**Using iOS SDK Sample App**

This guide will help you use the Sample app provided by Conviva along with iOS SDK.

**Getting Started**

Following are the features of provided Sample app

* The Sample app comes with a Custom player module.
* Conviva iOS SDK is already integrated, so in order to run the app, you don’t have to integrate it.
* For detailed integration steps, please follow the integration guide from Integration page of <https://community.conviva.com/>
* This readme will help you understand the UI element used in the app and their functionality.
* Two Sample app are available in both Objective C and Swift languages.
* Replace "CUSTOMER\_ID" with your own customer id in ViewController.m file
* Replace "TOUCHSTONE\_URL" with your own customer key in ViewController.m file
* Run pod install and launch the application

**Prerequisites and Downloading the app**

You need to download the sample app for Objective C/Swift from download page of <https://community.conviva.com/>:

Once downloaded, you can open the app with XCode.

After opening the app, run it and follow the actions in following order.

**Note: Since there are dependencies among the actions passed through UI, hence the flowing order is important to make sure that the app is working fine without producing any errors and reporting the metadata and metrics properly**

**Running the Sample App**

In the app’s UI, multiple options can be seen. The order in which the buttons are pressed is important. A walkthrough of Sample apps UI will make you understand better the flow in which you should complete your integration process.

The standard sequence of UI actions in order to report the metrics and metadata properly is:

**Init SDK Button**

This should be your first action. This action will instantiate the Conviva iOS SDK with the provided Customer Key and Gateway URL. By default, it takes the Customer ID value which is set in the Customer key UITextField and the Gateway URL value which will be set in the code. You can set/override the values for Customer Key and Gateway URL in initSDKClicked method.

**Create Session Button**

This will create the Conviva session with pre-defined values of metadata. In case you want to change the metadata values, do so in createMetadataObject method.

After these 2 actions, Conviva SDK should be iniitalized and Conviva Session should be created.

**Attach Player Button**

This action is used for instantiating the player. For this, attachPlayer

method is used. After instantiating the player, CISPlayerStateManager

class is instantiated. Instances of CISPlayerStateManager & Player are passed in PlayerInterface. Finally, current CISPlayerStateManager is attached to the session.

Also, if a player is already attached, don’t press this button again, if pressed again, it will show “detach current PlayerStateManager first” error. Right approach is to detach the already attached player then attach again.

**Player State Segment:**

Once you have attached player, you can report any of the 4 player states using this Player State Segment

**Detach Player Button**

This is used for detaching the already attached player. Do it when video player currently attached is no longer relevant for the current session.

**Clean Session Button**

Tap this button to clean up the session

**Clean Up SDK Button**

Tap this button to clean up the SDK.

**Report Error**

Tap this button to report any custom errors as described in the integration page.

**Ad Start & Ad End Buttons**

Tap this button to report the start and end of the ad playback.

**Ad roll selection Segment**

You can select when will the ad be played. i.e.:

* Before staring of the main content (Pre-roll),
* In between main content (Mid-roll) or
* after the completion of the main content (Post-roll)

**Clean PSM**

Tap this button to Clean up the Player state manager.

**Set Bitrate Button and UITextfield**

This button reports the Bitrate value entered through the UITextField next to Set Bitrate button. It should be set after attaching the player.

**Send Event Global Button**

This button reports the Custom event value entered through the UITextField next to Send Event Global Button

**Update Metadata Button**

This button updates missing content metadata for an existing monitoring session. In case you want to change the metadata values which are to be updated, do so in updatedMetadataObject method.