

## ARD-PLAYER

### *DOKUMENTATION PIXELUNG*

<b>Autor</b>	Klaus Panster
<b>Version</b>	0.7
<b>Stand</b>	11. Juni 2014
<b>Dateiname</b>	ARD-Player-Dokumentation-Pixelung-0.5.docx
<b>Ablageort</b>	„Release-Paket“ /docs/

## VERSIONIERUNG

---

Version	Datum	Status	Autor(en)	Beschreibung
0.1	25.11.2013	In Arbeit	Klaus Panster	<ul style="list-style-type: none"> <li>▪ <b>Erstellung des Dokuments</b></li> </ul>
0.2	09.12.2013	In Arbeit	Florian Diesner	<ul style="list-style-type: none"> <li>▪ <b>Qualitätswechsel nun bei Mausevents hinterlegt</b></li> <li>▪ <b>Qualitätswechsel per Tastatur aufgenommen</b></li> </ul>
0.3	07.02.2014	In Arbeit	Florian Diesner	<ul style="list-style-type: none"> <li>▪ <b>Super-Event für Timeshift aufgenommen</b></li> </ul>
0.4	28.01.2014	In Arbeit	Florian Diesner	<ul style="list-style-type: none"> <li>▪ <b>Super-Event für erstmaliges Abspielen</b></li> </ul>
0.5	25.03.2014	In Arbeit	Klaus Panster	<ul style="list-style-type: none"> <li>▪ <b>Parameterübergabe STREAM_LOAD</b></li> <li>▪ <b>Umbenennung PixelController in PlayerPixelController</b></li> </ul>
0.6	16.05.2014	In Arbeit	Klaus Panster	<ul style="list-style-type: none"> <li>▪ <b>Hinweis zur Schließung der Sicherheitslücke im PlayerPixelController</b></li> </ul>
0.7	11.06.2014	In Arbeit	Florian Diesner	<ul style="list-style-type: none"> <li>▪ <b>Super-Events für Qualitätswechsel, Zurück zu Live und Pluginwechsel</b></li> </ul>

# INHALTSVERZEICHNIS

---

<b>1</b>	<b>EINFÜHRUNG .....</b>	<b>4</b>
1.1	Allgemein .....	4
1.2	Der PlayerPixelController .....	4
<b>2</b>	<b>INITIALISIERUNG UND KONFIGURATION .....</b>	<b>5</b>
2.1	Installation .....	5
2.2	Allgemein .....	5
2.3	Einbindung per binding .....	5
2.3.1	Initialisierung.....	5
2.4	Einbindung per JSON .....	6
2.4.1	Initialisierung.....	6
2.4.2	Konfiguration.....	6
<b>3</b>	<b>PIXEL-EVENTS.....</b>	<b>7</b>
3.1	Standard-Pixel .....	7
3.1.1	Tastatur-Pixel .....	7
3.1.2	Maus-Pixel.....	7
3.1.3	Status-Pixel.....	8
3.2	Super-Pixel .....	8
<b>4</b>	<b>ANWENDUNGSBEISPIELE.....</b>	<b>10</b>

# 1 EINFÜHRUNG

---

## 1.1 Allgemein

Im vorliegenden Dokument werden die unterschiedlichen Vorgehensweisen zum Initialisieren und Konfigurieren der Pixelung des ARD-Players beschrieben. Die Pixelung kann auf zwei unterschiedlichen Wegen dem Player übergeben werden. So steht neben der programmatischen Methode, auch die Initialisierung über ein JSON-Objekt zur Verfügung. In den jeweiligen Pixel-Events können mehrere Methodenaufrufe gleichzeitig angestoßen werden.

## 1.2 Der PlayerPixelController

Der PlayerPixelController sammelt und bündelt alle geworfenen Player-Events und gibt diese über eine Schnittstelle nach außen weiter. Es ist somit möglich, zentral Player-Events abzugreifen und für den jeweiligen Zweck der Seite (Pixelung) zu verwenden.

Der Klasse „PlayerPixelController“ werden alle Player-Eigenschaften bei der Instanziierung mit übergeben, sodass beim Hören auf Events auch gesetzte Parameter oder Eigenschaften des Players innerhalb der Methode übergeben und abgefragt werden können.

## 2 INITIALISIERUNG UND KONFIGURATION

---

Im nachfolgenden Abschnitt werden die benötigten Schritte erläutert, um die Pixelung in den Player zu integrieren. Bitte beachten Sie, dass es sich hierbei lediglich um einen Leitfaden handelt.

### 2.1 Installation

Für die Pixelung ist die Datei `PlayerPixelController.js` exemplarisch so in die Website einzubinden:

```
<script  
src="base/js/com/netTrek/gundl/ardplayer/controller/PlayerPixelController.js"></script>
```

### 2.2 Allgemein

Das Listening des Players auf die Pixel-Events ist so aufgebaut, dass auf beide Implementierungswege parallel gehört werden kann. So ist es z.B. möglich, eine Standardkonfiguration auf JSON-Basis vorzunehmen, aber zusätzlich auf der Website auf dem gleichen Event eine andere Funktion zu binden. Wird das Event geworfen, so werden dann beide gebundenen Funktionen aufgerufen.

### 2.3 Einbindung per binding

#### 2.3.1 Initialisierung

Nach der Initialisierung des Players erfolgt die programmatische Einbindung des/der Listener(s) für das jeweilige Pixel-Event.

##### BEISPIEL

```
$(p).bind(PlayerPixelController.INTERACTION_INITIAL_PLAY, function (event) {  
    console.log("BINDING: " + PlayerPixelController.INTERACTION_INITIAL_PLAY);  
});
```

## 2.4 Einbindung per JSON

### Achtung:

**Ab der Version 3.6.2** ist aufgrund der Schließung einer Sicherheitslücke die Pixelung via **JSON** nicht mehr möglich!

### 2.4.1 Initialisierung

Nach der Instanziierung der **PlayerConfiguration** erfolgt der die Initialisierung der Pixelung per JSON und Übergabe an die PlayerConfiguration.

#### METHODE

```
void setPixelConfig = function (val)
```

#### PARAMETER

```
URL      Typ: String, relativerPfad zur Konfigurationsdatei
```

#### BESPIEL

```
pc.setPixelConfig("test/conf/pixel.js");
```

### 2.4.2 Konfiguration

Bei der Verwendung der JSON-Konfiguration werden JSON-Objekte in Form von Funktionen übergeben. Diese werden im Ajax-Request als String eingelesen und per eval() entsprechend neu interpretiert.

Einerseits erhöht sich somit die Variabilität der zentralen Verwaltung der Events, andererseits ist jedoch bei der Einbindung der Funktionen vom Entwickler absolute Präzision gefordert, um die korrekte Funktion des Players zu gewährleisten!

Um die Konformität des W3C zu gewährleisten, ist es zwingend erforderlich, die Datei mit den JSON-Objekten als **JavaScript-Datei (\*.js)** zu referenzieren!

#### BEISPIEL

```
(
  {
    player_loading: function (params) {
      console.log("PIXEL_load");
    }
  }
)
```

## 3 PIXEL-EVENTS

Alle pixelbaren Events sind direkt in der Klasse PlayerPixelController hinterlegt und werden durch entsprechende Strings referenziert.

### 3.1 Standard-Pixel

Die Standard-Pixel unterscheiden sich in Tastatur-Pixel, Maus-Pixel und Status-Pixel.

#### 3.1.1 Tastatur-Pixel

Tastatur-Pixel werden nur dann aktiviert, wenn das Event durch eine Tasteneingabe ausgelöst wurde.

Event	Beschreibung	Parameter
PlayerPixelController.IA_KEY_SUBTITLE_ACTIVATION	Aktivierung UT - Taste U	
PlayerPixelController.IA_KEY_SUBTITLE_DEACTIVATION	Dektivierung UT - Taste U	
PlayerPixelController.IA_KEY_VOLUME_MUTE	Ton aus - Taste M	
PlayerPixelController.IA_KEY_VOLUME_UNMUTE	Ton an - Taste M	
PlayerPixelController.IA_KEY_VOLUME_CHANGE	Lautstärkeänderung - Pfeiltasten hoch/runter	volume
PlayerPixelController.IA_KEY_SCRUBBAR_CHANGE_POSITION_FORWARD	Zeitleiste Sprung vorwärts - Pfeiltaste rechts	
PlayerPixelController.IA_KEY_SCRUBBAR_CHANGE_POSITION_REWIND	Zeitleiste Sprung rückwärts - Pfeiltaste links	
PlayerPixelController.IA_KEY_SCRUBBAR_CHANGE_POSITION_REWIND_TIMESHIFT	Zeitleiste Sprung rückwärts bei Timeshift – Pfeiltaste links	
PlayerPixelController.IA_KEY_PLAY	Play - Leertaste	
PlayerPixelController.IA_KEY_PAUSE	Pause – Leertaste	
PlayerPixelController.IA_KEY_PLAY_AGAIN	Erneut Abspielen – Leertaste	
PlayerPixelController.IA_KEY_QUALITY_CHANGE	Qualitätswechsel per Tastatur (1-4)	

#### 3.1.2 Maus-Pixel

Maus-Pixel werden nur dann aktiviert, wenn das Event durch ein Maus-Event (Klick) ausgelöst wurde.

Event	Beschreibung	Parameter
PlayerPixelController.IA_MOUSE_SUBTITLE_ACTIVATION	Aktivierung UT - Klick	
PlayerPixelController.IA_MOUSE_SUBTITLE_DEACTIVATION	Dektivierung UT – Klick	
PlayerPixelController.IA_MOUSE_VOLUME_MUTE	Ton aus – Klick	
PlayerPixelController.IA_MOUSE_VOLUME_UNMUTE	Ton an - Klick	
PlayerPixelController.IA_MOUSE_VOLUME_CHANGE	Lautstärkeänderung - Klick	volume
PlayerPixelController.IA_MOUSE_SCRUBBAR_CHANGE_POSITION_FORWARD	Zeitleiste Sprung vorwärts - Mausbewegung	
PlayerPixelController.IA_MOUSE_SCRUBBAR_CHANGE_POSITION_REWIND	Zeitleiste Sprung zurück - Mausbewegung	
PlayerPixelController.IA_MOUSE_SCRUBBAR_CHANGE_POSITION_REWIND_TIMESHIFT	Zeitleiste Sprung rückwärts bei Timeshift – Mausbewegung	
PlayerPixelController.IA_MOUSE_VIEW_COLOR_SETTINGS	Änderungen der Farbeinstellungen – Klick	
PlayerPixelController.IA_MOUSE_FULLSCREEN_ACTIVATION_BUTTON	Vollbild starten – Klick Button	
PlayerPixelController.IA_MOUSE_FULLSCREEN_ACTIVATION_VIEWPORT	Vollbild starten – Doppelklick auf Viewport	
PlayerPixelController.IA_MOUSE_FULLSCREEN_DEACTIVATION_BUTTON	Vollbild beenden – Klick Button	
PlayerPixelController.IA_MOUSE_FULLSCREEN_DEACTIVATION_VIEWPORT	Vollbild beenden – Doppelklick auf Viewport	
PlayerPixelController.IA_MOUSE_PLAY_BUTTON	Play – Klick Button	
PlayerPixelController.IA_MOUSE_PLAY_VIEWPORT	Play – Klick Viewport	
PlayerPixelController.IA_MOUSE_PAUSE_BUTTON	Pause – Klick Button	
PlayerPixelController.IA_MOUSE_PAUSE_VIEWPORT	Pause – Klick Viewport	
PlayerPixelController.IA_MOUSE_PLAY_AGAIN_BUTTON	Erneut abspielen – Klick Button	
PlayerPixelController.IA_MOUSE_PLAY_AGAIN_VIEWPORT	Erneut abspielen – Klick Viewport	
PlayerPixelController.IA_MOUSE_PLUGIN_CHANGE	Plugin-Wechsel (Flash/HTML5) – Klick	
PlayerPixelController.IA_MOUSE_PODCAST	Podcast – Klick	

### 3.1.3 Status-Pixel

Event	Beschreibung	Parameter
PlayerPixelController.STREAM_LOAD	Stream/Clip wurde geladen	Streamurl, isQualityChanged, isPluginChanged
PlayerPixelController.INTERACTION_INITIAL_PLAY	Stream/Clip wird zum ersten Mal nach Klick auf den Posterframe abgespielt. Hinweis: Nur bei autoplay=false. Alternativ ist der Super-Pixel SUPER_INITIAL_PLAY zu verwenden.	
PlayerPixelController.CLIP_ENDED	Stream/Clip ist am Ende	
PlayerPixelController.CLIP_ERROR	Fehler im Player – mit angezeigter Fehlermeldung	message
PlayerPixelController.PLAYER_LOADING	Player wird geladen / initialisiert	
PlayerPixelController.PLAYER_UNLOAD	Player wird ausgeworfen / entfernt	

## 3.2 Super-Pixel

Super-Pixel inkludieren alle Aufrufweisen der Pixel-Events. So wird das Super-Pixel z.B. beim Ausschalten der Lautstärke immer geworfen, sowohl bei Klick auf den Button, wie auch beim Tastendruck.



Event	Beschreibung	Paramter
PlayerPixelController.SUPER_SUBTITLE_ACTIVATION	Aktivierung UT	
PlayerPixelController.SUPER_SUBTITLE_DEACTIVATION	Dektivierung UT	
PlayerPixelController.SUPER_VOLUME_MUTE	Ton aus	
PlayerPixelController.SUPER_VOLUME_UNMUTE	Ton an	
PlayerPixelController.SUPER_VOLUME_CHANGE	Lautstärkeänderung	volume
PlayerPixelController.SUPER_SCRUBBAR_CHANGE_POSITION_FORWARD	Zeitleiste Sprung vorwärt	
PlayerPixelController.SUPER_SCRUBBAR_CHANGE_POSITION_REWIND	Zeitleiste Sprung rückwärts	
PlayerPixelController.SUPER_SCRUBBAR_CHANGE_POSITION_REWIND_TIMESHIFT	Zeitleiste Sprung rückwärts bei aktiviertem Timeshift	
PlayerPixelController.SUPER_FULLSCREEN_ACTIVATION	Vollbild starten	
PlayerPixelController.SUPER_FULLSCREEN_DEACTIVATION	Vollbild beenden	
PlayerPixelController.SUPER_PLAY	Play	
PlayerPixelController.SUPER_PAUSE	Pause	
PlayerPixelController.SUPER_REPLAY	Erneut Abspielen	
PlayerPixelController.SUPER_INITIAL_PLAY	Erstmaliges Abspielen	
PlayerPixelController.SUPER_QUALITY_CHANGE	Qualitätswechsel	
PlayerPixelController.SUPER_BACK_TO_LIVE	Zurück zu Live	
PlayerPixelController.SUPER_PLUGIN_CHANGE	Pluginwechsel	

## 4 ANWENDUNGSBEISPIELE

---

```
(  
  {  
    player_loading: function (params) {  
      console.log("PIXEL => player_loading (" + (params ? " " +  
JSON.stringify(params) + " " : "") + ") ");  
    },  
  
    stream_load: function (params) {  
      console.log("PIXEL => stream_load (" + (params ? " " +  
JSON.stringify(params) + " " : "") + ") ");  
    },  
  
    super_play: function (params) {  
      console.log("PIXEL => super_play (" + (params ? " " +  
JSON.stringify(params) + " " : "") + ") ");  
    },  
  
    super_pause: function (params) {  
      console.log("PIXEL => super_pause (" + (params ? " " +  
JSON.stringify(params) + " " : "") + ") ");  
    },  
  
    super_volume_change: function (params) {  
      console.log("PIXEL => super_volume_change (" + (params ? " " +  
JSON.stringify(params) + " " : "") + ") ");  
    }  
  }  
)
```