

# LiveText

Mit der LiveText Schnittstelle können dynamische Elemente wie Spielstände, Zeiten oder Wettquoten in Videos eingebettet und live dargestellt werden.

- Allgemeine Informationen
- Einrichtung eines Tags
- Ansteuerung des Textes
- KINEXON Connector

# Allgemeine Informationen

Mit dieser Funktion haben Sie die Möglichkeit einen oder mehrere Texte von außen auf der Bande live darzustellen und zu ändern.

Beispiele aus der Praxis sind unter anderem:

- Spielstände
- Uhrzeiten
- Wettquoten
- statistische Spielauswertungen
- dynamischer Content wie z.B. Twitter

Die Darstellung wird über einen zusätzlichen Layer über der jeweiligen Werbung realisiert. Das funktioniert sowohl über statischen Grafiken, also auch über Animationen.

LEDVIZ selbst führt hier intern eine Liste an Tags und zugehörigen Texten. Immer wenn der Text geändert wird, wird dieser an die Ausgabe weitergegeben. Die Ausgabe erzeugt eine transparente Textur und legt diese als Layer über die jeweiligen Grafik.

Die Liste an Tags wird global geführt und ist immer auf eine Grafik oder Animation bezogen. Egal wo in der Ausgabe diese Grafik eingesetzt wird, wird der Layer erstellt und dargestellt. Für eine Grafik oder Animation können gleichzeitig mehrere Tags erzeugt werden. Die Menge der Tags für die LiveText Layer ist nur durch die Performance des jeweiligen Rechners begrenzt. In der Praxis wurden hier bereits Sequenzen mit mehr als 25 LiveText Ausgaben in der Fussball-Bundesliga, gesteuert durch einen Twitter/Instagram-Crawler, realisiert.

Dadurch das sich die Darstellung auf Text beschränkt, wird die Darstellung von Bildern oder Emojis zur Zeit noch nicht unterstützt.

# Einrichtung eines Tags

Aktuell gibt es noch keine Einrichtungsmöglichkeit über LEDVIZ selbst, so dass die Tags für den LiveText Layer zur Zeit noch manuell im XML erfasst werden müssen.

Der Definitionsblock muss sich durch <TAGS> geklammert auf der obersten Ebene, genauso wie z.B. die Helligkeitszonen oder Segmente befinden. Hier ein Beispiel:

```
<TAGS>
  <TAG>
    <TAG-FILE>BNN_LIVE_Premium_15min.avi</TAG-FILE>
    <TAG-NAME>#TAG1</TAG-NAME>
    <TAG-VALUE>2: 0</TAG-VALUE>
    <TAG-DEFAULT>WILLKOMMEN</TAG-DEFAULT>
    <TAG-FONT>Tahoma</TAG-FONT>
    <TAG-SIZE>50</TAG-SIZE>
    <TAG-COLOR-RED>255</TAG-COLOR-RED>
    <TAG-COLOR-GREEN>255</TAG-COLOR-GREEN>
    <TAG-COLOR-BLUE>255</TAG-COLOR-BLUE>
    <TAG-BOLD>1</TAG-BOLD>
    <TAG-ITALIC>0</TAG-ITALIC>
    <TAG-CAPITALIZE>0</TAG-CAPITALIZE>
    <TAG-COUNT>4</TAG-COUNT>
    <TAG-Y>10</TAG-Y>
    <TAG-X>1024</TAG-X>
    <TAG-X2>2048</TAG-X2>
    <TAG-X3>4096</TAG-X3>
    <TAG-X4>8192</TAG-X4>
    <TAG-LINE-WIDTH>2550</TAG-LINE-WIDTH>
    <TAG-TIME-IN>2</TAG-TIME-IN>
    <TAG-TIME-OUT>20</TAG-TIME-OUT>
    <TAG-ALIGN>center</TAG-ALIGN>
  </TAG>
  ... (evtl. weitere Tags) ...
</TAGS>
```

Die einzelnen Untertags sind relativ selbsterklärend, werden aber zur Sicherheit hier nochmal aufgeführt:

|                                |                                |   |
|--------------------------------|--------------------------------|---|
| <b>&lt;TAG-FILE&gt;</b>        | (string)<br>z.B. "werbung.avi" | Der Name der Datei für die der Text-Layer erstellt werden soll  |
| <b>&lt;TAG-NAME&gt;</b>        | (string)<br>z.B. "TAG1"        | Der Name des Tags mit dem der Tag von außen her angesprochen wird. Dieser Name muss eindeutig sein und kann nur für diesen Tag verwendet werden.  |
| <b>&lt;TAG-VALUE&gt;</b>       | (string)<br>z.B. "2:0"         | Dieser Wert enthält den aktuellen gespeicherten Wert und muss nicht angegeben werden, es sei denn Sie möchten den LiveText Wert vorbelegen  |
| <b>&lt;TAG-DEFAULT&gt;</b>     | (string)<br>z.B. "WILLKOMMEN"  | Der Wert dieses Tags wird immer dann verwendet, wenn ein leerer Wert mitgeteilt wurde, also im Prinzip der Wert <TAG-VALUE> nicht belegt ist  |
| <b>&lt;TAG-FONT&gt;</b>        | (string)<br>z.B. "Tahoma"      | Der Windowsname der zu verwendenden Schriftart. Es muss hierbei der Name der Schriftart verwendet werden. Den genauen Namen können Sie den Installierten Schriftarten der Windowsinstallation entnehmen. Der Dateiname der Schriftart spielt hier keine Rolle, das Tag bezieht sich auf den Namen unter dem die Schriftart in Windows registriert ist |
| <b>&lt;TAG-SIZE&gt;</b>        | (integer)<br>z.B. 50           | Der Wert enthält die zu verwendende Schriftgröße. Zur besseren Lesbarkeit sollten hier Werte von >50 verwendet werden   |
| <b>&lt;TAG-COLOR-RED&gt;</b>   | (integer)<br>0 bis 255         | Der rote Farbwert für den zu verwendenden Text.   |
| <b>&lt;TAG-COLOR-GREEN&gt;</b> | (integer)<br>0 bis 255         | Der grüne Farbanteil  |
| <b>&lt;TAG-COLOR-BLUE&gt;</b>  | (integer)<br>0 bis 255         | Der blaue Farbanteil  |
| <b>&lt;TAG-BOLD&gt;</b>        | (bool)<br>z.B. 0 oder 1        | Bestimmt ob der Text fett oder normal gezeichnet wird.  |
| <b>&lt;TAG-ITALIC&gt;</b>      | (bool)<br>z.B. 0 oder 1        | Bestimmt ob der Text kursiv gezeichnet wird.  |

|   |  |  |
|---|--|--|
| <b>&lt;TAG-CAPITALIZE&gt;</b>   | (bool)<br>z.B. 0 oder 1                | Bestimmt ob der gesamt Text in Großbuchstaben umgewandelt werden soll.   |
| <b>&lt;TAG-COUNT&gt;</b>  | (integer)<br>0 bis 4                   | gibt die Anzahl der zu berücksichtigen <TAG-X?> Werte an.  |
| <b>&lt;TAG-X&gt;</b><br><b>&lt;TAG-X2&gt;</b><br><b>&lt;TAG-X3&gt;</b><br><b>&lt;TAG-X4&gt;</b> | (integer)                              | Dieser Wert bestimmt die horizontale Position in Pixeln innerhalb der Grafik. Der Wert 0 befindetets sich hierbei links.   |
| <b>&lt;TAG-Y&gt;</b>  | (integer)                              | Dieser Wert bestimmt die vertikale Position in Pixeln innerhalb der Grafik. Der Wert 0 befindetets sich hierbei oben. Bitte beachten Sie hierbei das die Schriftarten in der Regel etwas Freiraum oberhalb der Buchstaben besitzen und der Text nicht direkt pixelgenau an der Position beginnt. |
| <b>&lt;TAG-LINE-WIDTH&gt;</b>   | (integer)                              | Dieser Wert gibt die maximale Breite des Textes an.<br>Ist dieser Wert größer 0, dann wird der Text bei Erreichen des Wertes automatisch umgebrochen. Die Schrifgröße wird dann entsprechend angepaßt und verkleinert.   |
| <b>&lt;TAG-TIME-IN&gt;</b>  | (decimal)<br>z.B. 15.4                 | Dieser Wert bestimmt den Zeitpunkt, wann der Text während des Abspielens erscheinen soll. Der Wert repräsentiert die Zeit in Sekunden, wobei hier Dezimale Werte mit einem Punkt als Dezimalzeichen zulässig sind.   |
| <b>&lt;TAG-TIME-OUT&gt;</b>   | (decimal)<br>z.B. 15.4                 | Dieser Wert bestimmt den Zeitpunkt, wann der Text während des Abspielens ausgeblendet werden soll.   |
| <b>&lt;TAG-ALIGN&gt;</b>  | (string)<br>z.B. left, right, center   | Der Text kann an den durch <TAG-X> Koordinaten vorgegebenen Wert ausgerichtet werden. Zulässige Wert sind left, right und center   |
| <b>&lt;TAG-TYPE&gt;</b>   | (string)<br>z.B. text, progress, image | Der Standardwert ist Text, allerdings kann hier auch definiert werden, das es sich um ein Bild handelt. Der Wert in <TAG-VALUE> und <TAG-DEFAULT> wird dann als Verweis auf ein Bild interpretiert.  |
| <b>&lt;TAG-WIDTH&gt;</b><br><b>&lt;TAG-HEIGHT&gt;</b>   | (integer)                              | Größer der Ausgabe des referenzierten Bildes im Fall des <TAG-TYPE> "image"  |

|                         |                                |   |
|-------------------------|--------------------------------|---|
| <b>&lt;TAG-MIDI&gt;</b> | <code>(string),(string)</code> | Sollte eine Sequenz über einen Midi-Key für einen Aufruf verfügen, kann dieser Tags angewiesen werden, zusätzlich zum Dateinamen nur bei dieser Midi-Sequenzen angezeigt zu werden. Mehrere Midi-Keys können getrennt durch Komma eingegeben werden |
|-------------------------|--------------------------------|---|

Bitte beachten das der Renderer Zeichen, wie Return oder doppelte Leerzeichen vor der Darstellung entfernt. Um die Grafik zu positionieren empfiehlt es sich zuerst mit einem Wert von 0 für <TAG-TIME-IN> und einem hohen Wert, z.B. 999 für <TAG-TIME-OUT> zu beginnen und erst später die Abstimmung der Ein und Ausblendzeiten vorzunehmen.

# Ansteuerung des Textes

## Ansteuerung über UDP Schnittstelle

LEDVIZ hält hierfür permanent einen UDP Port offen. Alle Anfragen der Art beginnen mit einem Hashtag gefolgt von dem Tag-Namen. Die Trennung des Textes erfolgt durch eine Pipe |. Ein Beispiel eines Textes könnte also wie folgt aussehen:

```
|#TAG1| Hello world! |
```

Wird ein Tag-Element für einen LiveText empfangen wird dieses direkt an die Ausgabe weitergegeben. Das hat zur Folge das der Layer auch sofort aktualisiert wird. Der Text kann also während seiner Laufzeit jederzeit und mehrfach, während er ausgespielt wird, aktualisiert werden.

Die UDP Daten werden auf Port **28077** im UTF8 Format erwartet.

Aktuell wird noch an einer Eingabemaske unter Nutzung der bestehenden REST API gearbeitet. Neben dem Call über die RestAPI ist geplant eine Webseite mit verfügbaren Tags und Eingabe direkt als HTML Seite zur Verfügung zu stellen.

## Ansteuerung über den LEDVIZ

### Operational HUB

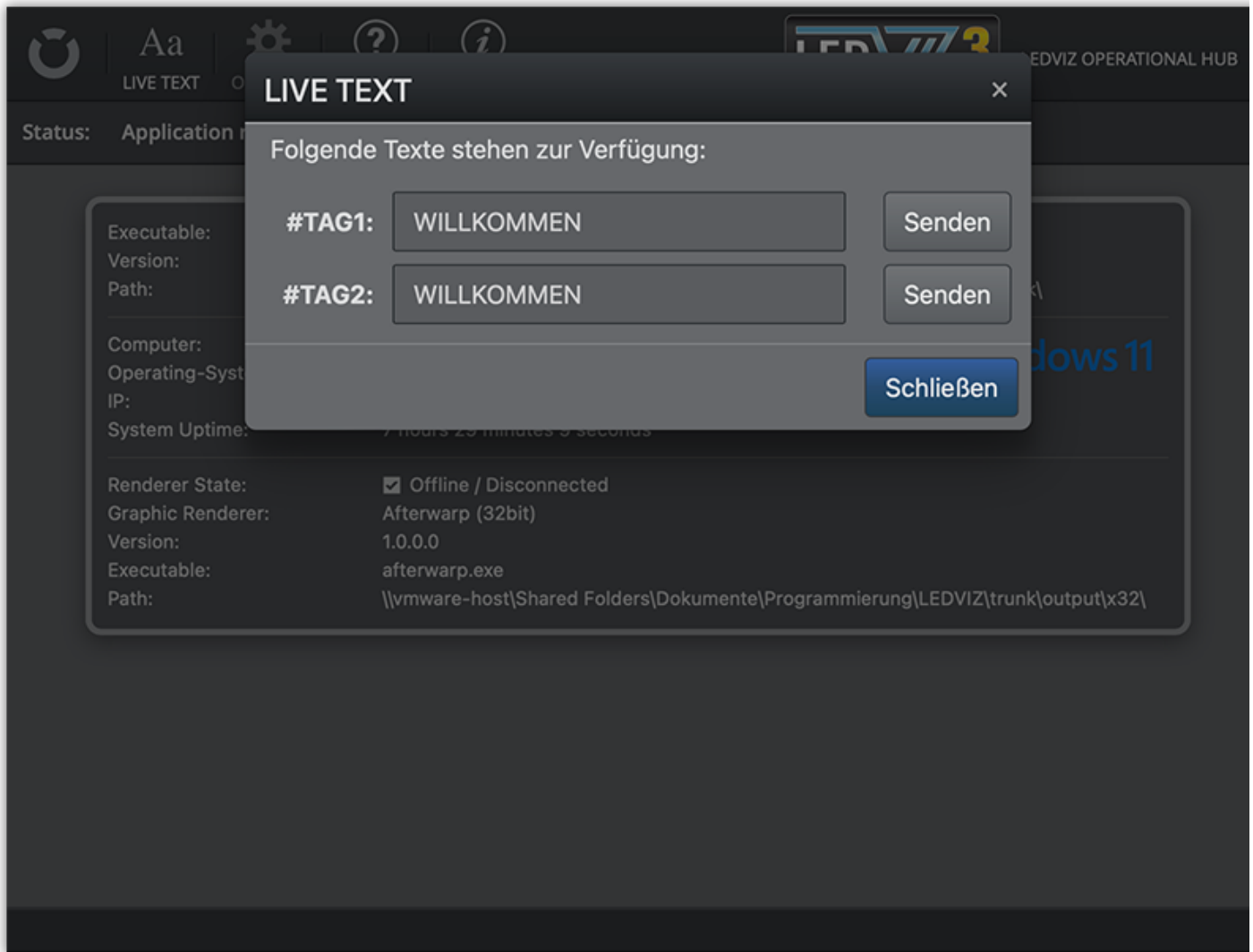
Der LEDVIZ Operational Hub ist eine Webseite die Funktionen von LEDVIZ als Webseite bereitstellt. Zugriff auf den Hub erfolgt über den Aufruf <http://localhost:19399>. Anstelle von localhost kann auch der spezifische Rechnername oder die IP angegeben werden. Somit ist eine Remote Eingabe der Texte über dritte Systeme jederzeit möglich.

Nach dem Aufruf des LEDVIZ Operational Hub im Browser sollte folgende Seite angezeigt werden:

The screenshot displays the LEDVIZ 3 Operational Hub interface. At the top, there is a navigation bar with icons for 'LIVE TEXT', 'OPTIONS', 'HELP', and 'ABOUT', along with the 'LEDVIZ 3' logo and the text 'LEDVIZ OPERATIONAL HUB'. Below the navigation bar, the status is shown as 'Application running and connected ...'. The main content area features a rounded rectangle containing system and application information:

|                   |  |
|-------------------|--|
| Executable:       | LEDVIZ.exe   |
| Version:          | 3.2.0.0  |
| Path:             | \\vmware-host\Shared Folders\Dokumente\Programmierung\LEDVIZ\trunk\            |
| Computer:         | DESKTOP-1QPV5DT  |
| Operating-System: | Windows 11   |
| IP:               | 192.168.67.142   |
| System Uptime:    | 7 hours 7 minutes 19 seconds   |
| Renderer State:   | <input checked="" type="checkbox"/> Offline / Disconnected                     |
| Graphic Renderer: | Afterwarp (32bit)  |
| Version:          | 1.0.0.0  |
| Executable:       | afterwarp.exe  |
| Path:             | \\vmware-host\Shared Folders\Dokumente\Programmierung\LEDVIZ\trunk\output\x32\ |

Hier kann über den Menüpunkt oben Links ein dialog für die Eingabe von Texten geöffnet werden. Es werden nur die Tags gelistet, die auch im aktuellen Projekt enthalten sind. Eingaben können hier direkt vorgenommen und gesendet werden.



## Sonstige Tools

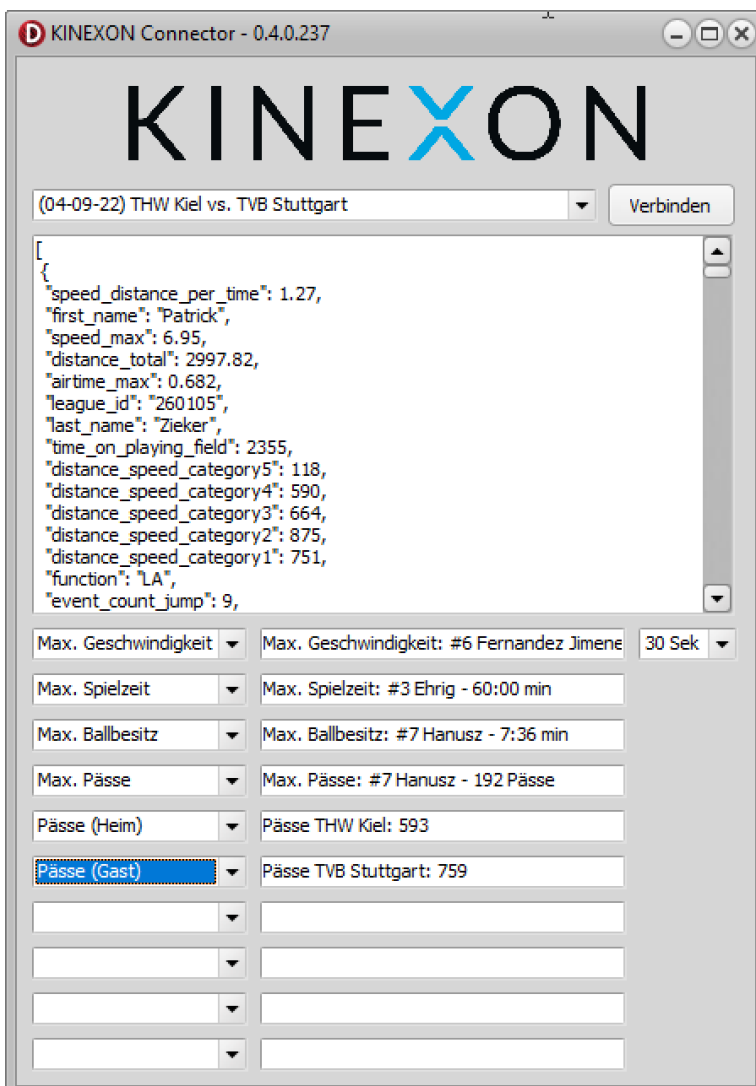
Zum Testen der Schnittstelle kann ein kleines Tool verwendet werden, das ursprünglich für die Bereitstellung von Wettquoten für ein Trainingsspiel des 1.FC Köln gegen den Watford FC erstellt wurde: <https://www.ledviz.de/download/LEDVIZlive.zip>. Das Tool verwendet die Tags:

`#QUOTE- WIN- FCK` | `#QUOTE- WIN- WTF` | `#QUOTE- TOR- FCK` | `#QUOTE- TOR- WTF`

# KINEXON Connector

Die Daten für die LiveText Schnittstelle können aus verschiedenen Quellen kommen. Unter anderem wurde bereits eine Schnittstelle zu Kinexon realisiert. Kinexon sammelt über Tracker verschiedenste statistische Daten z.B. in der Handball-Bundesliga.

Der Connector dient hier als Vermittler zwischen Kinexon und LEDVIZ. Die Daten werden in Realtime von der Kinexon Schnittstelle abgefragt und als Tags im LiveText Format an LEDVIZ weitergegeben. Hierbei werden die Daten aufbereitet und in Textform umformatiert. Ausgangspunkt sind die abgerufenen Daten im JSON Format der Kinexon REST-API.



The screenshot shows the KINEXON Connector application window. The title bar reads "KINEXON Connector - 0.4.0.237". The main window features the KINEXON logo at the top. Below the logo, there is a dropdown menu showing the match "(04-09-22) THW Kiel vs. TVB Stuttgart" and a "Verbinden" button. A large text area displays a JSON object with the following data:

```
{
  "speed_distance_per_time": 1.27,
  "first_name": "Patrick",
  "speed_max": 6.95,
  "distance_total": 2997.82,
  "airtime_max": 0.682,
  "league_id": "260105",
  "last_name": "Zieker",
  "time_on_playing_field": 2355,
  "distance_speed_category5": 118,
  "distance_speed_category4": 590,
  "distance_speed_category3": 664,
  "distance_speed_category2": 875,
  "distance_speed_category1": 751,
  "function": "LA",
  "event_count_jump": 9,
}
```

Below the JSON feed, there are several statistics displayed in a table-like format:

|                      |   |        |
|----------------------|---|--------|
| Max. Geschwindigkeit | Max. Geschwindigkeit: #6 Fernandez Jimene | 30 Sek |
| Max. Spielzeit       | Max. Spielzeit: #3 Ehrig - 60:00 min      |        |
| Max. Ballbesitz      | Max. Ballbesitz: #7 Hanusz - 7:36 min     |        |
| Max. Pässe           | Max. Pässe: #7 Hanusz - 192 Pässe         |        |
| Pässe (Heim)         | Pässe THW Kiel: 593                       |        |
| Pässe (Gast)         | Pässe TVB Stuttgart: 759                  |        |
|                      |   |        |
|                      |   |        |
|                      |   |        |
|                      |   |        |