Mixberry Web SDK v4.3

Developer Guide

TABLE OF CONTENTS

1. Introduction

The Mixberry Media SDK is a light client library that handles the communications between an audio based application and our ad servers, providing the functionality to request targeted streamed audio ads in real-time.

2. Definitions

Mixberry Web SDK

This is the Web SDK for Mixberry Media's audio ad network. The SDK provides a rich set of functionality for accessing Mixberry's server-side API calls.

MB_SDK_Web.js

This JavaScript file handles the communications between the publisher's web application and Mixberry. It must be included in your application to be able to pull ads from the Mixberry Media ad servers.

App Key

The App key is provided after the application registration process, upon Mixberry's review and approval of the app submission. The key must be unique for every application registered by the Publisher.

3. Compatibility

Platforms and browsers that are supported by Mixberry Media web SDK:

- Windows: Firefox, Chrome, Safari, IE8, IE9, IE10, IE11
- OSX: Safari, Firefox, Chrome
- Android: Android 2.2 Browser and later (Mobile Web).

Android Specific Considerations

Flash is required to be installed on the end user's device for ads to be delivered.

4. Getting Started:

Register as a Publisher

In order to implement Mixberry Media's audio ad service for your audio enabled application, you need to register as a "Publisher" at: http://www.mixberrymedia.com.

Define your Application or Web Site

Start by defining your web site or application name, description and category. Your application request would then await review and approval by Mixberry Media.

Obtain your App Key

Upon your application approval, log in to your portal to obtain your App Key.

Note:
The App key is mandatory and is used by the Mixberry Media library to identify which application requested the ad. Please make sure to use the correct App Key for each web site or application that you registered.

Download the SDK

Download the Web SDK from the Publisher's Dev Tools page within your portal. You will receive an archive containing the following files:

- MB_SDK_Web.js
- Jplayer.swf
- MB_SDK_Script.js
- MB_SDK_Web.css
- jquery.min.js
- mbComPlugin.swf
- MB_Pub_SDK_Web_v4.3.pdf (This document)

Note:
The minimum required version of jquery.min.js is 1.7.2.
If you want to use your own jquery.min.js file, it must be of version (1.7.2) or higher.
For versions 1.9 or later, make sure to add the following script:

```
<script src="http://code.jquery.com/jquery-migrate-1.2.1.js"></script>
```

5. SDK Integration

1. Create a new directory called "SDK", normally in your web root.
2. Add the downloaded files to the SDK directory. So now you must have the following file structure:
   - /SDK/MB_SDK_Web.js
   - /SDK/Jplayer.swf
   - /SDK/MB_SDK_Script.js
   - /SDK/MB_SDK_Web.css
   - /SDK/jquery.min.js
   - /SDK/mbComPlugin.swf
3. Include the SDK JavaScript library inside the <HEAD> </HEAD> tags of each page where you will utilize the SDK library.

```
<script type="text/javascript" src="./SDK/MB_SDK_Web.js"></script>
Your code should be like this: (Adjust the path to the SDK directory as
needed.)
<head>
.
.
<script type="text/javascript" src="./SDK/MB_SDK_Web.js"></script>
.
.
</head>
<body>
.
.
</body>
```

SDK Implementation

1. After integrating the Mixberry Media library into your web application, you need to call the SDK initializer, passing your App Key. It is recommended to call the initializer on the page load event to ensure that the page was loaded fully and that the SDK can be initialized without any errors.

```
<body onload="mbServiceInitialization ('YOUR_APP_KEY');">
```

2. Implement the mbOnAdDelivery (status) function:

```
function mbOnAdDelivery (status) {
      //PLACE YOUR CODE HERE
}
```

This method is called once the ad status is received. The status (*status*) contains info about the result of the operation. The status could be one of the following:

Result Code	Description
MB_AD_STATUS.SUCCESS = 0	The operation was successful.
MB_AD_STATUS.SKIPPED = 1	The ad was skipped.
MB_AD_STATUS.FAILED_DELIVERY = 2	The operation failed. The Ad cannot be delivered.
MB_AD_STATUS.FAILED_SDK_BUSY = 3	The operation failed. The SDK is busy serving another ad.
MB_AD_STATUS.FAILED_BROWSER_NOT_SUPPORTED = 4	Requests from iOS devices and Opera browsers are not supported.
MB_AD_STATUS.FAILED_FLASH_NOT_FOUND = 5	Requests from browser not supporting HTML5 where a flash player is not installed or enabled are not supported.

Table 1: List of results codes.

3. If you want to request an ad after you load the page directly, you may end up calling the request ad operation before the SDK is initialized fully and you will always get a failure operation. To override this you need to implement the mbOnServiceReady () method which will be called once the SDK is fully initialized:

```
function mbOnServiceReady (){
//Do any initialization: e.g. request Ad
}
```

4. Optionally, you can implement the mbOnAdRequested() function. This method will be called once the process of fetching starts and can be used to show the user a loading message.

```
function mbOnAdRequested (){
//Place your code here
}
```

5. Optionally, you can implement the mbOnAdReceived() function. This method will be called once the ad response is received

```
function mbOnAdReceived (){
//Place your code here
}
```

6. Optionally, you can implement the mbOnAdStarted() function. This method will be called once the ad starts playing

```
function mbOnAdStarted (){
//Place your code here
}
```

6. Requesting an Ad

To request an ad from the Mixberry Media servers, you need to set the ad criteria, which represent the specifications of the requested ad.

The process of requesting an ad is performed using the following method:

```
function mbRequestAd (adCriteria)
```

Ad Request Criteria

Mixberry Media supports two types of ad formats. The publisher can choose the ad type that best fits the app's behavior, along with banner position, skip offset and music genre (for music based apps)

Ad Type

Mixberry media supports two types of ads:

```
MB_AD_TYPE.AUDIO_ONLY: Audio only ad
MB_AD_TYPE.AUDIO_WITH_BANNER: Audio ad with a synchronized 300x250 banner.
```

Notes:

- If an ad type is not specified, the default value is AUDIO_WITH_BANNER.

- Changing this default value requires special approval by Mixberry Media.

Music Genre (Optional):

Music app Publishers can set the music genre parameter in order to match between the played song and the requested ad. To do so, we provide publishers with various types of genres such as ROCK, COUNTRY, BLUES, FOLK, ELECTRONIC, SOUL / RnB…etc. Genre values can be obtained from MB_MUSIC_GENRE associative array. For more information, please refer to Appendix B.

Interest (Optional):

Publishers can set the interest parameter in order to match between the content of the page and the requested ad. To do so, we provide publishers with various types of interests such as MUSIC, KIDS_AND_TEENS, LIFESTYLE, FOOD_AND_DRINK, TRAVEL, SPORTS, SHOPPING…etc. Interest values can be obtained from MB_INTEREST associative array. For more information, please refer to Appendix B.

Synchronized Companion Banner

You can also set the banner position by setting the parameter "position" to one of following values:

```
MB_BANNER_POSITION.TOP_LEFT
MB_BANNER_POSITION.TOP_RIGHT
MB_BANNER_POSITION.BOTTOM_LEFT
MB_BANNER_POSITION.BOTTOM_RIGHT (default)
```

Skip Ad

You can set a skip offset to allow users to skip the running ad after the specified period of time. The allowed values are [5-30]. If an invalid value is set, the SDK defaults to 5.

Notes:

- *If the ad length is less than the defined offset, the skip offset value is ignored.*
- *If skip offset is not defined, the user will not be able to skip the ad and the ad will play in its entirety.*
- *Audio only Ads cannot be skipped.*

Targeted Ads vs. General Ads

Based on the available user information, Mixberry Media servers deliver either a general or a targeted ad. As their names indicate, general ads can be sent to any user, whereas targeted ads target a certain group of users according to the information available about those users. The more information collected from the user the higher the chance of receiving a targeted ad. Targeted ads have two benefits over general ads: (1) The price of a targeted ad is typically higher, and (2) Targeted ads are typically more relevant to the user than general ads.

The user's information is set in the library using the method *mbSetUserInfo*. This method takes a bundle that maps keys and their values in a string-string manner. The keys are strings that are pre-defined in the Mixberry Media servers and are case-sensitive. If they are not entered correctly, they will be ignored, therefore rather than entering them manually, it is recommended to use the strings defined in the MB_USER_INFO associative arrays, which provides all the predefined keys that we use.

The following are the targeting parameters that can be used to request targeted ads:

1. Gender (GENDER)
   It takes its value from MB_GENDER associative array.
2. Language (LANGUAGE)

   It takes its value from MB_LANGUAGE associative array.
3. Age (AGE)

   Accepts positive integer values

Appendix A: Sample Code for Developers

The following examples demonstrate how to create a Mixberry Media Service object along with the related handlers.

Initialization Example:

```
<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Transitional//EN'
'http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd'>
<html xmlns='http://www.w3.org/1999/xhtml' lang='en' xml:lang='en'>
<head>
<title>SDK Initialization Sample</title>
<script type="text/javascript" src="SDK/MB_SDK_Web.js"></script>
</head>

<body onload ="mbServiceInitialization ('APP_KEY');">

</body>
</html>
```

To request an ad from Mixberry's network, the "mbRequestAd" method should be called in the controller. This call can be placed in any part of the application.

Example: Audio Ad with Banner
Mixberry Media's standard ads are audio ads with a synchronized 300x250 banner. The following example demonstrates the process of requesting this type of ad.

```
<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Transitional//EN'
'http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd'>
<html xmlns='http://www.w3.org/1999/xhtml' lang='en' xml:lang='en'>
<head>
<title>Combined Ad Example</title><script type="text/javascript"
src="SDK/MB_SDK_Web.js"></script>
<script type="text/javascript">
      function requestAdHandler() {
            var adCriteria = {
                        "adType" : MB_AD_TYPE.AUDIO_WITH_BANNER,

                  "position" : MB_BANNER_POSITION.TOP_LEFT,
            };

            mbRequestAd(adCriteria);
      }
</script>
</head>

<body onload=" mbServiceInitialization('APP_KEY');">  <a
href="javascript:requestAdHandler()">Get Ad</a>

</body>
</html>
```

Music Genre Example:
This example shows how to request an ad with music genre of type ROCK.

```
var adCriteria= {
            "genre" : MB_MUSIC_GENRE.ROCK
      };

mbRequestAd(adCriteria);
```

Interest Examples:
This example shows how to request an ad with Interest of type MUSIC

```
var adCriteria= {
            "interest": MB_INTEREST.MUSIC
      };

mbRequestAd(adCriteria);
```

This example shows how to request an ad with multiple Interest values (MUSIC, ARTS_AND_ENTERTAINMENT, EDUCATION)

```
var interestValues =[MB_INTEREST.MUSIC,MB_INTEREST.ARTS_AND_ENTERTAINMENT,
MB_INTEREST.EDUCATION];
var adCriteria= {
            "interest": intersetValues
      };

mbRequestAd(adCriteria);
```

Skip Ad Example:
This example shows how to request an ad with skip offset set to 10.

```
var adCriteria= {
             "adType" : MB_AD_TYPE.AUDIO_WITH_BANNER,
            "skipOffset" : 10
      };

mbRequestAd(adCriteria);
```

Targeted Ad Example:
In this example, the age group parameter is set to 27 to indicate that the Mixberry Media network should try to get an ad that targets 27 year-old users. Also, the language parameter is set to English and gender to female. The mbSetUserInfo should be called in the early stages, before calling the request ad.

```
var userInfo  = new Object(); // or just {}
userInfo[MB_USER_INFO.AGE] = 27;
userInfo[MB_USER_INFO.LANGUAGE] = MB_LANGUAGE.ENGLISH;
userInfo[MB_USER_INFO.GENDER] = MB_GENDER.FEMALE;
mbSetUserInfo(userInfo);
```

How to use the Mixberry Media SDK with links:

If your web site has media files links, and you want to play an ad before playing the media you can use:

```
var adCriteria= {
            "linkedMedia" : "http://MEDIA_LINK",
      };
mbRequestAd(adCriteria);
```

This will play the ad first and then redirect the browser to the provided link (http://MEDIA_LINK)

Note:

This API works for general purposes but was specifically provided for mobile websites that use links to play audio media streams through the mobile player instead of a player in the browser itself. So adding the above code will allow the web page to play an AD before the redirection process.

Appendix B: Targeting parameters predefined values

Gender, Language and Age group are targeting parameters (can be found in MB_USER_INFO associative array):

Gender

GENDER: The following list shows all supported gender types that can be found in MB_GENDER associative array.

MALE	
FEMALE	

Table 2: Gender parameter values

Language

LANGUAGE: The following list shows all supported language types that can be found in MB_LANGUAGE associative array.

ARABIC	ICELANDIC
SPANISH	INDONESIAN
ENGLISH	PERSIAN
MALAY	HEBREW
FILIPINO	ITALIAN
URDU	JAPANESE
FRENCH	DUTCH
GREEK	NORWEGIAN
PORTUGUESE	POLISH
HINDI	RUSSIAN
CHINESE	KOREAN
DANISH	SWEDISH
FINNISH	ZULU
HUNGARIAN	THAI

Table 3: Language parameter values

Age Group

AGE: represents the user's age and takes an integer value.

Music Genre Types

MB_MUSIC_GENRE: The following list shows all supported music genre types that can be found in this associative array.

ROCK	METAL
COUNTRY	NEW_AGE
BLUES	ENTERTAINMENT
FOLK	CLASSICAL_EARLY_COMMON_PRACTICE
ELECTRONIC	CLASSICAL_MODERN_CONTEMPORARY
SOUL_RNB	MIDDLE_EAST
POP	OCEANIA
JAZZ	LATIN
REGGAE	EUROPE
HIP_HOP	AFRICA

Table 4: Genre parameter values

Interest Types

MB_INTEREST: The following list shows all supported interest types that can be found in this associative array.

ARTS_AND_ENTERTAINMENT	BOOKS
MEDIA_AND_VIDEO	MEDICAL
SOCIAL_NETWORKING	GAMES
MUSIC	MEN_INTERESTS
KIDS_AND_TEENS	WOMEN_INTERESTS
LIFESTYLE	AUTOMOTIVE
FOOD_AND_DRINK	CAREERS
TRAVEL	FAMILY_AND_PARENTING
SPORTS	HOBBIES_AND_INTERESTS
SHOPPING	HOME_AND_GARDEN
HEALTH_AND_FITNESS	LAW_AND_GOVERNMENT
NEWS	PETS
TECHNOLOGY_AND_COMPUTING	REAL_AND_ESTATE
FINANCE	RELIGION_AND_SPIRITUALITY
BUSINESS	SCIENCE
EDUCATION	SOCIETY
WEATHER	STYLE_AND_FASHION

Table 5: Interest parameter values