|  |
| --- |
|  |

|  |  |
| --- | --- |
|  | November 21, 2012  3.18 SDK – Embedded Media Player |

Table of Contents

[Introduction 3](#_Toc342905778)

[Assumptions 3](#_Toc342905779)

[Setup 3](#_Toc342905780)

[Webpage 3](#_Toc342905781)

[<head/> setup 3](#_Toc342905782)

[<body/> setup 3](#_Toc342905783)

[Administration 4](#_Toc342905784)

[Add device in 3.x Admin UI Error! Bookmark not defined.](#_Toc342905785)

[Player API Methods 5](#_Toc342905786)

[Examples 6](#_Toc342905787)

[Player Configuration 7](#_Toc342905788)

[Examples 7](#_Toc342905789)

[Player Events 8](#_Toc342905790)

[Tutorial to Create a Simple Embedded Player 9](#_Toc342905791)

[Troubleshooting 10](#_Toc342905792)

[Appendix 10](#_Toc342905793)

[Revision History 10](#_Toc342905794)

# Introduction

This guide is to serve as a complement to the Barco Session API or Barco NMS API.

This guide explains how to integrate the Barco Media Player into an HTML webpage.

## Assumptions

This document assumes you have already configured the 3.x SDK API to retrieve media URLs from the Management Server

# Setup

## Webpage

### <head/> setup

In the HTML header, insert on of the following link sets to the JavaScript library files:

1. Standard compressed jQuery library:

<!-- 1. jquery library -->  
<script src="jquery-1.7.1.min.js"></script>

<script src="player.js" > </script>

-OR-

1. Uncompressed jQuery library with debugging capability:

<!-- 1. jquery library -->

<script src="jquery-1.7.1.js"></script>

<script src="player.js" > </script>

**NOTE:** After adding one of the links, the JavaScript Player will be loaded in your webpage. Please see *“Player API Methods”* below for API to the Media Player.

### <body/> setup

Insert one of the following snippets to insert the player into the webpage:

1. Standard:

<div class="barcoplayer"></div>

-OR-

1. Specific:

<div id=”xxxx” class="barcoplayer"></div>

Where “xxxx” is div id to find specific div in DOM

## Administration

The Embedded Player has to added as a device of type “UDP(Rx)” in the 3.x Admin UI. The UDP port at which the Player will receive the media has to be configured using the device stream URL.

### Add the Device

1. Login in to the Admin UI
2. Navigate to the Devices Tab
3. Click “New”
4. Select 3rd Party Device
5. Select UDP(Rx) Decoder
6. Enter a Device Name

### Configure Device Stream URL

1. From the Admin UI, navigate to the Devices Tab
2. Right-click on the UDP(Rx) Decoder you created above
3. Select the EditPort(EP) option
4. Navigate to the PreConfigure Dst Port section of the window
5. In the Enter Stream URL text box, enter:

udp://0.0.0.0:<port\_number>

**NOTE:** You can use a port number such as 2000.

# Player API Methods

Typically you get a handle of the player when page is loaded. For example:

$(document).ready(function()

{

var myPlayer = $('.barcoplayer');

//call other function of players

myPlayer.player(‘xxxx’);

}

Where xxxx is one of the following functions:

|  |  |
| --- | --- |
| **API** | **Description** |
| init() | Initialize Player configuration |
| conf() | Set user configuration |
| load() | Loads the Player ActiveX object as specified in the HTML div container, i.e. <div class=" barcoplayer ">… |
| unload() | Unloads the Player |
| play(controlURL) | Sends a request to specified control URL to start streaming or playback with response streamURL of the request |
| pause() | Pauses the currently playing clip or stream |
| resume() | Resumes the currently paused clip or stream |
| skipforward(skipOffset) | Timeshifts the video forward with the specified skipOffset. If no skipOffset is specified, the default value will be used (10000 ms) |
| skipbackward(skipOffset) | Timeshifts the video backwards with the specified skipOffset. If no skipOffset is specified, the default value will be used (10000 ms) |
| catchup() | Timeshifts the video to the current (live) time (PVR function) |
| stop() | Stops the current clip or stream |
| show() | Shows the Player (unhide) |
| hide() | Hides the Player while it continues to run in the background |
| mute() | Mutes the Player |
| unmute() | Unmutes the Player |
| getMediaURL() | Returns the current media URL |
| getParent() | Returns the Player’s containing DOM element |
| getVersion() | Returns the current version of the Player |
| id() | Returns the ID of the container of the Player |
| isFullscreen() | Returns true if the Player is in full screen mode |
| isHidden() | Returns true if the Player is hidden |
| isLoaded() | Returns true if the Player has finished loading |
| isPaused() | Returns true if the Player is paused |
| isPlaying() | Returns true if the Player is decoding (playing) |

## Examples

[1] //call player functions of init & play

myPlayer.player('init').player('play');

[2] //call init with a specific fps value & then play

myPlayer.player('init',{fps:30}).player(‘play’);

# Player Configuration

The player supports various configuration options:

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Default Value** | **Valid Entries** | **Description** |
| width | 200px | 0-screen width range | Player display width |
| height | 200px | 0-screen height range | Player display height |
| fps | 15 | 1-60 | The frame rate the Player will decode at |
| skipOffset | 10000ms |  | The skip forward / skip backward interval (in ms) |
| http | true | true, false | Specify if the stream transport protocol is HTTP (true) or UDP (false) |

## Examples

<script>  
myPlayer.player (‘init’,{  
   
 //set video fps

fps:15,

//set player display area

width:640,

height:360,  
   
 //millisecond, set offset for skip forward and skip back  
 skipOffset: 10000   
});  
</script>

# Player Events

You can use the subscribe method to execute your own JavaScript when something happens in the player. For example:

$.subscribe("onError", function ErrorHandler (){

//do your error process

});

Where ErrorHandler is a user defined function

|  |  |
| --- | --- |
| **Event** | **Description** |
| onLoad | When the player has finished loading |
| onUnload | When the player has unloaded |
| onStart | The Player has starting playing a video or stream |
| onPause | The Player has been paused |
| onResume | A paused video has been resumed |
| onCatchup | A PVR’ed video has caught up to real-time |
| onSeek | A video has been timeshifted (skip forward or skip backward) via PVR |
| onMute | When the Player has been muted |
| onUnmute | When the Player has been unmuted |
| onFullscreen | When the Player has entered fullscreen mode |
| onFullscreenExit | When the Player has exited from fullscreen mode |
| onError | The Player is in an error state |
| onStart | The Player has starting playing a video or stream |
| onStop | The Player has stopped playing a video or stream |
| onConnState | Connection state change event |
| onTimeLine | This event will be published whenever timeline operations are executed: skip forward, skip backward, resume start session, catchup |

Please refer to API document for the detail each attribute meaning.

# Tutorial to Create a Simple Embedded Player

The following tutorial provides an example of to create a simple Embedded Player:

<html>

<head>

<script src="jquery-1.7.1.min.js"></script>

<script src="player.js" > </script>

</head>

<body>

<div class="barcoplayer" ></div>

<div id="playercontrol">

<span id="play"> <BUTTON>Play</BUTTON> </span>

<span id="stop"> <BUTTON>Stop</BUTTON> </span>

</div>

<script>

$(document).ready(function () {

//get player

var myPlayer = $('.barcoplayer').player('init');

$('#play').bind("click", function () {

//start a session and play

myPlayer.player('play', <mediaURL retrieved 3.x API>);

});

$('#stop').bind("click", function () {

//stop the session

myPlayer.player('stop');

});

});

</script>

</body>

</html>

# Troubleshooting

If you having any issues getting the Player to display video, please check the following:

1. If the API user has the correct authorization to access the media
2. Whether a proper UDP(Rx) Decoder has been created in the Admin UI
3. Whether the appropriate Stream URL has been configured for the UDP(Rx) Decoder
4. Whether an ActiveX is loaded properly
   1. *You can check by right-clicking the mouse in the display area. You should see a popup dialog box with “Statistics/About” options*

**NOTE:** If you are still having trouble, you can get detailed information from the “Sessions” Tab of the Admin UI. *Refer to the Admin UI Setup Guide for more on the Sessions Tab.*

# Appendix

Pointer to reference player in 3.x SDK:

* SDKIMAGES/player/player.html

Pointer to JS library files:

* SDKIMAGES/player/player.js
* SDKIMAGES/player/sessionapi.js (loading by player.js)
* SDKIMAGES/player/player/jquery-1.7.1.js (standard library)

-OR-

* SDKIMAGES/player/player/jquery-1.7.1.min.js (minified library)

# Revision History

|  |  |
| --- | --- |
| **Revision** | **Description** |
| 1.0.1 | Reorganized API Methods, rewrote “Setup” section, shortened “Configuration” options list, inserted pointers to SDK files |
| 1.0.2 | Added “Tutorial” and “Troubleshooting” sections, Added new Methods and Events, added instructions to add UDP Rx in Admin UI, reworded several calls/events/configuration descriptions |
| 1.0.3 | Updated language related to “Setup,” referred to myPlayer for examples, renamed section headings |