

# Kofax Mobile Bill Pay

## Administrator's Guide

Version: 1.6.0

Date: 2018-02-23



© 2018 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>
Related documentation.....	5
Training.....	5
Getting Help for Kofax Products.....	5
Overview.....	7
What's new in this release?.....	7
New in prior releases.....	8
Image processing profile files.....	9
High-level application flow.....	10
Extracted fields.....	11
US biller database.....	11
Configure CityStateZip_CA and CityStateZip_CA_FR Databases.....	12
Integrate Custom CityStateZip_CA Database.....	12
Integrate Custom CityStateZip_CA_FR Database.....	12
Other biller databases.....	13
Set up other databases.....	13
US Postal Service database.....	14
Alternate biller and payer addresses.....	14
Parameters and settings.....	15
Installing and configuring Kofax Mobile Bill Pay.....	19
Basic installation.....	19
Configuring Kofax Mobile Bill Pay for Kofax Real-Time Transformation Interface.....	19
Configuring Kofax Mobile Bill Pay for Kofax TotalAgility.....	20
Scalability.....	20
Extraction.....	21
Find generic dates, amounts and phone numbers.....	21
Find specific kinds of dates, amounts and phone numbers by keyword.....	22
Re-evaluate confidences of values per scripted algorithms.....	23
Divide values into separate fields if required.....	26
Analyze repetitions and normalize confidences.....	26
Override values based on special conditions, if required.....	27
Find potential billers using lookup in industry biller database.....	28
Find potential billers using lookup in custom database (if configured).....	28
Decide whether to run rules-based address location.....	29

Find generic city / state / zip blocks using lookup (start of rules-based address location).....	29
Build complete addresses around city / state / zip blocks using scripted OCR analysis.....	30
Correct street names using postal database.....	30
Merge database and rules-based results.....	30
Send results to a multi-alternative, multi-subfield array fields.....	31
Using Kofax Mobile Bill Pay with companion products.....	34
Kofax Real-Time Transformation Interface.....	34
Example URL call.....	34
Kofax TotalAgility.....	40
Example URL calls.....	40
Maintaining Kofax Mobile Bill Pay.....	63
Performing database updates.....	63
Upgrading Kofax Mobile Bill Pay for Real-Time Transformation Interface.....	63
Uninstalling Kofax Mobile Bill Pay.....	63
Uninstalling Kofax Mobile Bill Pay for Real-Time Transformation Interface.....	63
Uninstalling Kofax Mobile Bill Pay for TotalAgility.....	64

# Preface

This guide includes information about installing and configuring Kofax Mobile Bill Pay to capture a bill image, extract and process payee information, and submit the information to a back-end bank system for payment.

This guide assumes that you have a thorough understanding of Windows standards, application and interfaces as well as Kofax Capture and Kofax Transformation Modules.

## Related documentation

In addition to this guide, Kofax Mobile Bill Pay documentation set includes the following:

- Documentation for this product is available [online](#).
- *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 Kofax Capture solution. Visit the Kofax website at [www.kofax.com](http://www.kofax.com) for complete details about the available training options and schedules.

## Getting Help for Kofax Products

Kofax regularly updates the Kofax Support site with the latest information about Kofax products.

Documentation for this product is available [online](#).

To access some resources, you must have a valid Support Agreement with an authorized Kofax Reseller/ Partner or with Kofax directly.

Use the tools that Kofax provides for researching and identifying issues. For example, use the Kofax Support site to search for answers about messages, keywords, and product issues. To access the Kofax Support page, go to [www.kofax.com/support](http://www.kofax.com/support).

The Kofax Support page provides:

- Product information and release news

Click a product family, select a product, and select a version number.

- Downloadable product documentation

Click a product family, select a product, and click **Documentation**.

- Access to product knowledge bases

Click **Knowledge Base**.

- Access to the Kofax Customer Portal (for eligible customers)

Click **Account Management** and log in.

To optimize your use of the portal, go to the Kofax Customer Portal login page and click the link to open the *Guide to the Kofax Support Portal*. This guide describes how to access the support site, what to do before contacting the support team, how to open a new case or view an open case, and what information to collect before opening a case.

- Access to support tools

Click **Tools** and select the tool to use.

- Information about the support commitment for Kofax products

Click **Support Details** and select **Kofax Support Commitment**.

Use these tools to find answers to questions that you have, to learn about new functionality, and to research possible solutions to current issues.

# Overview

Kofax Mobile Bill Pay is a Smart Mobile Component for Kofax Transformation Modules that uses the Kofax Transformation Modules Optical Character Recognition (OCR) engine, and the Kofax Search and Matching Server (KSMS). Kofax Mobile Bill Pay is delivered as a Kofax Transformation Modules project for the United States and Canada.

For Kofax TotalAgility, Kofax Mobile Bill Pay contains a Kofax Transformation project for reading and extracting data from bills.

Each project contains rules-based extraction methods to locate addresses and identify which address is the biller address.

## What's new in this release?

### **Payer name & address**

Kofax Mobile Bill Pay can now extract the name and address of the Payer (the person or organization who is paying the bill). Address blocks found on the bill are now classified as Biller vs Payer addresses and ranked by confidence based on the following criteria:

- The presence of certain keywords or phrases near to the address which indicate that this is a Biller (such as "remit to") or a Payer (such as "account information").
- The position of the address relative to the top or the bottom of the page
- The presence of personal first names in the Name position
- The presence of industry terms for a Biller (such as "bank") in the Name position

The most confident name and address block is returned to a new set of Payer fields.

The complete collection of confidence-ordered addresses is available for analysis via the multi-alternative Payers field, which is analogous to the existing Billers field.

### **Fields returned as blank**

In earlier versions of Kofax Mobile Bill Pay, some fields that contain no data are returned as a space instead of a blank field. The behavior has been changed to return blank fields by default. The original behavior can be restored by setting a new parameter. See the Parameters section for more details.

## New in prior releases

### **Integration with Kofax TotalAgility**

The Kofax Mobile Bill Pay project is available on the Kofax TotalAgility platform so that the application can leverage the native features of that platform such as multi-tenancy. It is possible to use a Web service API on the KTA platform to send an image and get the extraction results back.

### **Return version number**

A ProductVersion field is available that returns version information.

### **Improved setup of custom databases**

The integration of custom databases has been simplified, with an improved configuration to allow easier replacement of remote fuzzy databases in KSMS (Kofax Search and Matching Server).

### **Overall accuracy improvements**

Overall recognition accuracy has been improved by making better use of significant keywords and their locations on the document page.

### **Database updates**

The US Biller Database and the US postal databases have been updated.

### **AutoPayDate field**

There is now a separate AutoPayDate field. A new configurable option can choose to place the automatic payment date extracted from the bill in the new AutoPayDate field.

### **Support for Canadian French bills**

The same fields for English Canadian bills are now extracted from French language bills. This includes adding support for French language addresses, first and last names, and other language specific modifications to enhance recognition accuracy.

### **Support for phone numbers with alpha characters**

Kofax Mobile Bill Pay now recognizes phone numbers with alpha numeric characters and converts that to a pure numeric format in the phone number field. For example, 1-800-CALL-NOW will be returned as 1-800-225-5669 in the Phone Number field.

### **Due date logic**

When a bill contains a phrase such as "due upon receipt," a new configurable option can choose between returning an empty value for "Due Date" or making an effort to determine best possible "Due Date" by making inferences from other information extracted from the bill. Specifically it will return a relevant current date, such as the Bill Date or Statement Date. To maintain compatibility with earlier versions, returning an inferred date is the default behavior.



**LatePaymentDate field**

There is now a separate LatePaymentDate field. A new configurable option, can choose between putting the late payment date extracted from the bill in the new LatePaymentDate field instead of the previous DueDate field (prior implementation).

**Biller name extraction**

Bill Name extraction has been improved so that the results are more consistent across a range of extraction techniques. The new release of Bill Pay will, by default, attempt to extract the biller name as printed on the document whether the extraction is assisted with a database or found purely by using rules.

**Preferring the printed biller name**

Kofax Mobile Bill Pay will now prefer to extract the biller name as printed on the paper (with appropriate formatting) and only fall back to the third-party database name if rules-based extraction could not find a valid biller name. This clarifies the expectation of what should be returned in the biller name field, and makes the solution independent of the biller name specified in the third-party database record, which is not always succinct and might change release-to-release.

Rules-based extraction, and merging, is now always on by default, with DatabaseConfidenceToSkipFurtherProcessing set to 101. Extraction may be more time-consuming, but the power of merging will lead to greater accuracy because an address found at the same location confidently and independently by database and rules-based techniques is most likely to be correct.

The database is still used to arrive at the result when possible, and elements such as City, State and Zip are still preferred from the database.

A new parameter PreferPrintedBillerName, which defaults to True, can be set to False to achieve the behavior of the previous version of Kofax Mobile Bill Pay

## Image processing profile files

**ImagePerfectionProfile\_BillPay.txt**

An image processing profile is provided with Kofax Mobile Bill Pay. The file can be found at the root of the provided Transformation project folder: KofaxMobileBillPay-x.x.x\KTM Project, where x.x.x is the Kofax Mobile Bill Pay product version. The following text file is included:

ImagePerfectionProfile\_BillPay.txt.

This image processing profile is intended for use with the image of a letter-sized bill captured by a mobile camera as a single page (front or back). It will deskew, crop, and binarize the image. Alternatively, the coupon portion of the bill can be captured by itself, but must be detached for successful cropping.

**Note** This image perfection profile is supported only for use with Kofax Real-Time Transformation Interface 1.8 and later.

The image processing profile is as follows:

```
{
"imageperfectionsettings":
  "_DeviceType_2
  _DoBinarization__DoCropCorrection__DoSkewCorrectionAlt__Do90DegreeRotation_4
```

```
_LoadInlineSetting_[CSkewDetect.convert_to_gray.Bool=1]
_LoadInlineSetting_[CSkewDetect.scale_image_down.Bool=1]
_LoadInlineSetting_[CSkewDetect.scale_down_factor.Int=80]
_LoadInlineSetting_[CSkewDetect.document_size.Int=2]
_LoadInlineSetting_[CSkewDetect.correct_illumination.Bool=0]
_DoScaleImageToDPI_400"
}
```

- `_DeviceType_2`  
Specifies document images come from phone cameras.
- `_DoBinarization_`  
Added to output black and white images, as required by the bill extraction module.
- `_DoCropCorrection_`  
Added to crop the images.
- `_DoSkewCorrectionAlt_`  
Added to deskew the images based on content, because a torn bill coupon may have an edge that is not quite parallel to the printed content.
- `_Do90DegreeRotation_4`  
Added to turn the images right way up.
- `_LoadInlineSetting_[CSkewDetect.convert_to_gray.Bool=1]`  
`_LoadInlineSetting_[CSkewDetect.scale_image_down.Bool=1]`  
`_LoadInlineSetting_[CSkewDetect.scale_down_factor.Int=80]`  
Added to reduce processing time.
- `_LoadInlineSetting_[CSkewDetect.document_size.Int=2]`  
Added to tell the image processor the approximate size of the the larger dimension of the document, in this case 2 (medium).
- `_LoadInlineSetting_[CSkewDetect.correct_illumination.Bool=0]`  
Added to reduce processing time.
- `_DoScaleImageToDPI_400`  
Specifies output image is to be at 400 DPI resolution.

## High-level application flow

1. A user opens a mobile application that instructs them to capture a bill image on their mobile device.
2. The mobile application submits the image to the server where extraction is performed.
3. The extracted information, such as the amount due, is returned to the mobile device.
4. The mobile application displays the information to the user for review and possible correction.
5. The mobile application submits the verified information to the back-end bank system for payment.

## Extracted fields

The following fields are returned:

- ProductVersion
- Name
- AddressLine1
- AddressLine2
- City
- State
- Zip
- AccountNumber
- AmountDue
- DueDate
- LatePaymentDate
- AutoPayDate
- PhoneNumber
- PayerName
- PayerAddressLine1
- PayerAddressLine2
- PayerCity
- PayerState
- PayerZip
- Source
- PayerSource
- Billers
- Payers

## US biller database

The US Kofax Mobile Bill Pay project is integrated with a dedicated biller database to provide better extraction accuracy for addresses. The administrator has the option of setting confidence thresholds that, when met, will skip any subsequent rules-based address extraction methods in order to improve performance.

For example, if the threshold is set to a ninety percent confidence level, then when confidence in the database data exceeds or equals that value, the rules-based methods will not be run, thus saving time.

This can be achieved by setting the `DatabaseConfidenceToSkipFurtherProcessing` parameter - see [Parameters and settings](#).

## Configure CityStateZip\_CA and CityStateZip\_CA\_FR Databases

1. In Windows Explorer, right-click and open the Properties of the database files in this fix pack, and ensure they are not marked as Read-only or Blocked - this avoids problems importing or viewing the databases in Kofax Search and Matching Server (KSMS).
2. Open KSMS - Administration and log in to the appropriate host.
3. On the left panel, deactivate the CityStateZip\_CA and CityStateZip\_CA\_FR databases.
4. Browse to the file `CityStateZip_CA.zip` from Databases folder from this build. The CityStateZip\_CA, CityStateZip\_CA\_FR databases should be available and selected.
5. Select "Activate databases after import" and "Replace existing databases", then click Import. The CityStateZip\_CA, CityStateZip\_CA\_FR databases will be updated and should contain empty records.

## Integrate Custom CityStateZip\_CA Database

1. Create a dump from the relational database to produce a CSV-style file with the required columns:
  - City
  - Prov
  - P Code
  - Pref
  - A/C
  - T/Z
  - DST?
2. Import the CityStateZip\_CA database to KSMS as follows:
  - a. In the left panel of KSMS Administration, select the database you wish to replace with your own (any that have standard or custom configurations as described above). In this case, select the CityStateZip\_CA database.
  - b. **Click Import Settings**
  - c. Point to the dump file either on the local server or via a UNC / URL path and click **Upload**. Make sure to use commas (,) as the field delimiter.
  - d. **Click Import Database**.

## Integrate Custom CityStateZip\_CA\_FR Database

1. Create a dump from the relational database to produce a CSV-style file with the required columns:
  - City
  - Prov
  - P Code
  - Pref

- A/C
  - T/Z
  - DST?
2. Import the CityStateZip\_CA\_FR database to KSMS as follows:
    - a. In the left panel of KSMS Administration, select the database you wish to replace with your own (any that have standard or custom configurations as described above). In this case, select the CityStateZip\_CA\_FR database.
    - b. **Click Import Settings**
    - c. Point to the dump file either on the local server or via a UNC / URL path and click **Upload**. Make sure to use commas (,) as the field delimiter.
    - d. Click **Import Database**.

## Other biller databases

Bill Pay supports integration of two biller databases per locale - standard and custom:

- For the US locale, an industry biller database comes pre-integrated into the standard configuration (Kofax Billers), with a second configuration (Custom Billers) open for adding a supporting database to work in conjunction with this.
- For the CA locale, both the standard (Kofax Billers CA) and custom (Custom Billers CA) configurations are open for replacement since Bill Pay does not currently ship with a Canadian industry biller database.

## Set up other databases

You can perform the following steps to configure other biller databases. Note that the other database must have exactly the same column names. Therefore, simply pointing Kofax Search and Matching Server to the relational database directly will not work. A database dump from the relational database is required.

1. Create a dump from the relational database to produce a CSV-style file with the required columns.
  - Name
  - AddressLine1
  - AddressLine2
  - City
  - State
  - Zip
  - Class (does not require a value)
  - Masks (does not require a value)

**Note** At this point it is not necessary for the columns names to match, or even exist. Also, the column order is irrelevant.

2. Import the database to Kofax Search and Matching Server (KSMS) as follows.
  - a. On the left panel of KSMS Administration, select the database you wish to replace with your own (any that have standard or custom configurations as described above).
  - b. Click **Import Settings**.
  - c. Point to the dump file either on the local server or via a UNC / URL path and click **Upload**. (Edit the column names if they do not already match the names that Kofax Mobile Bill Pay requires.)
  - d. Click **Import Database**.
3. In the server variables (Real-Time Transformation Interface script variables or Kofax TotalAgility initialization variables), set the parameter `UseCustomDatabaseMatching` to TRUE if you edited one of the custom databases and need to use it in conjunction with the standard databases.

## US Postal Service database

Kofax Mobile Bill Pay is also integrated with the US Postal database to provide a spell check capability and the correction of street addresses as an optional formatting step. This helps normalize addresses extracted using the biller database or rules-based techniques.

## Alternate biller and payer addresses

The Billers and Payers fields expose all the alternative addresses extracted with different techniques within the Transformation project. Developers can use this data to decide what is presented to the device user within their application.

These are confidence-ordered results and they are referred to as "alternatives" even if there is only one result. The first listed alternative is the highest confidence, and its data is copied into the main fields.

A Source field and subfield for each Billers and Payers alternative will indicate the source of the data (Database, Document, or more specific values). The order of address alternatives is based on the generalized confidence score.

The following are possible Source subfield values for each alternative of the Billers and Payers fields:

### **Database**

The address is located using a successful fuzzy match with an official database containing only biller records. The confidence of the fuzzy match (the amount by which all words in the database record match those in the full text OCR of the document) contributes to the overall score but is not the only factor.

### **Document**

The address is located using rules. Firstly, a fuzzy match with an all-purpose database identifies the City, State and Zip location. The Name and Address lines are then extrapolated using OCR words in positions above the City, State and Zip (regular multi-line address).

### **DocSameLine**

The biller address is located using rules. Firstly, a fuzzy match with an all-purpose database identifies the City, State and Zip location. The Name and Address lines are then extrapolated using OCR words in positions to the left of the City, State and Zip (all address data on the same line).

### **DocDBLMiss**

The address is located using rules. Firstly, the Zip is identified using a straightforward format locator (regular expression) match. A word in the full page OCR is also detected nearby matching the format for a State, therefore this is considered a "miss" from the City, State and Zip database locator. The miss may be due to a failed match on the City or other factors. The Name and Address lines are extrapolated using OCR words in positions relative to the City, State and Zip.

### **DocRepairState**

The address is located using rules. Firstly, the Zip is identified using a straightforward format locator (regular expression) match. There is no valid nearby State value in the full page OCR, an occasional possibility due to the engine being optimized for speed, therefore is not technically a miss from the City, State and Zip database locator. Using zonal OCR (optimized for accuracy) in the localized area, a valid State is detected, therefore this is considered an OCR "repair". The Name and Address lines are extrapolated using OCR words in positions relative to the City, State and Zip.

### **Merged**

The address is located using a successful fuzzy match with a dedicated database containing only biller records. The exact same address data is found using rules-based techniques, confirming the address match, therefore the two address alternatives are merged into one and the confidence is increased.

## Parameters and settings

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

The following mobile parameters can be set in the project or the query string to maximize the returned data in the Kofax Mobile Bill Pay project:

### **ComprehensiveKeywordDetection**

When set to TRUE, this parameter allows for more flexibility in locating and identifying relevant keywords.

When set to FALSE keywords will be excluded if not exactly west, north (sometimes south) or within a reasonable distance of the next alternative. This mimics the behavior in earlier releases.

The default value will be FALSE for performance reasons.

### **CorrectStreetWithUspsDB**

This is the flag to enable the street name correction via the United States Postal Service database. Set to TRUE to perform a fuzzy match and spelling correction of the extracted street name with the state-specific USPS database, to correct any OCR errors and normalize the address format. Enabling this option will slightly increase processing time. The default value is FALSE.

### **Country**

The Country locale of the bills being processed. Set to CA to process Canadian bills (English or French language). The default value is US.

### **CustomerName**

This is the parameter with which your mobile application can pass the customer name to the Kofax Mobile Bill Pay project to improve accuracy. For example, if the customer name is identified in your mobile application, Kofax Mobile Bill Pay can better differentiate between the biller name and the customer name on a captured bill.

This parameter is handled as follows:

- The `CustomerName` parameter splits the name up into separate words, so "JOHN" and "SMITH" are fuzzily searched independently in the Name line
- The feature can decrement the confidence level by a maximum of 20% per word, depending on the fuzzy match confidence. These scores are cumulative for each identified word, so if both JOHN and SMITH match, it is possible to end up with as much as 40% decremented from the overall score. This means it's better to provide as many names as possible.
- Extra words in the reference customer name string that are not printed on the bill, such as "JOHN SMITH ELEPHANT" or "JOHN ELEPHANT SMITH" do not count against the match. They are simply ignored. So including middle names isn't a problem if they are not printed.
- There is no attempt to match any word under 4 characters, fuzzy or exact. So if the name is "JOHN LEE," "LEE" isn't searched. This also automatically excludes middle initials.

Based on the above, to use this feature provide first, middle and last names if available. No formatting of periods is required, though the dash should be removed from hyphenated names so that each name is separated by a space.

### **DatabaseConfidenceToSkipFurtherProcessing**

This is the threshold above which Kofax Mobile Bill Pay trusts the result from the fuzzy database match against an official billers database, and skips any further document / rules-based extraction of addresses. Set to a lower number to run rules-based extraction less often to improve performance, or set to 0 to disable rules-based extraction entirely. Note that when `MergeAddressesAtSameLocation` is TRUE, lowering this number is not recommended, as there will be fewer rules-based addresses available for merging. The default value is 101, which means that rules-based extraction is always run.

### **EnableSeparateAutoPayDate**

The scheduled automatic payment date is, by default, incorporated into the `DueDate` field and returned ahead of any earlier desired due date. Set this parameter to TRUE to enable a new field, `AutoPayDate`, and return the automatic payment date in this field and any other printed due date in the `DueDate` field. The default value is FALSE.

### **EnableSeparateMinAmountDue**

The minimum payment due is, by default, incorporated into the `AmountDue` field and returned ahead of any larger amount. Set this parameter to TRUE to enable a new field, `MinimumAmountDue`, and return the minimum amount due in this field and any other printed amount due in the `AmountDue` field. The default value is FALSE.



### **EnableSeparateLatePaymentDate**

The latest permitted payment date before incurring a late fee is, by default, incorporated into the DueDate field and returned ahead of any earlier desired due date. Set this parameter to TRUE to enable a new field, LatePaymentDate, and return the late payment date in this field and any other printed due date in the DueDate field. The default value is FALSE.

### **MergeAddressesAtSameLocation**

This is the flag to disable "merging" of two agreeing address results found at the same location from a fuzzy database match and rules-based extraction. Set to "FALSE" to stop merging the two agreeing address results. If you disable merging, you will then get both a "Database" and one of the rules-based alternative types for this given address. See the description of the "Merged" source for more details. The default value is TRUE.

### **PreferAKA**

This is the flag to disable the feature which normalizes all known AKAs of the biller name into one consistent name for that biller. Set this flag to TRUE to ensure the specific AKA is returned. The default value is FALSE.

### **PreferPrintedAddressLines**

When set to TRUE, this parameter will cause Kofax Mobile Bill Pay to prefer the address as it is printed on the bill over an address extracted from the biller databases.

The default value is FALSE, to maintain backwards compatibility.

### **PreferPrintedBillerName**

This is the flag to control whether the biller name as printed on the bill is preferred to the biller name from the third-party biller database record, when database and rules-based results are merged for the final return of the biller address. Set this flag to FALSE to prefer the biller name from the database, but note that this may be a legal or specific regional name and not the commonly-used company name. The default value is TRUE.

### **PreferUnformattedAccountNumber**

This is the flag to replace the formatted account number value with the original OCR text at the extracted location, including any original delimiters. Set this flag to TRUE to ensure that the unformatted account number is returned. The default value is FALSE.

### **ProcessImage**

This is the flag that enables image-processing server side, consequently this parameter does not require an "x" regardless of being used with Real-Time Transformation Interface or with Kofax TotalAgility .

### **RereadDocAddresses**

This is the flag to disable localized OCR rereads that are run to enhance extraction accuracy. Set to FALSE to disable the rereads. Disabling this will slightly improve processing time. The default value is TRUE.

**ReturnNoDateWhenDueOnReceipt**

This parameter controls whether the date is returned as blank if a phrase such as "due on receipt" is present. Set this to TRUE to blank the dates in this case. The default value is FALSE.

**ReturnZeroWhenNothingToPay**

This parameter controls whether the amount is returned as zero (0.00) if a phrase such as "do not pay" or "not a bill" is present. Set this FALSE to return an amount due in all cases. The default value is TRUE.

**UseBillerDatabaseMatching**

This is the flag to disable the fuzzy biller database matching. Set to FALSE to skip biller matching and rely solely on rules-based extraction techniques. The default value is TRUE.

**UseCustomDatabaseMatching**

This parameter can be used to toggle a custom biller database once configured. Set to TRUE to start pulling results from your configured custom database. The default value is FALSE.

**Language**

Set to EN or FR to guide Kofax Mobile Bill Pay so that it handles and formats text correctly for Canadian bills. If left blank, the software will detect the language automatically based on the characters present on the bill. The default value is blank.

**ReturnSpaceForBlankFields**

Set to TRUE to re-enable the legacy behavior of returning a space instead of a blank string when the value cannot be found on the bill. The default value is FALSE.

# Installing and configuring Kofax Mobile Bill Pay

Follow the instructions in this chapter to install a new instance of Kofax Mobile Bill Pay. To upgrade from an earlier version simply remove that version and continue with these instructions.

Kofax Mobile Bill Pay is distributed as a ZIP file that includes the following:

- `Billers.zip`, `BillerAKAs.zip`, `Billers_CA.zip`, and `BillerAKAs_CA.zip`, which are the databases required by the project (United States and Canada only).
- `CustomBillers.zip`, `CityStateZip_CA.zip`, and `CustomBillers_CA.zip`, which are empty custom databases that can be replaced with new data once imported.
- `CityStateZip.zip`, which is the database of cities, regions and postal codes used in rules-based extraction
- `USPS-States.zip`, which is the street address database.
- The Kofax Transformation Modules project folder which contains the project referenced in this guide.
- A Kofax TotalAgility folder containing the `KofaxBillPay.zip`.

**Important** To use Kofax Mobile Bill Pay, your mobile application must connect to the Kofax Real-Time Transformation Interface Web service, or the Kofax TotalAgility server. See the *Real-Time Transformation Interface Administrator's Guide* or the TotalAgility documentation for more information.

Each Real-Time Transformation Interface Web server must also have Kofax Transformation Modules and the Real-Time Transformation Interface, but does not need to have KSMS installed locally.

## Basic installation

1. Extract the file **KofaxMobileBillPay-x.x.x.ZIP** where x.x.x is the version you are installing.
2. Copy the entire contents of the extracted project folder to a shared drive where it is accessible to all your servers.

## Configuring Kofax Mobile Bill Pay for Kofax Real-Time Transformation Interface

1. Install the latest Kofax Transformation Modules patch. This patch supports the decryption of the Biller Databases. See the Kofax Support page, [www.kofax.com/support](http://www.kofax.com/support), for information about patches.
2. Verify that the Biller Directory databases and the ZIP code database are on a server with KSMS administration components installed.
3. Import the standard biller databases, the custom databases, the CityStateZip database, and the USPS-States database into KSMS. See the KSMS documentation for instructions.

4. If your Real-Time Transformation Interface server is not installed on a server where KSMS is installed, use the Kofax Transformation Modules Project Builder to edit the KTM project to point each remote fuzzy database on the Databases tab in Project Settings to the correct KSMS server location.
5. Add project mapping to the Real-Time Transformation Interface `web.config` file. Refer to Real-Time Transformation Interface documentation for more details.

## Configuring Kofax Mobile Bill Pay for Kofax TotalAgility

1. Login to the TotalAgility Designer.
2. Navigate to the Packages module -Select Import Package.
3. Select Import Package.
4. Browse to the `KofaxBillPay.zip` file in the Kofax TotalAgility Package folder.
5. Click Import to import the package
6. If your KSMS server is not installed on the same server as the TotalAgility Transformation Server, use the Kofax Transformation Designer to edit the `Kofax_Bill_Pay` or `Kofax_Bill_PayCA` shared transformation project to point to the correct KSMS server location. Refer to the TotalAgility documentation for details.

## Scalability

A single KSMS system (single KSMS node or group of nodes with KSMS load balancer) can handle a maximum of 128 concurrent connections by default. See the KSMS documentation for more information on configuring connection limits. Because Real-Time Transformation Interface spawns worker processes proportional to the number of concurrently executing requests and an individual Bill Pay request requires between 3-4 KSMS connections, the KSMS system can sustain a maximum load of 30 concurrent Real-Time Transformation Interface requests.

To scale to higher levels, it is recommended to have a dedicated KSMS system for each Real-Time Transformation Interface server processing Bill Pay requests and for load to be distributed to each Real-Time Transformation Interface server consistent with the limits above. Scaling vertically within the same Real-Time Transformation Interface server by adding more cores is not recommended - while Real-Time Transformation Interface may handle the OCR load, the generation of further worker processes based on additional concurrent requests will exceed the supported connection limit for the KSMS system.

For example, for a mobile environment with a maximum concurrent request load of 120 requests, the incoming requests must be divided by a load balancer in front of four independent Real-Time Transformation Interface servers, each pointing to an independent KSMS system. For a mobile environment with a maximum concurrent request load of 240 requests, the incoming requests must be divided by a load balancer in front of eight independent Real-Time Transformation Interface servers, each pointing to an independent KSMS system.

# Extraction







Extraction in Kofax Mobile Bill Pay is one of the more complex of the SMC products, mainly because of the size and variation of the documents being captured, and the variety of different data types being extracted.

The general flow is as follows:

1. Find generic dates, amounts and phone numbers.
2. Find specific kinds of dates, amounts and phone numbers by keyword.
3. Re-evaluate confidence of values by scripted algorithms.
4. Divide values into separate fields if required.
5. Analyze repetitions and normalize confidences.
6. Override values based on special conditions, if required.
7. Find potential billers using lookup in industry biller database.
8. Find potential billers using lookup in custom database (if configured).
9. Decide whether to run rules-based address location.
10. Find generic city / state / zip blocks using lookup (start of rules-based address location).
11. Build complete addresses around city / state / zip blocks using scripted OCR analysis.
12. Correct street names using postal database.
13. Merge database and rules-based results.
14. Send results to multi-alternative, multi-subfield array fields.

## Find generic dates, amounts and phone numbers























The first step of extraction uses format locators to find dates, amounts and phone numbers in a generic way - if pieces of the OCR match the given regular expression formats, they are collected for analysis.

-  FL\_Dates
-  SL\_Dates
-  FL\_Amounts
-  SL\_Amounts
-  FL\_PhoneNumbers
-  SL\_PhoneNumbers

You can see that a script locator follows each format locator. This takes the results from the format locator and eliminates values based on extra conditions if required.

## Find specific kinds of dates, amounts and phone numbers by keyword

The next step is to use the "keyword" feature of format locators to find surrounding words that indicate which type of date, amount or phone number has been discovered.

-  FL\_PhoneKeyword
-  FL\_ZipCodes
-  FL\_LongWords
-  SL\_CodeLineLongWords
-  FL\_CodeLineComponents
-  FL\_AmountDue\_Near
-  FL\_AmountDue\_Med
-  FL\_MinimumDue\_Near
-  FL\_MinimumDue\_Med
-  FL\_MinimumDue
-  AGL\_AmountDue
-  FL\_MisleadingValues
-  FL\_AccountNumber\_Near
-  FL\_AccountNumber\_Med
-  FL\_PolicyNumber\_Near
-  IGL\_AccountNumber
-  FL\_DueDate\_Near
-  FL\_DueDate\_Med
-  FL\_LatePaymentDate\_Near
-  FL\_LatePaymentDate\_Med
-  FL\_AutoPayDate\_Near
-  FL\_AutoPayDate\_Med

Each locator is dedicated to finding one specific piece of data, with the appropriate keywords defined in that locator. By porting the results from the generic locator that ran before, it reduces processing time and allows the search for dates, amounts, and phone numbers to be defined only once for easier maintenance.

You can see that with the exception of phone numbers, each locator is also dedicated to finding keywords Near or at Medium distance. This is because it is not possible to define the same keyword at both Near and Medium distance in a format locator.

So, after these locators have run, we have both raw values (with coordinates) plus keywords (with coordinates and confidences) related to each piece of data.

## Re-evaluate confidences of values per scripted algorithms

The format locator is too forgiving when it comes to deciding whether a keyword satisfies the Relation, Distance and Weight settings. Often it will give a keyword some kind of confidence even if it is not at the specified position. Therefore the confidences are re-evaluated in script by two algorithms:

- The standard algorithm is a rigid analysis which applies the confidences only of keywords that fit the exact criteria of the format locator - this runs every time
- The comprehensive algorithm is more refined and allows extra keywords to influence the confidence if they are connected to the primary keywords in a specific way - this runs only if a parameter is enabled

The algorithms run in the SL\_ConfidencePenaltyForRelation script locator.

The script locator loads each format locator and looks at how the keywords have been defined:

The screenshot shows the 'Properties of Format Locator' dialog box, specifically the 'Evaluation Settings' tab. The 'Keywords' section is active, showing a list of keywords with their configurations. The 'account' keyword is selected and highlighted in blue. The dialog includes a 'Keyword' input field with 'account', a 'Weight' slider set to -40, and a 'Distance' dropdown set to 'Medium'. A grid of directional buttons (NW, N, NE, W, E, SW, S, SE) is visible for defining the keyword's relation to other keywords. The main table lists various keywords with their 'Use', 'D', 'Keyword', 'Relation', 'Distance', 'Weight', 'Search Exact', and 'Whole Phrase' settings.

Use	D	Keyword	Relation	Distance	Weight	Search Exact	Whole Phrase
<input checked="" type="checkbox"/>		account	W, N	Medium	-40	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		amount	W, N	Medium	40	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		auto	W, N	Medium	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		autopay	W, N	Medium	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		balance	W, N	Medium	40	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		by	W, N	Medium	40	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		debited	W, N	Medium	80	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		due	W, N	Medium	80	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		due\$	W, N	Medium	80	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		duo	W, N	Medium	80	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		gross	W, N	Medium	80	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		installment	W, N	Medium	80	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		late	W, N	Medium	-100	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		mail	W, N	Medium	40	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		minimum	W, N	Medium	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		now	W, N	Medium	40	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		on	W, N	Medium	40	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		pay	W, N	Medium	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		payment	W, N	Medium	40	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		previous	W, N	Medium	-100	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		remittance	W, N	Medium	80	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		total	W, N	Medium	40	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>		withdrawn	W, N	Medium	80	<input type="checkbox"/>	<input checked="" type="checkbox"/>

### Keyword

These are short (in general, a single word) so they can "stack up" with each other for all possible variations of the phrase on unseen bills.

### Relation

The standard algorithm enforces that keywords are actually at the required relation before accumulating the scores:

- If the keyword is set to N, the script requires that at least some part of the keyword is directly above some part of the value.
- If the keyword is set to S, the script requires that at least some part of the keyword is directly below some part of the value.
- If the keyword is set to W, the script requires that at least some part of the keyword is directly to the left of some part of the value.

### Distance

The **standard algorithm** enforces that keywords are actually at the required distance before accumulating the scores:

- Near in the N or S direction is defined as "within one text line of the value."
- Medium in the N or S direction is defined as "within two text lines of the value."
- Near in the W direction is defined as "within  $\frac{1}{4}$  of the document's width."
- Medium in the W direction is defined as "within  $\frac{1}{2}$  of the document's width," allowing for extra empty space or padding words

### Weight

If the Relation and Distance criteria are met, the standard algorithm adds this Weight to the confidence of the related value. The weights take only three values:

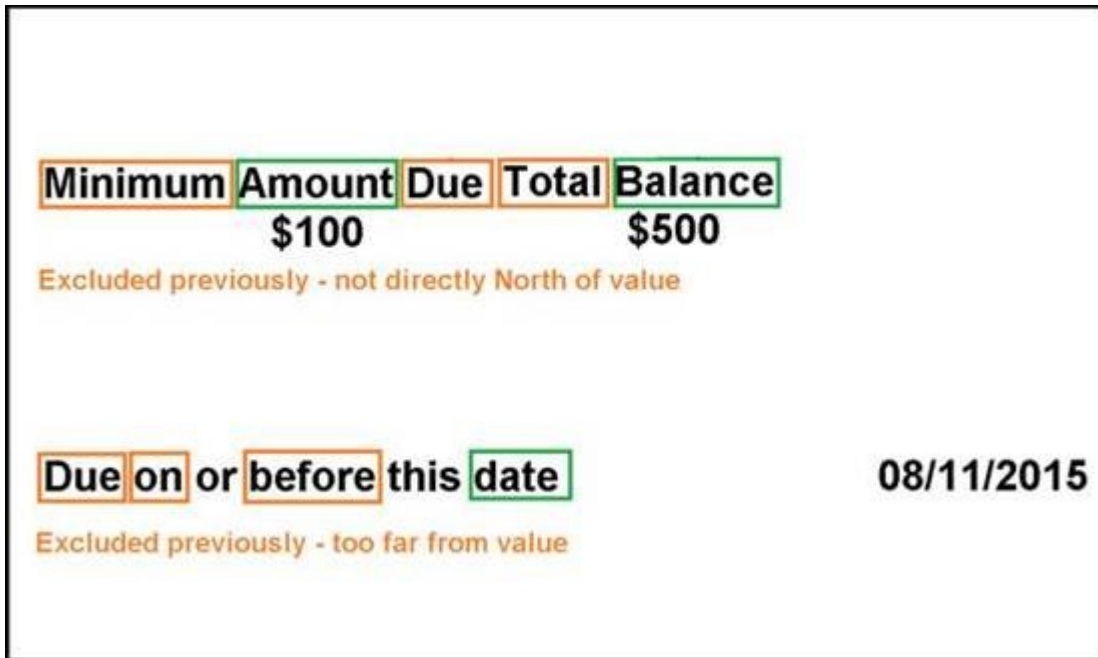
- 100 is a certain indicator that means we know absolutely what this value is based on finding the keyword nearby. An example would be "minimum" which almost always indicates this is the minimum amount to be paid. A negative example is "late," indicating that the value is for a late payment fee and should not be considered.
- 80 is a strong indicator - "due," for example is usually correct but not enough to be absolutely certain.
- 40 is a weak indicator which help us build up a score but are still fairly generic, words such as "amount" and "payment" are weak because there are many amounts and payments quoted on the document, but not necessarily the amount due. The weak indicators do strengthen values and provide fallbacks if no strong or certain indicators are located.

The comprehensive algorithm applies the score of some keywords that were previously excluded, due to not satisfying the standard rigid direction and distance requirements of the standard algorithm. These extra keywords are sometimes crucial in identifying the correct alternatives, or excluding the wrong alternatives.

A parameter, ComprehensiveKeywordDetection, enables this behavior when set to TRUE. The default value is FALSE for performance reasons.

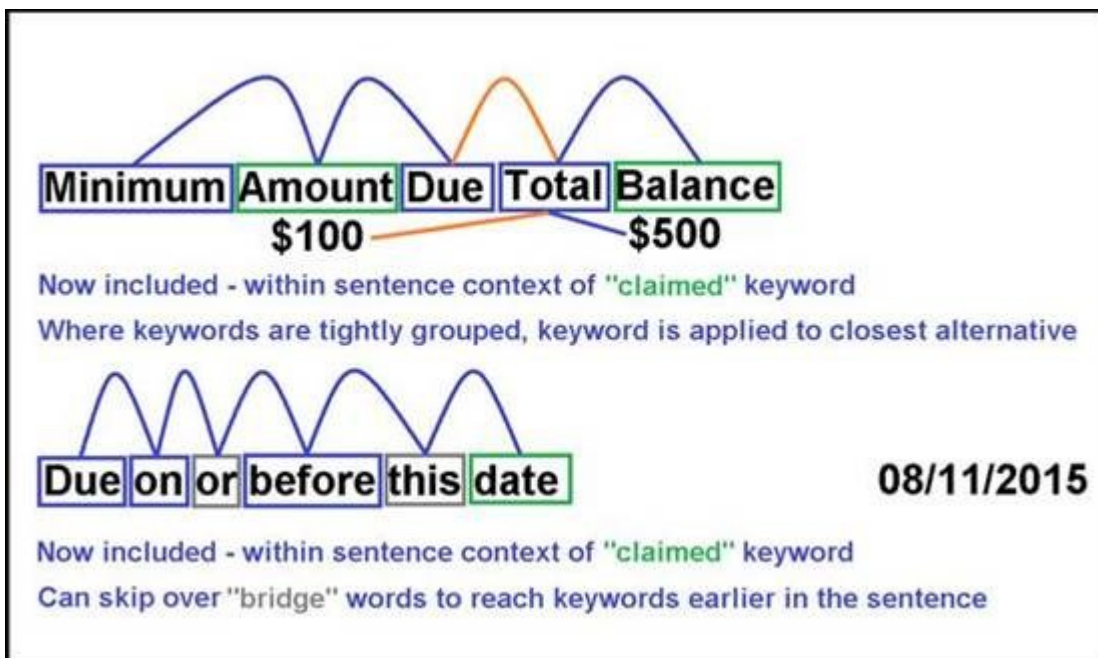
In the standard algorithm keywords are excluded if not exactly W, N (sometimes S) or within reasonable distance of alternative.





However, these "unclaimed" keywords are related to the "claimed" (used) keyword in that they are within "sentence context" - within a bunch of words closely grouped such that they form a sentence / phrase.

The comprehensive algorithm applies the scores (positive or negative) of keywords within sentence context of a "claimed" keyword. As long as one keyword satisfies the distance / direction conditions, its neighbors are applied too.



## Divide values into separate fields if required

By default, some fields have fallbacks depending on the type of data available on the bill.

### **Amount Due**

By default, Kofax Mobile Bill Pay will return amounts to the AmountDue field in the following order of priority:

- The minimum amount due, if one is specified on the bill.
- The total amount due, or balance.

If the EnableSeparateMinAmountDue parameter is set to TRUE, the MinimumAmountDue field returns the minimum payment amount if printed on the bill. The AmountDue field returns the total outstanding account balance if printed on the bill. The SL\_MinimumDue script locator divides the alternatives into different locators.

### **Due Date**

By default, Kofax Mobile Bill Pay will return dates to the Due Date field in the following order of priority:

- The auto pay date, if one is specified on the bill.
- The late payment date (latest date after which a customer will be charged a fee), if one is specified on the bill.
- The explicit due date.
- A bill or statement date.

If the EnableSeparateAutoPayDate parameter is set to TRUE, the AutoPayDate field returns the auto pay date if printed on the bill. The DueDate field returns the explicit due date if printed on the bill. The SL\_AutoPayDate script locator divides the alternatives into different locators.

If the EnableSeparateLatePaymentDate parameter is set to TRUE, the LatePaymentDate field returns the late payment date if printed on the bill. The DueDate field returns the explicit due date if printed on the bill. The SL\_LatePaymentDate script locator divides the alternatives into different locators.

## Analyze repetitions and normalize confidences

Repeated values are often a sign of the correct value. For example, the total balance is often repeated multiple times on the bill, whereas other incidental amounts are less frequent. Also, if multiple locators identify the same value, this should be reassuring.

Therefore, these script locators call a common helper function BoostRepeatedValues() to find repetitions and give a small boost:

- ☰ SL\_RepeatedAmounts
- ☰ SL\_RepeatedMinimumAmounts
- ☰ SL\_RepeatedAccountNumbers
- ☰ SL\_RepeatedDates
- ☰ SL\_RepeatedLatePaymentDates
- ☰ SL\_RepeatedAutoPayDates
- ☰ SL\_RepeatedPhoneNumbers

The function is smart enough to format the values before comparison and ensure values are not considered "different" due to noise or other factors. One factor in particular is redaction - some account numbers are redacted, removing the first 12 characters of a credit card number for example, but other parts of the bill may display the full number. These should all be considered the same number when it comes to boosting it due to repetition.

The final thing this function does is normalize the alternative confidences. Due to the weighting system from the standard and comprehensive keyword algorithms, some values may end up with a confidence above 100%. Since this is not acceptable as a return value, in this case the highest confidence value above 100% is scaled down to 100%, with other values scaled down by the same factor.

## Override values based on special conditions, if required

There are two conditions in which certain field values may need to be overridden.

### **Do not pay**

If the bill has already been paid or is just a statement and not an outstanding bill, a phrase such as "Do Not Pay" may be printed on the bill. Since an amount due may have been extracted anyway, the customer must set the parameter `ReturnZeroWhenNothingToPay` to `TRUE` to overwrite the `AmountDue` field with zero (0.00).

A format locator locates such a phrase and a script locator performs the action of editing the alternative value:

- `FL_DoNotPay`
- `SL_DoNotPay`

### **Due upon receipt**

If the bill is delinquent or the buyer would like to be paid to be paid immediately, a phrase such as "Due Upon Receipt" may be printed on the bill. Since a due date may have been extracted anyway, the customer must set the parameter `ReturnNoDateWhenDueOnReceipt` to `TRUE` to overwrite the `DueDate` field with blank.

A format locator locates such a phrase and a script locator performs the action of editing the alternative value:

- `FL_DueOnReceipt`
- `SL_DueOnReceipt`

## Find potential billers using lookup in industry biller database

Kofax Mobile Bill Pay ships with an industry biller database that is licensed to Kofax and updated by a third party on a monthly basis.

This is not just a database of all addresses, it is a database specifically for common billers. Therefore, a confident hit in the database lookup for this database usually means we have found the correct biller name and address. Database locators are configured to use remote fuzzy databases, assuming that the provided KSMS zip files have been imported and are activated (a mandatory configuration step for Kofax Mobile Bill Pay):

-  SL\_UseBillerDB
-  DBL\_Biller
-  DBL\_BillerAKA
-  DBL\_Biller\_CA
-  DBL\_BillerAKA\_CA

By default, this lookup runs every time as a starting point. It can be disabled by setting the parameter UseBillerDatabaseMatching to FALSE, and relying solely on rules-based address location, though this is not recommended unless the customer would prefer to use a custom database instead (see below).




The AKAs database is a way that Kofax manages billers that are listed in the database under multiple names. In general, the third party and Kofax attempt to include just a single address record for a given biller in the main database, with alternative / AKA names for that biller in the second database. This ensures records for the same biller do not clash during in the lookup, and also improves performance by reducing the size of the main database.

**Note** The biller and AKA database records are populated only for United States. At the time of writing, placeholder databases for Canada are provided but do not contain any records.

## Find potential billers using lookup in custom database (if configured)

Kofax Mobile Bill Pay provides an architecture for integrating a custom / secondary database of billers that the customer may have available. This functionality is more critical for Canada customers, due to the lack of an industry biller database. However, it may also be useful for the United States region, since a business may have a more specific set of records for their known clients.

The extraction works exactly the same way, with database locators pointing to remote fuzzy databases in KSMS.

-  SL\_UseCustomDB
-  DBL\_CustomBiller
-  DBL\_CustomBiller\_CA

Instructions for integrating custom databases can be found in the Kofax Mobile Bill Pay Administrator's Guide. As one of the steps, the parameter `UseCustomDatabaseMatching` must be set to `TRUE`, as by default these locators are not run. This parameter can be used to toggle the use of the custom database, once configured.

## Decide whether to run rules-based address location

If a database result has higher confidence than the `DatabaseConfidenceToSkipFurtherProcessing` parameter value, rules-based address location is skipped.

The default value of `DatabaseConfidenceToSkipFurtherProcessing` is 101. This means that rules-based address location always runs (because a database match can never achieve a confidence of 101). This is important to ensure an attempt is always made to extract the biller name (and address) from the document, and guarantees Kofax Mobile Bill Pay can always utilize the "merging" approach of combining the power of database and rules-based address location, to verify each other and take the best data from both.


Faster performance can be achieved by setting `DatabaseConfidenceToSkipFurtherProcessing` to a lower number such as 80, but it should be noted that doing so may cause unpredictable behavior for the accuracy of the biller name.

The logic that analyzes the `DatabaseConfidenceToSkipFurtherProcessing` parameter resides in the `SL_EnhanceBillerDB` script locator.

## Find generic city / state / zip blocks using lookup (start of rules-based address location)

The starting point of rules-based address location is to generically find city / state / zip triplets. These vary in format per locale, and very different database lookups are used for United States and Canada - the latter country uses two lookups since a bill may be in English or French language. However, the principal is the same - these are the indicators that we have an address block.

They are database locators pointing to remote fuzzy databases which are provided with the product and must be imported into KSMS before use:








-  `DBL_CityStateZip`
-  `DBL_CityStateZip_CA`
-  `DBL_CityStateZip_CA_FR`

The city / state / zip databases are usually updated with every significant (point) release of Kofax Mobile Bill Pay.

## Build complete addresses around city / state / zip blocks using scripted OCR analysis

The next step is to locate the address lines and biller name that sit above or to the left of the city / state / zip blocks, depending on whether it is a multi-line or single-line address.

This is the most complicated part of Kofax Mobile Bill Pay and the code is broken into a number of script locators that parse OCR text lines, repair individual words with zonal OCR re-reads, and attempt to sort the valid address lines from inter-dispersed garbage, based on their content and position:

-  SL\_RepairCSZCoordsFromDBL
-  SL\_BestCityStateZipPerLocation
-  FL\_BillerKeywords
-  FL\_CustomerKeywords
-  SL\_DynamicNegativeKeywords
-  FL\_DynamicNegativeKeywords
-  SL\_AddressBlocks

During this process, a confidence is assigned to the complete address block, based on the many traits of the text that was collected, including its format, the presence of telling items like a PO box, the presence of biller-related or customer-related keywords nearby, and other factors. Based on this confidence the addresses most likely to be the biller are pushed to the top of the list.

## Correct street names using postal database

A minor feature of Kofax Mobile Bill Pay is to use the USPS postal database to perform a fuzzy spell check on the extracted street name on a word-by-word basis to counteract small OCR errors and therefore improve accuracy via the SL\_PostalStreetCorrection script locator.

The performance impact of this feature was considered too large in relation to the improvement in accuracy, therefore this feature is disabled by default. Set the parameter CorrectStreetWithUspsDB to TRUE to enable this feature.

## Merge database and rules-based results

Merging is the operation of taking the results of the biller/custom database lookups, and the results from rules-based address location, and seeing if any of them agree. If certain elements such as the state or zip (including ZIP+4 for the United States), the PO box number, and the unit of the address line agree, and the position of the data also agrees, the address records are merged together, taking the best data from both, and their confidence boosted via the SL\_MergeDocdAndDb script locator.

This functionality can be disabled by setting the parameter MergeAddressesAtSameLocation to FALSE, but this is not recommended.

In the event of a merge, Kofax Mobile Bill Pay will prefer the biller name printed on the document (inside the rules-based address result) if it is found, controlled by the parameter `PreferPrintedBillerName` which defaults to `TRUE`.

To always prefer the database biller name during merging, set `PreferPrintedBillerName` to `FALSE`.

If no potential biller name is found printed on the document by rules, the database biller name will be preferred as fallback.

If the database biller name was preferred from any of these circumstances, and the value of the `PreferAKA` parameter is `TRUE`, Kofax Mobile Bill Pay will attempt to return a more specific name from the AKAs database.

Here is an example of how `PreferPrintedBillerName` and `PreferAKA` work together:

- Kofax Mobile Bill Pay tries to merge the database address with a rules-based address if the location and key data elements match
- If there was a matching rules-based address at the database address location, and the rules-based extraction located a potential biller name, that name will be returned.

**Note** `PreferAKA` will not be relevant in this scenario

- If there was no matching rules-based address at the database address location, or the rules-based extraction could not locate a biller name, the database biller name will be used.
  - If `PreferAKA` is `TRUE`, Kofax Mobile Bill Pay will try to return a more specific AKA name.
  - If `PreferAKA` is `FALSE`, Kofax Mobile Bill Pay will just return the top level database biller name.

Analogous to this, set the parameter `PreferPrintedAddressLines` to `TRUE` to return the address lines as printed on the document, instead of returning the lines as stored in the database.

## Send results to a multi-alternative, multi-subfield array fields

Kofax Mobile Bill Pay exposes the full comprehensive list of extracted addresses in the `Billers` and `Payers` fields.

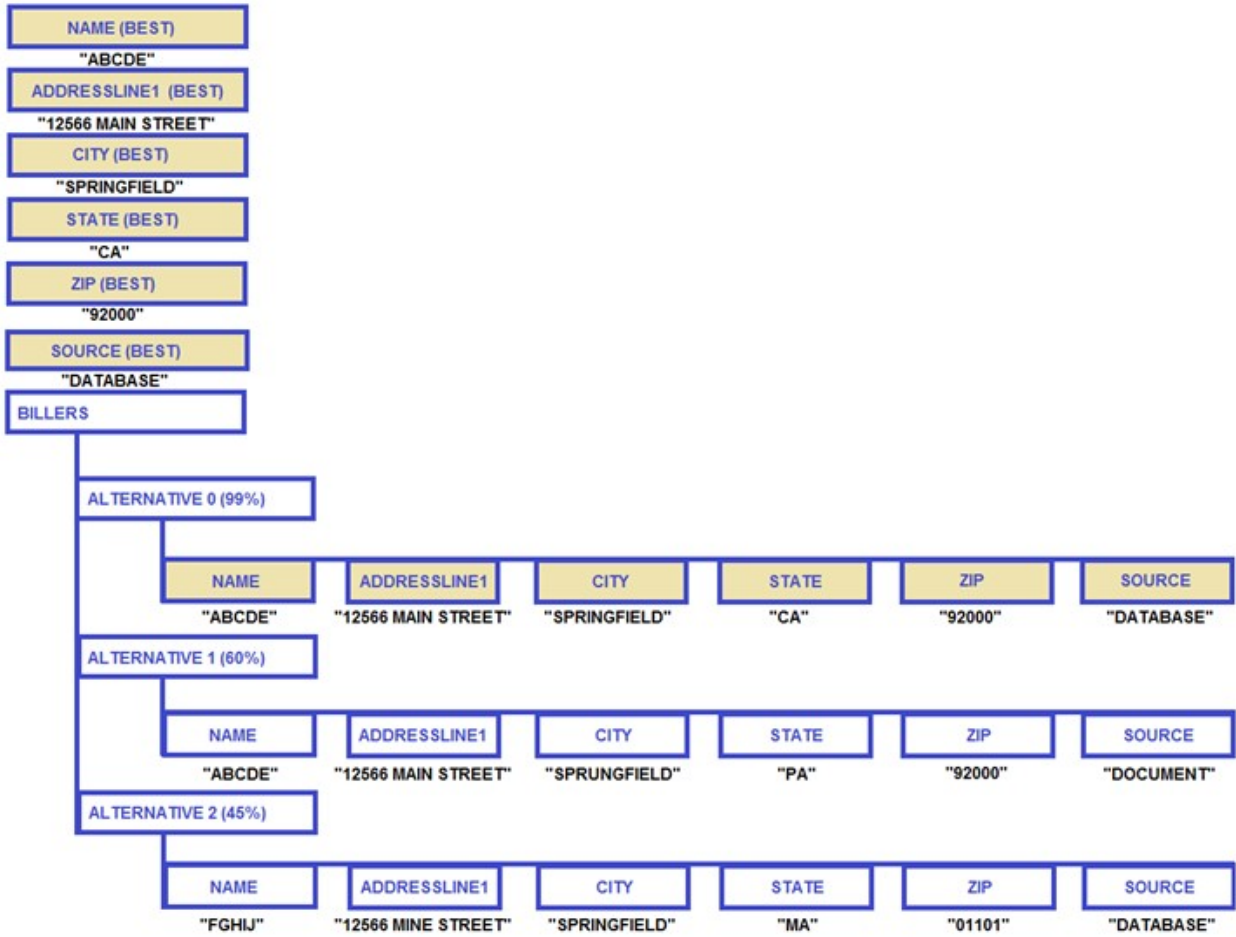
As described above, addresses are located based on rules-based analysis of the OCR ("Document"), or a fuzzy database search ("Database").

Confidence is calculated for each of these independently, and the complete address which is most confident has its component data ("subfields") copied into the top level index fields (see yellow fields on diagram below).

In addition to Kofax Mobile Bill Pay copying this best address alternative, the `Billers` and `Payers` fields have a hierarchical structure and are designed to expose all addresses evaluated during extraction. This is a data structure which can be de-serialized into an object structure on the client side.

The `Source` subfield of each `Billers` and `Payers` alternative indicates the source of the data ("Database", "Document" or more specific values). The order of address alternatives is based on the generalized confidence score.

An example of the Kofax Mobile Bill Pay return is shown below:

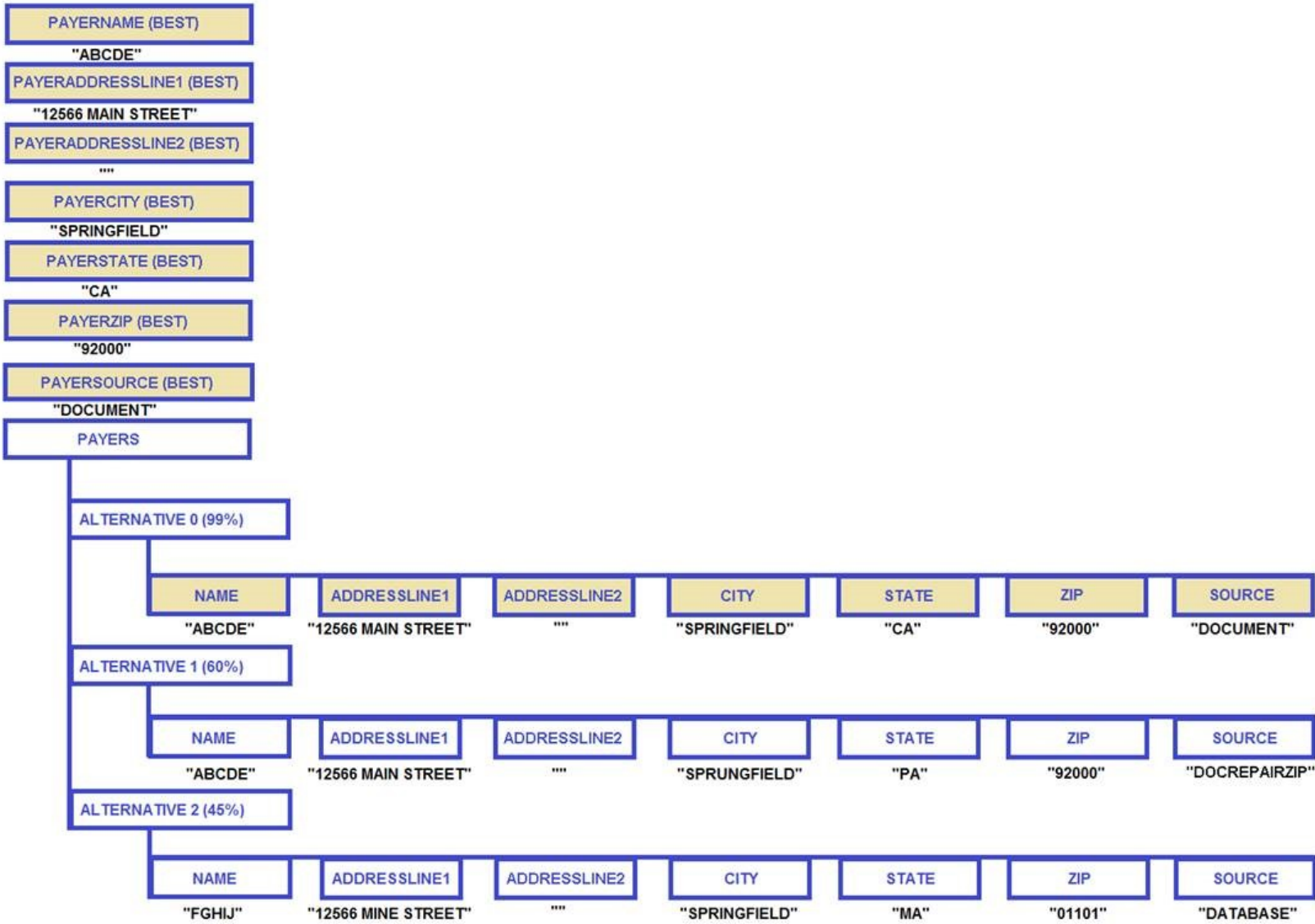


\*Only a subset of fields shown in diagram for explanatory purposes. The complete set of fields is listed in the **Extracted Fields** section.

In this example, the best alternative was from a fuzzy "Database" lookup with 99% confidence. The "Document" (rules-based) techniques located the (misspelled) City, (incorrect) State and Zip - due to various evaluation techniques this came out at a lower 60% confidence. A low confidence (45%) "Database" match is also returned - all considered matches will be returned. Unless there are many similar records in the biller database, this should not be a long list.

The Payer name and address fields have an analogous relationship:





\*Only a subset of fields shown in diagram for explanatory purposes. The complete set of fields is listed in the **Extracted Fields** section.

The multi-subfield Billers and Payers fields are created in the Document\_AfterExtract() event. Field alternatives must be requested in the call to the RTTI or KTA service in order to receive the full address arrays in the response.

# Using Kofax Mobile Bill Pay with companion products

## Kofax Real-Time Transformation Interface

By default, the server relays data extracted from the Kofax Transformation Modules project in a JSON format with a structure similar to the following sample.

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

### Example URL call

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

#### RequestURL

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

#### Request Headers

```
Content-Type: image/tiff
```

#### Request Body

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

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

#### Response

```
{
  "extractionClass": "Bill",
  "classificationResult": [{
    "class": "Bill",
    "confidence": 1.0
  }],
  "fields": [{
    "name": "ProductVersion",
    "text": "1.6.0.0.15",
    "valid": true,
```

```
"errorDescription": "",
"left": -1,
"top": -1,
"height": -1,
"width": -1,
"pageIndex": -1,
"confidence": 1.0,
"formattingFailed": false,
"fieldAlternatives": []
},
{
  "name": "Name",
  "text": "LAKE MISSION VIEJO ASSOCIATION",
  "valid": true,
  "errorDescription": "",
  "left": 227,
  "top": 1924,
  "height": 25,
  "width": 340,
  "pageIndex": 0,
  "confidence": 0.60000002384185791,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "AddressLine1",
  "text": "PO BOX 15385",
  "valid": true,
  "errorDescription": "",
  "left": 227,
  "top": 1951,
  "height": 20,
  "width": 159,
  "pageIndex": 0,
  "confidence": 0.60000002384185791,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "AddressLine2",
  "text": "",
  "valid": true,
  "errorDescription": "",
  "left": -1,
  "top": -1,
  "height": -1,
  "width": -1,
  "pageIndex": -1,
  "confidence": 0.60000002384185791,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "City",
  "text": "SANTA ANA",
  "valid": true,
  "errorDescription": "",
  "left": 226,
  "top": 1978,
  "height": 20,
  "width": 114,
  "pageIndex": 0,
  "confidence": 0.60000002384185791,
  "formattingFailed": false,
```

```
"fieldAlternatives": []
},
{
  "name": "State",
  "text": "CA",
  "valid": true,
  "errorDescription": "",
  "left": 349,
  "top": 1977,
  "height": 20,
  "width": 33,
  "pageIndex": 0,
  "confidence": 0.60000002384185791,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "Zip",
  "text": "92735-0385",
  "valid": true,
  "errorDescription": "",
  "left": 390,
  "top": 1977,
  "height": 20,
  "width": 131,
  "pageIndex": 0,
  "confidence": 0.60000002384185791,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "AccountNumber",
  "text": "I001578802501",
  "valid": true,
  "errorDescription": "",
  "left": 1049,
  "top": 1456,
  "height": 22,
  "width": 232,
  "pageIndex": 0,
  "confidence": 1.0,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "AmountDue",
  "text": "111.00",
  "valid": true,
  "errorDescription": "",
  "left": 1050,
  "top": 1525,
  "height": 20,
  "width": 79,
  "pageIndex": 0,
  "confidence": 1.0,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "MinimumAmountDue",
  "text": "",
  "valid": true,
  "errorDescription": "",
  "left": -1,
```

```
"top": -1,
"height": -1,
"width": -1,
"pageIndex": -1,
"confidence": 0.0,
"formattingFailed": false,
"fieldAlternatives": []
},
{
  "name": "DueDate",
  "text": "07/01/2013",
  "valid": true,
  "errorDescription": "",
  "left": 1121,
  "top": 134,
  "height": 21,
  "width": 105,
  "pageIndex": 0,
  "confidence": 0.9999999999999999,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "LatePaymentDate",
  "text": "",
  "valid": true,
  "errorDescription": "",
  "left": -1,
  "top": -1,
  "height": -1,
  "width": -1,
  "pageIndex": -1,
  "confidence": 0.0,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "AutoPayDate",
  "text": "",
  "valid": true,
  "errorDescription": "",
  "left": -1,
  "top": -1,
  "height": -1,
  "width": -1,
  "pageIndex": -1,
  "confidence": 0.0,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "PhoneNumber",
  "text": "1-949-770-1313",
  "valid": true,
  "errorDescription": "",
  "left": 223,
  "top": 633,
  "height": 26,
  "width": 148,
  "pageIndex": 0,
  "confidence": 0.4,
  "formattingFailed": false,
  "fieldAlternatives": []
},
}
```

```
{
  "name": "PayerName",
  "text": "Larry Wilson",
  "valid": true,
  "errorDescription": "",
  "left": 900,
  "top": 1851,
  "height": 21,
  "width": 146,
  "pageIndex": 0,
  "confidence": 0.34999999403953552,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "PayerAddressLine1",
  "text": "Laura Wilson",
  "valid": true,
  "errorDescription": "",
  "left": 900,
  "top": 1882,
  "height": 22,
  "width": 180,
  "pageIndex": 0,
  "confidence": 0.34999999403953552,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "PayerAddressLine2",
  "text": "123 Rodeo Dr",
  "valid": true,
  "errorDescription": "",
  "left": 900,
  "top": 1913,
  "height": 21,
  "width": 127,
  "pageIndex": 0,
  "confidence": 0.34999999403953552,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "PayerCity",
  "text": "MISSION VIEJO",
  "valid": true,
  "errorDescription": "",
  "left": 900,
  "top": 1945,
  "height": 26,
  "width": 155,
  "pageIndex": 0,
  "confidence": 0.34999999403953552,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "PayerState",
  "text": "CA",
  "valid": true,
  "errorDescription": "",
  "left": 1058,
  "top": 1942,
  "height": 27,
```

```
"width": 50,
"pageIndex": 0,
"confidence": 0.34999999403953552,
"formattingFailed": false,
"fieldAlternatives": []
},
{
  "name": "PayerZip",
  "text": "92692-5984",
  "valid": true,
  "errorDescription": "",
  "left": 1109,
  "top": 1946,
  "height": 23,
  "width": 141,
  "pageIndex": 0,
  "confidence": 0.34999999403953552,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "Source",
  "text": "Document",
  "valid": true,
  "errorDescription": "",
  "left": -1,
  "top": -1,
  "height": -1,
  "width": -1,
  "pageIndex": -1,
  "confidence": 0.60000002384185791,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "PayerSource",
  "text": "DocDBLMiss",
  "valid": true,
  "errorDescription": "",
  "left": -1,
  "top": -1,
  "height": -1,
  "width": -1,
  "pageIndex": -1,
  "confidence": 0.34999999403953552,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "Billers",
  "text": "",
  "valid": true,
  "errorDescription": "",
  "left": -1,
  "top": -1,
  "height": -1,
  "width": -1,
  "pageIndex": -1,
  "confidence": 0.0,
  "formattingFailed": false,
  "fieldAlternatives": []
},
{
  "name": "Payers",
```

```
"text": "",
"valid": true,
"errorDescription": "",
"left": -1,
"top": -1,
"height": -1,
"width": -1,
"pageIndex": -1,
"confidence": 0.0,
"formattingFailed": false,
"fieldAlternatives": []
}],
"sessionKey": "1f5e40d9-009e-4d5f-b709-49d65bb07cb9",
"environmentId": null,
"instanceId": null,
"documentId": "64e5a879-c317-421e-ad78-f4bd9dc411ba",
"words": null,
"processedImages": null,
"result": "Success",
"errorType": "None",
"errorDescription": null,
"errorCode": 0,
"autoProfileDetectResult": null,
"diagnosticResult": null,
"errorResult": {
  "result": "Success",
  "errorType": "None",
  "errorDescription": null,
  "errorCode": 0
},
"appStatsResult": {
  "sessionKey": "1f5e40d9-009e-4d5f-b709-49d65bb07cb9",
  "environmentId": null,
  "instanceId": null,
  "documentId": "64e5a879-c317-421e-ad78-f4bd9dc411ba"
}
}
```

## Kofax TotalAgility

### Example URL calls

#### Kofax TotalAgility

Here are a sample Request URL call and response for how to call Kofax TotalAgility for extraction of a bill:

#### RequestURL

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

**Note** The API: CreateJobWithDocumentsAndProgress2 is deprecated.



### 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": {}
          }
        ],
        "ReturnAllFields": true,
        "ReturnFullTextOcr": false,
        "RuntimeFields": null
      }
    ],
    "StoreFolderAndDocuments": false,
    "StartDate": null
  },
  "processIdentity": {
    "Id": null,
    "Name": "KofaxBillPaySync",
    "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": "KofaxBillPaySync",
  "Version": 0
},
"sessionId": "C640521793431F4486D4EF1586672385",
"variablesToReturn": {}
}

```

## Response

```

{
  "d": {
    "__type": "SyncJobWithDocuments:http://www.kofax.com/agility/services/sdk",
    "JobIdentity": {
      "__type": "JobIdentity:http://www.kofax.com/agility/services/sdk",
      "Id": "0454DB11B59D4E1E8371CCAD80D364D3"
    },
    "DocumentId": "878fc566-3af7-4fc6-9f53-a6c600e6379b",
    "FolderId": "8c7f72e7-148d-4056-a84f-a6c600e6379b",
    "ReturnedDocumentFields": [{
      "__type": "DocumentFields2:http://www.kofax.com/agility/services/sdk",
      "DocumentId": "878fc566-3af7-4fc6-9f53-a6c600e6379b",
      "ReturnedDocumentFields": [{
        "__type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
        "Id": "A352510AD91A4D678F7F541CDA9ABB1E",
        "TableRow": -1,
        "TableColumn": -1,
        "Value": "LAKE MISSION VIEJO ASSOCIATION",
        "Name": "Name",
        "Valid": true,
        "ErrorDescription": null,
        "ReviewForcedValid": false,
        "Verified": false,
        "Confidence": 0.6,
        "ReviewForcedValidDescription": null,
        "ExtractionConfident": false,
        "Left": 227,

```

```
"Top": 1924,
"Width": 340,
"Height": 25,
"PageIndex": 0,
"DocumentFieldAlternatives": [{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.6,
  "Text": "Lake Mission Viejo Association",
  "Height": 25,
  "Left": 227,
  "Top": 1924,
  "Width": 340
}],
"TableCells": []
},
{
  "_type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "2FA000BB87354BD0A95DACA5BB5307F1",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": "PO BOX 15385",
  "Name": "AddressLine1",
  "Valid": true,
  "ErrorDescription": null,
  "ReviewForcedValid": false,
  "Verified": false,
  "Confidence": 0.6,
  "ReviewForcedValidDescription": null,
  "ExtractionConfident": false,
  "Left": 227,
  "Top": 1951,
  "Width": 159,
  "Height": 20,
  "PageIndex": 0,
  "DocumentFieldAlternatives": [{
    "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": 0,
    "Confidence": 0.6,
    "Text": "PO Box 15385",
    "Height": 20,
    "Left": 227,
    "Top": 1951,
    "Width": 159
  ]
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.25,
  "Text": "22555 OLYMPIAD RD",
  "Height": 29,
  "Left": 146,
  "Top": 91,
  "Width": 240
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.2,
  "Text": "123 Rodeo",
  "Height": 22,
  "Left": 1122,
  "Top": 168,
  "Width": 129
}
```

```
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.05,
  "Text": "General Assessment",
  "Height": 23,
  "Left": 293,
  "Top": 300,
  "Width": 252
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0,
  "Text": "Lake Mission Viejo Association",
  "Height": 25,
  "Left": 227,
  "Top": 1924,
  "Width": 340
}],
"TableCells": []
},
{
  "__type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "E1030FE933194CB1812EFCE8E0C86673",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": null,
  "Name": "AddressLine2",
  "Valid": true,
  "ErrorDescription": null,
  "ReviewForcedValid": false,
  "Verified": false,
  "Confidence": 0.6,
  "ReviewForcedValidDescription": null,
  "ExtractionConfident": false,
  "Left": -1,
  "Top": -1,
  "Width": -1,
  "Height": -1,
  "PageIndex": -1,
  "DocumentFieldAlternatives": [],
  "TableCells": []
},
{
  "__type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "A4378366E9C147709553D335478CBFF8",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": "SANTA ANA",
  "Name": "City",
  "Valid": true,
  "ErrorDescription": null,
  "ReviewForcedValid": false,
  "Verified": false,
  "Confidence": 0.6,
  "ReviewForcedValidDescription": null,
  "ExtractionConfident": false,
  "Left": 226,
  "Top": 1978,
  "Width": 114,
  "Height": 20,
  "PageIndex": 0,
}
```

```
"DocumentFieldAlternatives": [{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.6,
  "Text": "SANTA ANA",
  "Height": 20,
  "Left": 226,
  "Top": 1978,
  "Width": 114
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.25,
  "Text": "MISSION VIEJO",
  "Height": 29,
  "Left": 146,
  "Top": 120,
  "Width": 160
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.2,
  "Text": "MISSION VIEJO",
  "Height": 26,
  "Left": 1122,
  "Top": 198,
  "Width": 158
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.05,
  "Text": "Payment: Check",
  "Height": 27,
  "Left": 293,
  "Top": 329,
  "Width": 199
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0,
  "Text": "Box",
  "Height": 19,
  "Left": 269,
  "Top": 1952,
  "Width": 0
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0,
  "Text": "Mission Viejo",
  "Height": 26,
  "Left": 326,
  "Top": 1722,
  "Width": 157
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0,
```

```
"Text": "Association",
"Height": 22,
"Left": 381,
"Top": 65,
"Width": 0
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0,
  "Text": "$0.36",
  "Height": 22,
  "Left": 1210,
  "Top": 1752,
  "Width": 0
}],
"TableCells": []
},
{
  "__type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "C02B3367FEF54DC6AB9EF323718F5636",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": "CA",
  "Name": "State",
  "Valid": true,
  "ErrorDescription": null,
  "ReviewForcedValid": false,
  "Verified": false,
  "Confidence": 0.6,
  "ReviewForcedValidDescription": null,
  "ExtractionConfident": false,
  "Left": 349,
  "Top": 1977,
  "Width": 33,
  "Height": 20,
  "PageIndex": 0,
  "DocumentFieldAlternatives": [{
    "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": 0,
    "Confidence": 0.6,
    "Text": "CA",
    "Height": 20,
    "Left": 349,
    "Top": 1977,
    "Width": 33
  }],
  {
    "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": 0,
    "Confidence": 0.25,
    "Text": "CA",
    "Height": 30,
    "Left": 308,
    "Top": 117,
    "Width": 52
  },
  {
    "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": 0,
    "Confidence": 0.2,
    "Text": "CA",
    "Height": 28,
    "Left": 1282,
```

```
"Top": 193,
"Width": 52
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.05,
  "Text": "No",
  "Height": 28,
  "Left": 494,
  "Top": 327,
  "Width": 48
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0,
  "Text": "Box",
  "Height": 19,
  "Left": 269,
  "Top": 1952,
  "Width": 41
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0,
  "Text": "CA",
  "Height": 28,
  "Left": 486,
  "Top": 1717,
  "Width": 51
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0,
  "Text": "Association",
  "Height": 22,
  "Left": 381,
  "Top": 65,
  "Width": 139
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0,
  "Text": "$0.36",
  "Height": 22,
  "Left": 1210,
  "Top": 1752,
  "Width": 76
}
}],
"TableCells": []
},
{
  "_type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "B6887195DB00427FB8A51D4C36123D28",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": "92735-0385",
  "Name": "Zip",
  "Valid": true,
  "ErrorDescription": null,
}
```

```
"ReviewForcedValid": false,
"Verified": false,
"Confidence": 0.6,
"ReviewForcedValidDescription": null,
"ExtractionConfident": false,
"Left": 390,
"Top": 1977,
"Width": 131,
"Height": 20,
"PageIndex": 0,
"DocumentFieldAlternatives": [{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.6,
  "Text": "92735-0385",
  "Height": 20,
  "Left": 390,
  "Top": 1977,
  "Width": 131
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.25,
  "Text": "92692",
  "Height": 21,
  "Left": 361,
  "Top": 123,
  "Width": 73
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.2,
  "Text": "92692",
  "Height": 21,
  "Left": 1335,
  "Top": 197,
  "Width": 75
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.05,
  "Text": "000915",
  "Height": 21,
  "Left": 543,
  "Top": 331,
  "Width": 90
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0,
  "Text": "15385",
  "Height": 20,
  "Left": 320,
  "Top": 1951,
  "Width": 66
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0,
```



```
"Text": "92692",
"Height": 21,
"Left": 538,
"Top": 1721,
"Width": 74
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0,
  "Text": "22555 019^805 ^45",
  "Height": 29,
  "Left": 146,
  "Top": 91,
  "Width": 240
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0,
  "Text": "^92614 61312013",
  "Height": 14,
  "Left": 897,
  "Top": 1787,
  "Width": 157
}],
"TableCells": []
},
{
  "__type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "78D17A2D49D4459BB6B93C3BE11991D0",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": "L1001578802501",
  "Name": "AccountNumber",
  "Valid": true,
  "ErrorDescription": null,
  "ReviewForcedValid": false,
  "Verified": false,
  "Confidence": 1,
  "ReviewForcedValidDescription": null,
  "ExtractionConfident": false,
  "Left": 1122,
  "Top": 63,
  "Width": 235,
  "Height": 22,
  "PageIndex": 0,
  "DocumentFieldAlternatives": [{
    "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": 0,
    "Confidence": 1,
    "Text": "L1001578802501",
    "Height": 22,
    "Left": 1122,
    "Top": 63,
    "Width": 235
  }],
  {
    "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": 0,
    "Confidence": 1,
    "Text": "I001578802501",
    "Height": 22,
    "Left": 1049,
```

```
"Top": 1456,
"Width": 232
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.285714,
  "Text": "000915",
  "Height": 21,
  "Left": 543,
  "Top": 331,
  "Width": 90
}],
"TableCells": []
},
{
  "_type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "80DD5136E64B453BA37DFE80DECD9CC0",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": "111.00",
  "Name": "AmountDue",
  "Valid": true,
  "ErrorDescription": null,
  "ReviewForcedValid": false,
  "Verified": false,
  "Confidence": 1,
  "ReviewForcedValidDescription": null,
  "ExtractionConfident": false,
  "Left": 1050,
  "Top": 1525,
  "Width": 79,
  "Height": 20,
  "PageIndex": 0,
  "DocumentFieldAlternatives": [{
    "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": 0,
    "Confidence": 1,
    "Text": "111.00",
    "Height": 20,
    "Left": 1050,
    "Top": 1525,
    "Width": 79
  }],
  {
    "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": 0,
    "Confidence": 0.621869,
    "Text": "5.00",
    "Height": 22,
    "Left": 543,
    "Top": 1087,
    "Width": 53
  }],
  "TableCells": []
},
{
  "_type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "5F8026E8AFC745F9BE3B5C209947C942",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": "07/01/2013",
  "Name": "DueDate",
  "Valid": true,
```

```
"ErrorDescription": null,
"ReviewForcedValid": false,
"Verified": false,
"Confidence": 1,
"ReviewForcedValidDescription": null,
"ExtractionConfident": false,
"Left": 1121,
"Top": 134,
"Width": 105,
"Height": 21,
"PageIndex": 0,
"DocumentFieldAlternatives": [{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 1,
  "Text": "07/01/2013",
  "Height": 21,
  "Left": 1121,
  "Top": 134,
  "Width": 105
}],
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.675676,
  "Text": "05/28/2013",
  "Height": 21,
  "Left": 1048,
  "Top": 1490,
  "Width": 104
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.636398,
  "Text": "08/01/2013",
  "Height": 20,
  "Left": 356,
  "Top": 1088,
  "Width": 82
},
{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.432432,
  "Text": "12/31/2013",
  "Height": 21,
  "Left": 326,
  "Top": 716,
  "Width": 81
}],
"TableCells": []
},
{
  "_type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "D73DE3DB04E141D0A87AA5C4D5631B80",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": "1-949-770-1313",
  "Name": "PhoneNumber",
  "Valid": true,
  "ErrorDescription": null,
  "ReviewForcedValid": false,
  "Verified": false,
```

```
"Confidence": 0.4,
"ReviewForcedValidDescription": null,
"ExtractionConfident": false,
"Left": 223,
"Top": 633,
"Width": 148,
"Height": 26,
"PageIndex": 0,
"DocumentFieldAlternatives": [{
  "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.4,
  "Text": "1-949-770-1313",
  "Height": 26,
  "Left": 223,
  "Top": 633,
  "Width": 148
}],
"TableCells": []
},
{
  "_type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "094C37DC170D4DFA9FFA8D9D15FCD2A3",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": "Document",
  "Name": "Source",
  "Valid": true,
  "ErrorDescription": null,
  "ReviewForcedValid": false,
  "Verified": false,
  "Confidence": 0.6,
  "ReviewForcedValidDescription": null,
  "ExtractionConfident": false,
  "Left": -1,
  "Top": -1,
  "Width": -1,
  "Height": -1,
  "PageIndex": -1,
  "DocumentFieldAlternatives": [{
    "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": -1,
    "Confidence": 0.6,
    "Text": "Document",
    "Height": -1,
    "Left": -1,
    "Top": -1,
    "Width": -1
  }],
  {
    "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": -1,
    "Confidence": 0.25,
    "Text": "DocDBLMiss",
    "Height": -1,
    "Left": -1,
    "Top": -1,
    "Width": -1
  },
  {
    "_type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": -1,
    "Confidence": 0.2,
    "Text": "DocDBLMiss",
```

```
"Height": -1,
"Left": -1,
"Top": -1,
"Width": -1
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": -1,
  "Confidence": 0.05,
  "Text": "DocRepairState",
  "Height": -1,
  "Left": -1,
  "Top": -1,
  "Width": -1
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": -1,
  "Confidence": 0,
  "Text": "DocRepairState",
  "Height": -1,
  "Left": -1,
  "Top": -1,
  "Width": -1
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": -1,
  "Confidence": 0,
  "Text": "DocDBLMiss",
  "Height": -1,
  "Left": -1,
  "Top": -1,
  "Width": -1
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": -1,
  "Confidence": 0,
  "Text": "DocRepairState",
  "Height": -1,
  "Left": -1,
  "Top": -1,
  "Width": -1
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": -1,
  "Confidence": 0,
  "Text": "DocRepairState",
  "Height": -1,
  "Left": -1,
  "Top": -1,
  "Width": -1
}
}],
"TableCells": []
},
{
  "__type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "2D298B9A75BB4A608EEA0828EA503406",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": null,
  "Name": "Billers",
```

```
"Valid": true,
"ErrorDescription": null,
"ReviewForcedValid": false,
"Verified": false,
"Confidence": 0,
"ReviewForcedValidDescription": null,
"ExtractionConfident": false,
"Left": -1,
"Top": -1,
"Width": -1,
"Height": -1,
"PageIndex": -1,
"DocumentFieldAlternatives": [{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.6,
  "Text": "Document",
  "Height": 47,
  "Left": 226,
  "Top": 1951,
  "Width": 295
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.25,
  "Text": "DocDBLMiss",
  "Height": 58,
  "Left": 146,
  "Top": 91,
  "Width": 288
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": -1,
  "Confidence": 0.2,
  "Text": "DocDBLMiss",
  "Height": 56,
  "Left": 1122,
  "Top": 168,
  "Width": 288
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.05,
  "Text": "DocRepairState",
  "Height": 56,
  "Left": 293,
  "Top": 300,
  "Width": 340
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": -1,
  "Confidence": 0,
  "Text": "DocRepairState",
  "Height": 47,
  "Left": 227,
  "Top": 1924,
  "Width": 340
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
```

```
"PageIndex": -1,
"Confidence": 0,
"Text": "DocDBLMiss",
"Height": -1,
"Left": -1,
"Top": -1,
"Width": -1
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": -1,
  "Confidence": 0,
  "Text": "DocRepairState",
  "Height": -1,
  "Left": -1,
  "Top": -1,
  "Width": -1
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": -1,
  "Confidence": 0,
  "Text": "DocRepairState",
  "Height": -1,
  "Left": -1,
  "Top": -1,
  "Width": -1
}],
"TableCells": []
},
{
  "__type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "8C102231291811E6AC79005056C00008",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": "1.6.0.0.0.14",
  "Name": "ProductVersion",
  "Valid": true,
  "ErrorDescription": null,
  "ReviewForcedValid": false,
  "Verified": false,
  "Confidence": 1,
  "ReviewForcedValidDescription": null,
  "ExtractionConfident": false,
  "Left": -1,
  "Top": -1,
  "Width": -1,
  "Height": -1,
  "PageIndex": -1,
  "DocumentFieldAlternatives": [{
    "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": -1,
    "Confidence": 1,
    "Text": "1.6.0.0.0.14",
    "Height": -1,
    "Left": -1,
    "Top": -1,
    "Width": -1
  }
  ],
  "TableCells": []
},
{
  "__type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "4C9B5CD5186B11E58FA9005056C00008",
```

```
"TableRow": -1,
"TableColumn": -1,
"Value": null,
"Name": "MinimumAmountDue",
"Valid": true,
"ErrorDescription": null,
"ReviewForcedValid": false,
"Verified": false,
"Confidence": 0,
"ReviewForcedValidDescription": null,
"ExtractionConfident": false,
"Left": -1,
"Top": -1,
"Width": -1,
"Height": -1,
"PageIndex": -1,
"DocumentFieldAlternatives": [],
"TableCells": []
},
{
  "_type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "6B95D47A72EF11E5AC46005056C00008",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": null,
  "Name": "LatePaymentDate",
  "Valid": true,
  "ErrorDescription": null,
  "ReviewForcedValid": false,
  "Verified": false,
  "Confidence": 0,
  "ReviewForcedValidDescription": null,
  "ExtractionConfident": false,
  "Left": -1,
  "Top": -1,
  "Width": -1,
  "Height": -1,
  "PageIndex": -1,
  "DocumentFieldAlternatives": [],
  "TableCells": []
},
{
  "_type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "7B6E7225783611E5AC46005056C00008",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": null,
  "Name": "AutoPayDate",
  "Valid": true,
  "ErrorDescription": null,
  "ReviewForcedValid": false,
  "Verified": false,
  "Confidence": 0,
  "ReviewForcedValidDescription": null,
  "ExtractionConfident": false,
  "Left": -1,
  "Top": -1,
  "Width": -1,
  "Height": -1,
  "PageIndex": -1,
  "DocumentFieldAlternatives": [],
  "TableCells": []
},
{
```



```
"__type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
"Id": "07B4BF258B4611E6901A005056C00008",
"TableRow": -1,
"TableColumn": -1,
"Value": "Larry Wilson",
"Name": "PayerName",
"Valid": true,
"ErrorDescription": null,
"ReviewForcedValid": false,
"Verified": false,
"Confidence": 0.35,
"ReviewForcedValidDescription": null,
"ExtractionConfident": false,
"Left": 900,
"Top": 1851,
"Width": 146,
"Height": 21,
"PageIndex": 0,
"DocumentFieldAlternatives": [{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.35,
  "Text": "Larry Wilson",
  "Height": 21,
  "Left": 900,
  "Top": 1851,
  "Width": 146
}],
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.331506,
  "Text": "Lake Mission Viejo Association",
  "Height": 24,
  "Left": 131,
  "Top": 1513,
  "Width": 341
}],
"TableCells": []
},
{
  "__type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "17DFF4658B4611E6901A005056C00008",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": "Laura Wilson",
  "Name": "PayerAddressLine1",
  "Valid": true,
  "ErrorDescription": null,
  "ReviewForcedValid": false,
  "Verified": false,
  "Confidence": 0.35,
  "ReviewForcedValidDescription": null,
  "ExtractionConfident": false,
  "Left": 900,
  "Top": 1882,
  "Width": 180,
  "Height": 22,
  "PageIndex": 0,
  "DocumentFieldAlternatives": [{
    "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": 0,
    "Confidence": 0.35,
    "Text": "Laura Wilson",
```

```
"Height": 22,
"Left": 900,
"Top": 1882,
"Width": 180
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.331506,
  "Text": "22555 OLYMPIAD RD",
  "Height": 25,
  "Left": 131,
  "Top": 1539,
  "Width": 219
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.050632,
  "Text": "Lake Mission Viejo Association",
  "Height": 24,
  "Left": 131,
  "Top": 1513,
  "Width": 341
}],
"TableCells": []
},
{
  "__type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "240F51248B4611E6901A005056C00008",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": "123 Rodeo",
  "Name": "PayerAddressLine2",
  "Valid": true,
  "ErrorDescription": null,
  "ReviewForcedValid": false,
  "Verified": false,
  "Confidence": 0.35,
  "ReviewForcedValidDescription": null,
  "ExtractionConfident": false,
  "Left": 900,
  "Top": 1913,
  "Width": 127,
  "Height": 21,
  "PageIndex": 0,
  "DocumentFieldAlternatives": [{
    "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": 0,
    "Confidence": 0.35,
    "Text": "123 Rodeo",
    "Height": 21,
    "Left": 900,
    "Top": 1913,
    "Width": 127
  }],
  "TableCells": []
},
{
  "__type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "305F2E888B4611E6901A005056C00008",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": "MISSION VIEJO",
```

```
"Name": "PayerCity",
"Valid": true,
"ErrorDescription": null,
"ReviewForcedValid": false,
"Verified": false,
"Confidence": 0.35,
"ReviewForcedValidDescription": null,
"ExtractionConfident": false,
"Left": 900,
"Top": 1945,
"Width": 155,
"Height": 21,
"PageIndex": 0,
"DocumentFieldAlternatives": [{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.35,
  "Text": "MISSION VIEJO",
  "Height": 21,
  "Left": 900,
  "Top": 1945,
  "Width": 155
}],
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.331506,
  "Text": "MISSION VIEJO",
  "Height": 24,
  "Left": 132,
  "Top": 1565,
  "Width": 145
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.050632,
  "Text": "22555 Olympiad",
  "Height": 24,
  "Left": 131,
  "Top": 1540,
  "Width": 180
}],
"TableCells": []
},
{
  "__type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "3EFBA3258B4611E6901A005056C00008",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": "CA",
  "Name": "PayerState",
  "Valid": true,
  "ErrorDescription": null,
  "ReviewForcedValid": false,
  "Verified": false,
  "Confidence": 0.35,
  "ReviewForcedValidDescription": null,
  "ExtractionConfident": false,
  "Left": 1065,
  "Top": 1946,
  "Width": 36,
  "Height": 20,
  "PageIndex": 0,
```

```
"DocumentFieldAlternatives": [{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.35,
  "Text": "CA",
  "Height": 20,
  "Left": 1065,
  "Top": 1946,
  "Width": 36
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.331506,
  "Text": "CA",
  "Height": 26,
  "Left": 280,
  "Top": 1561,
  "Width": 46
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.050632,
  "Text": "Rd",
  "Height": 26,
  "Left": 314,
  "Top": 1536,
  "Width": 43
}],
"TableCells": []
},
{
  "__type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "4C3327678B4611E6901A005056C00008",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": "92692-5984",
  "Name": "PayerZip",
  "Valid": true,
  "ErrorDescription": null,
  "ReviewForcedValid": false,
  "Verified": false,
  "Confidence": 0.35,
  "ReviewForcedValidDescription": null,
  "ExtractionConfident": false,
  "Left": 1109,
  "Top": 1946,
  "Width": 141,
  "Height": 23,
  "PageIndex": 0,
  "DocumentFieldAlternatives": [{
    "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": 0,
    "Confidence": 0.35,
    "Text": "92692-5984",
    "Height": 23,
    "Left": 1109,
    "Top": 1946,
    "Width": 141
  },
  {
    "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": 0,
```

```
"Confidence": 0.331506,
"Text": "92692",
"Height": 19,
"Left": 327,
"Top": 1564,
"Width": 67
},
{
  "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
  "PageIndex": 0,
  "Confidence": 0.050632,
  "Text": "22555 04^1012121 811",
  "Height": 25,
  "Left": 131,
  "Top": 1539,
  "Width": 219
}],
"TableCells": []
},
{
  "__type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "5A82B29E8B4611E6901A005056C00008",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": "Document",
  "Name": "PayerSource",
  "Valid": true,
  "ErrorDescription": null,
  "ReviewForcedValid": false,
  "Verified": false,
  "Confidence": 0.35,
  "ReviewForcedValidDescription": null,
  "ExtractionConfident": false,
  "Left": -1,
  "Top": -1,
  "Width": -1,
  "Height": -1,
  "PageIndex": -1,
  "DocumentFieldAlternatives": [{
    "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": -1,
    "Confidence": 0.35,
    "Text": "Document",
    "Height": -1,
    "Left": -1,
    "Top": -1,
    "Width": -1
  }],
  {
    "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": -1,
    "Confidence": 0.331506,
    "Text": "DocDBLMiss",
    "Height": -1,
    "Left": -1,
    "Top": -1,
    "Width": -1
  },
  {
    "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": -1,
    "Confidence": 0.050632,
    "Text": "DocRepairState",
    "Height": -1,
```

```
"Left": -1,
"Top": -1,
"Width": -1
}],
"TableCells": []
},
{
  "__type": "RuntimeField3:http://www.kofax.com/agility/services/sdk",
  "Id": "67CCD5108B4611E6901A005056C00008",
  "TableRow": -1,
  "TableColumn": -1,
  "Value": null,
  "Name": "Payers",
  "Valid": true,
  "ErrorDescription": null,
  "ReviewForcedValid": false,
  "Verified": false,
  "Confidence": 0,
  "ReviewForcedValidDescription": null,
  "ExtractionConfident": false,
  "Left": -1,
  "Top": -1,
  "Width": -1,
  "Height": -1,
  "PageIndex": -1,
  "DocumentFieldAlternatives": [{
    "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": 0,
    "Confidence": 0.35,
    "Text": "Document",
    "Height": 87,
    "Left": 900,
    "Top": 1882,
    "Width": 350
  }],
  {
    "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": 0,
    "Confidence": 0.331506,
    "Text": "DocDBLMiss",
    "Height": 50,
    "Left": 131,
    "Top": 1539,
    "Width": 263
  },
  {
    "__type": "FieldAlternative2:http://www.kofax.com/agility/services/sdk",
    "PageIndex": -1,
    "Confidence": 0.050632,
    "Text": "DocRepairState",
    "Height": 51,
    "Left": 131,
    "Top": 1513,
    "Width": 341
  }
}],
"TableCells": []
}],
"PageTextOcrs": []
}],
"ReturnedVariables": []
}
}
```

# Maintaining Kofax Mobile Bill Pay

## Performing database updates

Periodically, Kofax will create a fix pack to update the KSMS Biller Directory databases. Use the KSMS Administration interface to deploy the updated databases.

## Upgrading Kofax Mobile Bill Pay for Real-Time Transformation Interface

1. Backup your existing Kofax Mobile Bill Pay Kofax Transformation Modules (KTM) project file.
2. Extract the file **KofaxMobileBillPay-x.x.x.ZIP** where x.x.x is the version to which you are upgrading.
3. Copy the entire contents of the extracted project folder to a shared drive where it is accessible to all your Real-Time Transformation Interface servers as well as Kofax Transformation Modules and Kofax Capture.
4. Verify that the provided database are on a server with Kofax Search and Matching Server (KSMS) administration components installed.
5. Import the provided databases into KSMS. See the KSMS documentation for instructions.
6. If your Real-Time Transformation Interface server is not installed on a server where KSMS is installed, use the Kofax Transformation Modules Project Builder to edit the KTM project to point to the correct KSMS server location.
7. Add the KTM project to the Real-Time Transformation Interface configuration. See the *Real-Time Transformation Interface Administrator's Guide*.
8. If you customized your earlier Kofax Mobile Bill Pay KTM project, make the same modifications to the new Kofax Mobile Bill Pay project.

## Uninstalling Kofax Mobile Bill Pay

### Uninstalling Kofax Mobile Bill Pay for Real-Time Transformation Interface

To uninstall, delete the Kofax Mobile Bill Pay project, and the provided databases.

## Uninstalling Kofax Mobile Bill Pay for TotalAgility

Use the following steps to uninstall TotalAgility:

1. Remove the classification group.
2. Remove processes.
3. Remove the package form Kofax Mobile Bill Pay
4. Delete the package from TotalAgility Designer. (See the TotalAgility documentation. )