



Kofax Mobile Deposit Capture Administrator's Guide

Version: 2.0.0

Date: 2023-05-25

KOFAX

© 2023 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

Preface	4
Product documentation.....	4
Training.....	4
Getting help with Kofax products.....	4
Chapter 1: Overview	5
Image processing profile file.....	5
Parameters and settings.....	6
Project details.....	6
Extracted fields.....	6
Other fields.....	7
Image quality assurance (IQA).....	8
Chapter 2: Installation and configuration	13
Use Kofax Mobile Deposit Capture with companion products.....	14
Kofax Real-Time Transformation Interface.....	14
Kofax TotalAgility.....	15

Preface

This guide contains information about installing and configuring Kofax Mobile Deposit Capture, and assumes that you have a thorough understanding of Windows standards, applications, and interfaces, as well as Kofax Transformation and Kofax TotalAgility.

This guide is for solution integrators who are installing or configuring the Kofax Mobile Deposit Capture, or who need a description of the installation procedures and requirements.

Product documentation

In addition to this guide, other Kofax Mobile Deposit Capture documentation is available here:

https://docshield.kofax.com/Portal/Products/en_US/MDC/2.0.0-rh5yzytkt/MDC.htm

Also refer to the following documents:

- *Real-Time Transformation Interface Administrator's Guide*: Contains essential information about installing and configuring the Real-Time Transformation Interface.
- *Kofax TotalAgility Administrator's Guide*: Contains essential information about installing and configuring Kofax TotalAgility.

Training

Kofax offers both classroom and computer-based training that will help you make the most of your solution. Visit the Kofax website at www.kofax.com for complete details about the available training options and schedules.

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](#) home page and select **Customers > Knowledge Base**.

Chapter 1

Overview

Kofax Mobile Deposit Capture contains a Kofax Transformation project for extracting information from images of checks. It can be used with most commonly encountered check formats. It also includes a Kofax TotalAgility project that performs similar functions.

Kofax Mobile Deposit Capture requires Kofax Transformation 7.0.0.5 or later.

Kofax Mobile Deposit Capture provides an initial configuration of the Parascript CheckReader that is integrated with Kofax Transformation and Kofax TotalAgility. The specifics of the Parascript CheckReader configuration can be changed as needed.

Supported document types:

- Business checks
- Personal checks
- Cash tickets
- Deposit slips
- Money orders
- Traveler's checks
- Image Replacement Document (IRD)

Image processing profile file

One image processing profile is provided with the Kofax Mobile Deposit Capture project. This is for use with check images captured by mobile cameras to deskew, scale, and process both sides of the check to maximize the quality of the image.

An image processing profile is provided with Kofax Mobile Deposit Capture.

The file is available at the root of the provided Transformation project folder:


`KofaxMobileDepositCapture-2.0.0.0.0\KTM Project`. The following text file is included:
`ImagePerfectionProfile_CheckDeposit.txt`.

This image perfection profile contains the `_Do90DegreeRotation_9` token, which specifies the output image is to be auto-rotated so that the text is upright and then, if necessary to make the output be landscape orientation, the output image is rotated an additional 90 degrees clockwise. This profile also contains a `TokenReplaceList`, which is used to specify text changes to be made internally in the `imageperfectionsettings` string for processing images other than the first image in each Web service call.

The first image on a given call is always processed with the `imageperfectionsettings` string exactly as specified in the profile. The `TokenReplaceList` in this profile has the effect of replacing

the `_ProcessCheckFront_` token with `_ProcessCheckBack_` for processing the second image posted to Kofax Mobile Deposit Capture, presumably the back side of the check. This profile also has `FrontLengthAssistsAllPageLengths` enabled. `FrontLengthAssistsAllPageLengths` causes the longest dimension of the first processed image to be used to force second and subsequent images processed on the same Web service call to match. For a pair of check images, this has the effect of causing the long dimension of the processed check back to match that of the processed check front exactly.

Parameters and settings

 Use an x in front of a parameter name when using Real-Time Transformation Interface, unless stated otherwise. For example "Country" would become "xCountry".

You can set the following parameter:

PreferBooleans


Determines whether IQA Failure and Usability Failure results are returned as booleans (TRUE or FALSE, where TRUE indicates a failure) or tri-state values (YES, NO, or NOTDEFINED) which also indicate whether a specific test was actually performed. For boolean return, the associated result for a test that was not performed is returned as FALSE (no failure). Set this flag to FALSE to return the tri-state values. The default value is TRUE.

Project details

Kofax Mobile Deposit Capture supports processing of either an image of the front side of the check only, or an image with both sides of the check.

Extracted fields

The fields extracted from the check are listed in the following table.

 When an error description is returned for any of the `CheckCodeline_*` fields, that same error should appear in all of the fields that pertain to the check codeline.

Field Name	Description
CheckAmount	The amount value on the check. This contains the best estimate of the dollar value of the check obtained by considering both the LAR and CAR. This is the value that we recommend that the customer use as the amount of the check.
CheckCodeline	The number at the bottom of the check that includes routing number, account number, check number and check amount.
CheckDate	The date on the check—machine-printed or handwritten.
CheckNumber	The check number printed on the check.

Field Name	Description
CheckPayeeName	The payee name on the check.
CheckLAR	Legal Amount Recognition (LAR)—the written amount printed on the check.
CheckCAR	Courtesy Amount Recognition (CAR)—the numerical amount printed on the check.
CheckCodeline_AuxiliaryOnUs	Auxiliary On-UUs Field of MICR line.
CheckCodeline_EPC	External Processing Field (EPC) of MICR line.
CheckCodeline_Transit	Transit (Routing) number of MICR line.
CheckCodeline_OnUs1 CheckCodeline_OnUs2	The On-UUs fields are convenience parses provided where, ignoring any left-leading onus symbols in the OnUs field, onus1 is the part to the left of any remaining onus symbol and onus2 is the part to the right.
CheckCodeline_OnUs	Returns all characters from the MICR, from the transitNumber to the amount (if present), or to the right of the string if the amount is not present.
CheckCodeline_Amount	Amount field of MICR line.
CheckCodeline	MICR line.
ACH_AccountNumber	AccountNumber.
ACH_RoutingNumber	Same as transit number but has no symbols.
ACH_SerialNumber	Same as check number but can have additional spaces.

Other fields

Kofax Mobile Deposit Capture also makes use of these other fields.

Field Name	Description
CheckUsable	Indicates whether the check is considered usable in terms of pixel content, presence of OCR, readability and usability of the MICR including country-specific validations for the Transit (routing number) field.
ReasonForRejection	The primary reason for the failure if CheckUsable is returned FALSE. Further reasons can be located in the field alternatives collection for this field. For possible rejection reasons, see Overall usability decision .
RestrictiveEndorsement	A restrictive endorsement on the back of a check, if any, is returned in this field. Returns the value, confidence, and coordinates of the restrictive endorsement phrase, if detected, otherwise returns blank.
RestrictiveEndorsementPresent	A boolean to indicate whether the RestrictiveEndorsement field was returned. TRUE if a restrictive endorsement is found on the back of the check.
CheckMemoline	Returned by the ReturnCheckClassification feature. It returns the text in MEMO or FOR.
CheckPayorName AndAddress	CheckPayorNameAndAddress provides the name and address lines of the payor, semicolon delimited.

Field Name	Description
ISUS	ISUS is a boolean field which specifies if it is US or not for KTA Business rules and/or functionality.

Image quality assurance (IQA)

Each image quality test is returned as boolean "failure" fields, where true indicates the presence of the particular image defect. Each test uses thresholds to determine whether an image is acceptable.

- IQAFailure_UndersizeImage
- IQAFailure_FoldedOrTornDocumentCorners
- IQAFailure_FoldedOrTornDocumentEdges
- IQAFailure_DocumentFramingError
- IQAFailure_DocumentSkew
- IQAFailure_OversizeImage
- IQAFailure_PiggybackDocument
- IQAFailure_ImageTooLight
- IQAFailure_ImageTooDark
- IQAFailure_HorizontalStreaks
- IQAFailure_SpotNoise
- IQAFailure_ImageDimensionMismatch*
- IQAFailure_OutOfFocus


IQA Raw values:

- IQA_topleftCornerWidth
- IQA_topleftCornerHeight
- IQA_bottomleftCornerWidth
- IQA_bottomleftCornerHeight
- IQA_toprightCornerWidth
- IQA_toprightCornerHeight
- IQA_bottomrightCornerWidth
- IQA_bottomrightCornerHeight
- IQA_topEdgeTearWidth
- IQA_topEdgeTearHeight
- IQA_rightEdgeTearWidth
- IQA_rightEdgeTearHeight
- IQA_bottomEdgeTearWidth
- IQA_bottomEdgeTearHeight
- IQA_leftEdgeTearWidth
- IQA_leftEdgeTearHeight
- IQA_additionalLeftScanLinesWidth
- IQA_additionalTopScanLinesHeight
- IQA_additionalRightScanLinesWidth

- IQA_additionalBottomScanLinesHeight
- IQA_documentSkewAngle
- IQA_percentBlackPixels
- IQA_percentAverageImageBrightness
- IQA_percentAverageImageContrast
- IQA_StreaksCount
- IQA_StreaksHeight
- IQA_StreaksLocation
- IQA_SpotNoiseCount
- IQA_FrontRearWidthDifference
- IQA_FrontRearHeightDifference
- IQA_ImageFocusScore

In addition, Parascript provides a series of usability tests that are exposed through the following boolean fields:

- UsabilityFailure_CAR
- UsabilityFailure_LAR
- UsabilityFailure_Signature
- UsabilityFailure_PayeeName
- UsabilityFailure_Date
- UsabilityFailure_Codeline
- UsabilityFailure_PayeeEndorsement*
- UsabilityReadability_CAR
- Usability_Readability_LAR
- Usability_Readability_Date
- Usability_Readability_MICR

 Items marked with an asterisk are only applicable to the back of a check and are only tested if both a front and back are provided.

MICR parsing

Kofax Mobile Deposit Capture reads and parses the MICR line on a check. Extracted MICR components are:

- Amount Field
- OnUs Field
- OnUs1
- OnUs2
- Transit Field (Routing Number)
- EPC
- Auxiliary OnUs Field
- Transit symbol

- Amount symbol
- ACH_AccountNumber
- ACH_RoutingNumber
- ACH_SerialNumber

The script logic used for the field for parsing is shown in the following examples.

ACH_SerialNumber

1. If AuxOnUs exists -> checkNumber = AuxOnUs (business check)
2. If exists(onus2) and (len(onus1) > len(onus2)) -> checkNumber = onus2 (most common personal checks)
3. If exists(onus2) and (len(onus2) > len(onus1)) -> checkNumber = onus1 (most common Western Aux)
4. If len(onus1) >= 14 -> checkNumber = first four digits of onus1 (the industry's so-called "field4" referenced in X9.37) (the lesser common Western Aux including space, dash and undelimited)
5. Return null as default

ACH_AccountNumber

1. If AuxOnUs exists -> accountNumber = onus1 (business check)
2. If exists(onus2) and (len(onus1) > len(onus2)) -> accountNumber = onus1 (most common personal checks)
3. If exists(onus2) and (len(onus2) > len(onus1)) -> accountNumber = onus2 (most common Western Aux)
4. If len(onus1) >= 14 -> accountNumber = all digits from position 5 up of onus1 (the rest of it) (the lesser common Western Aux including space, dash and not delimited)
5. accountNumber = onus1 (Items that probably aren't personal checks or just don't have check numbers on them)

ACH_RoutingNumber

Same as Transit.

CheckCodeline_OnUs

Fields OnUS1+OnUS2.

The following field examples use a MICR line that was composed for illustration and example purposes.

CheckCodeline as returned:

```
{
  "name": "CheckCodeline",
  "text": "C111111CA22222222A33333333333333C444",
}
```


Parsed fields:

```
{
```

```
"name": "CheckCodeline",
"text": "111111222222222233333333333333444",
}
Parsed fields:
{
  "name": "ACH_AccountNumber",
  "text": "33333333333333",
},
{
  "name": "ACH_RoutingNumber",
  "text": "22222222",
},
{
  "name": "ACH_SerialNumber",
  "text": "111111",
},
{
  "name": "CheckCodeline_OnUs",
  "text": "33333333333333444",
}
```

In addition, the following four fields are parsed from the MICR code line :

- **Complete OnUs field:** (including any OnUs characters inside it) as a new KTM field. The previous version had Onus1 and Onus2 as individual components.
- **AccountNumber:** This is a dedicated AccountNumber business field where Kofax Mobile Deposit Capture has intelligently figured out the Account Number. The account number can be Auxiliary, or it can be a number in OnUs.
- **SerialNumber:** This is the SerialNumber field where Kofax Mobile Deposit Capture has determined the check's serial number as best it can. The serial number is the essentially the same as the check number, but it can have additional spaces. The serial number is assumed to be the number in the OnUs MICR field that either matches or seems most likely to match a typical serial number.

 The number in the AuxOnUs field, if present, is highly likely to be a check and/or serial number.

- **RoutingNumber:** The routing number is same as the transit number but has no symbols.

Overall usability decision

The field `CheckUsable` will return a boolean value representing the overall decision of whether this check is usable. Based on the Financial Services Technology Consortium reports, the most important factors are usability of the MICR and whether the image is too dark or too light. Thus the check is considered usable unless any of the following are true:

- Parascript does not identify the image as a check.
- A routing number is not present in the extracted codeline.
- Any of the following fields return true:
 - `UsabilityFailure_Codeline`
 - `IQAFailure_ImageTooLight`
 - `IQAFailure_ImageTooDark`

- `InvalidCodeline` (Mobile Deposit Capture performs validation on some MICR fields, such as EPC.)

If the `CheckUsable` field is set to false, Kofax Mobile Deposit Capture returns one of the following reasons.

- `ImageTooLight` (alert message: Check image is too light.)
- `ImageTooDark` (alert message: Check image is too dark.)
- `CodeLineUsabilityFailure` (alert message: Codeline is not usable.)
- `InvalidTransit` (alert message: Transit number is invalid.)
- `InvalidCodeline` (alert message: Codeline is invalid.)

Multi-field validation rule

The multi-field validation rule for `CheckCodeline` and its constituent components validate the complete codeline structure in terms of symbols and length of characters between these symbols. Additional validation is done to individual components just like the `Transit` field.

The Multi-field validation rule has advantages because the prior rule failed only if `Transit` (the routing number) did not match the checksum calculation. There could be other problems with the codeline (incorrect usage of symbols, incorrect length numbers between symbols) if they were not getting validated. So if routing number is good but the rest of the codeline was invalid, under the prior rule the codeline would still be regarded as valid. These special cases are now handled by the multi-field validation rule.

The Multi-field validation rule ensures that these symbols are correctly placed on the check (starting and ending) and also count the length of the number between these symbols.

For example: `C00000000000CA0000000000A500D000000000C`

The `CheckLocator` in Parascript uses the letters A, B, C, and D in place of special MICR symbols.

Chapter 2

Installation and configuration

Follow the instructions in this chapter to install a new instance of Kofax Mobile Deposit Capture 2.0.0.

Kofax Mobile Deposit Capture is distributed as a zip file that includes the following:

- The Kofax Transformation Modules project folder that contains the project referenced in this guide.
- The Kofax Mobile Deposit Capture 2.0.0 Kofax Transformation Modules project. This project is compatible with Kofax Transformation 7.0.0.5 or higher.
- Kofax TotalAgility package folder that contains the Kofax TotalAgility project import file.

1. Basic installation:

- a. Extract the file `KofaxMobileDepositCapture.x.x.x.zip` where x.x.x is the version you are installing.
- b. Copy the entire contents of the extracted project folder to a shared drive accessible to all your servers.

2. Copy the entire project folder to the system where it will be used.

3. Configure Kofax Mobile Deposit Capture for Real-Time Transformation Interface.

- a. Copy the project files to the server.
- b. In the Kofax Transformation Modules project folder, open `Kofax_Check_Deposit.fpr`.
- c. If using Real-Time Transformation Interface, add a reference to the project folder's fpr file to `web.config`. See Real-Time Transformation Interface documentation for more details. Configure the Transformation project for the intended platform.

4. Configure Kofax Mobile Deposit Capture in the Kofax TotalAgility interface.

- a. Install the latest version of Kofax TotalAgility 7.11 or above.
- b. Copy the `KofaxMobileDepositCapture.zip` file in the Kofax TotalAgility package folder from the project to the shared server where Kofax TotalAgility is installed.
- c. Log in to the Kofax TotalAgility Web application on the server where Kofax TotalAgility is installed.
- d. Click the **Packages** link on the home page.
- e. Click the **Import Package** link on the Packages page.
- f. Click **Browse** to select the above mentioned Kofax Mobile Deposit Capture Kofax TotalAgility package.
- g. Verify all the processes and other Kofax TotalAgility project components are imported.

- h. Click **Import**.
- i. Click **Close**.
- j. Verify that the Kofax Mobile Deposit Capture and that Kofax Mobile Deposit Capture sync process are displayed in the Process section.

Use Kofax Mobile Deposit Capture with companion products

Kofax Real-Time Transformation Interface

By default, the server relays data extracted from the Kofax Transformation project in a JSON format.

i It is also possible for the data to be returned as XML.

The Kofax Mobile Capture SDK includes functionality to be able to crop, classify, and clean up images of supported documents. See Kofax Mobile Capture SDK documentation for more detail. The following provides specific examples of how to call Real-Time Transformation Interface for extraction of a check.

RequestURL

`http://<servername>/mobilesdk/api/CheckDeposit`

Request headers

```
Accept: application/json
Content-Type: multipart/form-data; boundary=-----acebdf13572468
```

Request body

```
-----acebdf13572468
Content-Disposition: form-data; name="xCountry"

US
-----acebdf13572468
Content-Disposition: form-data; name="fieldNameHere"; filename="Laura_Wilson_Check.tif"
Content-Type: image/tiff

<Binary Image Data Will Be Here >
-----acebdf13572468--
```

Response

A JSON file is returned with the data extracted from the images. A text file is included that provides an example of a response. This example uses a MICR line created for illustration purposes.

Kofax TotalAgility

Here is a sample Request URL to call Kofax TotalAgility for extraction of a check:

RequestURL

```
http://<servername>/TotalAgility/Services/SDK/JobService.svc/json/  
CreateJobSyncWithDocuments
```

Request headers

```
Request Headers:  
Accept: application/json  
Host: <servername>  
Content-Type: application/json
```

Request body for TIFF

```
{  
  "jobWithDocsInitialization": {  
    "InputVariables": [  
      {  
        "Id": "ProcessImage",  
        "Value": false  
      }  
    ],  
    "Documents": [  
      {  
        "Base64Data": null,  
        "Data": null,  
        "DocumentGroup": null,  
        "DocumentName": null,  
        "DocumentTypeId": null,  
        "FieldsToReturn": null,  
        "FilePath": null,  
        "FolderId": null,  
        "FolderTypeId": null,  
        "MimeType": null,  
        "PageDataList": [  
          {  
            "Data": null,  
            "Base64Data": "<Image Data Will Be Here>",  
            "MimeType": "image/tiff",  
            "RuntimeFields": {}  
          },  
          {  
            "Data": null,  
            "Base64Data": "<Image Data Will Be Here>",  
            "MimeType": "image/tiff",  
            "RuntimeFields": {}  
          }  
        ],  
        "ReturnAllFields": true,  
        "ReturnFullTextOcr": false,  
        "RuntimeFields": null  
      }  
    ],  
    "StoreFolderAndDocuments": false,  
  }  
}
```

```
"StartDate": null
},
"processIdentity": {
  "Id": null,
  "Name": "KofaxCheckDepositSync",
  "Version": 0
},
"sessionId": "C640521793431F4486D4EF1586672385",
"variablesToReturn": {}
}
```

Request body for PDF

```
{
  "jobWithDocsInitialization": {
    "Documents": [
      {
        "Base64Data": "<Image Data Will Be Here>",
        "Data": null,
        "DocumentGroup": null,
        "DocumentName": null,
        "DocumentTypeId": null,
        "FieldsToReturn": null,
        "FilePath": null,
        "FolderId": null,
        "FolderTypeId": null,
        "MimeType": "application/pdf",
        "PageDataList": null,
        "ReturnAllFields": true,
        "ReturnFullTextOcr": false,
        "RuntimeFields": null
      }
    ],
    "InputVariables": [
      {
        "Id": "ProcessImage",
        "Value": false
      }
    ],
    "StartDate": null,
    "StoreFolderAndDocuments": false
  },
  "processIdentity": {
    "Id": null,
    "Name": "KofaxCheckDepositSync",
    "Version": 0
  },
  "sessionId": "C640521793431F4486D4EF1586672385",
  "variablesToReturn": {}
}
```