Substation Beta (SSB) v1

A typesetting-oriented subtitle format

File Format & Syntax

File encoding: Text-only, UTF-8
Line breaks: Windows (*CRLF*) & Unix (*LF*)
Data layout: Line-based
Ignore empty lines: Yes
Comments: Lines beginning with **//**
Syntax error warnings: Depends on renderer implementation

For more, follow **Sections**.

Sections

SBB scripts are separated in sections for different properties.
Sections begin with a header. It's a name, starting with character **#**.
Valid sections are...
META
Meta information.
FRAME
Target frame size.
STYLES
Styles to complement render data.
EVENTS
Render data & condition.

Meta

The meta section contains some side informations and has nothing to do what's rendered at the end. It's just interesting for editors.
Fields begin with a name, followed by "**:** " and the value.
Valid fields for meta are...
Title
Script title.
Author
Script author.
Description
Script description.
Version
Script version.

Frame

The frame section contains size informations for the target frame.
It's just relevant in the case that a script is made with a different video size than the video to render on later. On difference, rendering data will be scaled to fit.
Fields begin with a name, followed by "**:** " and the value.
Valid fields for frame are...
Width
Target frame width.
Height
Target frame height.

Styles

The styles section contains styles to prepend or insert text in-/to event text.
In fact, styles are just shortcuts for content, avoiding redundancy.
Styles begin with an unique identifier name, followed by "**:** " and the associated text.

Geometries

Geometries are the render source. Their appearance is influenced by styling tags.

Events

The events section contains rendering data and is the core of this format.
Each line is structured this way:

<START_TIME>-<END_TIME>|<STYLE>|<NOTE>|<TEXT>

<START_TIME> and **<END_TIME>** are timestamps for a video time range when to render.

Different geometry types are usable:

Text

Plain text. Line breaks are to write as **\n**, **{** are to escape with ****.

Example: *Hello world!\n\{Hallo Welt!}*

Points

Floating point number pairs as center coordinates for pixels and circles.

Example: *0 100 -50 -.125*

Path

Description for a 2D graphics path. Segments of one type begin with a specifier, followed by necessary values.

Type	Specifier	Values per segment
Movement	m	1 target point
Line	l	1 end point
Bezier curve	b	2 control points + 1 end point
Arc	a	1 angle center point + 1 angle (in degree)
Close	c	-

Example: *m 0 0 l 100 0 100 100.5 b 50 200 0 100 0 20 a 30 30 -45.5 c*

Timestamp structure from right to left:

milliseconds -> point -> seconds -> colon -> minutes -> colon -> hours

Units don't have to be written completely, so following two examples are valid:

12:3:4.56

0

Hours limit is 99, but that should be enough for nearly all purposes.

<STYLE> is a style identifier name, mentioned in **Styles**. The content of the choosen style will be prepended to **<TEXT>**.

<NOTE> is just a note for editors, nothing more.

<TEXT> is the description what to render. It's a combination of styling tags and geometries. Tags are enclosed by brackets **{}**, single ones parted by **;**, everything else are geometries. Additionally, content of styles can be inserted by **\\<STYLE_NAME>\\**. Insertions are limited by the renderer implementation.

Tags

Tags define styling parameters for geometries or the type of themself. They begin with a type name, mostly followed by **=** and one or more values.

Font

font-family=**Arial**

Font family.

font-style=

Font weigth, style & decoration. Value contains properties as characters: b (bold), i (italic), u (underline), s (strikeout).

font-size=**30**

Font size.

font-space=**0,0**

font-space-h=**0**

font-space-v=**0**

Font space between characters and lines.

Lines

line-width=**2**

Line width.

line-style=**round,round**

Line join & cap style. Join can be round or bevel, cap can be round or flat.

line-dash=**0**

Line dash pattern. Starts with an offset, followed by comma separated pattern (distances as numbers).

Geometry

geometry=**text**

Geometry type. Can be text, points or path.

Transformation

identity

Reset of transformation to identity.

translate=**0,0**

translate-x=**0**

translate-y=**0**

Prepend translation to transformation.

scale=**1**

scale=**1,1**

scale-x=**1**

scale-y=**1**

Prepend scale to transformation.

rotate-xy=**0,0**

rotate-yx=**0,0**

rotate-z=**0**

Prepend rotation on axis to transformation.

shear=**0,0**

shear-x=**0**

shear-y=**0**

Prepend shearing in horizontal & vertical direction to transformation.

transform=**1,0,0,1,0,0**

Multiplies current transformation with given 2x3 matrix.

xx 1)	xy 3)	x0 5)

Rastering

blend=**over**

Blending mode. Can be over, add, sub, mult, screen or differ.

blur=**0**

blur=**0,0**

blur-h=**0**

blur-v=**0**

Blurring strength in horizontal & vertical direction.

stencil=**off**

Stencil mode. Can be off, set, unset, in or out.

Animation

fade=**0**

fade=**0,0**

fade-in=**0**

fade-out=**0**

In- & outfade durations.

animate=**()**

animate=**0,0,()**

animate=**t,()**

animate=**0,0,t,()**

Animation by interpolating style properties over time.

Versions:

1) Interpolates from previous style properties to given style properties in round brackets.

mode=**fill**

Painting mode. Can be fill, wire or boxed.

deform=**x,y**

Geometry line points filter by two math formulas. Available variables are x (horizontal coordinate), y (vertical coordinate) and t (time progess in range 0-1). First formula for the new horizontal, second for the new vertical coordinate.

Position

position=

Geometry x- & y-coordinate.

align=**2**

Geometry alignment relativ to position, value orientated on keyboard numpad.

margin=**10**
margin=**10,10**
margin-h=**10**
margin-v=**10**

Margin to screen border.

direction=**ltr**

Geometry flow direction. Can be ltr (left-to-right), rtl (right-to-left) and ttb (top-to-bottom).

yx 2)	yy 4)	y0 6)

Color

color=**ffffff**
color=**ffffff,ffffff**
color=**ffffff,ffffff,ffffff,ffffff**

Solid, left-to-right or 4-corner gradient color for geometry filling in hexadecimal.

alpha=**ff**
alpha=**ff,ff**
alpha=**ff,ff,ff,ff**

Solid, left-to-right or 4-corner gradient opacity for geometry filling in hexadecimal.

texture=

Image for geometry filling.
(On animation, the progress status as integer will be appended to image filename.)

texfill=**0,0,clamp**

Image position and wrap mode. Wrap modes are clamp, repeat, mirror and flow.

line-color=**000000**

Solid color for lines.

line-alpha=**ff**

Solid opacity for lines.

2) Like last, but with time range.
3) Like first, but beginning with a math formula to edit the progress status. t is an available variable with the current progress in range 0-1.
4) Combination of second and third.

Karaoke

k=

Sylable duration. Increases next sylable start time.

kset=

Sets current sylable start time.

kcolor=**ff0000**

Sets active sylable's color.

kmode=**fill**

Sets active sylable's filling mode. Can be fill, solid or glow.

* Sizes are in pixels, angles in degree, times in milliseconds.

Example

```
#META
Title: My work
Author: Me
Description: Insert your description here.
Version: 1.0 (01.01.2014)

#FRAME
Width: 1280
Height: 720

#STYLES
Default: {align=5;font-size=40;font-style=b}

#EVENTS
0-10.0|Default|Write a note here.|{color=00ff00,0000ff;animate=(scale=2.5)}Hello world{color=ff0000}!
```