

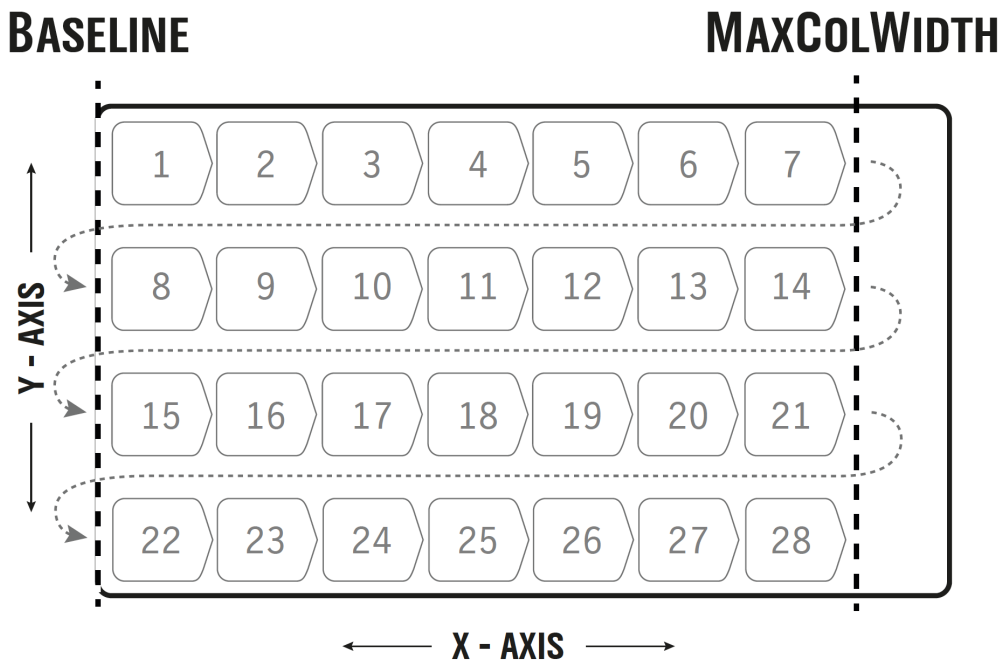
XML Projektdatei

Am Anfang einer jeden Konfiguration muss ein Mapping erstellt werden, also eine Konfiguration die dem System sagt, welchen Banner er wo darstellen muss. Für die Basics reicht der Dialog beim Erstellen eines neuen Projektes in der Regel aus. Es gibt aber Konfigurationen die z.B. über mehrere Bildschirme gehen. Hier muss man manuell in die Konfiguration eingreifen. Die folgenden Seiten geben einen Überblick über die Verfügbaren TAGS in der XML Datei und ihre Auswirkungen

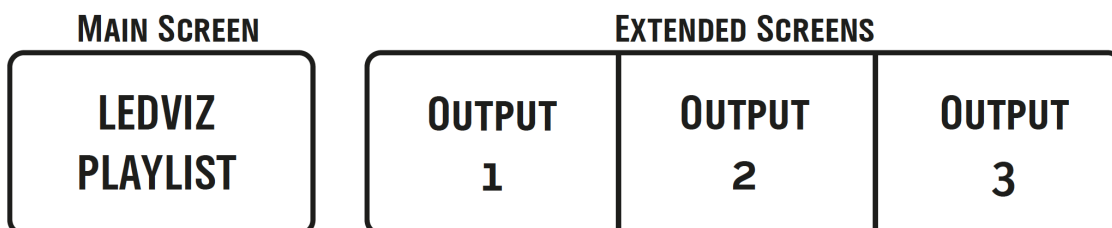
- [Individuelles XML Mapping](#)

Individuelles XML Mapping

Die Position von LED-Panels wird normalerweise automatisch berechnet. Auf die linke Seite ist die Grundlinie. Nach dem Erstellen einer Datei ist es immer Null ($x = 0$). Auf der rechten Seite wird es auf den errechneten rechten Rand begrenzt. Dieser Wert ist sichtbar in jedem XML unter dem Tag `<MAXCOLWIDTH>`



Um eine Konfiguration mit mehreren Bildschirmen zu handhaben, muss das Ausgabefenster auf einem oder mehreren erweiterten Bildschirmen so angepasst werden, dass es alle Bildschirme abdeckt. Da sich die Ausgabe immer über ein Fenster erstreckt, müssen hierzu die erweiterten Bildschirme nebeneinander in einer Reihenfolge von links nach rechts angeordnet werden. Diese Einstellungen können Sie in den Anzeigeeinstellungen von Windows vorgenommen werden.



Als Hardwarelösung kann hier eine Matrox TripleHead2Go oder DualHead2Go zum Einsatz kommen. Durch die Extender von Matrox wird die Auflösung an einem DVI / HDMI oder DSP Anschluss erweitert und anschließend wieder in einzelne Bildschirm Signale aufgeteilt. Alternativ, wenn die Grafikkarte mehrere Ausgänge hat, kann das Ausgabefenster von LEDVIZ auch in der Höhe oder Breite geändert werden, so dass es die komplette Ausgabe ausfüllt.

Ausgabebereiche neu Positionieren

Jede Anzeigefläche ist mit der linken Seite an der vorherigen Anzeigefläche ausgerichtet. Bei der Berechnung der Position wird der Y-Wert, also die vertikale Positionierung, solange beibehalten bis ein Zeilenumbruch erfolgen muss. Mit dem Zeilenumbruch wird die Anzeige um seine Höhe nach unten verschoben und wieder neu links mit $X = 0$ angesetzt. Es gibt grundsätzlich 3 Werte um eine Anzeigefläche direkt an eine andere Position zu bewegen:

<code><POSX> ? </POSX></code>	Weißt der Anzeigefläche einen neuen horizontalen X Wert zu
<code><POSY> ? </POSY></code>	Weißt der Anzeigefläche einen neuen vertikalen Y Wert zu
<code><BASEX> ? </BASEX></code>	Weißt der Berechnungsroutine einen neuen horizontalen X Wert für die linke Seite im Fall eines Zeilenumbruchs zu

Bitte beachten Sie, dass die XML Tags immer Großgeschrieben werden. Wenn die Werte einer Anzeigefläche geändert wurde, sind diese dann die Ausgangsbasis für alle weiteren Berechnungen, d.h. das folgende Element wird dort wieder rechts angesetzt usw. Die rechte Seite für den Zeilenumbruch wird dabei immer über folgende Formel berechnet:

$$\text{Rechte Seite} = \text{<BASEX>} + \text{<MAXCOLWIDTH>}$$

Das bedeutet, wenn Sie den `<BASEX>` Wert ändern, ändert sich automatisch auch immer die Position der rechten Seite, die den Zeilenumbruch erzeugt. Sie können den `<BASEX>` Wert so oft ändern wie Sie möchten, es gibt hier keine Begrenzungen.

Erzeugen eines Umbruchs in einer Anzeigefläche

Manchmal ist es notwendig, die Ausgabe einer Anzeigefläche zu unterbrechen und an einer anderen Stelle fortzusetzen. Einen Umbruch innerhalb einer Anzeigefläche kann durch folgenden XML Tag erzwungen werden:

<code><BREAK> ? </BREAK></code>	Definiert den Umbruch innerhalb einer Anzeigefläche nach definierter Anzahl an Pixeln
---	---

Nur der Umbruch selbst reicht an der Stelle aber noch nicht aus. Im gleichen Schritt müssen nun auch neue Koordinaten zum Fortsetzen der Ausgabe angegeben werden:

<code><BREAKX> ? </BREAKX></code>	Weißt einen neuen horizontalen X Wert für das Fortsetzen der Anzeigefläche zu
---	---

<code><BREAKY> ? </BREAKY></code>	Weißt einen neuen vertikalen Y Wert für das Fortsetzen der Anzeigenfläche zu
---	--

Ebenfalls kann hier auch direkt gleich die Zuweisung eines neuen `<BASEX>` Wertes vorgenommen werden, dann ab der umzubrechenden Stelle zu verwenden ist:

<code><BREAKBASEX> ? </BREAKBASEX></code>	Zuweisung eines neuen <code><BASEX></code> Wertes durch den Umbruch
---	---

Auch unterstützt werden bis zu 2 weitere Umbrüche:

<code><BREAK2> ? </BREAK2></code>	Definiert einen zweiten Umbruch innerhalb einer Anzeigenfläche nach definierten Anzahl an Pixeln
<code><BREAK2X> ? </BREAK2X></code>	Weißt einen neuen horizontalen X Wert für das fortsetzen der Anzeigenfläche zu
<code><BREAK2Y> ? </BREAK2Y></code>	Weißt einen neuen vertikalen Y Wert für das Fortsetzen der Anzeigenfläche zu

bzw.

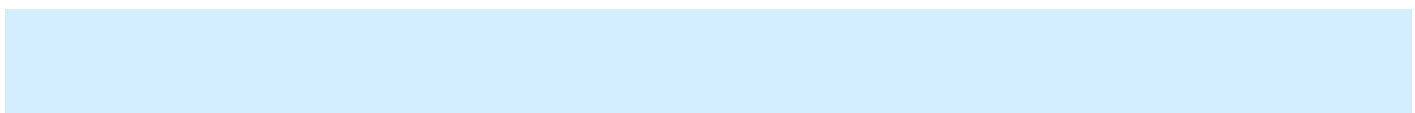
<code><BREAK3> ? </BREAK3></code>	Definiert einen dritten Umbruch innerhalb einer Anzeigenfläche nach definierten Anzahl an Pixeln
<code><BREAK3X> ? </BREAK3X></code>	Weißt einen neuen horizontalen X Wert für das fortsetzen der Anzeigenfläche zu
<code><BREAK3Y> ? </BREAK3Y></code>	Weißt einen neuen vertikalen Y Wert für das Fortsetzen der Anzeigenfläche zu

Erzeugen eines vorzeitigen Zeilenumbruchs

Für den Fall das die Ausgabe nicht am Standardwert des Zeilenumbruches geändert werden soll, sonder z.B. früher oder auch später, kann einer Anzeigenfläche ein individueller Rechter Rand für den Umbruch mitgeteilt werden.

<code><MAXX> ? </MAXX></code>	Zuweisung eines neuen Wertes für einen Zeilenumbruch
---	--

Dieser `<MAXX>` Wert gilt aber ausschließlich für das jeweilige Element das diesen Wert beinhaltet. Alle anderen Anzeigenflächen richten sie weiterhin nach den Standardwerten aus em XML



Wenn früher in der Zeile umgebrochen werden soll, und die anderen Werte wie <BASEX> sich nicht ändern, kann damit oft auch der einfache <BREAK> ersetzt werden. Wenn möglich, sollte dem <MAXX> Wert immer der Vorzug vor einem <BREAK> gegeben werden!

Sonderfälle

Da <POSX> oder <POSY> Werte mit einem 0 Wert, nicht angegeben werden müssen, gibt es einen Fall der zu einem logischen Problem bei der Verarbeitung der XML führt. Es ist kein direkter Rücksprung auf die Koordinaten $X = 0$ und $Y = 0$ möglich. In diesem Fall kann der Workaround angewendet werden und der <BASEX> Wert auf -1 und der <POSX> auf +1 gestellt werden. Mit der darauf folgenden Anzeigenfläche, können über die Definition des genauen <POSX> Wert diese dann wieder richtig gestellt werden. Es muss immer mindestens ein Wert von <POSX> oder <POSY> oder <BASEX> größer oder kleiner 0 angeben sein damit diese 3 Werte verarbeitet von LEDVIZ verarbeitet werden.

XML Beispiele

```
<SEGMENT1 id="1" >
  □<POSX>1920</POSX>
  □<POSY>0</POSY>
  □<BASEX>1920</BASEX>
  □<NAME>Panel 2</NAME>
  □<WIDTH>1080</WIDTH>
  □<HEIGHT>60</HEIGHT>
</SEGMENT1>
```

oder

```
<SEGMENT4 id="4" >
  □<BREAK>512</BREAK>
  □<BREAK>1920</BREAK>
  □<BREAKY>0</BREAKY>
  □<BREAKBASEX>1920</BREAKBASEX>
  □<NAME>Panel 5</NAME>
  □<WIDTH>1080</WIDTH>
  □<HEIGHT>60</HEIGHT>
</SEGMENT4>
```