Atalasoft MobileImage PhoneGap Plugin

Developer's Guide

Version: 3.4.0

Date: 2018-10-11

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

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

Getting help for Atalasoft products

Atalasoft regularly updates the Atalasoft Support site with the latest information about Atalasoft products.

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 www.atalasoft.com/support, where you can find a variety of resources and 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 Atalasoft MobileImage PhoneGap Plugin *Developer's Guide* provides an overview of developing Cordova applications using the Atalasoft Mobile Plugin for SDK. By adding the plugin to your mobile app, it can be used to capture, process bar code data received from mobile devices.

Note The earliest Cordova version supported by the Atalasoft MobileImage PhoneGap Plugin is version 5.0.

Chapter 2

The Atalasoft MobileImage PhoneGap Plugin for SDK

PhoneGap is an open source mobile application development framework, based upon the Apache Cordova project. See `cordova.apache.org` for documentation for details. The Atalasoft MobileImage PhoneGap Plugin for the mobile SDK in your mobile application can be used to capture and process images and bar code data received from mobile devices.

> **Note** PhoneGap is the Adobe branded version. Cordova is the generic open source Apache version. For the purposes of this guide, all references to "PhoneGap" can apply to either PhoneGap, or Cordova.

The plugin exposes much of the Atalasoft Mobile functionality, within a Cordova application. The plugin code calls existing SDK methods and sends the response back to the `KofaxCordovaPlugin` JavaScript code. `KofaxCordovaPlugin` contains native methods that are called from a hybrid application via `kfxMobilePlugin.js`

The following PhoneGap related files are provided:

File Name	Description
`kfxMobilePlugin.js` **Note** There are many additional Javascript files included in the `www directory`, which contain the classes referred to from the main Javascript file.	The plugin APIs are exposed via `kfxMobilePlugin.js`. All operations are exposed via the plugin objects:
`KofaxMobileSdkPlugin.jar`	This is the native part of the Android plugin. This part is responsible for interacting with native libraries.
`kfxMobileCordova.framework`	This is the native part of the iOS plugin. This part is responsible for interacting with native libraries.
`Plugin.xml`	This is the main part of the plugin. By using this, Cordova will install the plugin for the iOS and Android platforms.

The following should be taken into account when building an application using the plugin:

- Size of the UI control displayed may not be the same across all devices.
- Plugin calls are asynchronous. Consequently, it is good practice to put actions like "take picture" and "read bar code" in the success callback of the corresponding `addXXXView` method. Otherwise, on low-end devices, the API may not work as expected.
- All UI Controls will float on top of CordovaWebview. The same thing is true for any native controls added to an application using any plugin SDK feature.

- The developer has to manage memory issues. The plugin maintains an image array. If the image array has more than about 3 elements (depending on device memory capacity) the application may crash. Developers must be sure to remove unused images.
- The output image setting `jpegQuality` specifies the compression quality for the output jpg file created during image processing. This setting is applied only when both the following are true:
  - The mime type setting (`mimeType`) is set to `MIMETYPE_JPEG`
  - Representation is set to `IMAGE_REP_FILE`, or `IMAGE_REP_BOTH`

Create a new Atalasoft MobileImage PhoneGap Plugin application

Follow these steps to create and build a Atalasoft MobileImage PhoneGap Plugin application.

1. Create a new project with Cordova by running the following command:

   ```
   cordova create AtalasoftSampleApp com.kofax.sample AtalasoftSampleApp
   ```

   This creates a folder on your computer named `AtalasoftSampleApp`.
2. If there are any existing files in the `www` folder, copy and replace the existing contents in the `AtalasoftSampleApp/www` folder or create new files as required.
3. Change to the `AtalasoftSampleApp` folder.

   The commands for adding the plugin and platform must be executed in this folder.
4. Add KofaxPlugin by running the following command:

   ```
   cordova plugin add <kofaxPluginPath>
   ```

   Where <kofaxPluginPath> is the local system path. For example:

   ```
   .../Hybrid/PhoneGap/Plugins/com.kofax.mobile.plugins.sdk
   ```
5. For the Atalasoft SampleApp, add the necessary plugins with the following commands:

   ```
   cordova plugin add cordova.plugins.diagnostic
   cordova plugin add cordova-plugin-spinner-dialog
   cordova plugin add cordova-plugin-flashlight
   cordova plugin add cordova-plugin-file
   cordova plugin add cordova-plugin-screen-orientation
   cordova plugin add cordova-plugin-splashscreen
   cordova plugin add cordova-plugin-device
   cordova plugin add cordova-plugin-console
   cordova plugin add cordova-plugin-camera
   cordova plugin add cordova-plugin-dialogs
   ```
6. Add the appropriate platform by using either of these commands:
   - `cordova platform add android`
   - `cordova platform add ios`

Building the Android Atalasoft sample application

Follow these steps to build the Android Atalasoft SampleApp.

1. Copy the Android libraries from `Android/AtalasoftMobileImage_libs/aar` to `AtalasoftSampleApp/platforms/android/app/libs`.
2. In the `AtalasoftSampleApp/platforms/android/app` folder, open the build.gradle file and add the following lines in `dependencies {....}`.

```
repositories {
   flatDir {
      dirs 'libs'
   }
}
implementation (name: 'sdk-release', ext: 'aar')
```

> **Note** If you are using Android Studio 2. use `compile` instead of `implementation`.

3. Add the following lines in `Android { .... }`.

```
packagingOptions {
   exclude 'META-INF/LICENSE.txt'
   exclude 'META-INF/NOTICE.txt'
   exclude 'META-INF/DEPENDENCIES'
   exclude 'META-INF/LICENSE'
   exclude 'META-INF/NOTICE'
}
sourceSets.main {
   jniLibs.srcDir 'libs'
   jni.srcDirs=[] //disable automatic ndk-build call
}
```

4. Build the project by using the following command or open AndroidStudio and build the project to run on the device:

```
cordova build android
```

> **Note** If you are using a Mac, add `sudo`.

5. If the build fails, check the following:
   - If the build fails with a "com.android.tools.aapt2.Aapt2Exception: AAPT2 error: check logs for details" exception, do the following:
     a. In the `AtalasoftSampleApp/platforms/android` folder, create the gradle.properties file, if it does not already exist. Add the following code:

        ```
        android.enableAapt2=false
        ```

     b. In the `AtalasoftSampleApp/platforms/android/app` folder, create the build-extras.gradle file if it does not already exist. Add the following code:

        ```
        configurations.all {
           resolutionStrategy {
              force 'com.android.support:support-v4:27.1.0'
           }
        }
        ```

   - If the build fails with a "UnhandledPromiseRejectionWarning: Error: spawn EACCES" exception, run the following command:

     ```
     chmod 755 platforms/android/gradlew
     ```

6. Run the project with the following command.

```
cordova run android --device
```

Building the iOS Atalasoft sample application

Follow these steps to build the iOS Atalasoft SampleApp.

1. Open AtalasoftSampleApp.xcodeproj in the `AtalasoftSampleApp/platforms/ios` folder.

2. Unzip MobileSDK.zip file in the `/iOS/Frameworks` folder.
3. Copy the Atalasoft libraries from the `/iOS/Frameworks/MobileSDK` folder to the Frameworks section of the AtalasoftSampleApp.xcodeproj. Select **Copy items if needed** and click **Finish**.
4. Open the **Build Settings** for the project and add the following line under **Header Search Paths**.

   ```
   MobileSDK.framework/Headers
   ```
5. Build the project with the following command:

   ```
   cordova build ios
   ```
6. Run the project with the following command.

   ```
   cordova run ios --device
   ```

Required library and framework files

Please refer to *Getting Started with the SDK* chapter of the *Mobile SDK Developers* guide, for specific library and frameworks to include in order to use the Mobile SDK with your application.

Requirements

Certain frameworks must be included in your project.

For iOS

For an iOS project, the following frameworks are required.

- MobileSDK.framework
- SDKStrings.bundle
- uiimage.bundle

In order to integrate the Mobile SDK into an iOS app, you must:

1. Update the application project file to link with MobileSDK.framework.
2. Update the application project file to copy uiimages.bundle and SDKStrings.bundle into bundle resources, so they will be available at run time.
3. (Required only if your application uses an older version of SDKAPI.) In order to successfully locate the `kfxLibEngines`, `kfxLibLogistics`, `kfxLibUIControls`, and `kfxLibUtilities` header files, the application project file needs to be update to specify the location of `MobileSDK.framework` in the Header SearchPaths section.
4. Required for Xcode 7 and above.
   - If there are any existing dylib files, remove them and add the tbd equivalents.
   - Add a dictionary named 'App Transport Security Settings' and add a key value pair of 'Allow Arbitrary Loads' & 'YES' to the `info.plist` file.

For Android

If you want to apply this improvement for your PhoneGap app please update following files:

Update the libs folder in the Phonegap App with latest MobileSDK aar libs, including the *.so files. You can find these libs under `...\Android\MobileSDK_libs\aar and ...\Hybrid\PhoneGap\Plugins \com.kofax.mobile.plugins.sdk\lib\Android`.

Also update gradle file to include aar and jar files into build path.