Top Image Systems

Digital Business Starts Here.

mobiFLOW

SDK Implementation with Xamarin

Version 5

contact@topimagesystems.com

www.topimagesystems.com

12 January, 2018

Contents

Implementation with Xamarin

This document explains how to implement the mobiFLOW image capture library with Xamarin.

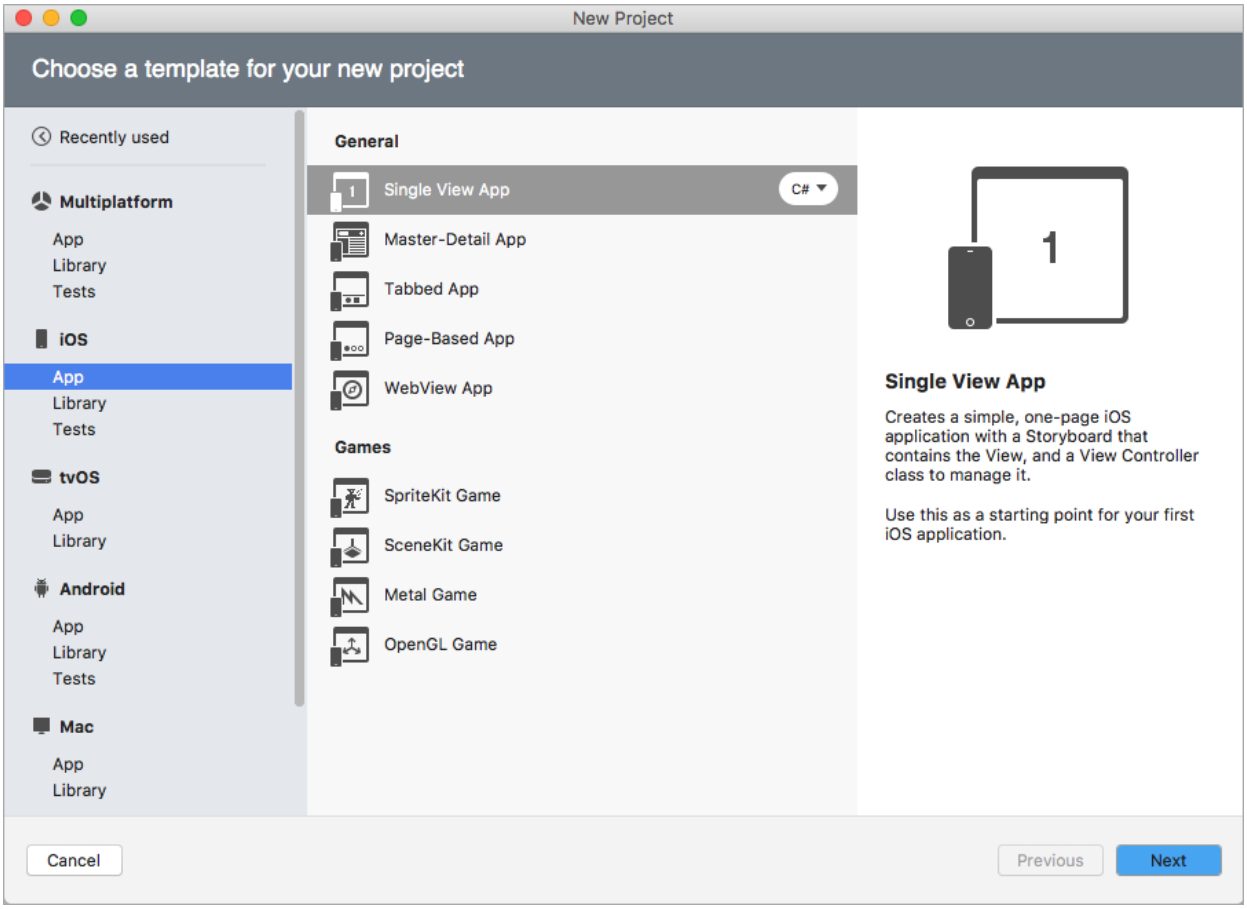
For detailed information on the mobiFLOW SDK, see the following guides:

- *mobiFLOW SDK Configuration Guide*
- *iOS Libraries SDK Guide*
- *Android Libraries SDK Guide*

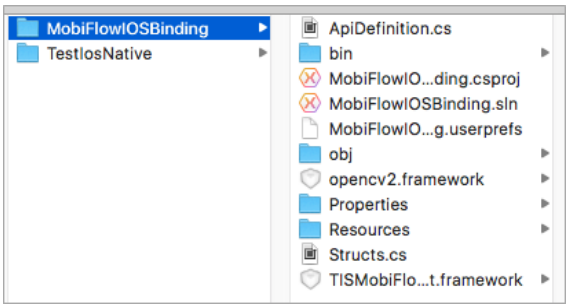
iOS native

Setup

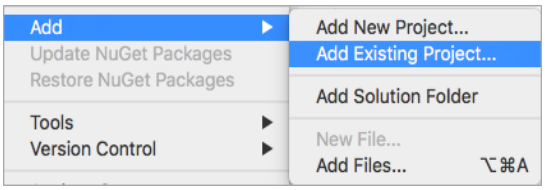
1. Create a new iOS project (Single View App), and follow the on-screen instructions.



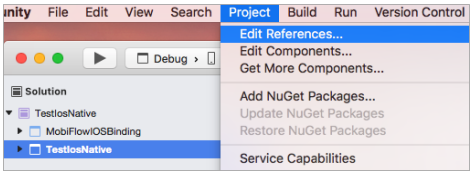
2. Copy the **MobiFlowIOSBinding** folder next to your project.



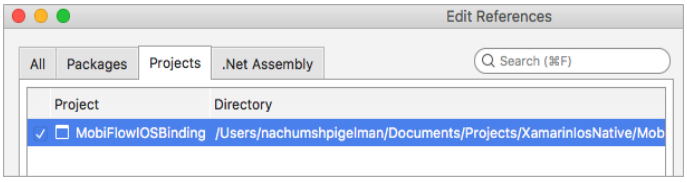
3. In the **MobiFlowIOSBinding** folder, delete the content of the **bin** and **obj** folders.
4. Add **MobiFlowIOSBinding.csproj** to your solution.



5. In your new project, add a reference to the **MobiFlowIOSBinding** project:
   a. Click on your project in the left menu and select **Project > Edit References...**



   b. In the **Edit References** window, click on the **Projects** tab, select **MobiFlowIOSBinding** and click **OK**.



Short example to start a session:

```
var sessionParams = new TISSessionParameters(TISDocumentType.Check);
var captureManager = new TISCaptureManagerViewController(sessionParams);
captureManager.CaptureManagerDelegate = customDelegate;

this.PresentViewController(captureManager, true, null);
```

A full example of the implementation can be found in the project *MobiFlowIOS.*

Note: Remember to add *MobiFlowSDK;* at the top of your ViewController that is implementing *TISMobiFlowDelegate*.

Known issue

When overriding the delegate functions:

- void CaptureManagerDidFinishWithResults (TISCaptureManagerViewController captureManagerViewController, [NullAllowed] TISProcessingResults imageResults);
- bool CaptureManagerDidOutputVideoFeedResultsForValidations (TISCaptureManagerViewController captureManagerViewController, [NullAllowed] TISProcessingResults imageResults);

You must change *TISProcessingResults* to the type of result you are expecting (could also stay as *TISProcessingResults*).

For example: if you are capturing a check, the result type should be *TISCheckProcessingResults*.

```
public override void CaptureManagerDidFinishWithResults(TISCaptureManagerViewController captureManagerViewController, TISCheckProcessingResults imageResults)
```

You must perform the change in the class *ApiDefinition.cs* located in the *MobiFlowIOSBinding* project, and also in your ViewController that overrides these delegate methods.

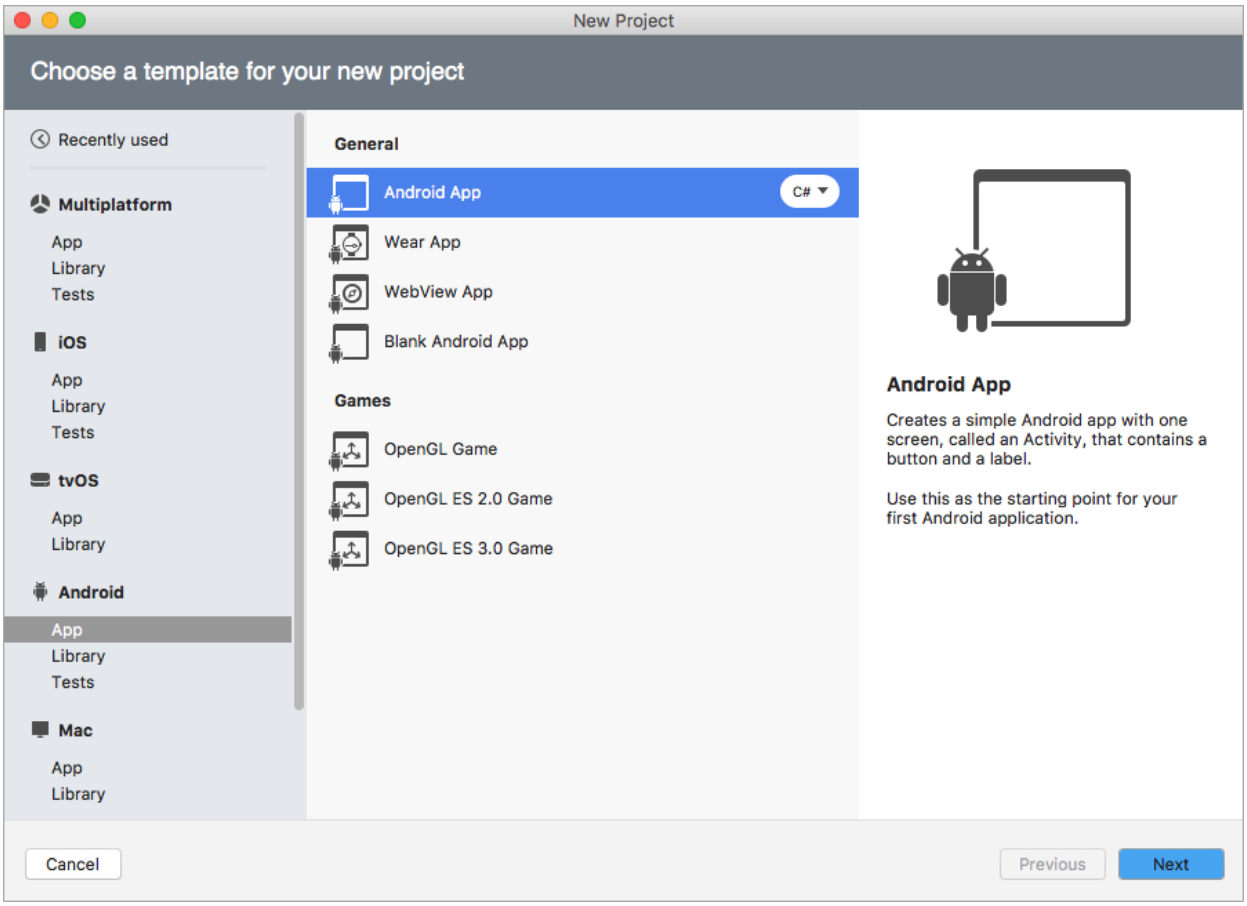
Results type available

- TISProcessingResults
- TISCheckProcessingResults
- TISCMC7CheckProcessingResults,
- TISCardProcessingResults
- TISCreditCardProcessingResults
- TISPassportProcessingResults

Android native

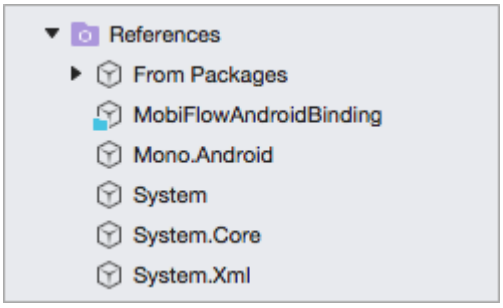
1. In the files you were given, enter the folder *MobiFlowAndroidBinding* and open *MobiFlowAndroidBinding.sln*.
2. Build the project: **Go to MobiFlowAndroidBinding > MobiFlowAndroidBinding > bin > (Debug/Release)** and copy **MobiFlowAndroidBinding.dll** aside; this will be used later.

3. Create a new Android App and follow the on-screen instructions.



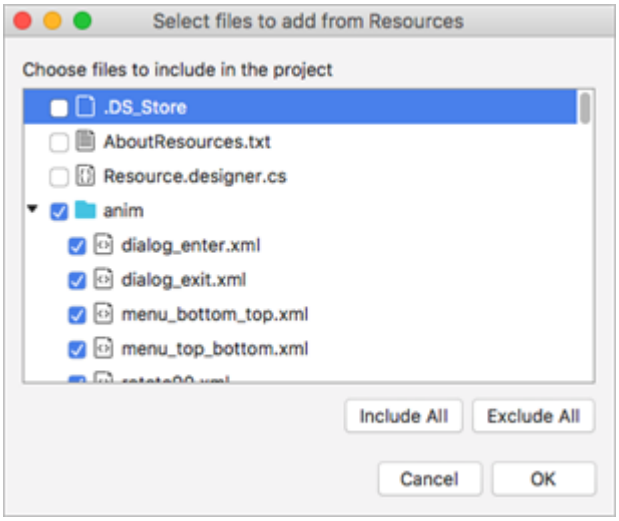
4. In your project, right-click **References** and select **Edit References…**.
5. Click the **.Net Assembly** tab and click the **Browse…** button.
6. Select the *.dll* file you saved in step 2.

   Your **References** should look like this:



7. In the **Resources** folder, delete all content from the project except **Resource.designer.cs** and **AboutResources.txt**.
8. Right-click on **Resources** and select **Add > Add files from folder…**.
9. Select the **Resources** folder from the demo project **MobiFlowAndroid** and click **Open**.

10. Click **Include All**, then deselect **Resource.designer.cs**, **AboutResources.txt** and **.DS_Store**(if it exists) and click **OK**.



11. Select **Copy the file to the directory** and click **OK**.
12. In the project, click once on Resources > values > attr.xml, then press DELETE and click **Remove from project**.
13. In *AndroidManifest.xml Source*, in the *<application>* section, add the following:

```
<activity android:name="com.topimagesystems.controllers.imageanalyze.CameraManagerController" android:configChanges="keyboardHidden|orientation|screenSize"></activity>
<activity android:name="com.topimagesystems.controllers.imageanalyze.CameraController" android:configChanges="keyboardHidden|orientation|screenSize" android:hardwareAccelerated="false"></activity>
<activity android:name="com.topimagesystems.controllers.imageanalyze.DynamicCaptureCameraController" android:configChanges="keyboardHidden|orientation|screenSize" android:hardwareAccelerated="false"></activity>
```

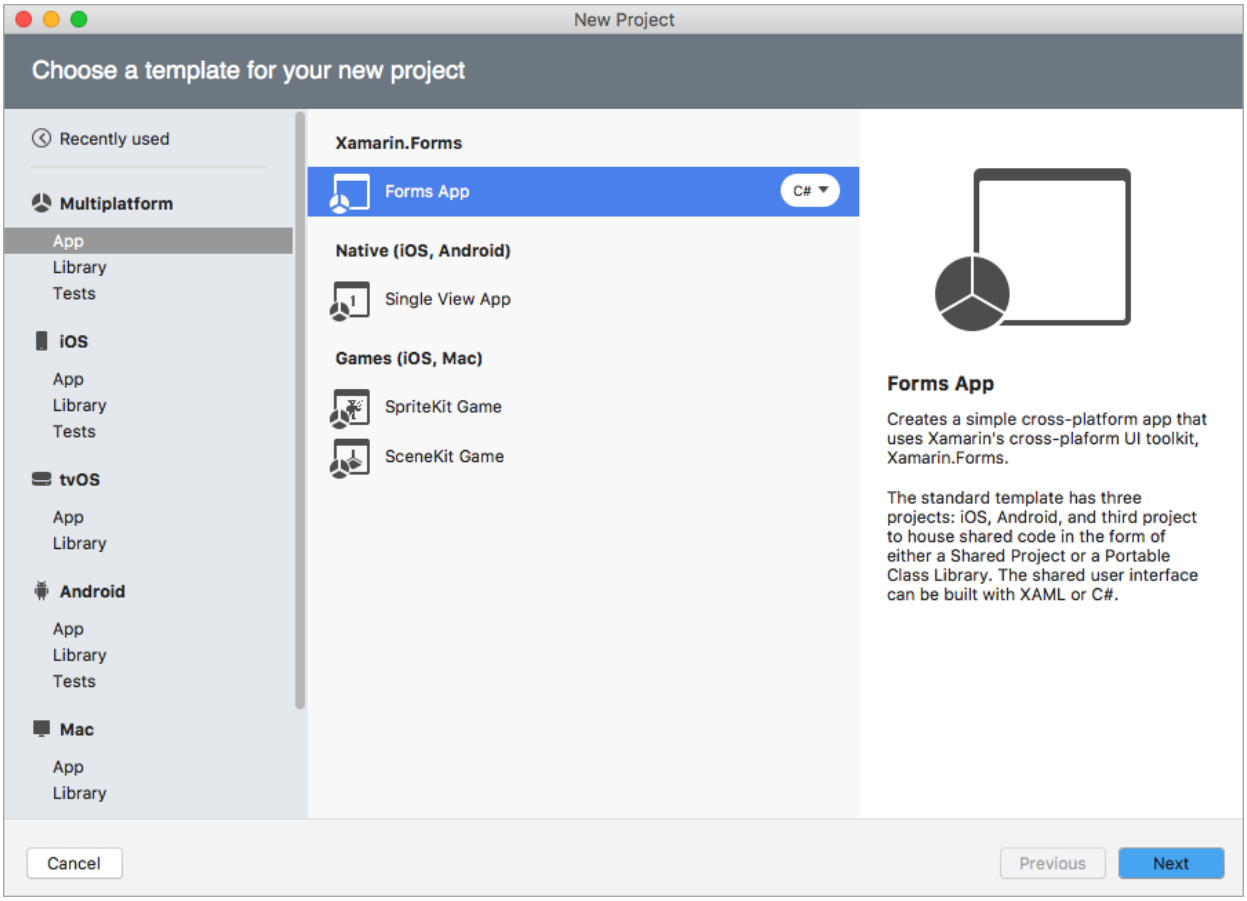
Short example to start a session:

```
MainActivity mobiListener;

CameraController.Listener = mobiListener;
CameraManagerController.SessionType = CaptureIntent.SessionType.Normal;
CaptureIntent captureIntent = new CaptureIntent(mobiListener);
CaptureIntent.BaseCaptureParams input = null;
input = captureIntent.GetCaptureParams(CaptureIntent.TISDocumentType.Check);
input.OcrType = Common.OCRType.E138b;

captureIntent.CaptureDocument(input);
```

A full example of the implementation can be found in the project *MobiFlowAndroid*.

Multiplatform Xamarin.Forms

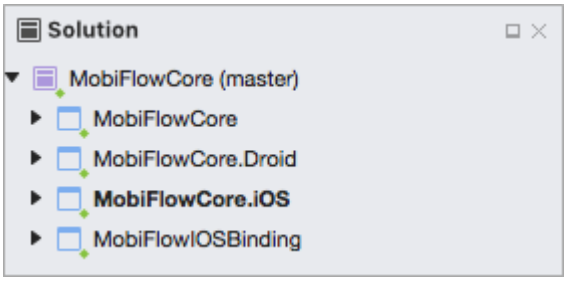
1. Create a new Forms App and follow the on-screen instructions.



Your solution should look like this:

2. Bind the project:
   - To bind an iOS project, follow the instructions for creating an iOS native project. Treat your *<Project Name>.iOS* as a native iOS project.

     When done, your solution should look like this:

   - To bind an Android project, follow the instructions for creating an Android native project, treat your *<Project Name>.Droid* as a native Android project.

A full example of a multiplatform project can be found in the folder *MobiFlowCore*. This project includes an interface example that can be used in iOS and Android projects. This way, the function that is called in your Forms project will be the same for iOS and Android.