

Kony FFI

Atalsoft MobileImage Kony FFI Developer's Guide

Version: 3.0.0

Date: 2016-06-14

©2013- 2016 Atalasoftware, 116 Pleasant St, Suite 321, Easthampton, MA 01027, U.S.A. All rights reserved.
Use is subject to license terms.

Third-party software is copyrighted and licensed from Atalasoftware's suppliers.

THIS SOFTWARE CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF ATALASOFTWARE, INC. USE, DISCLOSURE OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF KOFAX.

Atalasoftware, the Atalasoftware logo, and the Atalasoftware product names stated herein are trademarks or registered trademarks of Atalasoftware, Inc. in the U.S. and other countries. All other trademarks are the trademarks or registered trademarks of their respective owners. U.S. Government Rights Commercial software. Government users are subject to the Kofax. standard license agreement and applicable provisions of the FAR and its supplements.

You agree that you do not intend to and will not, directly or indirectly, export or transmit the Software or related documentation and technical data to any country to which such export or transmission is restricted by any applicable U.S. regulation or statute, without the prior written consent, if required, of the Bureau of Export Administration of the U.S. Department of Commerce, or such other governmental entity as may have jurisdiction over such export or transmission. You represent and warrant that you are not located in, under the control of, or a national or resident of any such country.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Table of Contents

Preface	4
Getting Help for Atalasoftware Products	4
Overview	5
Recommendations	5
Known Limitations	5
The Foreign Function Interface	6
The Utilities Class	6
Licenses	6
The UI Controls Class	8
The Camera Control Class	8
The Bar Code Control Class	15
The Credit Card Control Class	18
The Image Review Edit Control Class	21
The Image Array	24
ImageArray	24
The Image Processor	30
ImageProcessor	30
JSON Definitions	35
General Response Structure	35
Camera Options	36
Check Capture Experiences	36
Image Review Edit Options	39
Image Array Options	40
Barcode Capture Control Options	40
Quick Analysis Options	40
Building the Sample App	41
Using the Atalasoftware MobileImage Kony FFI in Other Apps	42
Manually Integrate the FFI Library and Mappings into the Kony Sample App	42
Create Your Own FFI on Top of the SDK	43

Preface

This guide includes the information you need to successfully integrate Atalasoft MobileImage Kony FFI into your mobile project.

Getting Help for Atalasoft Products

Atalasoft regularly updates the Atalasoft Support site with the latest information about Atalasoft products. To access some resources, you must have a valid Support Agreement with an authorized Atalasoft Reseller/Partner or with Atalasoft directly.

Use the tools that Atalasoft provides for researching and identifying issues. For example, use the Atalasoft Support site to search for answers about messages, keywords, and product issues. To access the Atalasoft Support page, go to <http://www.atalasoft.com/support>.

The Atalasoft Support page provides:

- Product information and release news
- Downloadable product documentation
- Access to product knowledge bases
- Access to support cases
- Direct contact information

Use these tools to find answers to questions that you have, to learn about new functionality, and to research possible solutions to current issues.

Chapter 1

Overview

The *Atalsoft MobileImage Kony FFI Developer's Guide* provides an overview of developing Atalsoft MobileImage Kony FFI implementations. This document includes information on creating a sample app that can be used to demonstrate the functionality of the SDK.

Recommendations

- Use the `getImageBase64` API for getting image data to display in any image widget.
- Use the `getImageRawBytes` API to send/upload an image to any Web server.

Known Limitations

- Kony applications do not support 64-bit applications.
- For Android, the response object from any method needs to be formatted as a JSON object in JavaScript, since the response from the Android FFI is returned as an Object (String).
- For Android, the application state is not maintained. This is a Kony limitation.
 - Whenever an app is launched by clicking the app icon, by default the launcher activity is launched.
 - This launcher activity is created dynamically while the application is being built.
- For iOS 9.0 and above:
 - Please set "Enable Bitcode" to NO.
 - If there are any existing dylib files, remove them and add tbd equivalents.
 - To the `info.plist` add a dictionary named 'App Transport Security Settings' and add a key value pair of 'Allow Arbitrary Loads' & 'YES'.
 - At the moment, Kony Studio (6.0.3 GA) doesn't support Android 6 (API level 23) and above.

Chapter 2

The Foreign Function Interface

A Foreign Function Interface (FFI) provides a mechanism for a program written in one language to call routines or use services in some other language. The Atalasoftware MobileImage Kony FFI has four classes: KofaxUtilities, KofaxUIControls, KofaxImageArray, and KofaxImageProcessor. The following sections describe these classes and their associated APIs.

The Utilities Class

This class handles all the license and error-related APIs.

Licenses

An instance of this class contains the properties and methods used to set license usage. Once you set the license, the other library API methods work as designed. The license remains valid while the app is still running. Setting the license is required to unlock the library's features.

Native Android License

Package name: `com.kofax.utilities`

Class Name: License

Native iOS License

Class Name: KofaxLicense

JavaScript Constructor

```
var License = Kofax.Utilities.License();
```

License API (FFI Mappings)

getSDKVersion()

Parameters

None

Return Value

JSON object

Description

Gets the version of the SDK being used.

Android Native Mapping

ava.lang.Object getSDKVersion ()

iOS Native Mapping

-(NSDictionary*) getSDKVersion

setMobileSDKLicense()

Parameters

License String

Return Value

JSON object

Description

Method to set a valid license obtained from Kofax.

Android Native Mapping

ava.lang.Object setMobileSDKLicense (java.lang.String license)

iOS Native Mapping

-(NSDictionary*) setMobileSDKLicense:(NSString*) license

getDaysRemaining ()

Parameters

None

Return Value

JSON object

Description

Gets the number of days remaining before the license expires.

Android Native Mapping

java.lang.Object getDaysRemaining ()

iOS Native Mapping

-(NSDictionary*) getDaysRemaining

Sample Code

This method returns a response dictionary that contains the status of the action performed.

```
//Creates an object of class 'License'  
var LicenseObject = new Kofax.Utilities.License();  
//Invokes method 'setMobileSDKLicense' on the object  
var returnedValue = LicenseObject.setMobileSDKLicense(
```

```
/**String*/ license);
```

The UI Controls Class

This class contains the Camera Control, Bar Code Control, Credit Card Control, and the Image Review Edit Control classes.

The Camera Control Class

This class handles all the capture-related APIs. Use this class to invoke the appropriate capture experience by passing the type of experience in the constructor. The JSON definition for the camera includes the `captureexperience` key, which specifies the type of experience.

There are two types of experiences: the camera JSON definition "0-CheckCapture Experience" and the "1-Uniform Guidance Experience" JavaScript Constructor:

Native Android

Package name: `com.kofax.capture`

Class Name: `Camera`

Native iOS

Class name: `KofaxCaptureControl`

Camera API

show(Options)

Parameters

JSON

Return Value

None

Description

Adds the Camera view depending on the type passed. The default type is Uniform Guidance.

Android Native Mapping

`show`

iOS Native Mapping

`showCamera`

set(Options)

Parameters

JSON

Return Value

JSON Object

Description

The Camera Options have both a type and set of options. Options for the default camera would be `{type: default, Options:{levelindicator:false}}`.

Android Native Mapping

setOptions

iOS Native Mapping

setOptions

get(Options)

Parameters

JSON

Return Value

JSON Object

Description

Retrieves the options on the current camera depending on the type. For example, if the camera currently shows animated guides, the result will contain the type and options related to that status.

Android Native Mapping

getOptions

iOS Native Mapping

getOptions

takePicture()

Parameters

NONE

Return Value

NONE

Description

Takes picture on the active camera and triggers the callback specified in `addImageCapturedListener ()` with the captured image.

Android Native Mapping

takePicture

iOS Native Mapping

takePicture

takePictureContinually()

Parameters

NONE

Return Value

NONE

Description

Takes pictures continually on the active Camera, and triggers a call back with the captured images.

Android Native Mapping

takePictureContinous

iOS Native Mapping

takePictureContinually

addFocusListener()

Parameters

focusListener - function

Return Value

NONE

Description

Function called on focus events.

Android Native Mapping

addFocusListener

iOS Native Mapping

addFocusListener

addStabilityListener()

Parameters

stabilityListener - function

Return Value

NONE

Description

Function called on stability events.

Android Native Mapping

addStabilityListener

iOS Native Mapping

addStabilityListener

addPagedetectionListener()

Parameters

pageDetectionListener - function

Return Value

NONE

Description

Function called on page detection events.

Android Native Mapping

addPageDetectionListener

iOS Native Mapping

addPageDetectionListener

addLevelnessListener()

Parameters

levelnessListener - function

Return Value

NONE

Description

Function called on levelness events.

Android Native Mapping

addLevelnessListener:

iOS Native Mapping

addLevelnessListener:

addImageCapturedListener()

Parameters

imageCapturedListener - function

Return Value

NONE

Description

Function called on image captured event.

Android Native Mapping

addImageCapturedListener

iOS Native Mapping

addImageCapturedListener

addCameraInitialisationListener()

Parameters

Function cameraInitialisationCallback

Return Value

NONE

Description

Function which gets called when the camera is added to the view and initialized.

Android Native Mapping

addCameraInitialisationListener

iOS Native Mapping

addCameraInitialisationListener

addCameraInitialisationFailedListener()

Parameters

Function cameraInitialisationFailedCallback

Return Value

NONE

Description

Function which gets called when the camera initialization failed when it is added to the view.

Android Native Mapping

addCameraInitialisationFailedListener

iOS Native Mapping

addCameraInitialisationFailedListener

removeFocusListener()

Parameters

NONE

Return Value

NONE

Description

Removes the listener added for focus events.

Android Native Mapping

removeFocusListener

iOS Native Mapping

removeFocusListener

removeStabilityListener()

Parameters

NONE

Return Value

NONE

Description

Removes the listener added for stability events.

Android Native Mapping

removeStabilityListener

iOS Native Mapping

removeStabilityListener

removePagedetectionListener()

Parameters

NONE

Return Value

NONE

Description

Removes the listener added for Page detection events.

Android Native Mapping

removePageDetectListener

iOS Native Mapping

removePageDetectListener

removeLevelnessListener()

Parameters

NONE

Return Value

NONE

Description

Removes the listener added for level events.

Android Native Mapping

removeLevelnessListener

iOS Native Mapping

removeLevelnessListener

removeImageCapturedListener()

Parameters

NONE

Return Value

NONE

Description

Removes the listener added for image captured events.

Android Native Mapping

removeImageCapturedListener

iOS Native Mapping

removeImageCapturedListener

removeCameraInitialisationListener()

Parameters

NONE

Return Value

NONE

Description

Removes the listener for camera initialization event.

Android Native Mapping

removeCameraInitialisationListener

iOS Native Mapping

removeCameraInitialisationListener

Sample Code

```
//Create an object of Camera class.
    camera = new Kofax.UI.Camera(cameraOptions.cameraoptions.captureexperience);
//Launch the camera with set of options Here cameraOptions is the JSON definition for the camera.
camera.
    camera.show(cameraOptions);
//Add camera initialization listener.This call back is received when camera is
initialized.Any actions to be performed after the initialization can be executed
in this listener viz. The TakePicture method can be called as the call back is received.
    camera.addCameraInitialisationListener(cameraInitialized);
//Add image captured listener.This call back is received when image is captured.
Any actions to be performed after the capture can be executed in this listener.
    camera.addImageCapturedListener(imagecaptured);
```

The Bar Code Control Class

This class handles all the bar code related APIs.

Native Android

Package name: com.kofax.capture

Class Name: Barcode

Native iOS

Class Name: KofaxBarcodeControl

JavaScript Constructor

```
var barcode = new Kofax.UI.BarcodeControl();
```

This view provides visual guidance to the end user to maximize the chances of correctly decoding a bar code. Once the view has been asked to read a bar code, it will search continuously until one is found.

Bar Code Control API**show(Options)****Parameters**

Options -- JSON Object

Return Type

NONE

Description

Shows bar code view.

Android Native Mapping

void show (java.util.Hashtable options)

iOS Native Mapping

show: (NSDictionary*) options

setOptions(Options)

Parameters

Options - JSON Object

Return Type

JSON object

Description

Options to set on bar code.

Android Native Mapping

void setOptions (java.util.Hashtable barcodeOptions)

iOS Native Mapping

setOptions: (NSDictionary*) options

getOptions()

Parameters

NONE

Return Type

JSON object

Description

Retrieves the properties of the Barcode Capture Control and sends them to the callback function.

Android Native Mapping

java.lang.Object getOptions ()

iOS Native Mapping

getOptions

readBarCode()

Parameters

NONE

Return Type

JSON object

Description

Reads a single bar code from the camera view.

Android Native Mapping

void readBarcode ()

iOS Native Mapping

readBarcode

addBarcodeFoundListener()

Parameters

barcodeFoundListener - function

Return Type

NONE

Description

Function called when a bar code is found.

Android Native Mapping

void addBarcodeFoundListener (com.konylabs.vm.Function barcodeFoundListener)

iOS Native Mapping

addBarcodeFoundListener:(id *)callBack

removeBarcodeFoundListener()

Parameters

NONE

Return Type

NONE

Description

Removes the listener added for the bar code found event.

Android Native Mapping

void removeBarcodeFoundListener ()

iOS Native Mapping

removeBarcodeFoundListener

remove()**Parameters**

NONE

Return Type

NONE

Description

Removes the bar code control.

Android Native Mapping

void remove ()

iOS Native Mapping

remove

Sample Code

```
//Creates an object of class 'BarcodeControl'  
var barcode = new Kofax.UI.BarcodeControl();  
  
//Add bar code found listener.Called when barcode is found  
barcode.addBarCodeFoundListener(barcodeFoundListener);  
  
//Shows the barcodes with the specified options  
barcode.show(barcodeOptions);  
  
//Reads the barcode and triggers barcode found event  
barcode.readBarcode();
```

The Credit Card Control Class

This class handles credit card related APIs.

Native Android

Package name: com.kofax.capture

Class Name: CreditCardWrapper

Native iOS

Class Name: KofaxCreditCardControl

JavaScript Constructor

```
var creditcard = new Kofax.UI.CreditCardControl();
```

Credit Card Control API

show()

Parameters

NONE

Return Type

NONE

Description

Shows credit card view.

Android Native Mapping

void show ()

iOS Native Mapping

-(void) show

addCardExtractedListener()

Parameters

cardExtractedListener-Function

Return Type

NONE

Description

Function called when credit card data is extracted.

Android Native Mapping

void show () void addCardExtractedListener (com.konylabs.vm.Function cardExtractionListener)

iOS Native Mapping

addCardExtractedListener:(id)callBack

addCreditCardExtractionFailedListener()

Parameters

creditCardExtarctionFailedListener-Function

Return Type

NONE

Description

Function called when credit card extraction fails.

Android Native Mapping

void addCreditCardExtractionFailedListener (com.konylabs.vm.Function
creditCardExtractionFailedListener)

iOS Native Mapping

addCreditCardExtractionFailedListener:(id)callBack

removeCardExtractedListener()

Parameters

NONE

Return Type

NONE

Description

Removes the listener for the credit card extracted event.

Android Native Mapping

void removeCardExtractedListener ()

iOS Native Mapping

removeCardExtractedListener

removeCreditCardExtractionFailedListener()

Parameters

NONE

Return Type

NONE

Description

Removes the listener for the credit card extraction failed event.

Android Native Mapping

void remove ()

iOS Native Mapping

removeCreditCardExtractionFailedListener

Sample Code

```
//Creates an object of class 'CreditCardControl'  
var creditCard = new Kofax.UI.CreditCardControl();  
  
//Add card extracted listener.Called when credit card extraction is successful  
creditCard.addCardExtractedListener(cardExtractionListener);  
  
//Invokes method 'show' on the object
```

```
creditCard.show();
```

The Image Review Edit Control Class

This class handles ImageEdit related APIs.

Native Android

Package name: `com.kofax.preview`

Class Name: `ImageReviewEditControl`

Native iOS

Class Name: `KofaxImageReviewEdit`

JavaScript Constructor

```
var imageReviewEdit = new Kofax.UI.ImageReviewEdit();
```

Image Review Edit API

show(Options)

Parameters

Options-JSONObject

Return Type

NONE

Description

Shows the Image Review and Edit.

Android Native Mapping

`void show java.util.Hashtable imageEditReviewOptions)`

iOS Native Mapping

`showImageReviewEdit:(NSDictionary *)options`

setOptions(Options)

Parameters

Options-JSONObject

Return Type

JSON object

Description

Options to set on the Image Review Edit.

Android Native Mapping

void setOptions (java.util.Hashtable ierOptions)

iOS Native Mapping

setImageReviewEditOptions:(NSDictionary*)options

getOptions()

Parameters

NONE

Return Type

JSON object

Description

getImageReviewEditOptions returns the options on the Image Review Edit.

Android Native Mapping

java.lang.Object getOptions ()

iOS Native Mapping

getImageReviewEditOptions

setImage ()

Parameters

image ID - String

Return Type

JSON object

Description

Displays the Image Object in Image Review Edit.

Android Native Mapping

java.lang.Object setImage (java.lang.String imageID)

iOS Native Mapping

setImageToImageReviewEdit:(NSString*)imageId

getImage ()

Parameters

NONE

Return Type

JSON object

Description

getImageOfImageReviewEdit returns the image of Image Review Edit.

Android Native Mapping

java.lang.Object getImage ()

iOS Native Mapping

getImageOfImageReviewEdit()

clearImage ()

Parameters

NONE

Return Type

NONE

Description

Clears the image in Image Review Edit.

Android Native Mapping

void clearImage ()

iOS Native Mapping

clearImageOfImageReviewEdit()

remove ()

Parameters

NONE

Return Type

NONE

Description

Removes the Image Review Edit.

Android Native Mapping

void remove ()

iOS Native Mapping

removeImageReviewEdit()

addCropTetragonListener()

Parameters

Function cropTetragonListnerCallback

Return Type

NONE

Description

Function which gets called when Tetragon is selected, and the user clicks Done in the ImageEditReview control.

Android Native Mapping

```
void addCropTetragonListener (com.konylabs.vm.Function cropTetragonListener)
```

iOS Native Mapping

```
addCropTetragonListener:(id)callBack
```

Sample Code

```
//create an object of class ImageReviewEdit
var imageReviewEdit = new Kofax.UI.ImageReviewEdit();

//Set the capturedImage to ReviewEdit
imageReviewEdit.setImage(capturedImageID);

//Show crop rectangle
imageReviewEdit.showCropRectangle(true);

//Add crop tetragon listener.Called when cropping is done
imageReviewEdit.addCropTetragonListener(cropTetragon);

//show EditReview with the specified options
imageReviewEdit.show(options);
```

The Image Array

This section describes the ImageArray class in detail.

ImageArray

This class handles all image related operations such as getting base64 from an image, "image write to file" and so on.

Native Android

Package name: com.kofax.engines

Class Name: ImageArray

Native iOS

Class Name: KofaxImageArray

JavaScript Constructor

```
var imageArray = Kofax.ImageArray.ImageArray();
```


ImageArray API

ImageArray()

Parameters

NONE

Return Value

Default constructor

Description

ImageArray().

Android Native Mapping

ImageArray

iOS Native Mapping

-(id) init

getImageIDs()

Parameters

NONE

Return Value

JSON Array

Description

Returns all Image IDs as JSON array.

Android Native Mapping

java.lang.Object getImageIDs ()

iOS Native Mapping

-(NSDictionary*) getImageIDs

removeImages(JSONArray)

Parameters

imageIDs- JSONArray

Return Value

NONE

Description

Removes the images for given IDs.

Android Native Mapping

void removeImages (java.util.Hashtable imageIDs)

iOS Native Mapping

-(void) removeImages:(NSDictionary*) imageIDs

removeAllImages()

Parameters

NONE

Return Value

NONE

Description

Clears the ImageArray.

Android Native Mapping

void removeAllImages ()

iOS Native Mapping

-(void) removeAllImages

writImageToFile(JSONObject)

Parameters

JSON Object

Return Value

NONE

Description

Writes image to a specific file path.

Android Native Mapping

java.lang.Object writImageToFile (java.util.Hashtable filePathDetails)

iOS Native Mapping

-(NSDictionary*) writImageToFile:(NSDictionary*) filePathDetails

readImageFromFile(ImageID)

Parameters

imageID-String

Return Value

NONE

Description

Reads the image bitmap from a file.

Android Native Mapping

java.lang.Object readImageFromFile (java.lang.String imageID)

iOS Native Mapping

-(NSDictionary*) readImageFromFile:(NSString*) imageID

clearImageBitmap(ImageID)

Parameters

imageID-String

Return Value

NONE

Description

Clears the bitmap for the given image ID.

Android Native Mapping

java.lang.Object clearImageBitmap (java.lang.String imageID)

iOS Native Mapping

-(NSDictionary*) clearImageBitmap:(NSString*) imageID

deleteImageFromDisk(ImageID)

Parameters

imageID-String

Return Value

NONE

Description

Deletes the image from the disk for the given image ID.

Android Native Mapping

java.lang.Object deleteImageFromDisk (java.lang.String imageID)

iOS Native Mapping

-(NSDictionary*) deleteImageFromDisk:(NSString*) imageID

setImageProperties(Options)

Parameters

Options - JSON Object

Return Value

NONE

Description

Sets the properties for the given image ID.

Android Native Mapping

java.lang.Object setImageProperties (java.util.Hashtable jsonProperties)

iOS Native Mapping

-(NSDictionary*) setImageProperties:(NSDictionary*) jsonProperties

getImageProperties(imageID)

Parameters

imageID-String

Return Value

JSON Object

Description

Gets the properties of the given Image ID in JSON format.

Android Native Mapping

java.lang.Object getImageProperties (java.lang.String imageID)

iOS Native Mapping

-(NSDictionary*) getImageProperties:(NSString*) imageID

getImageRawBytes(imageID)

Parameters

imageID-String

Return Value

JSON Array with raw bytes object

Description

Returns the image raw bytes(blob) for the given image ID.

Android Native Mapping

java.lang.Object getImageBase64 (java.lang.String imageID)

iOS Native Mapping

-(NSDictionary*) getImageRawBytes:(NSString*) imageID

getImageBase64(imageID)

Parameters

imageID-String

Return Value

String

Description

Returns the base64 string of the given image ID.

Android Native Mapping

java.lang.Object getImageBase64 (java.lang.String imageID)

iOS Native Mapping

-(NSDictionary*) getImageBase64:(NSString*) imageID

getImageFromBase64(base64)

Parameters

Base64 - String

Return Value

JSON Object

Description

Returns the image object in JSON format.

Android Native Mapping

java.lang.Object getImageFromBase64 (java.lang.String strBase64)

iOS Native Mapping

-(NSDictionary*) getImageFromBase64:(NSString*) strBase64

getImageFromFilePath(Options)

Parameters

Options -- JSON Object

Return Value

JSON Object

Description

Returns the image object in JSON format.

Android Native Mapping

java.lang.Object getImageFromFilePath (java.util.Hashtable filePathDetails)

iOS Native Mapping

-(NSDictionary*) getImageFromFilePath:(NSDictionary*) filePathDetails

Sample Code

```
//Creates an object of class 'ImageArray'
var ImageArrayObject = new Kofax.ImageArray.ImageArray();
//Invokes method 'getImageIDs' on the object
var returnedValue = ImageArrayObject.getImageIDs();
```

The Image Processor

The following section describes the ImageProcessor object in detail.

ImageProcessor

The ImageProcessor object is an image processing engine that can process a single image according to settings contained in either an ImagePerfectionProfile object, or a BasicSettingsProfile object, but not both simultaneously.

Native Android

Package name: com.kofax.engines

Class Name: Processor

Native iOS

Class Name: KofaxImageProcessor

JavaScript Constructor

```
var ImageProcessorObject = new Kofax.ImageProcessor.ImageProcessor();
```

ImageProcessor API

getOptions()**Parameters**

NONE

Return Value

JSON object containing options

Description

getImageProcessorOptions retrieves the options for the Image Processor.

Android Native Mapping

java.lang.Object getOptions ()

iOS Native Mapping

-(NSDictionary*) getOptions

specifyProcessedImageFilepath()

Parameters

Filepath - String

Return Value

JSON object

Description

Specifies the file path for the Image Processor.

Android Native Mapping

java.lang.Object specifyProcessedImageFilepath (java.lang.String filePath)

iOS Native Mapping

-(NSDictionary*) specifyProcessedImageFilepath:(NSString*) filePath

getProcessedImageFilepath()

Parameters

NONE

Return Value

JSON object

Description

Retrieves the file path of the Image Processor.

Android Native Mapping

java.lang.String getProcessedImageFilepath ()

iOS Native Mapping

-(NSString*) getProcessedImageFilepath

processImage(imageID,Options)

Parameters

Image ID - String, Options --JSONObject

Return Value

JSON object

Description

Processes the specified image.

Android Native Mapping

java.lang.Object processImage (java.lang.String imageID, java.util.Hashtable imageProcessorOptions)

iOS Native Mapping

-(NSDictionary*) processImage:(NSString*) imageID withOptions:(NSDictionary*) imageProcessorOptions

cancelProcessing()

Parameters

NONE

Return Value

NONE

Description

Cancels the image processing.

Android Native Mapping

void cancelProcessing()

iOS Native Mapping

void cancelProcessing()

doQuickAnalysis(imageID,Options)

Parameters

ImageID - String, Options -JSONObject

Return Value

JSON object

Description

Checks the image quality and returns the QuickAnalysis Feedback.

Android Native Mapping

void doQuickAnalysis (java.lang.String imageID, java.util.Hashtable options)

iOS Native Mapping

-(void) doQuickAnalysisForImage:(NSString*) imageID withSettings:(NSDictionary*) options

getQuickAnalysisOptions()

Parameters

NONE

Return Value

JSON object containing options

Description

Returns a JSON object.

Android Native Mapping

java.lang.Object getQuickAnalysisOptions ()

iOS Native Mapping

-(NSDictionary*) getQuickAnalysisOptions

addImageOutEventListener()

Parameters

imageOutEventListner - function

Return Value

NONE

Description

Function called when an image is processed.

Android Native Mapping

void addImageOutListener (com.konylabs.vm.Function imageOut)

iOS Native Mapping

-(void) addImageOutEventListener:(Callback*) imageOut

addProcessProgressListener()

Parameters

processProgressListener - function

Return Value

NONE

Description

Function called for "image processing progress."

Android Native Mapping

void addProcessProgressListener (com.konylabs.vm.Function processProgressCallback)

iOS Native Mapping

-(void) addProcessProgressListener:(Callback*) processProgressCallback

addAnalysisCompleteListener()

Parameters

analysisCompleteListener - function

Return Value

NONE

Description

Function called for "image analysis progress."

Android Native Mapping

void addAnalysisCompleteListener (com.konylabs.vm.Function analysisCompleteCallback)

iOS Native Mapping

-(void) addAnalysisCompleteListener:(Callback*) analysisCompleteCallback

removeImageOutEventListener()

Parameters

NONE

Return Value

NONE

Description

Removes the listener added for the image processed.

Android Native Mapping

void removeImageOutListener ()

iOS Native Mapping

-(void) removeImageOutEventListener

removeProcessProgressListener()

Parameters

NONE

Return Value

NONE

Description

Removes the listener added for "image process progress."

Android Native Mapping

void removeProcessProgressListener ()

iOS Native Mapping

-(void) removeProcessProgressListener

removeAnalysisCompleteListener()**Parameters**

NONE

Return Value

NONE

Description

Removes the listener added for "image analysis progress."

Android Native Mapping

void removeAnalysisCompleteListener ()

iOS Native Mapping

-(void) removeAnalysisCompleteListener

Sample Code

```
//Creates an object of class 'ImageProcessor'
var ImageProcessorObject = new Kofax.ImageProcessor.ImageProcessor();
//Invokes method 'doQuickAnalysis' on the object
ImageProcessorObject.doQuickAnalysis(imageID, options);
```

JSON Definitions

This section contains definitions for JSON formatted items.

General Response Structure

All JSON objects returned by the APIs have the structure defined here. Only if success is TRUE does the response contain valid data.

```
{
  success:true/false,
  errorInfo:{
    errorMsg: String,
    errorDesc: String
  },
  response:{
    method specific key: value
  }
}
```

Camera Options

The JSON definitions in this section consist of three main keys used to configure the SDK camera:

- Cameraoptions
- checkcaptureoptions
- documentcaptureoptions

Cameraoptions

These options are common irrespective of the selected type of experience. The following JSON structure contains default values that can be configured in the application. These values are set to properties in the `kfxKUIImageCaptureControl` class in the Atalasoft MobileImage Kony FFI.

iOS

Framework: `kfxLibUIControls`

Class: `kfxKUIImageCaptureControl`

Android

Package: `com.kofax.kmc.kui.uicontrols`

Class: `ImageCaptureView`

Check Capture Experiences

There are two types of experience: `UniformGuidance` and `Check Capture Experience`. The experience shown to the user depends on the value of the JSON capture experience key.

CheckCaptureExperience(captureexperience:0)

If the selected experience is "checkcapture," values from `checkcaptureoptions` are set as properties in the `kfxKUICheckCaptureExperience` class in the Mobile SDK.

iOS

Framework: `kfxLibUIControls`

Class: `kfxKUICheckCaptureExperience`

Android

Package: `com.kofax.kmc.kui.uicontrols.captureanimations`

Class: `CheckCaptureExperience`

CheckCaptureExperience(captureexperience:1)

If the selected experience is "uniform guidance," values from `checkcaptureoptions` are set as properties in the `kfxKUIDocumentCaptureExperience` class in the Mobile SDK.

iOS

Framework: `kfxLibUIControls`

Class: kfxKUIDocumentCaptureExperience

Android

Package: com.kofax.kmc.kui.uicontrols.captureanimations

Class: DocumentCaptureExperience

Check Capture Options

```

cameraoptions : {
    flashmode : "OFF",
    deviceDeclinationPitch : 0,
    deviceDeclinationRoll : 0,
    levelThresholdPitch : 7,
    levelThresholdRoll : 7,
    stabilityDelay : 75,
    videoFrame : false,
    continuousCaptureMode : false,
    captureExperience : 1
},
checkCaptureOptions : {
    tutorialEnabled : true,
    vibrationEnabled : true,
    outViewFinderColor : "#00000000",
    checkDetectionSettings : {
        checkSide : 0,
        targetFrameAspectRatio : 2.18,
        targetFramePaddingPercent : 5,
        targetFrameCenter : {
            x : null,
            y : null
        },
        zoomMinFillFraction : 0.65,
        zoomMaxFillFraction : 1.5,
        turnSkewAngleTolerance : 10,
        tolerance : 0.15
    },
    captureCriteria : {
        stabilityThresholdEnabled : true,
        stabilityThreshold : 95,
        focusEnabled : true,
        rollThresholdEnabled : false,
        rollThreshold : 7,
        pitchThreshold : 7,
        pitchThresholdEnabled : false
    },
    capturedMessage : {
        backgroundColor : "#70000000",
        textColor : "#01DF01",
        message : "Done",
        visible : true
    },
    centerMessage : {
        backgroundColor : "#70000000",
        textColor : "#FFFFFF",
        message : "Center the Check",
        visible : true
    },
    userInstructionMessage : {
        backgroundColor : "#00000000",
        textColor : "#FFFFFF",
        message : "Fill viewable area with check",
        visible : true
    },
},

```

```

zoominmessage : {
backgroundcolor : "#70000000",
textcolor : "#FFFFFFF",
message: "Move Closer",
visible : true
},
zoomoutmessage : {
backgroundcolor : "#70000000",
textcolor : "#FFFFFFF",
message: "Move Back",
visible : true
},
holdsteadymessage: {
backgroundcolor: "#70000000",
textcolor: "#FF0000",
message: "Hold Steady",
visible: true
}
}

```

Uniform Guidance Options

```

cameraoptions : {
flashmode : "OFF",
devicedeclinationpitch : 0,
deviceDeclinationRoll : 0,
levelthresholdpitch : 7,
levelthresholdroll : 7,
stabilitydelay : 75,
videoframe : false,
continuouscapturemode:false,
captureexperience:1
},
documentcaptureoptions : {
capturecriteria : {
focusenabled : true,
pitchthreshold : 15,
pitchthresholdenabled : true,
rollthreshold : 15,
rollthresholdenabled : true,
stabilitythreshold : 95,
stabilitythresholdenabled : true
},
capturedmessage : {
backgroundcolor : "#70000000",
textcolor : "#01DF01",
message: "Done",
visible :true
},
centermessage : {
backgroundcolor : "#70000000",
textcolor : "#FFFFFFF",
message:"Center the document",
visible : true
},
documentdetectionsettings : {
targetframecenter : {
x : 0.00,
y : 0.00
},
longaxisthreshold : 85,
shortaxisthreshold : 85,
targetframepaddingpercent : 5,
targetframeaspectratio:0.75,
tolerance : 0.15,
turnskewangletolerance : 10,
zoommaxfillfraction : 1.5,
zoomminfillfraction : 0.2
},
},
}

```

```

holdsteadymessage : {
  backgroundcolor : "#70000000",
  textcolor : "#FF0000",
  message: "Hold Steady",
  visible : true
},
userinstructionmessage : {
  backgroundcolor : "#00000000",
  textcolor : "#FFFFFFF",
  message: "Fill the area with the document",
  visible : true
},
zoominmessage : {
  backgroundcolor : "#70000000",
  textcolor : "#FFFFFFF",
  message: "Move Closer",
  visible : true
},
zoomoutmessage : {
  backgroundcolor : "#70000000",
  textcolor : "#FFFFFFF",
  message: "Move Back",
  visible : true
},
outerviewfindercolor : "#70000000",
tutorialenabled : false,
vibrationenabled : true
}

```

Image Review Edit Options

The following JSON definition specifies the tetragon corners used to crop the image. Crop line, crop corner color, and crop style can also be configured.

iOS

Framework: kfxLibUIControls

Class: kfxKUIImageReviewAndEdit

Android

Package: com.kofax.kmc.kui.uicontrols

Class: ImgReviewEditCntrl

```

{
  Tetragon : {
    show : true,
    TopLeft : {
      x : 0,
      y : 0
    },
    TopRight : {
      x : 0,
      y : 0
    },
    BottomLeft : {
      x : 0,
      y : 0
    },
    BottomRight : {
      x : 0,
      y : 0
    }
  },
}

```

```
Crop :{
  lineColor : "#AABBCDD",
  cornerColor : "#AABBCDD",
  lineStyle : "LINE_STYLE_SOLID"
```

Image Array Options

```
writeImageToFile method JSON is
{
  filePath : "Give Filepath Here",
  imageID: "Give Image ID Here"
}
getImageFromFilepath method JSON is
{
  filePath: "Give Filepath Here",
  mimeType: "MIMETYPE_JPG"
}
setImageProperties method JSON is
{
  imageID: "Give imageID Here",
  mimeType: "MIMETYPE_JPG",
  dpi: 75,
  tag: 1,
  filePath: "Give Filepath Here",
  createDateTime: "Give ISO 8601 Date Format String Here",
  jpegQuality: 75
}
```

Barcode Capture Control Options

The following options are used to configure BarCodeCaptureControl.

iOS

Framework: kfxLibUIControls

Class: kfxKUIBarCodeCaptureControl

Android

Package: com.kofax.kmc.kui.uicontrols

Class: BarCodeCaptureView

```
{
  searchDirection:["HORIZONTAL","VERTICAL"],
  symbologies:["PDF417","CODE39"],
  guidingLine:"LANDSCAPE"
}
```

Quick Analysis Options

Use the quickanalysis settings to get feedback on image quality.

iOS

Framework: kfxLibEngines

Class: kfxKEDQuickAnalysisSettings

Android

Package: com.kofax.kmc.ken.engines.data

Class: QuickAnalysisSettings

```

{
    QuickAnalysisSettings = {
        enableBlurDetection = true;
        enableGlareDetection = true;
        enableSaturationDetection = true;
        generateImage = true;
        glareDetectedThreshold = "0.02";
        glareDetectionIntensityFraction = "0.03";
        glareDetectionIntensityThreshold = 230;
        glareDetectionMinimumGlareAreaFraction = "0.01";
        glareDetectionNumberOfTiles = 100;
    };
}

```

Building the Sample App

To build and use the sample application with the integrated FFI, follow these instructions. Atalasoftware MobileImage Kony FFI Mobile SDK

1. Import the sample application into Kony Studio.
2. Import `Kofax-Kony-FFI.zip` file (available in the SDK release folder) into the Atalasoftware MobileImage Kony FFI sample app.
 1. Right-click the sample application in Kony Studio and import a custom library.
 2. Select `Kofax-Kony-FFI.zip`.
3. Build the Kony application at least once by selecting a supported platform (Android or iOS).
4. Go to the Navigator tab in Kony Studio.
5. Select the Kony application.
6. Refresh the Kony application once.
7. Go to the `Resources>Customlibs>lib>Android` (or iOS) folder.
8. Paste the SDK libs from the `Jar` folder (or framework files for iOS).

Note For Android do not copy `repackaged.jar`.

9. For Android, copy the values folders (e.g. `values-de`) from the localization folder (`Android\MobileSDK_libs\jar\localization`) of the Mobile SDK to this path: `resources/mobile/native/android/` in the Navigator tab. Then, edit the `strings.xml` file in all the values folders to add a string by including this line: `<string name="app_name">Your app name</string>`.
10. For Android, copy or merge the drawables folders (e.g. `drawable-xhdpi`) from the drawables folder (`Android\MobileSDK_libs\jar\drawables`) of the Mobile SDK to this path: `resources/mobile/native/android/` in the Navigator tab.
11. A `Libkonyjsvm.so` file for each platform is bundled with the FFI. Replace the `Libkonyjsvm.so` file originally included in Kony Studio with the appropriate platform specific file that came with the Atalasoftware MobileImage Kony FFI.
12. Make a clean build of the Kony app. (For iOS, if the app is not launched automatically in Xcode, directly launch the project from the Kony workspace.)

Using the Atalasoftware MobileImage Kony FFI in Other Apps

In order to create a new Kony application and use/integrate the Atalasoftware MobileImage Kony FFI into it, follow these instructions.

1. Create an application using Kony Studio.
2. Import the `Kofax-Kony-FFI.zip` file (available in the SDK release folder) into the Atalasoftware MobileImage Kony FFI sample app.
 1. Right-click the sample application in Kony Studio and import a custom library.
 2. Select the `Kofax-Kony-FFI.zip` file.
3. Build the Kony application at least once by selecting a supported platform (Android or iOS).
4. Go to the Navigator tab in Kony Studio.
5. Select the Kony application.
6. Refresh the Kony application once.
7. Go to the `Resources>Customlibs>lib>Android` (or iOS) folder.
8. Paste the SDK libs from the `Jar` folder (or framework files for iOS).

Note For Android do not copy `repackaged.jar`.

9. For Android, Copy the values folders (e.g. `values-de`) from the localization folder (`Android\MobileSDK_libs\jar\localization`) of the Mobile SDK to this path: `resources/mobile/native/android/` in the Navigator tab. Then, edit the `strings.xml` file in all the values folders to add a string by including this line: `<string name="app_name">Your app name</string>`.
10. For Android, copy or merge the drawables folders (e.g. `drawable-xhdpi`) from the drawables folder (`Android\MobileSDK_libs\jar\drawables`) of the Mobile SDK to this path: `resources/mobile/native/android/` in the Navigator tab.
11. A `Libkonyjsvm.so` file for each platform is bundled with the Atalasoftware MobileImage Kony FFI. Replace the `Libkonyjsvm.so` file originally included in Kony Studio with the appropriate platform specific file that came with the Atalasoftware MobileImage Kony FFI.
12. Make a clean build of the Kony app. (For iOS, if the app is not launched automatically in Xcode, directly launch the project from the Kony workspace.)

Manually Integrate the FFI Library and Mappings into the Kony Sample App

1. Create an application using Kony Studio.
2. In the applications tab, select your Kony application and select **Integrate Third Party > Manage Custom Libraries**.
3. Right-click the JavaScript FFI Definition and give it a valid Javascript namespace.
4. Copy the Mobile SDK libraries into the platform appropriate folders.
5. Right-click the Javascript name space and create a Javascript class.
6. In the Native Mappings tag, click the import button to import the required native FFI library.
7. Map the Javascript class names to the native FFI class names as well as the Javascript method names with the FFI method names.
8. Clean up the application as needed.
9. Write the Javascript file in the `modules/JS` folder of the application with the business logic.
10. Run the application. (For iOS, if the app is not launched automatically in xCode, directly launch the project from the Kony workspace.)

11. Repeat steps 3-10 as needed to update the FFI integrations.

Create Your Own FFI on Top of the SDK

To create your own FFI (iOS or Android) on top of the Atalsoft MobileImage SDK and use it in a Kony application, refer to the following Kony documentation.

URL: http://docs.kony.com/konyonpremises/#../Subsystems/Studio_User_Guide/Content/Foreign_Function_Interface_JS.htm%3FTocPath%3DKony%2520Studio|Studio%2520User%2520Guide|Integrating%2520with%2520Third-party%2520Libraries%2520Using%2520FFI|_____0