Kofax mobiFlow

SDK with Xamarin Administrator’s Guide

Version: 6.0.0

Date: 2020-10-23

KOFAX

Table of Contents

Preface

This guide describes how to implement the mobiFlow image capture library with Xamarin.

For detailed information on mobiFlow SDK, refer to the following guides:

- *Kofax mobiFlow SDK Developer's Guide*
- *Kofax mobiFlow iOS Developer's Guide*
- *Kofax mobiFlow Android Developer's Guide*

Getting help with Kofax products

The Kofax Knowledge Base repository contains articles that are updated on a regular basis to keep you informed about Kofax products. We encourage you to use the Knowledge Base to obtain answers to your product questions.

To access the Kofax Knowledge Base, go to the Kofax website and select **Support** on the home page.

Note The Kofax Knowledge Base is optimized for use with Google Chrome, Mozilla Firefox or Microsoft Edge.

The Kofax Knowledge Base provides:

- Powerful search capabilities to help you quickly locate the information you need.

  Type your search terms or phrase into the **Search** box, and then click the search icon.
- Product information, configuration details and documentation, including release news.

  Scroll through the Kofax Knowledge Base home page to locate a product family. Then click a product family name to view a list of related articles. Please note that some product families require a valid Kofax Portal login to view related articles.
- Access to the Kofax Customer Portal (for eligible customers).

  Click the **Customer Support** link at the top of the page, and then click **Log in to the Customer Portal**.
- Access to the Kofax Partner Portal (for eligible partners).

  Click the **Partner Support** link at the top of the page, and then click **Log in to the Partner Portal**.
- Access to Kofax support commitments, lifecycle policies, electronic fulfillment details, and self-service tools.

  Scroll to the **General Support** section, click **Support Details**, and then select the appropriate tab.

Product documentation

By default, the Kofax mobiFlow documentation is available online. However, if necessary, you can download the documentation to use offline.

Online documentation

The product documentation for Kofax mobiFlow 6.0.0 is available at the following location:

https://docshield.kofax.com/Portal/Products/en_US/mobiFlow/6.0.0-tss0pu9zau/mobiFlow.htm

Offline documentation

To access the documentation offline, download the documentation .zip files from the Kofax Fulfillment Site and extract them on a local drive available to your users.

Build sample applications

This section describes how to build sample applications for Android and iOS.

Build MobiFlowAndroidBinding project

1. Copy mobiFlow.Android-release.aar file from KofaxmobiFlowSDK (`Android/aars`) to KofaxmobiFlowSDK (`Hybrid/Xamarin/MobiFlowAndroidBinding/MobiFlowAndroidBinding/Jars`).
2. Open MobiFlowAndroidBinding project from `Hybrid/Xamarin/MobiFlowAndroidBinding/MobiFlowAndroidBinding.sln` in Visual studio.
3. Build the project.
   MobiFlowAndroidBinding.dll file is generated in `Hybrid/Xamarin/MobiFlowAndroidBinding/MobiFlowAndroidBinding/bin/Debug` folder.
4. Close Visual studio.

Build MobiFlowCore sample application for Android

1. Open MobiFlowCore project from `Hybrid/Xamarin/MobiFlowCore/MobiFlowCore.sln` in Visual studio.
2. In the MobiFlowImplementation.cs file update the license at line number 49.
3. Run MobiFlowCore.Droid target on device.

Build MobiFlowAndroid sample application

1. Open MobiFlowAndroid project from `Hybrid/Xamarin/MobiFlowAndroid/MobiFlowAndroid.sln` in Visual studio.
2. In the MainActivity.cs file update the license at line number 144.
3. Run the project on device.

Build MobiFlowiOS sample application

1. Unzip the Frameworks from KofaxmobiFlowSDK `(\iOS\mobiFlow\Frameworks)`.
2. Copy opencv2.framework and KofaxmobiFlowWidget.framework from KofaxmobiFlowSDK (`iOS/mobiflow/Frameworks`) to KofaxmobiFlowSDK (`Hybrid/Xamarin/MobiFlowiOSBinding`).
3. Copy Resources folder from KofaxmobiFlowSDK (`iOS/Mobiflow`) to KofaxmobiFlowSDK (`Hybrid/Xamarin/MobiFlowiOSBinding`).

4. Open MobiFlowIOS project from `Hybrid/Xamarin/MobiFlowIOS/MobiFlowIOS.sln` in Visual studio.
5. Update the license in ViewController.cs file.
6. Change Bundle Identifier in Info.plist.
7. Run the project on the device.

Build MobiFlowCore sample application for iOS

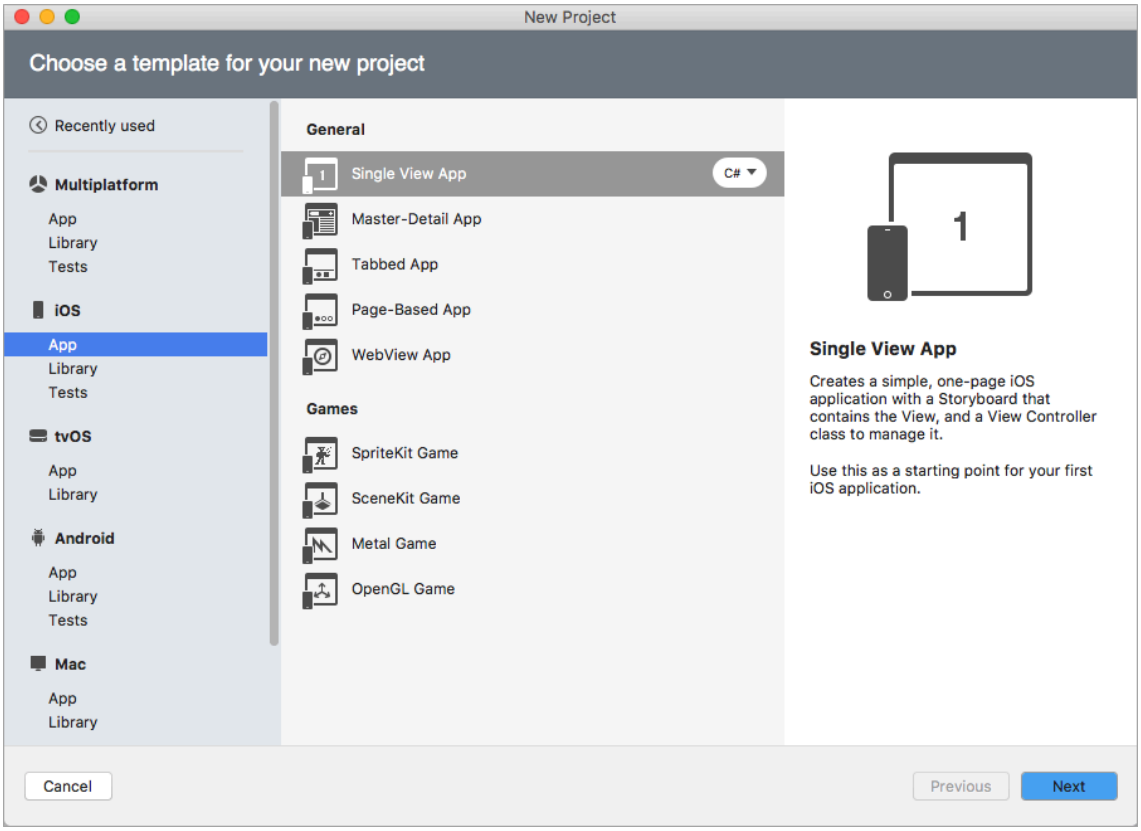
1. Unzip the Frameworks from KofaxmobiFlowSDK (`\iOS\mobiFlow\Frameworks`).
2. Copy opencv2.framework and KofaxmobiFlowWidget.framework from KofaxmobiFlowSDK (`iOS/mobiflow/Frameworks`) to KofaxmobiFlowSDK (`Hybrid/Xamarin/MobiFlowiOSBinding`).
3. Copy Resources folder from KofaxmobiFlowSDK (`iOS/mobiflow`) to KofaxmobiFlowSDK (`Hybrid/Xamarin/MobiFlowiOSBinding`).
4. Open MobiFlowCore project from `Hyrbid/Xamarin/MobiFlowCore/MobiFlowCore.sln` in Visual studio.
5. Update the license in MobiFlowImplementation.cs file.
6. Change Bundle Identifier in Info.plist file.
7. Run MobiFlowCore.iOS target on the device.

iOS native

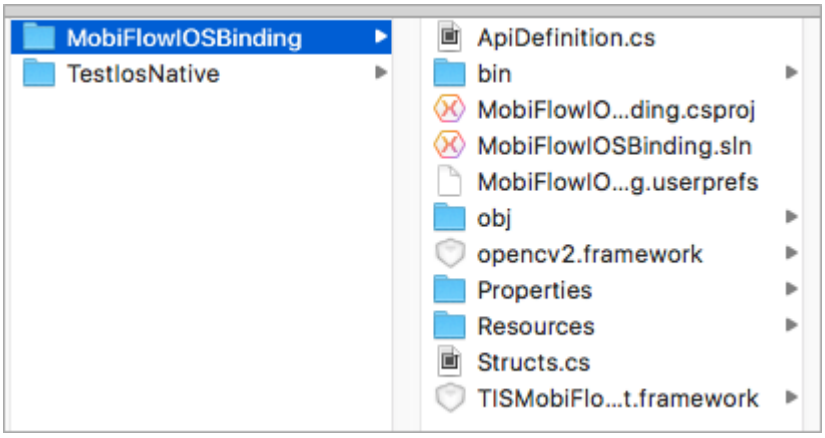
This section describes how to bind an iOS project using Xamarin.

Setup

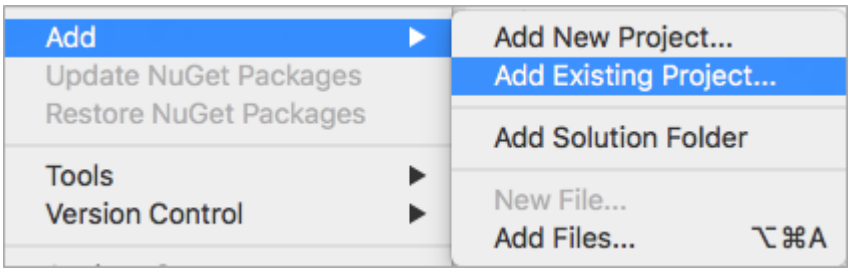
1. Create a new iOS project (Single View App), and follow the on-screen instructions.



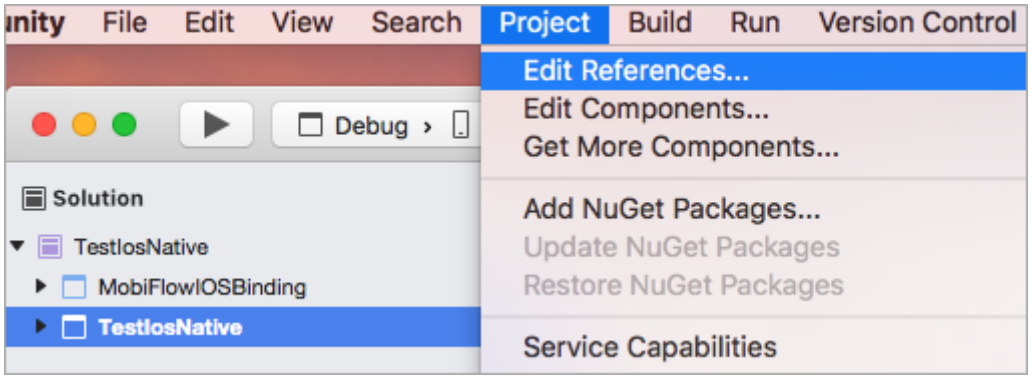
2. Copy the **MobiFlowIOSBinding** folder next to your project.



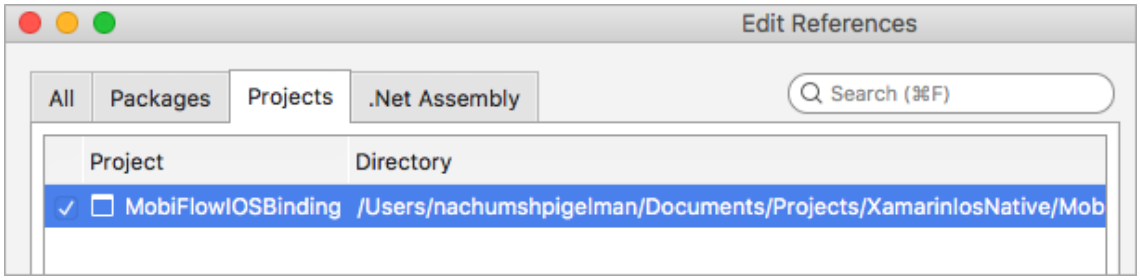
3. In the **MobiFlowIOSBinding** folder, delete the contents of the **bin** and **obj** folders.
4. Add **MobiFlowIOSBinding.csproj** to your Solution.



5. In your new project, to add a reference to the MobiFlowIOSBinding project do the following:
   a. Select your project, and then click **Project** > **Edit References**.



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



Example: 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 MobiFlowIOS project.

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

Delegate functions override

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

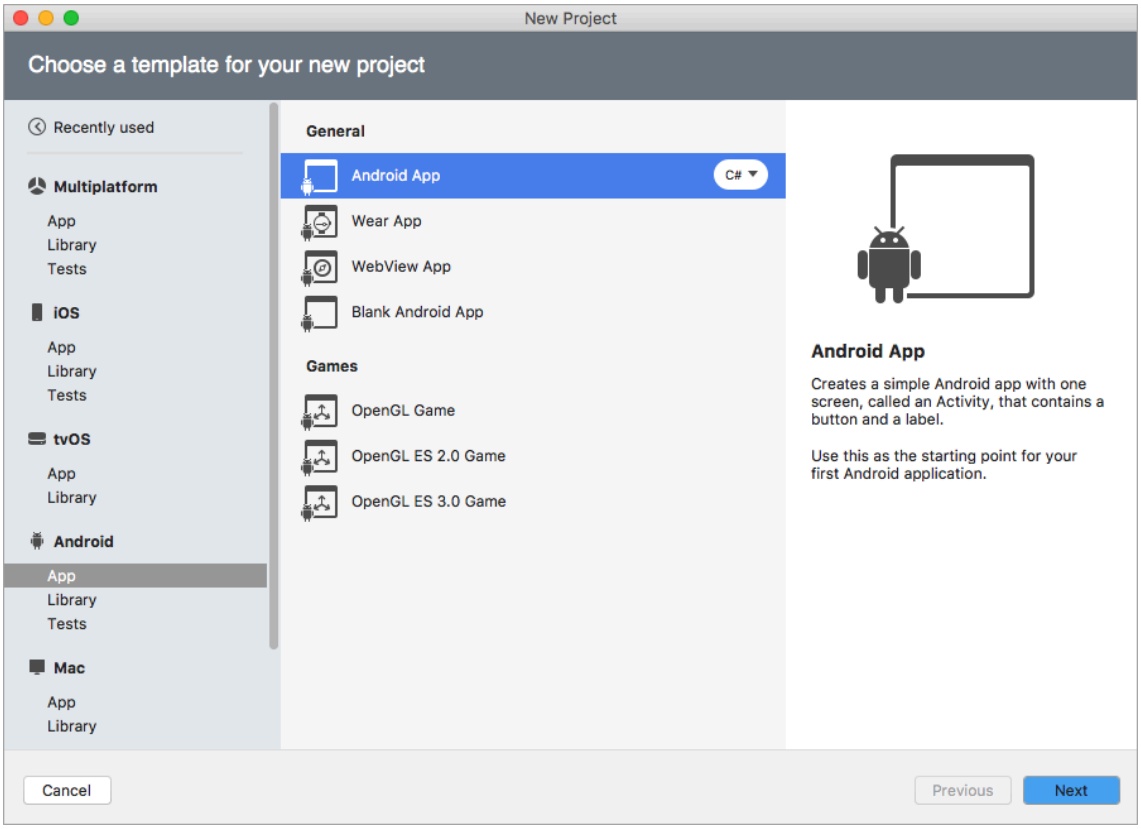
- TISProcessingResults
- TISCheckProcessingResults
- TISCMC7CheckProcessingResults
- ISCardProcessingResults
- TISCreditCardProcessingResults
- TISPassportProcessingResults

Android native

This section describes how to bind an Android project using Xamarin.

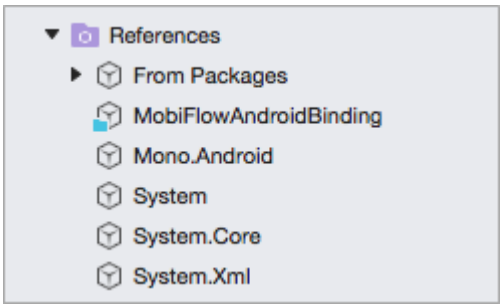
1. In the files you were given, go to the MobiFlowAndroidBinding folder and open MobiFlowAndroidBinding.sln.
2. To build the project, go to `MobiFlowAndroidBinding\MobiFlowAndroidBinding\bin\(Debug/Release)` and copy the MobiFlowAndroidBinding.dll aside.

   This will be used later.
3. Create a new Android App and follow the on-screen instructions.

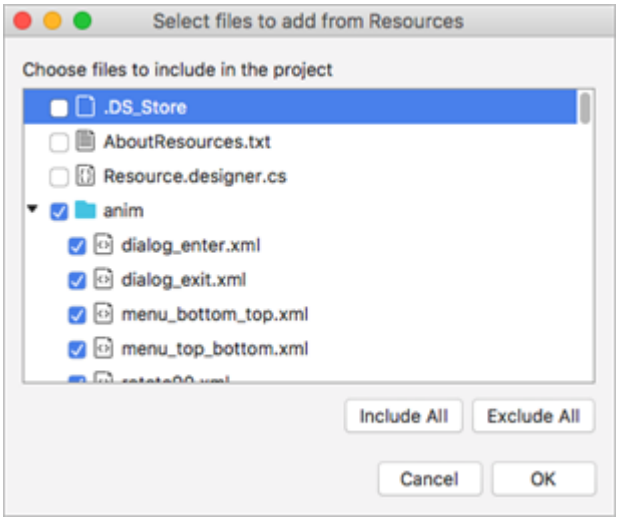


4. In your project, right-click **References** and select **Edit References**.
5. Select the **.Net Assembly** tab and click **Browse**.

6. Select the .dll file you have saved in the step 2.
   Your **References** menu should look as given below:



7. In the Resources folder, delete all the contents from the project except Resource.designer.cs and AboutResources.txt files.
8. Right-click the Resources folder and select **Add** > **Add files from folder**.
9. Select the Resources folder from the demo project MobiFlowAndroid and click **Open**.
10. Click **Include All**, and then clear **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 **Resources** > **values** > **attr.xml**, press Delete, and then click **Remove from project**.
13. In AndroidManifest.xml Source, in the <application> section, add the following:

```
<activity
 android:name="com.topimagesystems.controllers.imageanalyze.CameraManagerContro
ller"
android:configChanges="keyboardHidden|orientation|screenSize"></activity>
<activity
 android:name="com.topimagesystems.controllers.imageanalyze.CameraController"
android:configChanges="keyboardHidden|orientation|screenSize"
 android:hardwareAccelerated="false"></act
ivity>
```

```
<activity
 android:name="com.topimagesystems.controllers.imageanalyze.DynamicCaptureCameraCon
troller"
android:configChanges="keyboardHidden|orientation|screenSize"
 android:hardwareAccelerated="false"></act
ivity>
```

Example: Short example to start a session:

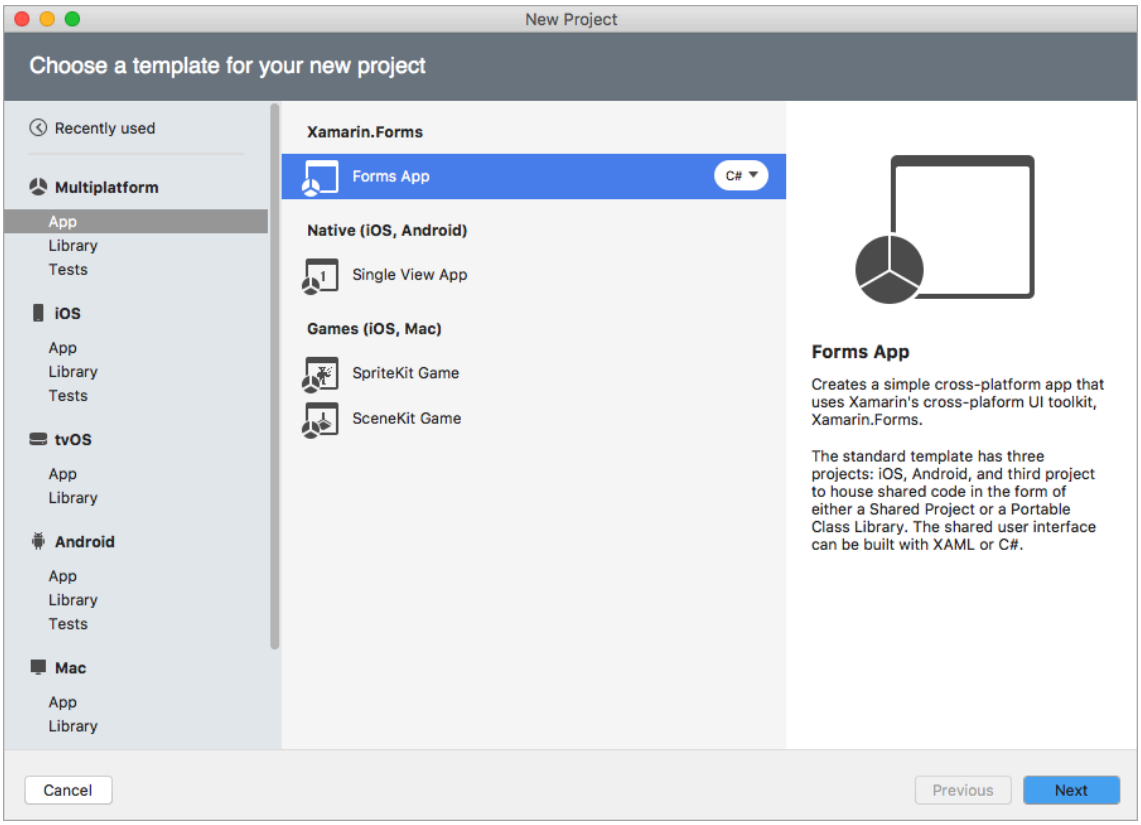
```
MainActivity mobiListener;
CameraController.Listener = mobiListener;
CameraManagerController.SessionType = CaptureIntent.SessionType.Normal;
CaptureIntent captureIntent = new CaptureIntent(mobiListener);
CaptureIntent.BaseCaptureParams input = null;
input = captureIntent.GetCaptureParams(CaptureIntent.TISDocumentType.Check);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

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

Multiplatform Xamarin.Forms

This section describes how to bind a project for multiplatform using Xamarin.

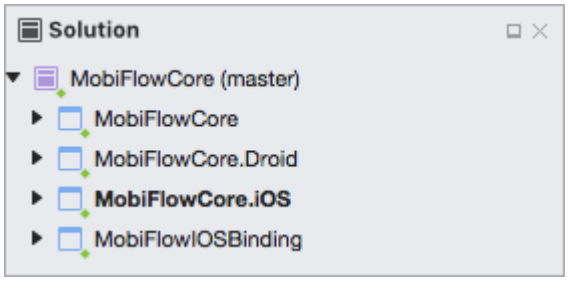
1. Create a new Forms App and follow the on-screen instructions.



Your **Solution** should look as given below:

2. Bind the project as follows:
   - 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 you have finished, the **Solution** should look as given below:

   - 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.