

1. Syntaxis

2. SubripPlus script lines

- Title
- Style definition
- Measurement unit
- Colour values

3. Basic keywords for style attributes

4. Subtitle item format

- Item number
- Time stamp format
- Tolerance
- One time stamp per subtitle item
- Line break
- Comment line between events

5. Example

6. Conversion program (windows)

- Capabilities
- Error messages
- Substation alpha import
- SubripPlus, subrip and subviewer import
- SubripPlus, subrip, substation alpha and subviewer output
- Known issues

7. Build-in styles

8. Copyright

Appendix: Inline layout codes

The latest version of this document is published at: <https://github.com/BertKerkhof/SrtPlus>
Also at that location find the latest version of the SubripPlus conversion program (windows),
as well as other documents to make you familiar with the terms and concepts used.



Emmen, September 15th 2019
kerkhof.bert@gmail.com

[1] Syntaxis

1	<p>Scripts are plain text files.</p> <p>This means they can be manually edited using any notepad text editor that can save a file in Unicode. Choose a text editor that displays line numbers the SubripPlus validator refers to script line numbers rather than subtitle item numbers. Any of the usual file line-endings are accepted. The windows conversion program writes the usual line ending sequence.</p>
2	<p>The syntax builds upon the popular Subrip (.srt) and Subviewer (.sbv) file format</p> <p>Minimal SubripPlus statements at the beginning of existing Subrip and Subviewer scripts will open up many layout features.</p>
3	<p>To begin with: the conversion program adds a default style statement to an imported subtitle</p> <p>This default statement displays the most used settings such as font characteristics and placement of the subtitle at the video.</p>
4	<p>The Default keyword is followed by a '=' mark.</p> <p>At the right side of the equation are keywords with their property values. The settings can be edited at will. Besides the displayed keywords, more than 40 are at disposal (SubripPlus) to use many layout attributes. The keywords and their meaning is described in this document.</p>
5	<p>Words in each statement are separated by spaces</p> <p>So don't use commas.</p>
6	<p>Command lines in SubripPlus cannot be split.</p> <p>Each SubripPlus style statement contains all its information in a single line.</p>
7	<p>Create an additional style with a style statement</p> <p>Numbers are allowed in style names and also a dot. Create as many styles as you like.</p>
8	<p>Created styles can be referenced in following style statements</p> <p>In this way one can build variants. It is good practice to give derived styles the same root name followed by a dot and a sub-name for specific features.</p>
9	<p>The styles are followed by subtitle items</p> <p>Each subtitle item consist of a time stamp line, followed by one or more text lines. The text lines are followed by a blank line to close the item, such as in the familiar Subrip (.srt) and SubViewer (.sbv) script.</p>
10	<p>In SubripPlus there is only one time stamp per item, the start time.</p> <p>End time is automatically calculated, based on the length of the item text. Optionally set a fixed duration within a style. SubripPlus accepts scripts with two time stamps per item for import. Usual time stamp formats with one digit for the hours and a dot as a separator for milli-seconds are accepted. The .srt+ file format uses two digits for hours and a comma as centi-seconds separator.</p>
11	<p>Directly after the time stamp, at the same line, the item can be given a layout style</p> <p>Refer to a pre-defined style with one single word. If no style is specified, the default style applies.</p>
12	<p>The SubripPlus v3 script format is different from earlier versions</p> <p>The first draft is only two weeks old and the user base is small. So no effort is done to maintain compatibility. Earlier version scripts may not process without some handwork.</p>

[2] Script lines	
SubripPlus script lines appear at the start of script, before the first subtitle text item. There are four SubripPlus language elements. All are optional:	
Title = <name>	Title refers to the video for which the subtitles are made. The keyword has to start at the beginning of a line. Is followed by an equal sign and the name. No quotes before and after the name.
Style <name> = <style definition>	The Style keyword should start at the first position of the line. It is followed by the style name, an equal sign and subsequently a style definition. The style name consists of characters and digits. One or more dots can be used. Style refers to the layout of subtitle items later on in the script.
;	Comment. A semicolon must be the first character in the line. Any text can follow. Use as much comment lines as you need. Comment is not visible when you play the subtitle script together with the video, is used in the script file only.
	Blank line. Anywhere blank lines can be added, basically to improve readability.

After the first subtitle text item, the Title and Style constructs are not recognized.

Style definition

A series of keywords and values set the layout attributes. Together they make up the style definition. Styles defined in the SubripPlus part of the script, can be referenced in the subtitle items later on. They specify the appearance and position at the video.

The keywords and values are separated by spaces. Values can be a string or a number. Strings should not be enquoted. Numbers are always integers, no fractions. The style name and also the keywords are case insensitive. A basic keyword directly sets a single attribute to a number, such as a colour value. Other keywords describe the attribute with a word, for example Blue. Also a keyword can be an earlier defined style that rules multiple elements of the appearance. Even one keyword can prepare for a complete corporate style.

Measurement unit

Attributes such as size of a font, spacing between characters, margins and also the amount of outline and shadow are dimensioned in *PixelEquivalents*. This unit of measurement is 1/720 of the video height. So a PixelEquivalent is not dependent on video resolution.

Colour values

The formats used to specify font- and other colours are of type RGB[A]. This means that optionally an RGB value can be followed by an Alpha value. RGB values are a series of three hexadecimal characters. The characters govern the intensity of red, green and blue. Zero means black and F is the highest intensity. Alpha specifies the opacity of the colour.

To state colour more precisely, the RRGGBB[AA] format is available. The colour specification is always preceded by the # hash character.

Any of the settings in the style, (except shadow/outline) can be overridden by inline layout codes, these are marks in the subtitle text. The inline control are documented in the last paragraph of this document.

[3] Basic keywords for style attributes

Font	The font name as used by the video player. On some operating systems this can be case-sensitive.
Size	The size of the font, measured in <i>pixel equivalents</i> . A PixelEquivalent is defined as 1/720 of the display video height. So when you have a video with a height of 720 pixels, a PixelEquivalent is the same as one pixel. For a 360pixel video, PixelEquivalent is equal to half a point.
Colour	<p>This is the colour that a subtitle will normally appear in. Colour is stated as RGB[A] in three or four hexadecimal numbers, preceded by the hash character '#'. This format well known from html is supported, with and without trailing hexadecimal number for the alpha channel.</p> <p>Each digit represents an amount of colour: '0' means black and 'F' is maximum intensity. With opacity, a background shows up. To state the opacity, add a fourth digit. To specify colour more precise, the #RRGGBB format is available. If you want this together with opacity, add the alpha level in two digits as #RRGGBBAA.</p>
SecondaryColour, OutlineColour, BackColour	These set the colours for several attributes.
Bold, Italic, Underscore, Strikeout	Defines whether text has this attribute (1) or not (0). You can have text which is both bold and italic and all the other mark-ups. These keywords used without a trailing value means they are on.
ScaleX, ScaleY	Modifies the width and height of the font. In percent.
Spacing	Extra space between characters. In pixel equivalents. A pixel equivalent in SubripPlus is defined as 1/720 of the video height.
Angle	Rotation of the subtitle. Between -180 and 180 degrees.
BorderStyle, Transparent, Opaque	Defines whether text is displayed with outline and dropshadow (1, transparent) or in an opaque box (3). Keywords Opaque and Transparent are shortcut keywords that have the same function. The last ones don't need a trailing value although 1 (on) and 0 (off) are available.
Outline, Shadow	If BorderStyle is transparent, then Outline specifies the width of the outline around the text. Shadow specifies the amount of drop shadow. In pixel equivalents. A pixel equivalent is 1/720 of the video height.
xAlign, yAlign, Align	This sets how text is justified within the onscreen margins. xAlign values may be 1=Left, 2=Centered, 3=Right. yAlign values are 1=bottom, 2=mid, 3= top Keyword Align is a legacy keyword from earlier implementations and is a combination of xAlign and yAlign.
MarginL, MarginR, MarginV	Defines the left, right and vertical margin in pixel equivalents. It is the distance from the edge. This defines the area in which the subtitle text will be displayed. For a mid-title, the value of MarginV is ignored.

Encoding	Not used in version 3. SubripPlus is based on unicode.
Layer	Normally zero. A non-zero value means the viewer can optionally choose to display subtitles that are styled in that layer. An extra layer can be used to display additional subtitles for specific target audiences such as the hearing impaired.
Name	The name of the character who speaks the dialogue. An application can have additional meaning for this field.
Effect	<p>Transition Effect. Either empty, or contains information for one of the three transition effects implemented with SubripPlus.</p> <p>The effect names are case sensitive and must appear exactly as shown. The effect names do not have quote marks around them.</p> <p>"Karaoke" means that the text will be successively highlighted one word at a time.</p> <p>"Scroll up;y1;y2;delay[:fadeawayheight]" means that the text/picture will scroll up. The parameters after the words "Scroll up" are separated by semicolons.</p> <p>The y1 and y2 values define a vertical region on the video in which the text will scroll. The values are in pixels, and it doesn't matter which value (top or bottom) comes first. If the values are zeroes then the text will scroll up the full height of the video.</p> <p>The delay value can be a number from 1 to 100, and it slows down the speed of the scrolling - zero means no delay and the scrolling will be as fast as possible.</p> <p>"Banner;delay" means that text will be forced into a single line, regardless of length, and scrolled from right to left across the video.</p> <p>The delay value can be a number from 1 to 100, and it slows down the speed of the scrolling - zero means no delay and the scrolling will be as fast as possible.</p> <p>"Scroll down;y1;y2;delay[:fadeawayheight]"</p> <p>"Banner;delay[:lefttoright;fadeawaywidth]"</p> <p>lefttoright 0 or 1. This field is optional. Default value is 0 to make it backwards compatible.</p> <p>When delay is greater than 0, moving one pixel will take (1000/delay) second.</p> <p>(WARNING: Avery Lee's "subtitler" plugin reads the "Scroll up" effect parameters as delay;y1;y2)</p> <p>fadeawayheight and fadeawaywidth parameters can be used to make the scrolling text at the sides transparent.</p>
Duration	This sets the display time of a subtitle item to a fixed duration. In centi-seconds.

[4] Subtitle text item format

The subtitle items are called *events*. The events follow the Title and Style specifications. One script can have as much subtitle items as needed.

Item number

Subtitles have no item number. Error messages reference the line number of the script. For compatibility reasons, SubripPlus can import Subrip scripts with item numbers. In scripts written with SubripPlus, these are disappeared.

The hh:mm:ss,cc time stamp format

Each event starts with a time-stamp that specify hours, minutes, seconds and centi-seconds. They define the time at which the text will appear onscreen, relative to the start of the video.

Note that SubripPlus uses two digits for the hours. For compatibility reasons, the SubripPlus standard is tolerant enough to read other formats, see the table. SubripPlus writing is always with two digits for the hours, a comma as fraction separator and two digits for the centi-seconds.

Tolerance for various time stamp formats				
Format	Digits for hours	Fraction separator	Fraction digits	Time stamp separator
SubripPlus (*srt+)	2	comma	2	none (single stamp)
Subviewer (*sbv)	2	dot	2	comma
Subrip (*srt)	2	comma	3	--> (vector)
Substation (*ass, *.ssa)	2	dot	2	comma

One time stamp per subtitle item

In draft version 3 of the SubripPlus standard, the end-time to display the item is calculated by artificial intelligence, based on the text length and a minimum display time. For styles in layer zero, the display time is limited by the start time of the next item. Items in styles with a higher layer can display concurrently.

For compatibility reasons the SubripPlus standard has the ability to read scripts with two time stamps. In that case, the two time-stamps are separated by the characters ' --> ' like in the popular subrip format or just a comma such as in the Subviewer format. The information in the second stamp is overided by the robot. Many practical samples prove that robotics do a better job to optimize readability.

Item style specification

On the same line the time stamp can be followed by a single style name, which defines how the event is displayed. Each style refers to a style that is predefined at the start of the script. Note that only one style name per event is allowed. If no style is specified, the Default style is used. The default style used by the script author is stored in the script.

Line break

Next lines contain the text of the event, like in the subrib (*.srt) subtitles format. Each event ends with a blank line. When importing subviewer scripts, the subviewer specific [br] linebreak is replaced by a line break that is common for the operating system at hand, such as CRLF for windows, LF for linux, CR for apple. Each subtitle item closes with a blank line.

Comment line between events

Comments between subtitle items are allowed directly above the timestamp line. A comment line starts with a ; semicolon character. When translating subtitles, comment lines can be used for supervision.

[5] Example of a SubripPlus script:

; This is a SubripPlus v3 created srt+ script
; (c) SubripPlus is designed by Bert Kerkhof
; Add your comment

Title = Red (2010).dutch

Style Default = font Arial size 20 red transparent outline 2 shadow 1 bottom center

Style Heading = size 80 red mid center

00:00:55,40 Heading
Red (2010)

00:01:49,40
943-66-2291,
Afdeling Pensioenen?

00:01:54,86
{\i1}Mr Moses, uw contactpersoon{\i0}
{\i1}komt eraan.{\i0}

00:02:15,34
Hallo, met Sarah.
- Met Frank Moses.

00:02:18,93
Frank, hoe gaat het?

00:02:21,97
Stil.

00:02:25,14
Ik bel omdat ze die cheque
weer zijn vergeten.

00:02:30,23
Ongelofelijk dat ze 't nog
steeds niet hebben geregeld.

00:02:33,31
Ik stuur een nieuwe.

00:02:35,69
{\i1}Het spijt me vreselijk.{\i0}
- Ach, wat doe je eraan?

00:02:41,61
Hoe gaat 't met je avocado?

[6] Conversion program

The SubripPlus conversion program also functions as subtitle format validator, so can be used to test the syntax of various script format.

Conversion and syntax validation program v3 capabilities			
	Import	Export	Syntax validation
SubripPlus (*.srt+)	Yes	Yes	Yes
Subrib (*.srt)	Yes	Yes	Yes
Substation alpha v4+ (*.ass)	Yes	Yes	Yes
Substation alpha v4 (*.ssa)	Yes	No	Yes
Subviewer (*.sbv)	Yes	No	Yes

Error messages	
Substation alpha import	
1	<i>Missing section</i> Three sections of the substation alpha scripts have data for import in SubripPlus. The exact spelling of the section headers is: [Script Info] [V4 Styles] or [V4+ Styles] [Events] All of these sections are optional and won't stop the validation process. After correcting of the spelling, the sections will be read.
2	<i>Title: is missing</i> Enter a Title in the Script Info section with the Title: command
3	<i>PlayResY: missing, 288 is assumed</i> With the assumed video height, generally the subtitles will display too big. Enter the correct video height with a PlayResY: command
4	<i>Unrecognized style lines</i> The command lines displayed not have the valid syntax and are ignored. It may have to do with the spelling. The line it concerns is displayed together with a script line number.

	Correcting the error will solve the problem
5	<i>Format: statement missing</i> In the V4+ Styles section, enter a Format: statement to specify field names. In the absence of the specification, a commonly used list will be used.
6	<i>Invalid field: <name></i> The line with the style Format statement is shown, together with the script line number. With an invalid field name, a field will be ignored. Correcting the fieldname will solve the problem.
7	<i>Style: Default is missing</i> It is assumed that every substation script has a Default style and that it is the first one. If not so, a default with commonly used values will be used.
8	<i>Colour value ignored <value></i> Instead a default for this field is assumed.
9	<i>Out of range <value> for <keyword></i> For the attributes there is a minimum and maximum value. The value is out of boundaries.
10	<i>Format: statement is missing</i> Like in the V4+ styles section, the Events section needs a Format statement It is to specify subtitle item elements to be read. In the Event section, enter a Format: statement to specify field names. In the absence, a commonly used list will be used.
11	<i>Required event field Start / Test / Style missing</i> The line with the erroneous event Format statement is shown. After the field is included in the list, the subtitle items will be read
12	<i>Unknown event field <name></i> This field in the Format list is ignored so its data are not imported The line with the unknow field in the event Format statement is shown. Enter a correct field name in accordance with a name listed in the V4 Styles section Format statement.
13	<i>Unknown event style <name></i> The style name in a subtitle Dialogue line does not occur in the Style collection defined in the V4+ Styles section. For this subtitle the Default style is assumed. The line with the unknown style is displayed together with the script line number. Correct the error by choosing the correct style name or add style data to the V4+ Styles section.
SubripPlus, subrip and subviewer import	
1	<i>Unrecognized style lines</i> The SubripPlus lines displayed not have the valid syntax and are ignored. It may have to do with the spelling. The line it concerns is displayed, together with a script line number. Correct the error and the problem is solved.
2	<i>Unknown event style <name></i> The style name in a subtitle test line does not occur in the Style collection defined at the start of the script file and is not a build in SubripPlus style.

	<p>If it happens, the Default style is assumed.</p> <p>The line with the unknown style is displayed together with the script line number.</p> <p>Correct the error by choosing the correct style name or add style data to script.</p>
SubripPlus, subrip, substation alpha and subviewer export	
	Since all error checking is done at import, no error or error message is expected in the export module.

Known issues for version 3 as of Friday September 13th 2019	
1	<p><i>Missing font colour conversion</i></p> <p>Inline html layout and <\font> specifications in import subrip (.srt) files are not converted to the SubripPlus layout code and vice versa.</p>
2	<p><i>No check on the validity of inline layout codes.</i></p> <p>Only correctly formatted tags will convert properly.</p>
3	<p><i>Missing upscaling from inline codes to item wide layout style</i></p> <p>Inline layout codes for Bold, Italic, Underline and/or StrikeOut, if they rule the appearance of the complete item, are not converted to SubripPlus styles. Only html codes are converted to similar inline SubripPlus ones.</p>
4	<p><i>No fixed duration in subrip export</i></p> <p>In subrip (.srt) export, a Duration that can be defined per style, has no effect on the calculated end-time written. When using the *.srt output, a warning is issued.</p>
5	<p><i>Only Unicode import</i></p> <p>Scripts written in regional 8bit character code are rejected.</p> <p>An Encode value other than zero stated in a substation alpha 4 or 4+ import file, has no effect.</p>
6	<p><i>Slow SubripPlus import</i></p> <p>The conversion process takes a few seconds.</p>
Updates will be published at: https://github.com/BertKerkhof/SrtPlus	

[7] Build-in styles

SubripPlus comes with build-in shortcuts. They intent to give words to values, as words are more easy to memorize. These are not yet the last words to form a collection of templates and certainly not enough to make up a corporate identity corporate. It is just a begin.

	Style name	Definition
	left	xalign 1
	center	xalign 2
	right	xalign 3
	bottom	yalign 1
	mid	yalign 2
	top	yalign 3
	opaque	borderstyle 1
	transparent	borderstyle 0
	white	colour #FFFFFF00
	black	colour #00000000
	red	colour #FF000000
	green	colour #00FF0000
	blue	colour #0000FF00
	cyan	colour #00FFFF00
	yellow	colour #FFFF0000
	purple	colour #FF00FF00

[8] License

Copyright 2019 Bert Kerkhof

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Appendix: Inline layout codes

The intention of SubripPlus is to generate the layout per event, to prevent from cumbersome tasks. However in some parts of a subtitle, one wants a more detailed layout. This is not at all impossible with SubripPlus. In fact, today's video players such as VLC and Media Player Classic and also modern television devices support more layout than yet is contained in styles. This paragraph may be useful for those of you who wish to learn the inline style override codes, so you can type them in manually. It may be the route to design new styles to apply with events. Please report your findings and send in your developed styles!

All Override codes appear within braces { } except the newline \n and \N codes.

All override codes are always preceded by a backslash \

Several overrides can be used within one set of braces.

Each override affects all text following the override. To apply an override only to selected text, you need a second "cancelling" override after the selected text, to "undo" the effect of the first override.

Some overrides automatically apply to all the text - currently this is just alignment overrides, but more may be added later (for example shadow/outline overrides).

Override code	Parameters	Description
n		Soft line break - the line <i>may</i> break here.
N		Hard / forced line break
h		Non-breaking space
b i u s	0 1	Bold / italic / underline / strikethrough off (0) or on (1)
bord	width	Border
shad	depth	Shadow
be	blur amount	Outline blur ("blur edges")
fn	Font name	Changes the font.
fs	size	Font size
fscx fscy	Percent	Font scaling
fsp	pixels	Font spacing
fr/frz frx fry	degrees	Rotation around the x/y/z-axes.
fe	charset	Encoding number
c/1c 2c 3c 4c	colour	Primary / secondary / outline / shadow colour
alpha/1a 2a 3a 4a	alpha value	Primary / secondary / outline / shadow transparency
an	alignment code	Alignment
k kf/K ko	duration (x0.01s)	The duration of time, in centiseconds, the succeeding text will be displayed, until the end of the line or the next k override code. This creates a karaoke effect.
q	integer	Wrapping style

r	[style]	Resets all previous override styles, applying the default style. To apply a different style, put it as a parameter.
t	([<start>,<end>],[<accel>,<styles>])	Applies a transition animation to the line. The start and end times are in milliseconds, and the styles can be any combination of the available animatable styles.
pos	(x,y)	Sets the position of the subtitle anchor, where 0,0 is the top-left corner. Note that the subtitle anchor also depends on the line's alignment.
move	(x1,y1,x2,y2[,t1,t2])	Animates the subtitle position at the specified time or at the entirety of the subtitle time from point 1 to point 2. Time is in milliseconds, relative to the beginning of the subtitle. The subtitle anchor depends on the line's alignment.
org	(x,y)	Relocates the rotation origin of the subtitle. By default, the subtitle rotates about its anchor.
fade	a1,a2,t1,t2,t3,t4	Creates a three-part fade animation.
fad	fade in duration, fade out duration	Sets a fade-in and fade-out duration. 0 will effectively disable the effect.
clip	x1,y1,x2,y2	Clips a subtitle inside a rectangle.
clip	[scale,] drawing	Clips a subtitle inside a drawing.

--