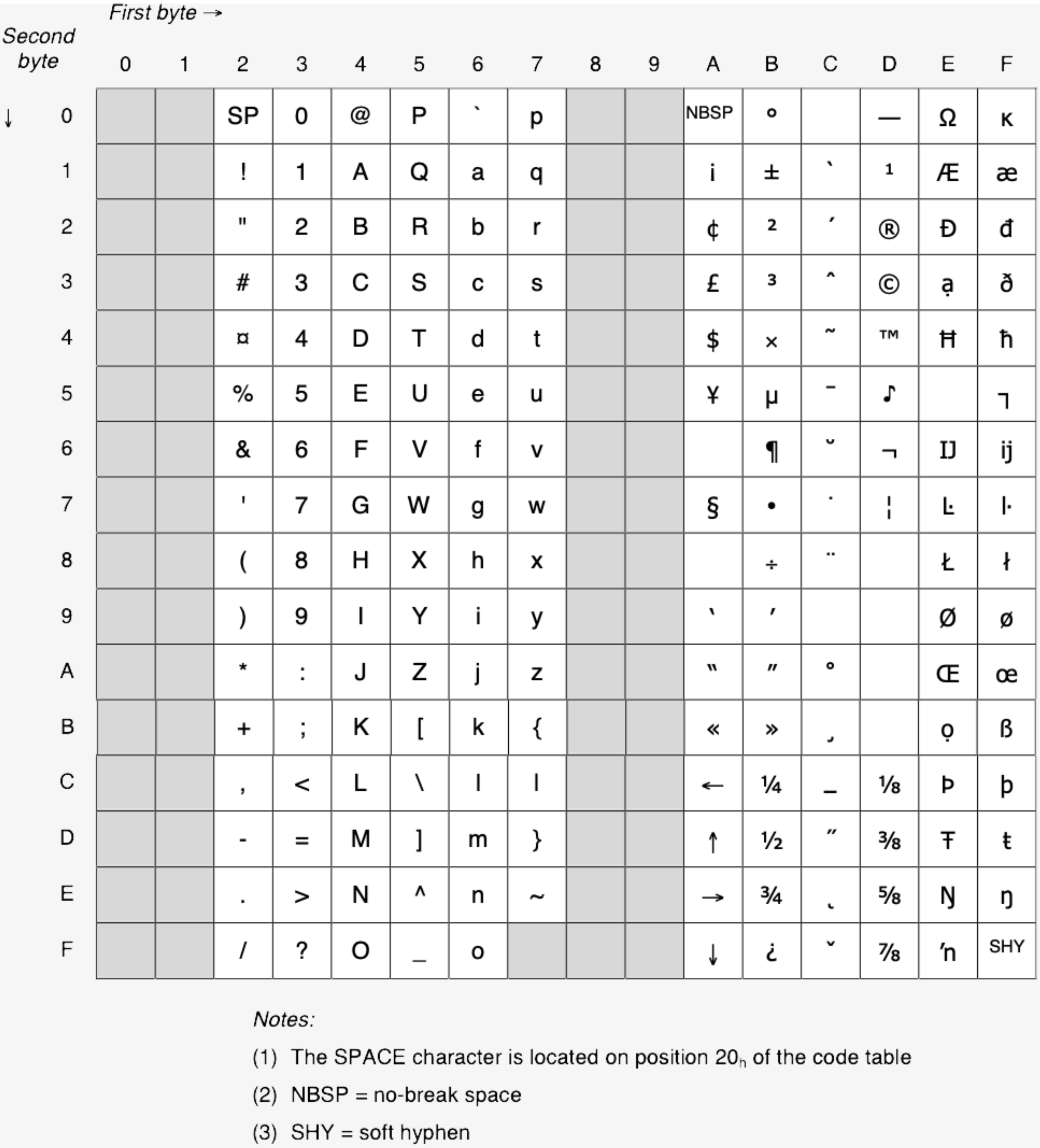
Should requirement:

For every `<tt:span>` with background applied, make the background height equal to the calculated line height regardless of other specifications. This is to help ensure no gap exists between lines.

APPENDICES

28 Appendix 1: STL and Teletext character sets

STL files must use characters in code table 00 - Latin alphabet within TTI blocks. Reproduced from EBU TECH. 3264-E.



First byte →

Second byte ↓	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			SP	0	@	P	`	p			NBSP	°		—	Ω	ĸ
1			!	1	A	Q	a	q			¡	±	`	¹	Æ	æ
2			"	2	B	R	b	r			¢	²	´	®	Đ	đ
3			#	3	C	S	c	s			£	³	ˆ	©	ạ	ð
4			¤	4	D	T	d	t			$	×	˜	™	Ħ	ħ
5			%	5	E	U	e	u			¥	µ	¯	♪		˥
6			&	6	F	V	f	v				¶	˘	¬	Ĳ	ĳ
7			'	7	G	W	g	w			§	•	˙	¦	Ŀ	ŀ
8			(	8	H	X	h	x				÷	¨		Ł	ł
9			)	9	I	Y	i	y			‘	’			Ø	ø
A			*	:	J	Z	j	z			“	”	˚		Œ	œ
B			+	;	K	[	k	{			«	»	¸		ọ	ß
C			,	<	L	\	l	|			←	¼	_	⅛	Þ	þ
D			-	=	M	]	m	}			↑	½	˝	⅜	Ŧ	ŧ
E			.	>	N	^	n	~			→	¾	˛	⅝	Ŋ	ŋ
F			/	?	O	_	o				↓	¿	ˇ	⅞	ŉ	SHY

Notes:

(1) The SPACE character is located on position 20_h of the code table

(2) NBSP = no-break space

(3) SHY = soft hyphen

Teletext output must signal and use the Latin G0 character set with English National option sub-set as defined in ETS 300 706.

Note that some mapping is required between the STL and Teletext character sets.

29 Appendix 2: Sample files

This is an example of a *prepared* subtitle file. This is not a complete file: multiple instances of elements have been removed and long values shortened. Not all possible elements are included (for example, elements required for live subtitles are not included).

Sample file: EBU-TT v1.0 pre-prepared

Sample file (raw XML) `ABCD123A02-1-preRecorded.xml`

When this sample EBU-TT file is converted for distribution as EBU-TT-D and IMSC 1 Text Profile it generates the following output:

Sample file: EBU-TT-D and IMSC 1 Text Profile distribution subtitle document.

Sample file (raw XML) `ABCD123A02-1-preRecorded.ebuttd.xml`

30 Appendix 3: BBC metadata XSD

This is the XSD for the BBC metadata section of the EBU-TT document. It includes elements for audio description and signs-language documents that can be omitted for subtitle files. To validate the document fully an EBU-TT schema should also be used.

Sample file: XML Schema Definition for BBC EBU-TT metadata

31 Appendix 4: Quick EBU-TT-D how-to

This section provides a step-by-step guide for making an EBU-TT-D file using a template for *online distribution only*. These instructions assume no prior knowledge and if followed closely will produce a valid but minimal file. You can then use this file as a basis for additional styling such elements such as colour.

Note that these instructions are for creating a *bare-bones file that does not include many of the features required by the BBC*. All subtitles will appear in white text on a black background and centred at the bottom of the screen. This minimal formatting excludes features like colour (to identify speakers), positioning (to avoid obscuring important information) and cumulative subtitles. You should therefore check with the commissioning editor that this minimal file is suitable.

This is important: Do not follow these instructions if you need to deliver subtitles for broadcast or if the presentation requires more than simple white-on-black text centred at the bottom of the screen. Consult the rest of this Guidelines document for these cases.

1. **Prepare the text.** If available, begin with a transcript file so you don't have to type in the text. Add labels if required (e.g. to describe action).

2. **Add line breaks and timings.** This is commonly done with an authoring tool. Ideally, the tool should allow you to configure all of the features that determine line length (line padding, region definition, cell resolution, font family and font size). This will allow you to preview the subtitles as reliably as possible (the final appearance will be determined by the user's system). If your tool does not support these features, use a WYSIWYG tool to define a subtitle region of 71.25% of the width of the video (for a 16:9 video). Use a wide font such as Verdana to minimise the risk of text overflowing the region when rendered in the final display font. It is not recommended to control line length using a character count limit only: this is a crude method that does not take into account the width of individual letters and fonts. Although 37 characters would fit most of the time, in some cases they might not (e.g. too many 'M's and 'W's). If you use this method you should test your subtitles in different browsers and operating systems before delivery.

   - If you don't have access to an authoring tool, you can use a simple text editor, although this method is slow and error-prone. Create a paragraph with manual line breaks for each subtitle and add timings for each paragraph. In this case you can only control line length by counting characters per line, and you should test your file thoroughly on different browsers and systems before delivery.

   Timings must be relative to a programme begin time of `00:00:00.000`

3. **Save or export the subtitles as a simple text file.** The file should include nothing but the subtitle text with line breaks and timings.

4. **Format timings.** Timings must be in the format HH:MM:SS followed by a fraction (e.g. `00:01:29.265`). In EBU-TT-D, the `begin` time of the subtitle is inclusive, but the `end` time is exclusive. This means that if you want one subtitle to follow another without any gaps, you should set the `end` time of the first subtitle to be the same as the `begin` time of the following subtitle.

5. **Format lines.** Ensure that lines are not too long and that a `<tt:br/>` tag is present for every line break within a subtitle. Remove unnecessary line breaks and white space at the beginning or end of a subtitle.

6. **Escape characters.** Replace special characters with their escaped version as detailed in Encoding characters.

7. **Create the span elements.** Wrap each subtitle in a `<tt:span>` element with a `style` attribute, so you have a list of subtitles like this:

```
<tt:span style="spanStyle">First line<tt:br/>second line</tt:spa
<tt:span style="spanStyle">This subtitle has one line</tt:span>
<tt:span style="spanStyle">Next subtitle...</tt:span>
```

8. **Create the paragraph elements.** Wrap each of the spans in a `<tt:p>` element. Each must have begin and end times and an identifier (which must begin with a letter). In this minimal example `region` and `style` attributes are fixed for all subtitles so they are set in the container `div` . The identifier must be unique for each subtitle. For the begin and end times use the timings you've prepared. You will end up with something like this:

```
<tt:p xml:id="subtitle1" begin="00:00:10.000" end="00:00:20.000'
 <tt:span style="spanStyle">First line<tt:br/>second line</tt:sp
</tt:p>
<tt:p xml:id="subtitle2" begin="00:00:20.000" end="00:00:20.748'
<tt:span style="spanStyle">This subtitle has one line</tt:span>
</tt:p>
<tt:p xml:id="subtitle3" begin="00:00:21.12" end="00:00:21.54">
<tt:span style="spanStyle">Next subtitle...</tt:span>
</tt:p>
```

9. **Place the subtitles inside a template.** Save a copy of the EBU-TT-D

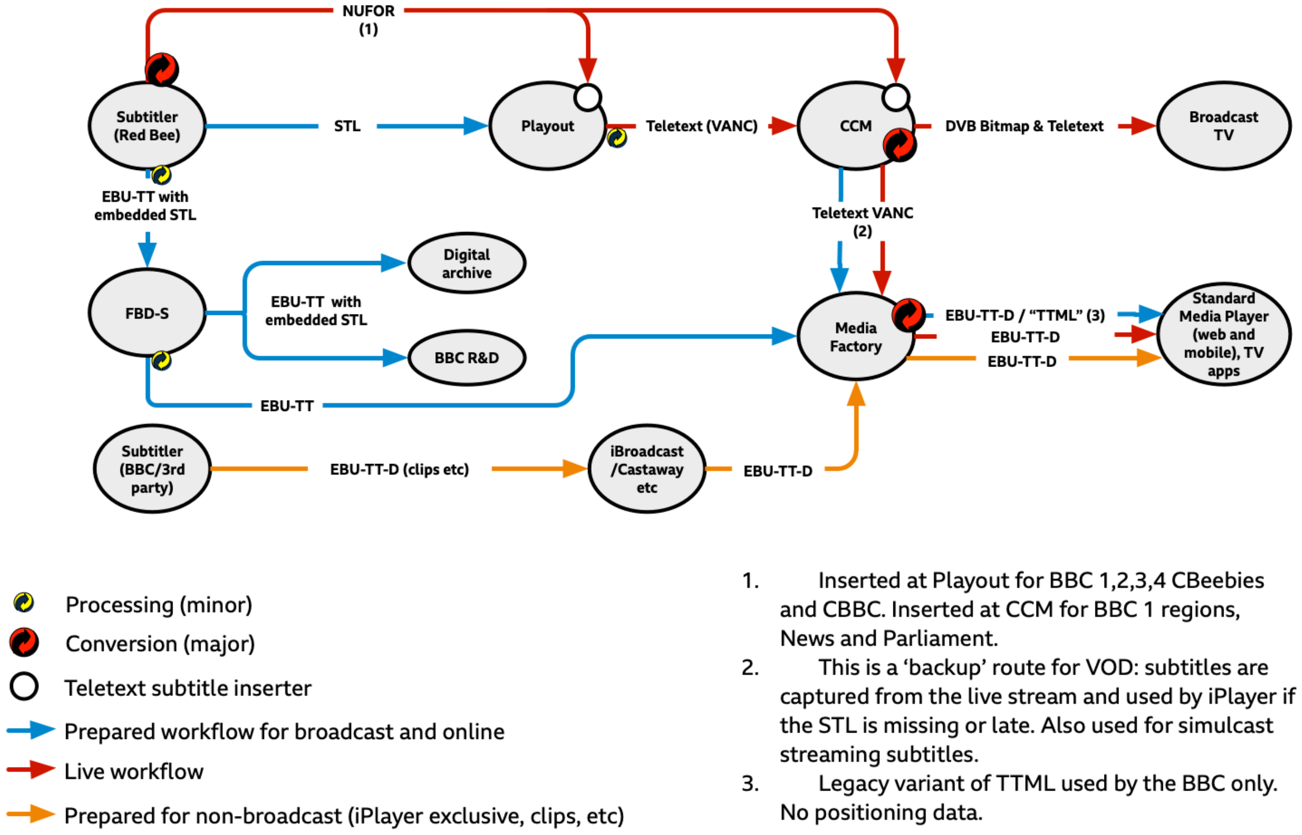
template and open it with a simple text editor (avoid word processors such as Word). Copy the list of paragraph elements you created in the previous step and paste it between `<tt:div>` and `</tt:div>`, replacing the entire comment line (from <!-- to --> inclusive).

10. **Update the copyright.** Enter the correct year in the copyright element in the template: `<ttm:copyright>BBC 2021</ttm:copyright>`

11. **Save.** Save the file with a `.ebuttd.xml` file extension. For the file name, see EBU-TT-D file.

32 Appendix 5: BBC subtitle workflows

This diagram is a high-level view of current subtitle workflows:



1. Inserted at Playout for BBC 1,2,3,4 CBeebies and CBBC. Inserted at CCM for BBC 1 regions, News and Parliament.
2. This is a 'backup' route for VOD: subtitles are captured from the live stream and used by iPlayer if the STL is missing or late. Also used for simulcast streaming subtitles.
3. Legacy variant of TTML used by the BBC only. No positioning data.

33 Appendix 6: References

- 'BBC Access Services Presentation & Style Guidelines' (internal document).

2012

- ‘bbc.co.uk Online Subtitling Editorial Guidelines V1.1’. 2009. (archived via the Wayback Machine)
- BBC R&D White Paper 287 (PDF): A Survey of UK Television Viewing Conditions. 2015
- ‘Specification of the EBU Subtitling data exchange format', TECH. 3264-E. 1991
- 'EBU-TT Part 1 Subtitling Format Definition', Tech3350 (version 1.0)
- 'EBU-TT Part 1 - Subtitle Format Definition', Tech3350 (version 1.1)
- 'EBU-TT Part 1 - Subtitle Format Definition', Tech3350 (version 1.2)
- 'EBU-TT part M, Metadata Definitions', Tech3390
- 'EBU-TT-D Subtitling Distribution Format', Tech3380
- 'Enhanced Teletext specification', ETS 300 706
- ‘ITC Guidance on Standards for Subtitling’. 1999. (Word document download)
- 'The quality of live subtitling - Improving the viewer experience'. 2013, Ofcom.
- 'TTML Profiles for Internet Media Subtitles and Captions 1.0.1 (IMSC1)'

EBU-TT-D Application Samples provided by Institut für Rundfunktechnik.

TTML examples provided by W3C Timed Text Working Group.