



# Kofax Mobile Capture SDK Best Practices Guide

Version: 3.8.0

Date: 2023-01-23

**KOFAX**

© 2022 Kofax. All rights reserved.

Kofax is a trademark of Kofax, Inc., registered in the U.S. and/or other countries. All other trademarks are the property of their respective owners. No part of this publication may be reproduced, stored, or transmitted in any form without the prior written permission of Kofax.

# Table of Contents

<b>Preface</b> .....	<b>5</b>
<b>Chapter 1: Overview</b> .....	<b>6</b>
<b>Chapter 2: Capturing identity documents with the Kofax Mobile Capture SDK</b> .....	<b>7</b>
ID demo images.....	8
ID image processing.....	8
ID data extraction.....	9
ID verification and facial recognition.....	9
QuickExtract and ID MRZ and bar code extraction.....	9
<b>Chapter 3: Capturing passports with the Kofax Mobile Capture SDK</b> .....	<b>10</b>
QuickExtract and passports.....	11
Passport demo images.....	11
Passport image processing.....	12
Passport data extraction.....	12
Passport ID verification and facial recognition.....	12
<b>Chapter 4: Capturing checks with the Kofax Mobile Capture SDK</b> .....	<b>13</b>
Check demo images.....	14
Check image processing.....	14
Check data extraction.....	14
<b>Chapter 5: Capturing credit cards with the Kofax Mobile Capture SDK</b> .....	<b>15</b>
Credit card demo images.....	16
Credit card image processing.....	16
Credit card data extraction.....	16
<b>Chapter 6: Capturing bills with the Kofax Mobile Capture SDK</b> .....	<b>17</b>
Bill demo images.....	18
Bill image processing.....	18
Bill data extraction.....	19
<b>Chapter 7: General image processing recommendations</b> .....	<b>20</b>
<b>Chapter 8: Memory management</b> .....	<b>21</b>
<b>Chapter 9: Security recommendations</b> .....	<b>22</b>
Obfuscation.....	22
Certificate pinning.....	22
Required permissions.....	22
Security for Android apps.....	22
<b>Chapter 10: Licensing</b> .....	<b>23</b>

**Chapter 11: Troubleshooting..... 24**  
    Frequently asked questions.....24

# Preface

This guide provides recommended best practices for using the Kofax Kofax Mobile Capture SDK when building mobile applications. Following these best practices ensures a smooth integration with a variety of supported use cases and helps avoid many issues developers or users might encounter.

## Chapter 1

# Overview

Applications typically interact with the Kofax Mobile Capture SDK to perform the following activities:

- Capture one or more images of a business document.
- Clean up the captured images by automatically cropping or correcting for skew and other distortions.
- Extract information from the images, either directly with the Kofax Mobile Capture SDK or by transmitting the image data to a backend server solution.

The image capture experiences provided by the Kofax Mobile Capture SDK guide the application user in producing images of documents that are ideal for image processing and data extraction. We recommend that the guidance provided by the capture experiences be used.

The goal of image processing is to remove any unnecessary data from the image by cropping; correct any tilt, skew, or angles; and adjust the image to fit a specific aspect ratio.

## Chapter 2

# Capturing identity documents with the Kofax Mobile Capture SDK

When capturing IDs, follow these recommendations:

- The Capture Control for the application should be set to use imageMode to produce high-resolution images. High-resolution images ensure the best OCR results.
- Use the Fixed Aspect Ratio Capture Experience when capturing IDs with a known aspect ratio. For ID-1 size documents (3 3/8 x 2 1/8 inches, 85.60 x 53.98 millimeters) use the default settings. For other ID sizes, specify the dimensions. For portrait-oriented IDs, the width will be smaller than the height. For landscape-oriented images, the width will be greater than the height. The Fixed Aspect Ratio Capture experience can return a glare-free image. Enable glareDetectionEnabled for poor lighting conditions and glare on the card.

The following images show examples of capturing the front and back of IDs.



## ID demo images

To help the user understand which side of the ID they should be capturing, your application should use demo images.



## ID image processing

Note the following recommendations for ID image processing:

- The Kofax Mobile Capture SDK has a class designed specifically for processing and extracting ID and passport images:
  - Android: `OnDeviceIdExtractor`
  - iOS: `kfxKOEIDExtractor.h`
- QuickAnalysis should be performed on any image to be used for data extraction. Doing this ensures that only high quality images are used for extraction, which produce the best results.
- You now have the option to switch to legacy blur detection.
- Target frame cropping is recommended when using the Fixed Aspect Ratio Capture Experience. This feature improves cropping accuracy by removing the majority of the background.
- On-device processing is recommended to reduce the time required to upload the image to the server. For the correct image processing string, see the *Kofax Kofax Mobile Capture SDK Developer's Guide*.
- Processing on the server enables the developer to log the raw image, processing string, and processing results. This data can be used for debugging issues.
- Engines have an API to remove glare. Users can send images to get a glare-free image. For best performance, images fed to the API should be captured at different tilt angles with respect to the group.



## ID data extraction

On-device extraction is recommended when the developer wants to minimize the initial application download size or provide the option to perform extraction when Wi-Fi or cellular data is not available. When using on-device extraction, note the following:

- To reduce the application's initial download size, the application should be designed to perform on-demand downloading of the Mobile ID Models using QuickUpdater.
- For regions where Wi-Fi or cellular data connections are unreliable, the Mobile ID models can be compiled into the application for use when the device is offline.

For information on server-side data extraction, see the *Kofax Mobile ID Administrator Guide*.

## ID verification and facial recognition

See the *Kofax Mobile ID Verification Administrator Guide* for information about ID verification and facial recognition.

## QuickExtract and ID MRZ and bar code extraction

If you only require the information in the bar code or MRZ on the ID, you can use the QuickExtract feature to return this data quickly. QuickExtract does not require capturing or processing an image, though an image will be returned if requested. Additionally, the facial image can be returned if present and requested.

The QuickExtractAgent allows reading data from images acquired by the application developer, either from the Kofax Mobile Capture SDK or from other sources. When using the QuickExtractAgent with images captured from the Kofax Mobile Capture SDK, video mode should be used to ensure that images are of sufficient resolution to read data while not being too high resolution to adversely impact performance.

## Chapter 3

# Capturing passports with the Kofax Mobile Capture SDK

When capturing passports, follow these recommendations:

- The Capture Control for the application should be set to use imageMode to produce high-resolution images. High-resolution images ensure the best OCR results.
- It is recommended to use the Passport Capture Experience. This experience is designed to actively search for MRZ and guide the user to capture a quality image. The Capture Experience should be landscape oriented to best match the orientation of a passport as shown in this example.





## Passport image processing

Note the following recommendations for passport image processing:

- The Kofax Mobile Capture SDK has a class designed specifically for processing and extracting ID and passport images:
  - Android: `OnDeviceIdExtractor`
  - iOS: `kfxKOEIDExtractor.h`
- QuickAnalysis should be performed on any image to be used for data extraction. Doing this ensures that only high quality images are used for extraction, which produce the best results.
- On-device processing is recommended to reduce the time required to upload the image to the server. For the correct image processing string, see the *Kofax Kofax Mobile Capture SDK Developer's Guide*.
- Processing on the server enables the developer to log the raw image, processing string, and processing results. This data can be used for debugging issues.

## Passport data extraction

On-device extraction is recommended when the developer wants to minimize the initial application download size or provide the option to perform extraction when Wi-Fi or cellular data is not available. When using on-device extraction, note the following:

- To reduce the application's initial download size, the application should be designed to perform on-demand downloading of the Mobile ID Models using the QuickUpdater.
- For regions where Wi-Fi or cellular data connections are unreliable, the Mobile ID models can be compiled into the application for use when the device is offline.

For information on server-side data extraction, see the *Kofax Mobile ID Administrator Guide*.

## Passport ID verification and facial recognition

See the *Kofax Mobile ID Verification Administrator Guide* for information about ID verification and facial recognition.

## Chapter 4

# Capturing checks with the Kofax Mobile Capture SDK

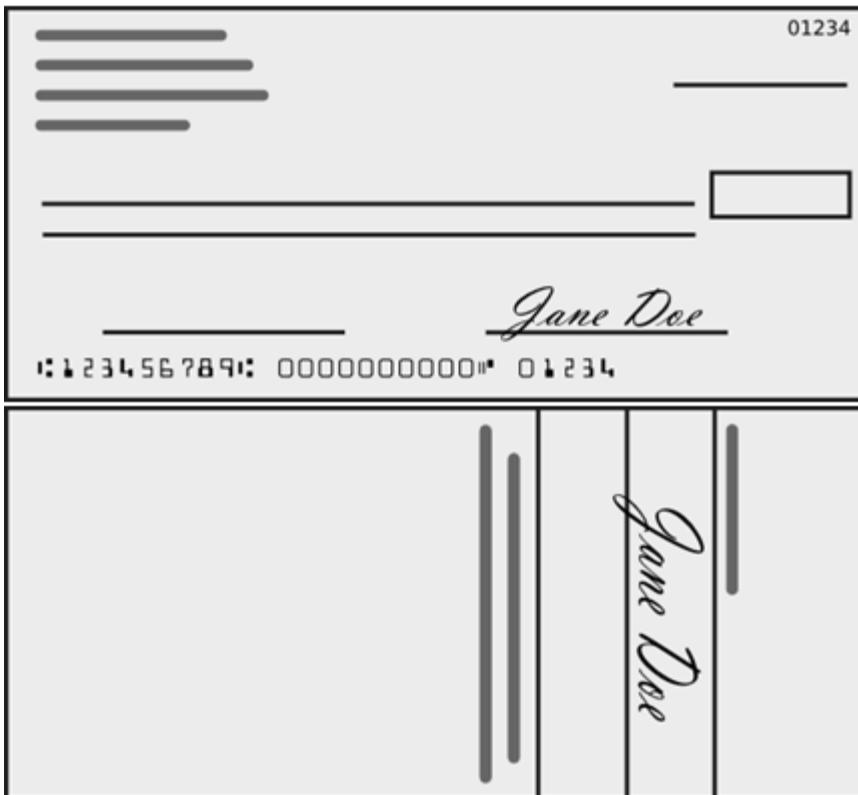
When capturing checks, follow these recommendations:

- The Capture Control for the application should be set to use imageMode to produce high-resolution images. High-resolution images ensure the best OCR results.
- It is recommended to use the Check Capture Experience. This experience provides guidance to the user based on detecting MICR text on the check. The aspect ratio of the front captured image can be used in assisting with capturing the back image. This is necessary because there is not much text on the back that can be used to detect the edges of the check.



## Check demo images

To help the user understand which side of the check to be captured, we recommend that your application use demo images. Demo images are provided with Kofax Mobile Capture SDK.



## Check image processing

QuickAnalysis should be performed on any image to be used for data extraction. Doing this ensures that only high quality images are used for extraction, which produce the best results.

## Check data extraction

For on-device extraction of the front of a check, include the `_ProcessCheckFront_` token to extract MICR information and include it in the processed image's metadata. You can then use the `MicrParser` class to verify that the extracted data is valid. This is the extent of device-side extraction.

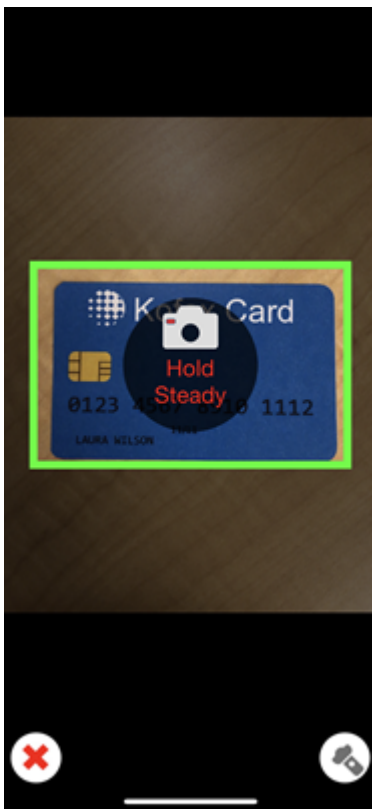
## Chapter 5

# Capturing credit cards with the Kofax Mobile Capture SDK

When capturing credit cards, follow these recommendations:

- The Capture Control for the application should be set to use imageMode to produce high-resolution images. High-resolution images ensure the best OCR results.
- It is recommended that you use the Fixed Aspect Ratio Capture Experience for capturing credit cards when you know the aspect ratio of the document. For ID-1 size documents (3 3/8 x 2 1/8 inches, 85.60 x 53.98 millimeters) use the default settings.

For other credit card sizes, specify the dimensions as width x height. For portrait-oriented credit cards, the width will be smaller than the height. For landscape-oriented images, the width will be greater than the height. For credit and debit cards, use portrait captures.



## Credit card demo images

To help the user understand which side of the credit card to be captured, we recommend that your application use demo images. Demo images are provided with the Kofax Mobile Capture SDK.



## Credit card image processing

Note the following recommendations for credit card image processing:

- QuickAnalysis should be performed on any image to be used for data extraction. Doing this ensures that only high quality images are used for extraction, which produce the best results.
- Target frame cropping is recommended when using the Fixed Aspect Ratio Capture Experience. This feature improves cropping accuracy by removing the majority of the background.
- On-device processing is recommended to reduce the time required to upload the image to the server. For the correct image processing string, see the *Kofax Mobile Credit and Debit Card Framework Administrator Guide*.
- Processing on the server enables the developer to log the raw image, processing string, and processing results. This data can be used for debugging issues.

## Credit card data extraction

For information about server-side extraction of credit and debit cards, see the *Kofax Mobile Credit and Debit Card Framework Administrator Guide*.

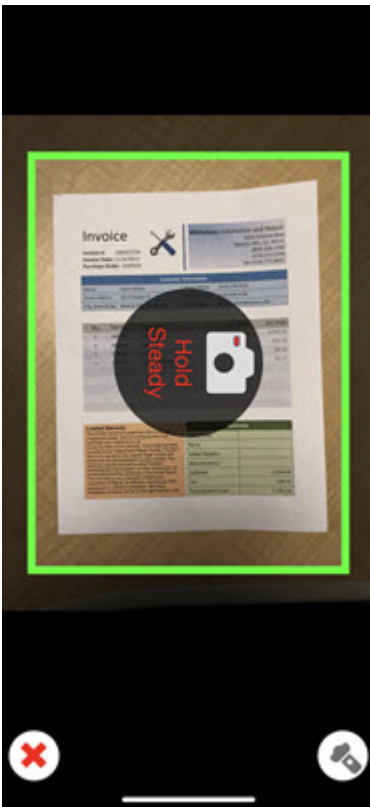


## Chapter 6

# Capturing bills with the Kofax Mobile Capture SDK

When capturing bills, follow these recommendations:

- The Capture Control for the application should be set to use imageMode to produce high-resolution images. High-resolution images ensure the best OCR results.
- It is recommended that you use the Fixed Aspect Ratio Capture Experience when you know the aspect ratio of the bill. If you do not know the aspect ratio of a bill, use the Document Capture Experience.



## Bill demo images

To help the user understand how to capture the bill, we recommend that your application use demo images. Demo images are provided with the Kofax Mobile Capture SDK.



## Bill image processing

Note the following recommendations for bill image processing:

- QuickAnalysis should be performed on any image to be used for data extraction. Doing this ensures that only high quality images are used for extraction, which produce the best results.
- Target frame cropping is recommended when using the Fixed Aspect Ratio Capture Experience. This feature improves cropping accuracy by removing the majority of the background.
- On-device processing is recommended to reduce the time required to upload the image to the server. For the correct image processing string, see the *Kofax Mobile Bill Pay Administrator Guide*.
- Processing on the server enables the developer to log the raw image, processing string, and processing results. This data can be used for debugging issues.

## Bill data extraction

Note the following:

- This document type does not have device-side extraction options.
- For information about server-side extraction of bills, see the *Kofax Mobile Bill Pay Administrator Guide*

## Chapter 7

# General image processing recommendations

Follow these recommendations for image processing:

- Avoid using `ImageProcessorConfiguration.documentDimensions` if you do not know the real document size.
- Set `ImageProcessorConfiguration.outputDPI` higher than 500 dpi. It will increase processed image file size.
- Use `ImageProcessorConfiguration.targetFrameCropType = TargetFrameCropType.TARGET_FRAME_CROP_ON` only if the image was captured using the Kofax Mobile Capture SDK.

## Chapter 8

# Memory management

To use memory efficiently, note the following:

- If applications retain more than two images in memory, write the images to device storage to avoid an out-of-memory condition.
- When a bitmap representation of an image is no longer needed, call the clear image bitmap API to release memory:
  - iOS: `clearImageBitmap()`
  - Android: `imageClearBitmap()`
- For Android, in the `AndroidManifest.xml` file, set `largeHeap=true`.

## Chapter 9

# Security recommendations

## Obfuscation

Obfuscation is recommended to protect your code and prevent an attacker from reverse engineering your software.

## Certificate pinning

Certificate pinning is recommended to prevent SSL certificate misissuance.

## Required permissions

Applications that use the Kofax Mobile Capture SDK can require some access permissions to function properly.

- Access to camera: Required for capture functionality.
- Access to device storage (read/write): Required to use images from device storage.

## Security for Android apps

We recommend using HTTPS for apps running on Android OS 9 or above. If you need to use HTTP, add or change the following setting in the app's manifest file:

```
android:usesCleartextTraffic = "true"
```

## Chapter 10

# Licensing

The Kofax Mobile Capture SDK is protected by an encrypted license. When activating your license, check the remaining duration on the license. If the number of days reaches zero, the license has expired, and the Kofax Mobile Capture SDK will no longer function.

## Chapter 11

# Troubleshooting

When debugging an application, check the following:

- Some Kofax Mobile Capture SDK API may have been deprecated and removed in the newest release. Check documentation for finding the replacement.
- Make sure that the device has enough disk storage because some features, like storing and processing images, write data to the device.
- Set a reasonable timeout for server communications and other long-time operations.
- Always check for null pointers in callbacks and error codes.


## Frequently asked questions

This section describes methods, classes, and properties to use for common situations or to address problems. For more information, see the Kofax Mobile Capture SDK API reference for your type of device.

### **App crashes after capturing a number of images with the `OutOfMemory` exception.**

This may be caused by insufficient device memory. Cache the captured images in the file system as opposed to keeping it in memory. You can compress images to TIFF or JPEG. Use these Kofax Mobile Capture SDK methods when using images:

- `image.imageWriteToFile`: Store captured images to a device storage.
- `image.imageReadFromFile`: Retrieve the image file from a device storage.

 We recommend that the user call `image.clearImageBitmap` whenever they are finished with an image object. This ensures that memory from the image's bitmap is freed immediately.

### **Images are processed correctly with the correct resolution and other attributes, but they seem to be corrupted when they are saved to files.**

Use Kofax Mobile Capture SDK methods to save images to storage. You have these options:

- Use `image.setImageFilePath` and `image.imageWriteToFile` to store images to device storage and `image.imageReadFromFile` to retrieve them.
- Process images with `IMAGE_REP_BOTH` or `IMAGE_REP_FILE` representation, and specify a path to the processed image with `imageProcessor.specifyProcessedImageFilePath`.



**Images are processed correctly with the correct resolution and other attributes, but they seem to be corrupted when they are sent to a TotalAgility or Kofax Real-Time Transformation Interface server.**

The Kofax Mobile Capture SDK provides an API to connect with servers and send images as objects or as bytes array. The API handles image objects correctly and does not require any knowledge on image encoding and decoding. See the Kofax Mobile Capture SDK API reference for your type of device for information about the `kfxKLOServerExtractor` and `KFXCaptureServer` classes.

**I would like to hide UI guidance or to modify them.**

The `kfxKUICaptureMessage` class enables you to customize the font, color, position, and other attributes of UI guidance. To hide guidance, set the properties to nil or null, such as `Message.messageIcons = nil`.

**I want auto capture to work with documents of non-standard, but known, aspect ratio.**

If you are going to capture only documents with a predefined aspect ratio, use `FixedAspectRatioCaptureExperience`. To set desired aspect ratio, use `KFXFixedAspectRatioDetectionSettings.targetFrameAspectRatio`.

**I want auto capture to work even when the mobile device is not parallel to the document and with any angle.**

The settings that are used for automatic capture are in the classes `DocumentBaseCaptureExperienceCriteriaHolder` and `DocumentCaptureExperienceCriteriaHolder`. You can control pitch and roll with `CaptureExperienceCriteriaHolder.setPitchThresholdEnabled`. Use `CaptureExperienceCriteriaHolder.setRollThresholdEnabled` and specify suitable threshold values.

**The device is having problems capturing documents.**

Use the debug capture setting, `setDiagnosticsViewEnabled()` in `DocumentBaseCaptureExperience`.