Release Notes

mobiFLOW 5

Top Image Systems

New Features

This section describes the new functionality provided as part of this release.

Credit card capture and OCR

Added an option to read the credit card number. Expiry date OCR is not fully supported.

Manual capture

Added an option to capture the image manually at the press of a button.

When this option is enabled, users do not have to follow the capture guidelines, and image capture is possible at wider angles and at a further distance from the document.

Due to the less strict capture process, we recommend using the crop controller when manual capture is enabled.

Crop controller

Added another optional step after the image was taken and before processing starts.

When this option is enabled, users can confirm the image quality and correct the cropping rectangle found by the algorithm.

Xamarin integration support

The release package now contains a new Xamarin project with C# bindings. Applications that run on Xamarin can now use mobiFLOW for iOS and Android.

Support in remote repositories

mobiFLOW for iOS and Android has been uploaded to JCenter and CocoaPods. The calling application can now integrate the SDK without downloading and integrating the actual code.

> **Note:** User interface customization on Android is limited when using this option.

Better debugging and automation support

Added an option to save full session details (video feed images and still images, if these exist) so that they can be inserted for debugging purposes.

When the SDK is working in record mode, every video frame or stills are saved to the device storage.

In play mode, the SDK processes the saved images and outputs a log file with the result.

New Android showcase

Added a new Android showcase with a modern user interface and the option to share all session data (images and the log file).

Improvements and Bug Fixes

This section describes the issues that have been found and fixed or improved during testing of this release.

Improved Swift binding with the SDK

iOS only

The Objective-C class that communicates with the SDK methods has been removed. It is now possible to call the SDK methods directly from Swift.

User interface enhancement when the screen capture is presented and dismissed

iOS only

Animation is now smoother when the screen capture is presented to the user and when the user dismisses the capture.

Adding custom animation is now possible.

Removed dependency on PCH file

iOS only

To reduce imports, the necessary includes are now included on each file and not globally.

Capture orientation validation on portrait capture

iOS and Android

To prevent users from capturing the image with the wrong orientation, the SDK will now present a hint and the frame will remain red until the phone and the document have the same orientation.

Same capture caption string used for multiple text views

Android only

The capture caption string ID *TISFlowPleaseCaptureCheckFront* was used for multiple text views.

The capture message *TISFlowPleaseCaptureCheckFront* ID has been changed to *TISFlowFrontCaption* and *TISFlowBackCaption*.

The position of the text was changed to the center of the screen.

New initialization in iOS

There are two new methods to initialize *TISCaptureManagerViewController*:

- (nullable instancetype) initWithSessionParameters:(nonnull TISSessionParameters*)sessionParameters andCustomView:(nullable UIViewController*)customViewController;
- (nullable instancetype) initWithSessionParameters:(nonnull TISSessionParameters*)sessionParameters;

The main difference is that you must now initialize *TISSessionParameters* and your custom View Controller before you initialize *TISCaptureManagerViewController*.

Note: The previous method of initializing *TISCaptureManagerViewController* will not work.

Known Issues

This section describes any known issues that are present in the release.

Resources images for iPad

iOS only

Currently, images are adjusted to the iPhone screen resolution. When working with an iPad, the images may be too small in the iPad adjusted view. This will be fixed in a later version.

IQA issues

IQA tests require high quality images, and there is a trade-off between the strictness of the capture and the image quality. A higher resolution image that was captured at minimum capture angles will give the best result. On the other hand, video mode and soft capture may affect IQA performance.

Folded corners

When IQA is enabled, validation of the corners only considers the area as part of the validation; the width and height of the corners are not validated.

To avoid false positive scenarios and to enable easy capture, small corners of less than 1 inch will not count as a folded corner.

Check piggyback

Piggyback will be blocked by the aspect ratio check. However, in soft capture the SDK enables a larger aspect ratio threshold, so capture may be possible, and piggyback will be detected if there is contrast between the two checks.

Manual capture and crop controller

IQA is limited when using the crop controller or manual capture.

Corner detection, skew detection and number of spots are disabled.

Limitation on video feed processing

Video feed processing is only supported on iPhone 4S and later for smartphones, on iPad 3 and later for tablets, and iPad Mini and later for mini tablets. If the feature is enabled in the SDK for devices that are not supported, the behavior will be as if it was disabled and the capture will be done in stills mode.

OCR recognition and image quality will be better using stills capture.

Full page capture with low end cameras

Since full size image capture is done from a higher distance above the document, the SDK will choose the highest resolution available from the device to ensure good image quality.

Cameras below 8 megapixels or low-quality cameras will not produce good enough images for full capture OCR.

Blur detection issues

Blur detection was tested to work mostly on bill payments and full-page documents. If this feature is turned on for other document types, it may be too sensitive in video mode and make the capture more difficult. The recommendation is to use it mostly for bills or full-page capture at the moment.

Recognition rates on bills when using video feed processing

Android Only

Due to the large variety of Android devices and different camera resolutions, the recognition rates on some devices may be not as good as the latest devices that are currently on the market. It is therefore recommended to use stills capture mode for devices that were not tested beforehand, or for which the camera resolution is unknown.

Signature on MICR line

TISFlowWarningMICRInterupted is only fired on checks with CMC7 MICR lines. It is not fired on checks with other MICR lines or E13B lines, and it is not fired on bill payments with OCR-A lines.

Samsung Galaxy S5 focus issues

Android Only

There are focus issues with all Samsung Galaxy S5 models that can cause high blur rates with stills capture, and may cause low recognition rates for bills when capturing in video mode. The recommendation for this device is to use stills for bills.

Blur detection is always On for this device (even if set to Off by the SDK initialization) in order to catch cases when the device is out of focus and return it into focus again.

Sony Xperia Z3 focus issues

The Sony Xperia camera sometimes throws an exception when trying to focus.

There is a specific handling for these cases. When it happens, the SDK will order the camera to restart and the screen will blink once for a few milliseconds.

Support for passport capture

Using capture with the Passport document type is only supported on devices that support video capture (according to the specification document), since the MRZ recognition is done on the video feed. The SDK does not block this option; it is the responsibility of the calling app to check whether the device supports the resolution for video mode and to display a relevant error message.

Using fragments

Android Only

To avoid *IllegalStateException*, it is best to load the fragment from the code and not use the *<fragment />* tag in the XML layout.

MRZ for cards only works for front capture

If the session is configured to scan front and back in one session, the SDK will only perform OCR for the front side.

To perform OCR on the back side, split the capture and capture front and back separately, using *frontOnly* for both sides. When you capture the back side, set the *ocrType* to MRZ and leave it OFF for the actual front side.

HP Fortify Test on iOS

The report contains two medium severity issues and two low severity issues.

The two medium issues refer to the way the SDK decrypts strings. It is false positive, since decrypting the string is an extra validation that does not have to be performed, and those strings do not contain any sensitive data.

The two low severity issues refer to temporary files that do not affect the SDK release package.

Tested Environments

The table below shows the device and operating system combinations used for testing the Service Pack. The devices used were selected to achieve maximum coverage based on the criteria defined in the mobiFLOW Test Strategy document.

Android Devices	Operating System Version
Samsung Galaxy S8	7.0
Samsung Galaxy S7	6.0.1
Samsung Galaxy S6	5.1
Samsung Galaxy S5	4.4.2
Samsung Galaxy Note 5	6.0.1
LG G3	5.0.1
Sony Xperia Z3	5.0.2
Nexus 5	6.0.1
Nexus 5X	8.0.0
HTC One M9	5.1

Android Devices	Operating System Version
Huawei P8	5.0.1

iOS Devices	Operating System Version
iPhone 7+	10.3.2
iPhone 6S+	10.0.2
iPhone 6	11
iPhone 6+	8.4
iPhone 5S	8.0.2
iPhone 5	8.4
iPhone 6S+	10.0.2