BOOM VIDEO PLAYER INTEGRATION GUIDE WITH IOS 1.0

Document Version 1.0
Last Updated – January 20, 2015

Contact

Level 4 / 86 Liverpool Street

Sydney NSW, 2000

Tel: 02 9264 0001

Fax: 02 8580 5109

Email: info@boomvideo.com.au

Table of Contents

What's in this Guide?

This integration guide provides information about how to integrate the BOOM Video Player with iOS. It describes the following procedures:

- Building the project and initializing the video player
- Adding optional functionalities (Offer List or Video player)

It also provides a sample code for the BOOM Video Player and Offer list integration.

Prerequisites

Minimum skill level required

- iOS developer with intermediate skill level

Environment setup

- Xcode 6 or above
- iOS 7.0 or above (device/simulator)
- You will need 'BOOMGUID' for your game, which will be provided by BOOM. You can contact BOOM for the BOOMGUID at info@boomvideo.com.au

Sample Application

Here is a sample iOS video player example. Click the URL below to see the example.

https://github.com/BOOMGitRepo/BOOMVideo_iOS_Example

Steps to Integrate BOOM Preroll Video Player with iOS

Step 1: Build the Project

1. Download the following **`.a`** (static library) file and i**`nclude`** folder:

   http://boom.boomvideo.tv/alpha/app/iOS/BoomiOSVideoPlayerLibrary.zip

2. Extract and copy the downloaded **`.a`** file and **`include`** folder into the project.
3. Include the following frameworks into your Xcode project:

   JavaScriptCore.framework
   AddressBook.framework
   AssetsLibrary.framework
   CoreLocation.framework
   CoreMotion.framework
   CoreText.framework
   CoreGraphics.framework
   Security.framework
   SystemConfiguration.framework
   MediaPlayer.framework
   Social.framework
   UIKit.framework
   Foundation.framework

4. Download Google Plus SDK from the following link:

   https://developers.google.com/+/mobile/ios/getting-started

   And include following files in project:

   GoogleOpenSource.framework
   GooglePlus.framework
   GooglePlus.bundle

5. In the “Build Setting” tab of the project’s Target, add the flag “-**ObjC”** under the “Other Linker Flags” option inside “Linking” section.
6. Set the BOOMGUID:

   Create a global constant boomGuid before @interface of the class.

```
#define boomGuid      @"2900bbd3-f8f9-4862-b65f-ce9dbe165f09"
```

where “2900bbd3-f8f9-4862-b65f-ce9dbe165f09” is test BOOMGUID. Please replace with BOOMGUID provided for your game

7. Import `BMResourceManager.h` in your class and add the following block of code on Button click or where you want to add Preroll player

```
[BMResourceManager showVideoForGUID:boomGuid withType:BMPreroll];
```

8. For Google+ sharing, developer needs to generate a unique “Client ID” with respect to their “Bundle Identifier”. Go to the following link and follow the “Step 1. Creating the Google Developers Console project” to create “Client ID” and “Step 3. Add a URL type” for properly handling client ID in your project.
   https://developers.google.com/+/mobile/ios/getting-started

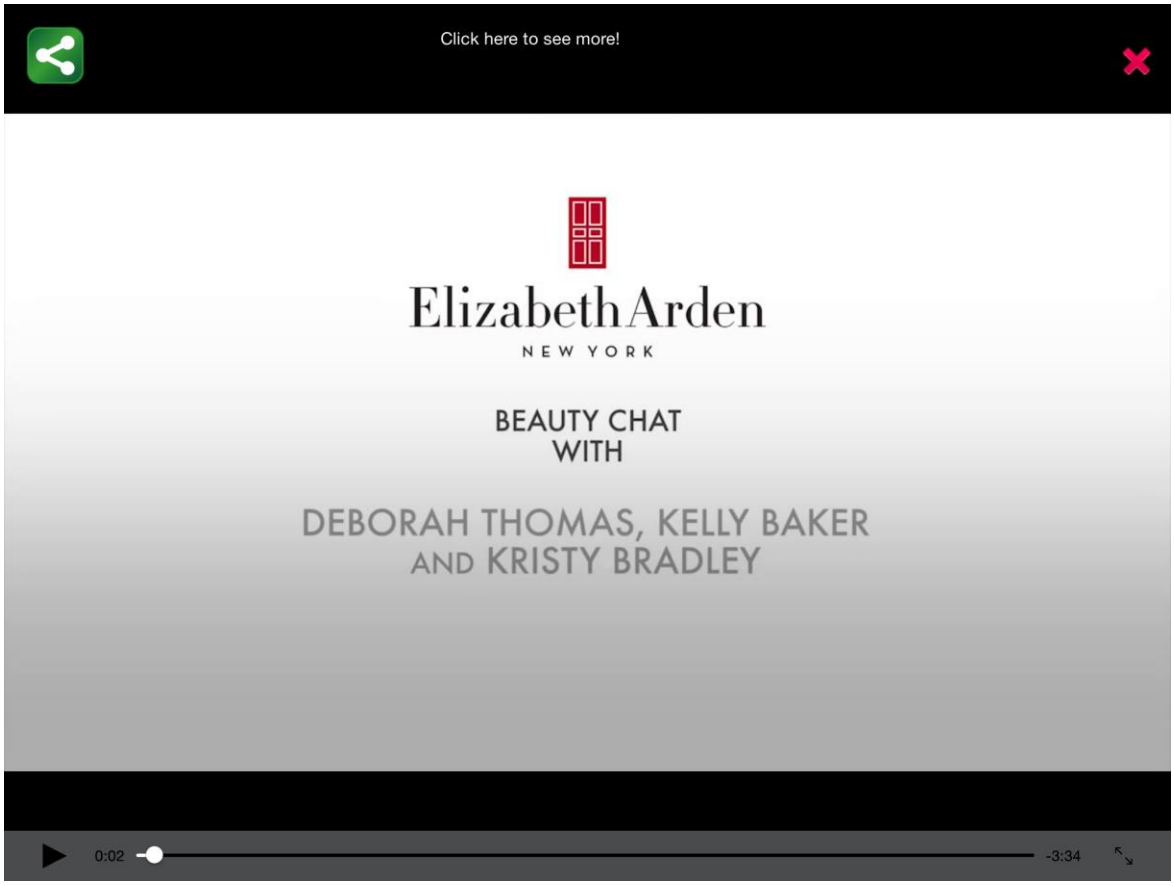
8.1 Add a new plist file in your project with name - “boomVideoSharingData”.

8.2 Add a Client ID generated by the link as a String value type for key- “googlePlusClientId”.

8.3 In AppDelegate.m file of your project, import and add following code:

```
#import <GooglePlus/GooglePlus.h>
- (BOOL)application: (UIApplication *)application openURL: (NSURL *)url
 sourceApplication: (NSString *)sourceApplication
 annotation: (id)annotation {
    return [GPPURLHandler handleURL:url
 sourceApplication:sourceApplication
 annotation:annotation];
}
```

BOOM Preroll Video Player after Integration



Step 2: Add Callback

1. Import BMResourceManager class and confirm BoomVideoTrackerDelegate protocol in `.h` file of the class.

   For example:

   ```
   #import “BMResourceManager.h”

   @interface MainViewController : UIViewController <BoomVideoTrackerDelegate>
   ```

2. After the `@interface` line of `.h` class, declare a property of `BMResourceManager` class.

   For example:

```
@property (nonatomic, strong) BMResourceManager *resourceManager;
```

3. In .m file of the class, initialize BMResourceManager singleton class inside viewDidLoad method and give self to videoTrackerInfoDelegate of BMResourceManager.
   For example:

```
self.resourceManager = [BMResourceManager sharedInstance];
self.resourceManager.videoTrackerInfoDelegate = self;
```

4. Implement the required delegate method in `.m` file of the class
   - (void)boomVideoTrackCallbackWithEvent:(BOOMEventErrorCode) eventCode withData:(NSDictionary*)detailData;

```
- (void)boomVideoTrackCallbackWithEvent:(BOOMEventErrorCode) eventCode
withData:(NSDictionary*)detailData {
    // BOOMEventErrorCode - It will give the event codes
    // detailData - It is a dictionary which will contain three key-value
pairs:
    // 1. eventName - Name of Event
    // 2. videoPercentage - Percent of video seen by user
    // 3. pointsRevealed - Points to give, according to the business rule
    NSLog(@"Tracking Details->%@",detailData);
}
```

Here are various result codes and their description sdk will give:

Sr.no	Result Codes	Description
1)	kVideoStarted	SDK will give **kVideoStarted** result code when video has loaded for the first time

2)	kVideoPaused	SDK will give **kVideoPaused** result code when video is paused
3)	kVideoCompletedwithPercentage	SDK will give **VideoCompletedwithPercentage** result code when video is closed or completely seen by user, with the percentage of video seen in dictionary
4)	kSuccessfulSharedOnFacebook	SDK will give **kSuccessfulSharedOnFacebook** result code when user shares video on facebook successfully
5)	kSuccessfulSharedOnTwitter	SDK will give **kSuccessfulSharedOnTwitter** result code when user shares video on twitter successfully
6)	kSuccessfulSharedOnGooglePlus	SDK will give **kSuccessfulSharedOnGooglePlus** result code when user shares video on Google+ successfully
7)	kSurveyCompleted	SDK will give **kSurveyCompleted** result code when user has finished survey from offerlist
8)	kSurveyNotCompleted	SDK will give **kSurveyNotCompleted** result code when user has cancelled survey from offerlist
9)	kRedirectedToBlog	SDK will give **kRedirectedToBlog** result code when user clicks on blog link.

10)	kRedirectedToInstall	SDK will give **kRedirectedToInstall** result code when user clicks on installation of application link.
11)	kRedirectedToInstagram	SDK will give **kRedirectedToInstagram** result code when user clicks on instagram link.
12)	kRedirectedToSlideshare	SDK will give **kRedirectedToSlideshare** result code when user clicks on slideshare link.
13)	kRedirectedToAnnotation	SDK will give **kRedirectedToAnnotation** result code when user clicks on annotation link.
14)	kNetworkNotAvailableError	SDK will give **kNetworkNotAvailableError** result code when network is not available
15)	kUnknownError	SDK will give **kUnknownError** result code when there is an UNKNOWN ERROR

Sample Code for BOOM Preroll Video Player Integration

https://github.com/BOOMGitRepo/BOOMVideo_iOS_Example

Steps to Integrate BOOM Reward Video Player with iOS

Step 1: Build the Project

1. Download the following **`.a`** (static library) file and i**`nclude`** folder:

   http://boom.boomvideo.tv/alpha/app/iOS/BoomiOSVideoPlayerLibrary.zip

2. Extract and copy the downloaded **`.a`** file and **`include`** folder into the project.
3. Include the following frameworks into your Xcode project:

   JavaScriptCore.framework
   AddressBook.framework
   AssetsLibrary.framework
   CoreLocation.framework
   CoreMotion.framework
   CoreText.framework
   CoreGraphics.framework
   Security.framework
   SystemConfiguration.framework
   MediaPlayer.framework
   Social.framework
   UIKit.framework
   Foundation.framework

4. Download Google Plus SDK from the following link:

   https://developers.google.com/+/mobile/ios/getting-started

   And include following files in project:

   GoogleOpenSource.framework
   GooglePlus.framework
   GooglePlus.bundle

5. In the “Build Setting” tab of the project’s Target, add the flag “-**ObjC”** under the “Other Linker Flags” option inside “Linking” section.
6. Set the BOOMGUID:

   Create a global constant boomGuid before @interface of the class.

```
#define boomGuid      @"2900bbd3-f8f9-4862-b65f-ce9dbe165f09"
```

where “2900bbd3-f8f9-4862-b65f-ce9dbe165f09” is test BOOMGUID. Please replace with BOOMGUID provided for your game

7. Import `BMResourceManager.h` in your class and add the following block of code on Button click or where you want to add Reward player

```
[BMResourceManager showVideoForGUID:boomGuid withType:BMReward];
```

8. For Google+ sharing, developer needs to generate a unique “Client ID” with respect to their “Bundle Identifier”. Go to the following link and follow the “Step 1. Creating the Google Developers Console project” to create “Client ID” and “Step 3. Add a URL type” for properly handling client ID in your project.
   https://developers.google.com/+/mobile/ios/getting-started

8.1 Add a new plist file in your project with name - “boomVideoSharingData”.

8.2 Add a Client ID generated by the link as a String value type for key- “googlePlusClientId”.

8.3 In AppDelegate.m file of your project, import and add following code:

```
#import <GooglePlus/GooglePlus.h>
- (BOOL)application: (UIApplication *)application openURL: (NSURL *)url
 sourceApplication: (NSString *)sourceApplication
 annotation: (id)annotation {
    return [GPPURLHandler handleURL:url
 sourceApplication:sourceApplication
 annotation:annotation];
}
```

BOOM Reward Video Player after Integration

Step 2: Add Callback

1. Import BMResourceManager class and confirm BoomVideoTrackerDelegate protocol in `.h` file of the class.

   For example:

   ```
   #import “BMResourceManager.h”

   @interface MainViewController : UIViewController <BoomVideoTrackerDelegate>
   ```

2. After the `@interface` line of `.h` class, declare a property of `BMResourceManager` class.

   For example:

   ```
   @property (nonatomic, strong) BMResourceManager *resourceManager;
   ```

3. In .m file of the class, initialize BMResourceManager singleton class inside viewDidLoad method and give self to videoTrackerInfoDelegate of BMResourceManager.
   For example:

```
self.resourceManager = [BMResourceManager sharedInstance];
self.resourceManager.videoTrackerInfoDelegate = self;
```

4. Implement the required delegate method in `.m` file of the class
   - (void)boomVideoTrackCallbackWithEvent:(BOOMEventErrorCode) eventCode withData:(NSDictionary*)detailData;

```
- (void)boomVideoTrackCallbackWithEvent:(BOOMEventErrorCode) eventCode
withData:(NSDictionary*)detailData {
    // BOOMEventErrorCode - It will give the event codes
    // detailData - It is a dictionary which will contain three key-value
pairs:
    // 1. eventName - Name of Event
    // 2. videoPercentage - Percent of video seen by user
    // 3. pointsRevealed - Points to give, according to the business rule
    NSLog(@"Tracking Details->%@",detailData);
}
```

Here are various result codes and their description sdk will give:

Sr.no	Result Codes	Description
1)	kVideoStarted	SDK will give **kVideoStarted** result code when video has loaded for the first time
2)	kVideoPaused	SDK will give **kVideoPaused** result code when video is paused

3)	kVideoCompletedwithPercentage	SDK will give **VideoCompletedwithPercentage** result code when video is closed or completely seen by user, with the percentage of video seen in dictionary
4)	kSuccessfulSharedOnFacebook	SDK will give **kSuccessfulSharedOnFacebook** result code when user shares video on facebook successfully
5)	kSuccessfulSharedOnTwitter	SDK will give **kSuccessfulSharedOnTwitter** result code when user shares video on twitter successfully
6)	kSuccessfulSharedOnGooglePlus	SDK will give **kSuccessfulSharedOnGooglePlus** result code when user shares video on Google+ successfully
7)	kSurveyCompleted	SDK will give **kSurveyCompleted** result code when user has finished survey from offerlist
8)	kSurveyNotCompleted	SDK will give **kSurveyNotCompleted** result code when user has cancelled survey from offerlist
9)	kRedirectedToBlog	SDK will give **kRedirectedToBlog** result code when user clicks on blog link.
10)	kRedirectedToInstall	SDK will give **kRedirectedToInstall** result code when user clicks on installation of application link.

11)	kRedirectedToInstagram	SDK will give **kRedirectedToInstagram** result code when user clicks on instagram link.
12)	kRedirectedToSlideshare	SDK will give **kRedirectedToSlideshare** result code when user clicks on slideshare link.
13)	kRedirectedToAnnotation	SDK will give **kRedirectedToAnnotation** result code when user clicks on annotation link.
14)	kNetworkNotAvailableError	SDK will give **kNetworkNotAvailableError** result code when network is not available
15)	kUnknownError	SDK will give **kUnknownError** result code when there is an UNKNOWN ERROR

Sample Code for BOOM Reward Video Player Integration

https://github.com/BOOMGitRepo/BOOMVideo_iOS_Example

Steps to Integrate BOOM Offerlist Video Player with iOS

Step 1: Build the Project

1. Download the following **`.a`** (static library) file and i**`nclude`** folder:

   http://boom.boomvideo.tv/alpha/app/iOS/BoomiOSVideoPlayerLibrary.zip

2. Extract and copy the downloaded **`.a`** file and **`include`** folder into the project.
3. Include the following frameworks into your Xcode project:

   JavaScriptCore.framework
   AddressBook.framework
   AssetsLibrary.framework
   CoreLocation.framework
   CoreMotion.framework
   CoreText.framework
   CoreGraphics.framework
   Security.framework
   SystemConfiguration.framework
   MediaPlayer.framework
   Social.framework
   UIKit.framework
   Foundation.framework
4. Download Google Plus SDK from the following link:

   https://developers.google.com/+/mobile/ios/getting-started

   And include following files in project:

   GoogleOpenSource.framework
   GooglePlus.framework
   GooglePlus.bundle
5. In the “Build Setting” tab of the project’s Target, add the flag “-**ObjC”** under the “Other Linker Flags” option inside “Linking” section.
6. Set the BOOMGUID:

   Create a global constant boomGuid before @interface of the class.

```
#define boomGuid      @"2900bbd3-f8f9-4862-b65f-ce9dbe165f09"
```

where “2900bbd3-f8f9-4862-b65f-ce9dbe165f09” is test BOOMGUID. Please replace with BOOMGUID provided for your game

7. Import `BMResourceManager.h` in your class and add the following block of code on Button click or where you want to add Offerlist player

```
[BMResourceManager showVideoForGUID:boomGuid withType:BMOfferList];
```

8. For Google+ sharing, developer needs to generate a unique “Client ID” with respect to their “Bundle Identifier”. Go to the following link and follow the “Step 1. Creating the Google Developers Console project” to create “Client ID” and “Step 3. Add a URL type” for properly handling client ID in your project.
   https://developers.google.com/+/mobile/ios/getting-started

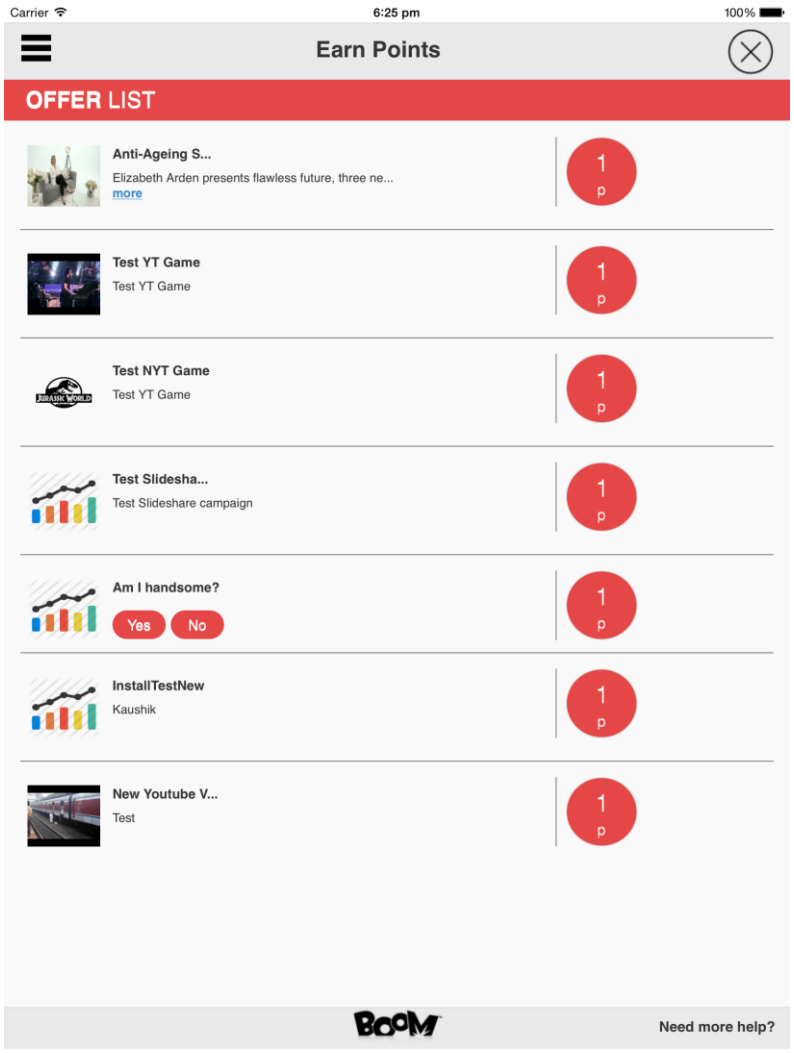
8.1 Add a new plist file in your project with name - “boomVideoSharingData”.

8.2 Add a Client ID generated by the link as a String value type for key- “googlePlusClientId”.

8.3 In AppDelegate.m file of your project, import and add following code:

```
#import <GooglePlus/GooglePlus.h>
- (BOOL)application: (UIApplication *)application openURL: (NSURL *)url
 sourceApplication: (NSString *)sourceApplication
 annotation: (id)annotation {
    return [GPPURLHandler handleURL:url
 sourceApplication:sourceApplication
 annotation:annotation];
}
```

BOOM Offerlist Page after Integration



Step 2: Add Callback

1. Import BMResourceManager class and confirm BoomVideoTrackerDelegate protocol in `.h` file of the class.

   For example:

```
#import “BMResourceManager.h”

@interface MainViewController : UIViewController <BoomVideoTrackerDelegate>
```

2. After the `@interface` line of `.h` class, declare a property of `BMResourceManager` class.
   For example:

```
@property (nonatomic, strong) BMResourceManager *resourceManager;
```

3. In .m file of the class, initialize BMResourceManager singleton class inside viewDidLoad method and give self to videoTrackerInfoDelegate of BMResourceManager.
   For example:

```
self.resourceManager = [BMResourceManager sharedInstance];
self.resourceManager.videoTrackerInfoDelegate = self;
```

4. Implement the required delegate method in `.m` file of the class
   - (void)boomVideoTrackCallbackWithEvent:(BOOMEventErrorCode) eventCode withData:(NSDictionary*)detailData;

```
- (void)boomVideoTrackCallbackWithEvent:(BOOMEventErrorCode) eventCode
withData:(NSDictionary*)detailData {
    // BOOMEventErrorCode - It will give the event codes
    // detailData - It is a dictionary which will contain three key-value
pairs:
    // 1. eventName - Name of Event
    // 2. videoPercentage - Percent of video seen by user
    // 3. pointsRevealed - Points to give, according to the business rule
    NSLog(@"Tracking Details->%@",detailData);
}
```

Here are various result codes and their description sdk will give:

Sr.no	Result Codes	Description

1)	kVideoStarted	SDK will give **kVideoStarted** result code when video has loaded for the first time
2)	kVideoPaused	SDK will give **kVideoPaused** result code when video is paused
3)	kVideoCompletedwithPercentage	SDK will give **VideoCompletedwithPercentage** result code when video is closed or completely seen by user, with the percentage of video seen in dictionary
4)	kSuccessfulSharedOnFacebook	SDK will give **kSuccessfulSharedOnFacebook** result code when user shares video on facebook successfully
5)	kSuccessfulSharedOnTwitter	SDK will give **kSuccessfulSharedOnTwitter** result code when user shares video on twitter successfully
6)	kSuccessfulSharedOnGooglePlus	SDK will give **kSuccessfulSharedOnGooglePlus** result code when user shares video on Google+ successfully
7)	kSurveyCompleted	SDK will give **kSurveyCompleted** result code when user has finished survey from offerlist
8)	kSurveyNotCompleted	SDK will give **kSurveyNotCompleted** result code when user has cancelled survey from offerlist

9)	kRedirectedToBlog	SDK will give **kRedirectedToBlog** result code when user clicks on blog link.
10)	kRedirectedToInstall	SDK will give **kRedirectedToInstall** result code when user clicks on installation of application link.
11)	kRedirectedToInstagram	SDK will give **kRedirectedToInstagram** result code when user clicks on instagram link.
12)	kRedirectedToSlideshare	SDK will give **kRedirectedToSlideshare** result code when user clicks on slideshare link.
13)	kRedirectedToAnnotation	SDK will give **kRedirectedToAnnotation** result code when user clicks on annotation link.
14)	kNetworkNotAvailableError	SDK will give **kNetworkNotAvailableError** result code when network is not available
15)	kUnknownError	SDK will give **kUnknownError** result code when there is an UNKNOWN ERROR

Sample Code for BOOM Offerlist Video Player Integration

https://github.com/BOOMGitRepo/BOOMVideo_iOS_Example