ABBYY®

ABBYY Mobile Capture SDK

Developer's Guide

Table of Contents

Introducing ABBYY Mobile Capture

Welcome to ABBYY Mobile Capture.

ABBYY Mobile Capture is a software development kit that provides flexible methods of mobile data capture. The Mobile Capture SDK will automatically capture the image for further back-end processing or recognize the data from the document in real-time on the mobile device, requiring minimal interaction from the user.

Key features:

The ABBYY Mobile Capture SDK can power your applications with:

Out-of-the-box image capture: Easily add image capture with UI components by utilizing our API, to automatically capture the best quality image suitable for OCR for further back-end processing.

Automatic document detection: Detects document boundaries, crops and corrects perspective.

On device OCR: Automatically recognizes text from a static image or on the smartphones' camera preview screen from video stream by simply pointing the camera on the document or object.

Customizable data capture: Extract any specific data from a document by setting a regular expression describing the required content. Capture machine-readable zones (MRZ) or international bank account numbers (IBAN) by simply applying predefined profiles.

Out-of-the-box document capture: Easily add ready-made functionality to extract important fields from specific documents: passports, IDs, driver licenses, bank cards and others.

Ready-to-use business card reading: Allows automatic and convenient extraction of contact data from business cards by simply pointing the camera at the card to use within your mobile CRM or lead management app or for customer onboarding.

Out-of-the-box image capture scenario: Implement image capture by adding just a few lines of code to your app, using API that can draw UI, handle phone camera and perform image capture.

Translation: Provides built-in translation dictionaries; word-by-word and phrase-by-phrase.

Benefits:

- *Increase your customer retention rates:* Provide your customers with a seamless customer experience with a friendly mobile onboarding solution, meeting customers in their preferred channel with accurate results and minimal steps for the end user.
- *Get ahead of the competition:* Provide a better customer experience by minimizing the efforts by the end user to capture and deliver data within the onboarding experience with seamless accurate back-end integration to process the required information.
- *Optimize your development resources:* Easily integrate a pre-built comprehensive mobile capture solution into your mobile application.

Guided Tour

This section will help you to get started using ABBYY Mobile Capture.

- How to Add the Library to Your Xcode Project
- Implementation of image capture scenario, using the User Interface API
- Step-by-step guides to the simple scenarios:
  - How to Capture Image from Camera
  - How to Capture Text from Camera
  - How to Recognize Text on Photos
  - How to Capture Data from Documents
  - How to Capture a Custom Data Field
- Code Samples

How to Add the Library to Your Xcode Project

To create an application which uses ABBYY Mobile Capture SDK, you will need to add the library to your project, copy resource files and sign the framework. The configuring can be done mostly manually or by scripts. This is required for new projects only — packaged examples work out of the box.

Building application in manual mode

1. Add the frameworks from the **libs** folder to your project. Please keep strictly to the following order:
   1) AbbyyRtrSDK.framework
   2) AbbyyZlib.framework
   3) CustomAllocator.framework
   4) FineMachineLearning.framework
   5) FineObj.framework
   6) Image.Mobile.framework
   7) MobileImaging.framework
2. Add the **license** file to your project (simply drag and drop it into your project window).
3. Select your project in the **Target** group and open the **Build Phases** tab. In the **Link Binary With Libraries** section, click "+" and add the **AbbyyRtrSDK.framework**.
4. Open the **General** tab and add all the frameworks to the section **Embedded Binaries**.
5. Now you need to add the resource files and set up the copying rules. See Distribution Kit for a detailed description of the necessary resources. To add the resource files do the following:
   1) Go to **Build Phases** and add a new **Copy Files** phase.
   2) In the **Destination** field, specify **Resources**.
   3) In the **Subpath** field, specify **bcr**, **dictionaries**, **patterns**, **translation** and other required resource files.
6. Finally, add framework signing:
   1) In **Build Phases**, add a new **Run Script** phase.
   2) Run the **copy_frameworks.sh** script that removes the frameworks for the non-active CPU architectures (the complete list depends on the project settings), and sign the resulting framework. This is a required step before uploading your application to App Store.

```
/bin/sh "${SRCROOT}/../libs/copy_frameworks.sh"
```

Building application using scripts

1. Add the **license** file to your project (simply drag and drop it into your project window).

2. Add the **AbbyyRtrSDK.framework** from the **libs** folder to your project.
3. Select your project in the **Target** group and open the **Build Phases** tab. In the **Link Binary With Libraries** section, click "+" and add the **AbbyyRtrSDK.framework**.
4. Add all the other frameworks to your project using script:
   a. In **Build Phases**, add a new **Run Script** phase.
   b. Run the **copy_frameworks.sh** script to add all the frameworks to you project. It will also remove from the frameworks the non-active CPU architectures (the complete list depends on the project settings) and sign the resulting framework. This is a required step before uploading your application to App Store.

```
/bin/sh "${SRCROOT}/../libs/copy_frameworks.sh"
```

5. Now you need to add the resource files and set up the copying rules:
   a. In **Build Phases**, add a new **Run Script** phase.
   b. Run the **copy_assets.py** script to automatically copy all resource files to corresponding destinations and add necessary dictionaries. Your scenario may require only certain assets, therefor the script provides customizable settings: keys. Set the key value to copy only specified for the scenario source files. See the script file for details.

```
python "${SRCROOT}/../assets/copy_assets.py"
```

Important! *Your application needs an Internet connection to gather the information about the current state of the library.*

How to Capture Image from Camera

This guide helps you to implement the whole image capture scenario including the user interface part with just a little effort.

The purpose of image capture scenario is to enable your application to capture the image from the smartphone camera preview frames. Once you begin capturing, the Mobile Capture engine will automatically receive new camera frames, detecting the quality assessment of the captured image to OCR and filtering out low quality photos. This process is continued until the result reaches the required stability level. Accessible image is cropped and justified. Then it can be compressed and exported to the processing server.

There are three optimal variations of the scenario, providing different instruments for image capture integration into your project:

- **Multipage image capture scenario**, including API for ready-to-use user interface implementation. This is the easiest to integrate case. Use this API to implement an onboarding document capture application, to capture large document batches with no page limit or fixed number of documents with different page size. Test such supported features as interface elements and working with photo gallery and customize them according to your needs.
- **One-page image capture scenario** with API for user interface implementation. This case is useful for scenarios, when a single page capture with basic user interface elements should be performed.
- **Image capture API** for specific cases when a custom user interface is implemented and only the processing mechanisms are required.

Below you can find detailed "How to..." for each of these scenarios, demonstrating ABBYY Mobile Capture API usage.

How to Implement Multipage Image Capture with User Interface

Note: *Before you begin, see How to Add the Library to Your Xcode Project.*

1. Add the **NSCameraUsageDescription** and **NSPhotoLibraryUsageDescription** keys into the info.plist file for requesting access to the device's camera and to the user's photo library.
2. Create an instance of the **AUICaptureController** interface for managing user interface and starting capture scenario. On this step you can use the properties of this interface to tune the user interface appearance and the settings of the camera.
3. To get access to the processing mechanisms for chosen scenario create an **RTREngine** object using the **sharedEngineWithLicenseData:** method. The method requires an **NSData** object containing your license data. For example, you can use **dataWithContentsOfFile:** to create a data object, then pass this object to the **sharedEngineWithLicenseData:** method.
4. Create an instance of an interface inherited from the **AUICaptureScenario** for managing the capture scenario. Choose the inherited interface for the multipage image capture scenario: **AUIMultiPageImageCaptureScenario** interface. Pass the previously created **RTREngine** object and a full path to a folder, that will be a page storage, as input parameters to the constructor.
5. Use the properties of the **AUIMultiPageImageCaptureScenario** interface for flexible tuning of your application. I.e., depending on your application goal you can tune the presets of the scenario user interface and image count requirements:
   - define the **requiredPageCount** property for fixing an exact number of images to be captured or allow unlimited image capture;
   - to customize interface string resources, do the following:
     - define custom string resources in the **AUIMultiPageResourceType** enumeration
     - set these resources to the user interface setting using the **captureScenario:stringForResourceType:forPageAtIndex:** method of the **AUIMultiPageUISettings** protocol
     - set the **uiSettings** property to the **AUIMultiPageUISettings**.
6. Implement the delegate interface, corresponding to the scenario object, and its methods. I.e. for **AUIMultiPageImageCaptureScenarioDelegate** interface you should implement these methods:
   a. The **captureScenario:didFinishWithResult:** method returns the **AUIMultiPageImageCaptureResult** object providing access to the result images
   b. The **captureScenario:onCloseWithResult:** method notifies that the scenario was canceled by user
   c. The **captureScenario:didFailWithError:** method delivers error messages.
7. Add the instance of the interface as a **delegate** property of the **AUIMultiPageImageCaptureScenario** object.
8. Set the created scenario object as a **captureScenario** property of the **AUICaptureController** object. Scenario will start immediately at the camera start.
9. If any error occurs while processing, the capture scenario becomes temporary passive and the **paused** property of the **AUICaptureController** object is set to **true**.
10. Process the messages sent by scenario to the delegate object. The result will be delivered as an **AUIMultiPageImageCaptureResult** object providing access to the result images.

How to Implement One-Page Image Capture with User Interface

Note: *Before you begin, see How to Add the Library to Your Xcode Project.*

1. Add the **NSCameraUsageDescription** and **NSPhotoLibraryUsageDescription** keys into the info.plist file for requesting access to the device's camera and to the user's photo library.
2. Create an instance of the **AUICaptureController** interface for managing user interface and starting capture scenario. On this step you can use the properties of this interface to tune the user interface appearance and the settings of the camera.
3. To get access to the processing mechanisms for chosen scenario create an **RTREngine** object using the **sharedEngineWithLicenseData:** method. The method requires an **NSData** object containing your license data. For example, you can use **dataWithContentsOfFile:** to create a data object, then pass this object to the **sharedEngineWithLicenseData:** method.

4. Create an instance of an interface inherited from the **AUICaptureScenario** for managing the capture scenario. Choose the inherited interface for the image capture scenario: **AUIImageCaptureScenario** interface. Pass the previously created **RTREngine** object as an input parameter to the constructor.
5. Implement the delegate interface, corresponding to the scenario object, and its methods. I.e. for **AUIImageCaptureScenarioDelegate** interface you should implement these methods:
   a. The **captureScenario:didCaptureImageWithResult:** method returns the result image as a **AUIImageCaptureResult** object
   b. The **captureScenario:didFailWithError:** method delivers error messages
   c. The **captureScenarioDidCancel:** method notifies that the scenario was canceled.
6. Add the instance of the interface as a **delegate** property of the **AUIImageCaptureScenario** object.
7. Set the created scenario object as a **captureScenario** property of the **AUICaptureController** object. Scenario will start immediately at the camera start.
8. If any error occurs while processing, the capture scenario becomes temporary passive and the **paused** property of the **AUICaptureController** object is set to **true**.
9. Process the messages sent by scenario to the delegate object. The result will be delivered as an **AUIImageCaptureResult** object, storing the captured image.

How to Implement Image Capture Processing

Note: *Before you begin, see How to Add the Library to Your Xcode Project.*

To implement the image capture scenario, follow these steps:

1. Implement a delegate conforming to the **RTRImageCaptureServiceDelegate** protocol. The delegate will handle messages from the image capture service. Here are the recommendations on what its methods should do:
   - The **onBufferProcessedWithStatus:result:** method is where you work with the results, display them to the user, etc.
   - The **onError:** method is for handling processing errors.
   - The **onWarning:** method can optionally be used to show warnings to the user.
2. Create an **RTREngine** object using the **sharedEngineWithLicenseData:** method. The method requires an **NSData** object containing your license data. For example, you can use **dataWithContentsOfFile:** to create a data object, then pass this object to the **sharedEngineWithLicenseData:** method.
3. Use the **createImageCaptureServiceWithDelegate:** method of the **RTREngine** object to create a background image capture service. Only one instance of the service per application is necessary: multiple threads will be started internally.
4. Implement a delegate that adopts the **AVCaptureVideoDataOutputSampleBufferDelegate** protocol. Instantiate an **AVCaptureSession** object, add video input and output and set the video output delegate. When the delegate receives a video frame via the **captureOutput:didOutputSampleBuffer:fromConnection:** method, pass this frame on to the image capture service by calling the **addSampleBuffer:** method.
   We recommend using the `AVCaptureSessionPreset1920x1080` preset for your **AVCaptureSession**.
   Also note that your video output must be configured to use the `kCVPixelFormatType_32BGRA` video pixel format.
5. Process the messages sent by the service to the **RTRImageCaptureServiceDelegate** delegate object. The result will be delivered via the **onBufferProcessedWithStatus:result:** method. It also reports the result stability status, which indicates if the result is available and if it is likely to be improved by adding further frames (see the *Status* parameter).
6. Process the delivered result using the **RTRImagingCoreAPI** functionality:
   - load the captured image to an **RTRCoreAPIImage** protocol instance;

- crop it with the **RTRCoreAPICropOperation** in case the lens correction is required.

7. Export the processed result to the server in one of available formats: JPG, JPG 2000, PNG or PDF. It is recommended to use the *RTRCoreAPIExportCompressionNormalLevel* compression mode for saving high quality while reducing image size.
8. When pausing or quitting the application, call the **stopTasks** method to stop processing and clean up image buffers. The image capture service keeps its configuration settings and necessary resources. The processing will start automatically on the new call to the **addSampleBuffer:** method.

See the description of classes and methods in the API Reference section.

How to Capture Text from Camera with iOS

This guide walks you through a simple real-time text capture scenario, in which the user points the device's camera at the text to be recognized.

How it Works

The purpose of Mobile Capture SDK for OCR development is to enable your application to capture information directly from the smartphone camera preview frames, without actually snapping a picture. Once you start capturing, the Mobile Capture SDK engine will automatically receive new camera frames and process them, using each new frame to verify and improve the recognition result from the previous frame. This process is continued until the result reaches the required stability level. Combining several images enables Mobile Capture SDK to recognize text even in situation when it is hard to obtain a still photo of suitable quality for recognition.

Note that Mobile Capture SDK also supports recognizing text on an image that was already saved to a file, which allows it to process existing photos, scanned texts, and so on. See How to Recognize Text on Photos for the description of this scenario.

Implementation

Note: *Before you begin, see How to Add the Library to Your Xcode Project.*

To implement the real-time text capture scenario, follow these steps:

1. Implement a delegate conforming to the **RTRTextCaptureServiceDelegate** protocol. The delegate will handle messages from the text capture service. Here are the recommendations on what its methods should do:
   - The **onBufferProcessedWithTextLines:resultStatus:** method is where you work with the results, display them to the user, etc.
   - The **onError:** method is for handling processing errors.
   - The **onWarning:** method can optionally be used to show warnings to the user.
2. Create an **RTREngine** object using the **sharedEngineWithLicenseData:** method. The method requires an NSData object containing your license data. For example, you can use dataWithContentsOfFile: to create a data object, then pass this object to the **sharedEngineWithLicenseData:** method.
3. Use the **createTextCaptureServiceWithDelegate:** method of the **RTREngine** object to create a background text capture service. Only one instance of the service per application is necessary: multiple threads will be started internally.
4. Configure the text capture service:

- If you are using a recognition language different from English, specify it using the **setRecognitionLanguages:** method. Multiple languages are also supported, although setting too many languages may decrease recognition performance.
- Your application can automatically translate the recognized text. To enable translation, add a dictionary using the **setTranslationDictionary:** method.
  Note that when a dictionary is set, recognition results are returned in the target language, and text in the source language is no longer available.
- It is also recommended to call the **setAreaOfInterest:** method to specify the rectangular area on the frame where the text is likely to be found. For example, your application may show a highlighted rectangle in the UI into which the end users will try to fit the text they are capturing. The best result is achieved when the area of interest does not touch the boundaries of the frame but has a margin of at least half the size of a typical printed character.

5. Implement a delegate that adopts the AVCaptureVideoDataOutputSampleBufferDelegate protocol. Instantiate an AVCaptureSession object, add video input and output and set the video output delegate. When the delegate receives a video frame via the captureOutput:didOutputSampleBuffer:fromConnection: method, pass this frame on to the text capture service by calling the **addSampleBuffer:** method.
   We recommend using the `AVCaptureSessionPreset1280x720` preset for your AVCaptureSession. Also note that your video output must be configured to use the `kCVPixelFormatType_32BGRA` video pixel format.

6. Process the messages sent by the service to the **RTRTextCaptureServiceDelegate** delegate object. The result will be delivered via the **onBufferProcessedWithTextLines:resultStatus:** method. It also reports the result stability status, which indicates if the result is available and if it is likely to be improved by adding further frames (see the *resultStatus* parameter). Use it to determine whether your application should stop processing and display the result to the user. We do not recommend using the result until the stability level has reached at least RTRResultStabilityAvailable.
   The result consists of one or more text lines represented by objects of the **RTRTextLine** class. Each **RTRTextLine** contains information about the bounding quadrangle of a single line of text, and the recognized text as a string.
   Work with the results on your side.

7. When pausing or quitting the application, call the **stopTasks** method to stop processing and clean up image buffers. The text capture service keeps its configuration settings (language, area of interest) and necessary resources. The processing will start automatically on the new call to the **addSampleBuffer:** method.

See the description of classes and methods in the API Reference section.

How to Recognize Text on Photos

This guide explains how Mobile Capture SDK can be used as a common OCR solution, recognizing text on existing images.

How it Works

Mobile Capture SDK provides access to single image processing functions, enabling the generic OCR functionality. This scenario works with any image file you can load to memory. It does not require access to the camera on the device.

Implementation

Note: *Before you begin, see How to Add the Library to Your Xcode Project.*

To implement the image recognition scenario, follow these steps:

1. Create an **RTREngine** object using the **sharedEngineWithLicenseData:** method. The method requires an NSData object containing your license data. For example, you can use dataWithContentsOfFile: to create a data object, then pass this object to the **sharedEngineWithLicenseData:** method.
2. Use the **createCoreAPI** method of the **RTREngine** object to create a recognizer object which conforms to the **RTRCoreAPI** protocol.
3. If you want to change recognition settings, use the **textRecognitionSettings** property of the recognizer object (**RTRCoreAPITextRecognitionSettings** protocol).
   - If you are using a recognition language different from English, specify it using the **setRecognitionLanguages:** method. Multiple languages are also supported, although setting too many languages may decrease recognition performance.
   - It is also recommended to call the **setAreaOfInterest:** method to specify the rectangular area of the image where to search for text. For example, your application may provide controls that allow user to select a smaller part of image for recognition if needed. Also, best results are achieved when the area of interest does not cover the whole image but has a margin of at least half the size of a typical printed character.
4. The **processingSettings** property of the recognizer object (**RTRCoreAPIProcessingSettings** protocol) allows you to set the number of processing threads.
5. Recognition starts with a call to the **recognizeTextOnImage:onProgress:onTextOrientationDetected:error:** method. It requires you to implement the following callbacks (passed as arguments to this method):
   - A progress callback (*onProgress*) that receives estimated completion percentage and warnings. This callback should return a BOOL value. The return value can be used to interrupt processing: return TRUE to terminate the current operation, FALSE to continue.
   - A callback that informs you when the image orientation is detected (*onTextOrientationDetected*).
   - A callback to handle errors (*error*).

   Please note, that the method is synchronous (blocking) and should not be used on UI thread.
6. When finished, the **recognizeTextOnImage:onProgress:onTextOrientationDetected:error:** method will return an array of **RTRTextBlock** objects which contain the results of recognition for the text areas found on the image. Each **RTRTextBlock** is an array containing one or more text lines represented by **RTRTextLine** objects. Each **RTRTextLine** contains information about the bounding quadrangle for a single line of text and the recognized text.
   Work with the results on your side.

See the description of classes and methods in the API Reference section.

How to Capture Data from Documents

This guide describes the procedure you need to follow to create an application which captures data from a specified type of document, without snapping a photo.

How it Works

In data capture scenarios, the processing quality is improved by the fact that we know which kind of data fields may be expected on the document. When you start capturing, you specify the type of document you are going to recognize (a data capture profile). The Mobile Capture SDK engine will automatically receive new camera frames and process them, trying to apply corresponding result schemes. The engine uses each new frame to verify and improve the recognition result from the previous frame. This process is

continued until a specific result scheme is matched and the result reaches the required stability level.

For some data capture profiles, there are two or more corresponding result schemes. The difference between a data capture profile and a result scheme is the following:

- A data capture profile is the general type of document you specify to the engine — for example, a bank card or some document with a machine-readable zone (MRZ).
- A result scheme is a more specific identifier of the recognized document, returned by the engine — for example, an embossed or unembossed bank card, or a specific MRZ (from a passport, visa, travel document, and so on).

The profile you specify determines which result schemes may be applied during recognition, and the result scheme determines which document fields will be recognized and returned as the result. Data capture profiles and corresponding result schemes supported in Mobile Capture library are detailed in Data Capture Profiles; see also the summary below in Supported ID Documents.

Note that Mobile Capture SDK also allows you to create custom data capture profiles for documents that are not supported out-of-the-box. See How to Capture a Custom Data Field for the description of this scenario.

Supported Documents

Mobile Capture SDK provides predefined data capture profiles for many types of data, including:

- machine-readable zone (MRZ) in various documents,
- international bank account numbers (IBAN),
- business card details,
- bank card details,
- data from ID documents:
  - ID cards,
  - passports,
  - driver's licenses, and other.

Recognizing with predefined profiles does not require you to set specific rules or specify regular expressions that should match document fields. You simply specify a data capture profile (the general type of a document) and get recognized data with a more specific result scheme identifying the recognized document.

MRZ

Mobile Capture SDK can automatically detect and recognize the machine-readable zone (MRZ) on various ID documents: passports, ID cards, travel documents, and other. For details on supported MRZ types and recognized data, see MRZ profiles.

For example, when recognizing a 2-line or 3-line MRZ of a passport or an ID document, Mobile Capture SDK will recognize and extract the following data:

- Document type and subtype
- Document number
- The country where the document was issued
- Document holder's first and last name, date of birth, sex and nationality

- Document holder's personal number
- Document expiry date

IBAN

Mobile Capture SDK allows to automatically detect and extract international bank account numbers for Germany, France, Spain, and the United Kingdom. IBAN can be extracted from any document.

Bank card

Mobile Capture SDK can capture data from debit and credit cards, embossed and unembossed.

When recognizing a bank card, Mobile Capture SDK will detect and extract the card number, cardholder's full name, and date of expiry.

ID documents

Mobile Capture SDK can automatically extract data from various ID documents such as ID cards, driver's licenses, passports, and other documents from different countries (see Data Capture Profiles for detailed information).

For example, when recognizing the front side of a German ID card, Mobile Capture SDK will detect and extract the following data:

- Document number
- Document holder's first and last name, nationality, date and place of birth
- RFID number
- Document expiry date

The rest of the data in the German ID card scheme is recognized from the back side of the card; note that the data capture profile you specify and the result data scheme are the same for both card sides.

Business card

Mobile Capture SDK can automatically extract data from various business cards.

On the business card, recognized by Mobile Capture SDK, the following data will be detected and extracted:

- First Name/Last Name
- Phone and/or mobile phone number
- Fax number
- Web address
- Mailing and E-mail address
- Company name
- Job title

The recognition languages of the business card can be specified via data capture profile settings. Please note, that capturing business cards with non-Latin scripts requires English language for E-mail and Web address recognition.

Important! *For best business cards recognition accuracy Full High Definition (Full HD) camera stream is required.*

Implementation

Note: *Before you begin, see How to Add the Library to Your Xcode Project.*

To implement the document data capture scenario, follow these steps:

1. Implement a delegate conforming to the **RTRDataCaptureServiceDelegate** protocol. The delegate will handle messages from the data capture service. Here are the recommendations on what its methods should do:
   - The **onBufferProcessedWithDataScheme:dataFields:resultStatus:** method is where you work with the results, display them to the user, etc.
   - The **onError:** method is for handling processing errors.
   - The **onWarning:** method can optionally be used to show warnings to the user.

2. Create an **RTREngine** object using the **sharedEngineWithLicenseData:** method. The method requires an NSData object containing your license data. For example, you can use **dataWithContentsOfFile:** to create a data object, then pass this object to the **sharedEngineWithLicenseData:** method.

3. Use the **createDataCaptureServiceWithDelegate:profile:** method of the **RTREngine** object to create a background recognition service. Set the type of document you are going to capture using the *profile* parameter — for example, `"IBAN"` or `"MRZ"`. The service is created and will further work with this profile (for a full list of available profiles, see Data Capture Profiles).
   Only one instance of the service per application is necessary: multiple threads will be started internally.

4. We recommend calling the **setAreaOfInterest:** method to specify the rectangular area on the frame where the document is likely to be found. For example, your application may show a highlighted rectangle in the UI into which the end user will try to fit the page they are capturing. The best result is achieved when the area of interest does not touch the boundaries of the frame but has a margin of at least half the size of a typical printed character.

5. Implement a delegate that adopts the AVCaptureVideoDataOutputSampleBufferDelegate protocol. Instantiate an AVCaptureSession object, add video input and output and set the video output delegate. When the delegate receives a video frame via the captureOutput:didOutputSampleBuffer:fromConnection: method, pass this frame on to the data capture service by calling the **addSampleBuffer:** method.
   We recommend using the `AVCaptureSessionPreset1280x720` preset for your AVCaptureSession. Also note that your video output must be configured to use the `kCVPixelFormatType_32BGRA` video pixel format.

6. Process the messages sent by the service to the **RTRDataCaptureServiceDelegate** delegate object. The result will be delivered via the **onBufferProcessedWithDataScheme:dataFields:resultStatus:** method:

   - an **RTRDataScheme** object; use its **id** property to determine what recognition scheme has been applied to the document (some profiles provide two or more recognition result schemes), and its **name** property to display a human-readable description to the user, if needed. For details on recognition schemes corresponding to the profile you selected, see Data Capture Profiles.

Important! *If **nil** is passed instead of a valid **RTRDataScheme** object, the data scheme has not yet been matched, which may mean that the document the user is trying to recognize is not a passport. In this case, the results are not usable.*

   - an array of **RTRDataField** objects, each representing one of the fields found and recognized. An **RTRDataField** object provides the identifier and the human-readable name for the field, the field text, and its location.
   - the result stability status, which indicates if the result is available and if it is likely to be improved by adding further frames. Use it to determine whether the application should stop processing and display the result to the user. We do not recommend using the result until the stability level has reached at least RTRResultStabilityAvailable and the data scheme has been matched.

7. Save the results for the recognized page. Call the **stopTasks** method to stop processing and clean up image buffers. The data capture service keeps its configuration settings (such as area of interest) and necessary resources. The processing will start automatically on the new call to the **addSampleBuffer:** method.

See the description of classes and methods in the API Reference section.

How to Capture a Custom Data Field with iOS

This section contains a step-by-step guide to creating an application that captures a single custom data field.

How it Works

With Mobile Capture SDK you can create custom data capture profiles for documents that are not supported out-of-the-box. In the corresponding result schemes you define custom data fields. (Currently, only one scheme per profile is supported, and only one field may be defined in the scheme). To tell the recognition engine that some text string is a data value (a field value), you will have to specify a regular expression that should match the strings you are looking for. The value may be a date, some code with a known format, and so on: the more specific the data is, the easier it would be to capture it.

This guide uses an alphanumeric code as an example of data that can be captured. Code format is the following: it contains 15 characters that are either digits or capital letters, and the first two characters are always digits. Example: 69KL46D7WF2AR5U.

Implementation

Note: *Before you begin, see How to Add the Library to Your Xcode Project.*

Implementing the delegates

1. Implement a delegate conforming to the **RTRDataCaptureServiceDelegate** protocol. The delegate will handle messages from the data capture service. Here are the recommendations on what its methods should do:
   - The **onBufferProcessedWithDataScheme:dataFields:resultStatus:** method is where you work with the results, display them to the user, etc.
   - The **onError:** method is for handling processing errors.
   - The **onWarning:** method can optionally be used to show warnings to the user.
2. Implement a delegate that adopts the AVCaptureVideoDataOutputSampleBufferDelegate protocol. Instantiate an AVCaptureSession object, add video input and output and set the video output delegate. When the delegate receives a video frame via the captureOutput:didOutputSampleBuffer:fromConnection: method, pass this frame on to the data capture service by calling the **addSampleBuffer:** method of the **RTRDataCaptureService** object.
   We recommend using the `AVCaptureSessionPreset1280x720` preset for your AVCaptureSession. Also note that your video output must be configured to use the `kCVPixelFormatType_32BGRA` video pixel format.

Loading the library and setting up the service

1. Create an **RTREngine** object using the **sharedEngineWithLicenseData:** method. The method requires an NSData object containing your license data. For example, you can use dataWithContentsOfFile: to create a data object, then pass this object to the **sharedEngineWithLicenseData:** method.

2. Use the **createDataCaptureServiceWithDelegate:profile:** method of the **RTREngine** object to create a background recognition service. The *profile* parameter should be left empty.
   Only one instance of the service per application is necessary: multiple threads will be started internally.

3. Call the **configureDataCaptureProfile** method of the **RTRDataCaptureService** object to create an **RTRDataCaptureProfileBuilder** object. Create a data scheme builder using the **addScheme:** method.

The scheme builder will allow you to specify a human-readable name for the scheme and to add field definitions.

4. Use the **addField:** method to create a new field builder. Use **setName:** to add a human-readable field name and **setRegEx:** to specify the regular expression that should match the field text. The *regEx* parameter `@"[0-9]{2}[0-9A-Z]{13}"` — match 2 digits followed by 13 characters which are digits or capital letters.

Note: *For details on regular expression syntax supported in ABBYY Mobile Capture SDK, see the Regular Expressions section.*

An alphanumeric code needs no additional check besides the regular expression. However, there is the option of implementing a block which would perform additional validation after the data has passed the regular expression check, for example, calculate the field's checksum (see the **setPredicateBlock:** method).

5. Call the **checkAndApply** method of the **RTRDataCaptureProfileBuilder** object to submit the profile for use in the data capture service. If an error is returned at this stage, it is probable the regular expression has mistakes in the syntax, please check it again.

6. We recommend also calling the **setAreaOfInterest:** method to specify the rectangular area on the frame where the field is likely to be found. For example, your application may show a highlighted rectangle in the UI into which the end user will try to fit the page they are capturing. The best result is achieved when the area of interest does not touch the boundaries of the frame but has a margin of at least half the size of a typical printed character.

Processing

1. Process the messages sent by the service to the **RTRDataCaptureServiceDelegate** delegate object. The result will be delivered via the **onBufferProcessedWithDataScheme:dataFields:resultStatus:** method:
   - An **RTRDataScheme** object. Its **id** property should return the same identifier you specified in the custom profile.

Important! *If **nil** is passed instead of a valid **RTRDataScheme** object, the data scheme has not yet been matched, which may mean that there is no data of the required type in the area of interest. In this case, the results are not usable.*

   - An array containing, in this case, one **RTRDataField** object which represents the extracted field. It provides the identifier and the human-readable name for the field, the field text, and its location.
   - The result stability status, which indicates if the result is available and if it is likely to be improved by adding further frames. Use it to determine whether the application should stop processing and display the result to the user. We do not recommend using the result until the stability level has reached at least RTRResultStabilityAvailable and the data scheme has been matched.
2. Save the results. Call the **stopTasks** method to stop processing and clean up image buffers. The data capture service keeps its configuration settings (the custom profile, the area of interest) and necessary resources. The processing will start automatically on the new call to the **addSampleBuffer:** method.

See the description of classes and methods in the API Reference section.

Code Samples

The ABBYY Mobile Capture SDK distribution package includes several code samples that show API usage and provide examples of typical scenarios.

The code samples are found in the root folder of the distribution package. All samples are provided in Objective-C and/or Swift programming languages.

Sample scenario	Folder name	Description
Text Capture	**sample-textcapture** **sample-textcapture-swift**	A simple text capture scenario. The only setting available to the user is the text language.
Data Capture	**sample-datacapture**	The general data capture scenario showing how to capture a predefined document and a custom data field.
Image Capture	**sample-imagecapture** **sample-imagecapture-swift**	This simple image capture scenarios demonstrate how to automatically capture an image from the smartphone video preview frames.
	sample-ui-imagecapture **sample-ui-imagecapture-swift**	The image capture scenario for capturing single page from the video-stream, using the special API for user interface implementation.
	sample-ui-imagecapture-multipage **sample-ui-imagecapture-multipage-swift**	The sample code implementing a multipage image capture scenario with tuned user interface.
Core API	**sample-coreapi** **sample-coreapi-swift**	The sample demonstrates a simple scenario of a single image processing with the core API.

Configuring the code samples

The samples can be opened and built right from where they are in the downloaded distribution package. To work with any of the code samples you need to do only a little configuring first.

1. Please change the bundle ID before building, modifying or otherwise using any of the samples.

2. All samples expect that the license file (named **license**) is found into the **assets** folder located in the distribution package root. Copy your license to this folder and rename the file if necessary (a license obtained from your supplier may have a different name).
   You can also change the license file name or path in the sample code: see the RTRViewController implementation.

API Reference

This section describes the Objective-C API of the ABBYY OCR library that allows you to create an application with OCR functionality for iOS.

RTREngine class

The main ABBYY Mobile Capture SDK class which serves to initialize the library and create a background recognition service. It is a singleton class: only one instance may exist at a time. Repeated attempts to create an **RTREngine** object will return the same object.

Properties

Name	Type	Description
extendedSettings	RTREngineSettings, read-only	Additional settings for ABBYY Mobile Capture SDK engine which apply to all processing scenarios.

Methods

Name	Description
+ sharedEngineWithLicenseData:	Creates the **RTREngine** object or returns its existing instance.
- createCoreAPI	Creates a core API object which provides access to low-level single image processing functions.
- createDataCaptureServiceWithDelegate:profile:	Creates a background service for data capture.
- createTextCaptureServiceWithDelegate:	Creates a background service for text recognition.
- createImageCaptureServiceWithDelegate:	Creates a background service for image capture.

Name	Description
- **dataSchemesForProfile:error:**	Returns full list of supported data schemes for specified profile.
- **languagesAvailableForOCR**	Returns the set of languages which can be used for text recognition.
- **languagesAvailableForBCR**	Returns the set of languages which can be used for text recognition.

sharedEngineWithLicenseData: method of the RTREngine class

Creates or returns the **RTREngine** object. Repeated calls to this method will result in the same object instance.

```
+ (instancetype)sharedEngineWithLicenseData:(NSData*)licenseData;
```

Parameters

licenseData

The license data to initialize ABBYY Mobile Capture SDK.

Return values

The method returns an instance of the **RTREngine** object, or **nil** if object creation failed.

createCoreAPI method of the RTREngine class

Creates a core API object which provides access to low-level single image processing functions.

```
- (id<RTRCoreAPI>)createCoreAPI;
```

Return values

The method returns an instance implementing the **RTRCoreAPI** protocol.

createDataCaptureServiceWithDelegate:profile: method of the RTREngine class

Creates a background recognition service to run in data capture mode. Only one instance of the service is necessary per application: multiple threads for processing will be started internally.

Before a call to this method, implement the **RTRDataCaptureServiceDelegate** protocol to work with processing results and handle errors and warnings.

```
- (id<RTRDataCaptureService>)createDataCaptureServiceWithDelegate:
(id<RTRDataCaptureServiceDelegate>)delegate profile:(NSString*)profile;


- (id<RTRDataCaptureService>)createDataCaptureServiceWithDelegate:
(id<RTRDataCaptureServiceDelegate>)delegate profile:(NSString*)profile
           settings:(RTRExtendedSettings*)settings;
```

Parameters

delegate

The delegate object that implements the **RTRDataCaptureServiceDelegate** protocol for interacting with the service.

profile

The name of a data capture profile (data scheme) to use. For the available predefined profiles see Data Capture Profiles.
Use an empty string or **nil** to configure your own profile for custom data field capture with the help of the **configureDataCaptureProfile** method of the **RTRDataCaptureService** protocol. You can also configure the predefined recognition languages using this method, if the data capture profile is set to BusinessCards.

settings

[optional] Extended service configuration settings represented by an **RTRExtendedSettings** object.

Return values

The method returns an instance implementing the **RTRDataCaptureService** protocol.

createTextCaptureServiceWithDelegate: method of the RTREngine class

Creates a background recognition service to run in text capture mode. Only one instance of the service is necessary per application: multiple threads for processing will be started internally.

Before a call to this method, implement the **RTRTextCaptureServiceDelegate** protocol to work with processing results and handle errors and warnings.

```
- (id<RTRTextCaptureService>)createTextCaptureServiceWithDelegate:
(id<RTRTextCaptureServiceDelegate>)delegate;


- (id<RTRTextCaptureService>)createTextCaptureServiceWithDelegate:
(id<RTRTextCaptureServiceDelegate>)delegate
           settings:(RTRExtendedSettings*)settings;
```

Parameters

delegate

The delegate object that implements the **RTRTextCaptureServiceDelegate** protocol for interacting with the service.

settings

[optional] The extended service configuration settings represented by an **RTRExtendedSettings** object.

Return values

The method returns an instance implementing the **RTRTextCaptureService** protocol.

createImageCaptureServiceWithDelegate: method of the RTREngine class

Creates a background recognition service to run in image capture mode. Only one instance of the service is necessary per application: multiple threads for processing will be started internally.

Before a call to this method, implement the **RTRImageCaptureServiceDelegate** protocol to work with processing results and handle errors and warnings.

```
- (id<RTRImageCaptureService>)createImageCaptureServiceWithDelegate:
(id<RTRImageCaptureServiceDelegate>)delegate;


- (id<RTRImageCaptureService>)createImageCaptureServiceWithDelegate:
(id<RTRImageCaptureServiceDelegate>)delegate
           settings:(RTRExtendedSettings*)settings;
```

Parameters

delegate

The delegate object that implements the **RTRImageCaptureServiceDelegate** protocol for interacting with the service.

settings

[optional] The extended service configuration settings represented by an **RTRExtendedSettings** object.

Return values

The method returns an instance implementing the **RTRImageCaptureService** protocol.

languagesAvailableForOCR method of the RTREngine class

Returns the set of languages which can be used for text recognition in the current application (for which the necessary resources are available).

```
- (NSSet*)languagesAvailableForOCR;
```

Return values

Returns a set of strings containing internal language names. See Available Languages for a complete list of languages and the corresponding internal names.

languagesAvailableForBcr method of the RTREngine class

Returns the set of languages which can be used for business cards recognition in the current application (for which the necessary resources are available).

```
- (NSSet*)languagesAvailableForBCR;
```

Return values

Returns a set of strings containing internal language names. See Available Languages for a complete list of languages and the corresponding internal names.

dataSchemesForProfile:error: method

Returns full list of supported data schemes for specified profile.

```
- (nullable NSArray<RTRDataScheme*>*)dataSchemesForProfile:(NSString*)
profile error:(NSError**)error;
```

Parameters

profile

The list of data schemes will be returned for this profile. Profile name should be passed. To investigate available profiles, see the Data Capture Profiles article.

error

The error that has occurred.

Return values

The method returns the **RTRDataScheme** array. See the **fields** property of the **RTRDataScheme** object to get information about all available fields for concrete data scheme.

Services API

RTRDataCaptureService protocol

A background data capture service protocol. Inherits from the **RTRRecognitionService** protocol.

This protocol is adopted by the data capture service object returned by the **createDataCaptureServiceWithDelegate:profile:** method. Its methods are used to tune the processing settings, pass video frames from the camera to the background processing engine, and release the resources afterwards.

The data capture service requires a delegate that conforms to the **RTRDataCaptureServiceDelegate** protocol. The service informs the delegate when the result is ready, sends progress information, warnings and errors.

Methods

Name	Description
- **addSampleBuffer:**	Sends the video frame obtained from camera to the service. Inherited from the **RTRRecognitionService** protocol.
- **configureDataCaptureProfile**	Creates a profile builder object with which you will be able to configure the data capture service to recognize fields of a specific type.
- **setAreaOfInterest:**	Sets the search area on the frame. Inherited from the **RTRRecognitionService** protocol.
- **stopTasks**	Stops processing and releases the resources used by the service. Inherited from the **RTRRecognitionService** protocol.

addSampleBuffer: method of the RTRDataCaptureService protocol

Sends the video frame obtained from camera to the service.

Call this method to pass on the video sample buffer received by an **AVCaptureVideoDataOutputSampleBufferDelegate** object via the **captureOutput:didOutputSampleBuffer:fromConnection:** method. The service will pick the frames it needs from the sequence of the frames you supply.

Note: *The video output must be configured to use the kCVPixelFormatType_32BGRA video pixel format. Other pixel formats are currently not supported.*

```
- (void)addSampleBuffer:(CMSampleBufferRef)sampleBuffer;
```

Parameters

sampleBuffer

A **CMSampleBuffer** object containing the video frame data.

configureDataCaptureProfile method of the RTRDataCaptureService protocol

Creates a profile builder object with which you will be able to configure the data capture service to recognize fields of a specific type. This is the first step for capturing a custom field, without using any of the predefined data capture profiles, and makes sense only if you have left the profile name parameter empty when creating the data capture service.

```
- (id<RTRDataCaptureProfileBuilder>)configureDataCaptureProfile;
```

Return values

The method returns an instance implementing the **RTRDataCaptureProfileBuilder** protocol, or **nil** if a profile may not be configured (e.g. you have already specified a profile name on creating the data capture service).

setAreaOfInterest: method of the RTRDataCaptureService protocol

Sets the search area on the frame.

The size of the area of interest affects performance and the speed of convergence of the result. The best result is achieved when the area of interest does not touch the boundaries of the frame but has a margin of at least half the size of a typical printed character.

```
- (void)setAreaOfInterest:(CGRect)areaOfInterest;
```

Parameters

areaOfInterest

The rectangle specifying the area of interest in the image coordinates. Pass CGRectZero as this parameter to select the default area of interest that covers the whole frame (not recommended).

stopTasks method of the RTRDataCaptureService protocol

Stops processing and cleans up image buffers. The service keeps its configuration settings and necessary resources, so the processing will start automatically when the service receives a new frame.

```
- (void)stopTasks;
```

RTRDataCaptureServiceDelegate protocol

The protocol for a delegate object to receive results, status information, warnings and errors from the data capture service. Inherits from the **RTRRecognitionServiceDelegate** protocol. The methods are to be implemented on the client side.

Methods

Name	Description
- **onBufferProcessedWithDataScheme:dataFields:resultStatus:**	Notifies the delegate that a frame was recognized, delivers the result and status information.
- **onError:**	Notifies the delegate about an error. Inherited from the **RTRRecognitionServiceDelegate** protocol.
- **onWarning:**	Optional method. Informs the delegate about warnings from the service. Inherited from the **RTRRecognitionServiceDelegate** protocol.

onBufferProcessedWithDataScheme:dataFields:resultStatus: method of the RTRDataCaptureServiceDelegate protocol

Notifies the delegate that a frame was recognized, delivers the result and its stability status.

The result stability status should be used to determine if the accuracy is high enough for the result to be used for any practical purposes. We recommend not to use the data in any way until the stability level has reached at least RTRResultStabilityAvailable and the data scheme has been matched. When stability of the result has reached the desired level, the service may be stopped by calling the **stopTasks** method of the **RTRDataCaptureService** protocol.

This method is to be implemented on the client side. The implementation of this method will probably contain assessing the result plausibility, displaying the results to the user or using them in any other way you need.

```
- (void)onBufferProcessedWithDataScheme:(RTRDataScheme*)dataScheme
dataFields:(NSArray<RTRDataField*>*)dataFields
           resultStatus:(RTRResultStabilityStatus)resultStatus;
```

Parameters

dataScheme

Information on the data scheme applied to the recognized frame, represented by a **RTRDataScheme** object.

Important! *If* ***nil*** *is passed instead of a valid* ***RTRDataScheme*** *object, the data scheme has not yet been matched, which may mean that the document the user is trying to recognize does not fit the data capture profile with which the data service was created. In this case, the results are not usable.*

dataFields

The result as an array of data fields, represented by **RTRDataField** objects.

resultStatus

The estimate of how stable the result is, represented by an **RTRResultStabilityStatus** enumeration constant. It is not guaranteed that it ever reaches the desired level for a particular scene.

onError: method of the RTRDataCaptureServiceDelegate protocol

Notifies the delegate about an error.

```
- (void)onError:(NSError*)error;
```

Parameters

error

The error that has occurred.

onWarning: method of the RTRDataCaptureServiceDelegate protocol

Informs the delegate about warnings. This method is optional.

```
- (void)onWarning:(RTRCallbackWarningCode)warningCode;
```

Parameters

warningCode

The warning that has occurred, represented by an **RTRCallbackWarningCode** enumeration constant.

RTRDataCaptureProfileBuilder protocol

The protocol for a builder object which allows you to configure a custom data capture profile.

Methods

Name	Description
- **checkAndApply**	Submits the configured profile for use in the data capture service.
- **addScheme:**	Creates a new scheme in the data capture profile. Using the scheme builder you will then be able to add the data fields and define the rules to which they should conform.
- **setRecognitionLanguages:**	Sets the languages to be used for field recognition.

checkAndApply method of the RTRDataCaptureProfileBuilder protocol

Submits the configured profile for use in the data capture service.

This method should be called after all your modifications to the profile are completed. If this method call is successful, the service is ready to capture custom data fields as specified by the profile.

```
- (NSError*)checkAndApply;
```

Return values

The method returns **nil** if the profile was applied successfully. If there were some problems (for example, the regular expression is not valid), the error object is returned instead.

addScheme: method of the RTRDataCaptureProfileBuilder protocol

Creates a new scheme in the data capture profile. Using the scheme builder you will then be able to add the data fields and define the rules to which they should conform.

Note: *Currently, only one scheme may exist in the profile, and only one field may be defined in the scheme.*

```
- (id<RTRDataSchemeBuilder>)addScheme:(NSString*)id;
```

Parameters

id

The scheme identifier.

Return values

The method returns an instance implementing the **RTRDataSchemeBuilder** protocol.

setRecognitionLanguages: method of the RTRDataCaptureProfileBuilder protocol

Sets the languages to be used for recognition.

By default, only the English language is set. Setting the correct languages for your text will improve recognition accuracy. However, setting too many languages may decrease performance.

Use the **languagesAvailableForOCR** method to check which languages are supported in your application.

```
- (id<RTRDataCaptureProfileBuilder>)setRecognitionLanguages:(NSSet*)
recognitionLanguages;
```

Parameters

recognitionLanguages

The set of languages to be used for recognition, each language represented by its internal name as a string. See Available Languages for the list of languages and corresponding internal names.

Return values

The method returns the same **RTRDataCaptureProfileBuilder** object on which it was called.

RTRDataSchemeBuilder protocol

The protocol for a scheme builder object which lets you add fields to the scheme.

Note: *Currently, only one scheme may exist in the profile, and only one field may be defined in the scheme.*

Methods

Name	Description
- **addField:**	Adds a new field.

Name	Description
	The rules to which the data should conform may be specified later via the field builder object.
- **setName:**	Sets the scheme name.

addField: method of the RTRDataSchemeBuilder protocol

Adds a new field.

The rules to which the data should conform may be specified later via the field builder object.

Note: *Currently, only one scheme may exist in the profile, and only one field may be defined in the scheme.*

```
- (id<RTRDataFieldBuilder>)addField:(NSString*)id;
```

Parameters

id

The field identifier.

Return values

The method returns an instance implementing the **RTRDataFieldBuilder** protocol.

setName: method of the RTRDataSchemeBuilder protocol

Sets the scheme name.

```
- (id<RTRDataSchemeBuilder>)setName:(NSString*)name;
```

Parameters

name

The new scheme name.

Return values

The method returns the same **RTRDataSchemeBuilder** object on which it was called.

RTRDataFieldBuilder protocol

The protocol for a field builder object which allows you to set the name and rules for the data field.

Properties

Name	Type	Description
id	NSString*, read-only	Field identifier.

Methods

Name	Description
- **setName:**	Sets a human-readable name for the field.
- **setPredicateBlock:**	Sets the user-implemented validation block which will be called for further verification of the data (e.g. calculating the checksum) after it has passed the regular expression check.
- **setRegEx:**	Sets the regular expression to match the field data.

setName: method of the RTRDataFieldBuilder protocol

Sets the field name.

```
- (id<RTRDataFieldBuilder>)setName:(NSString*)name;
```

Parameters

name

The new field name.

Return values

The method returns the same **RTRDataFieldBuilder** object on which it was called.

setPredicateBlock: method of the RTRDataFieldBuilder protocol

Sets the validation block which will be called for further verification of the data (e.g. calculating the checksum) after it has passed the regular expression check.

```
- (id<RTRDataFieldBuilder>)setPredicateBlock:(RTRFieldPredicateBlock)
predicateBlock;
```

Parameters

predicateBlock

The user-implemented validation block of the type **RTRFieldPredicateBlock**. May be **nil**, which means the data will not be verified.

Return values

The method returns the same **RTRDataFieldBuilder** object on which it was called.

setRegEx: method of the RTRDataFieldBuilder protocol

Sets the regular expression that should match the field's text.

Note: *For details on regular expression syntax supported in ABBYY Mobile Capture SDK, see the Regular Expressions section.*

Important! *If the field contains two or more matches for the specified regular expression, the engine will extract and return only the first one.*

```
- (id<RTRDataFieldBuilder>)setRegEx:(NSString*)regEx;
```

Parameters

regEx

A string describing the regular expression.

Return values

The method returns the same **RTRDataFieldBuilder** object on which it was called.

RTRFieldPredicateBlock

This is a type definition for a user-defined validation block which will be called for further verification of the data (e.g. calculating the checksum) after it has passed the regular expression check.

A typical use for the validation block would be to calculate a checksum.

```
typedef BOOL (^RTRFieldPredicateBlock)(NSString* value);
```

Parameters

value

The string with the recognized text of the field.

Return values

The block must return TRUE if the data is correct, FALSE otherwise.

RTRImageCaptureService protocol

A background image capture service protocol. Inherits from the **RTRRecognitionService** protocol.

This protocol is adopted by the image capture service object returned by the **createImageCaptureServiceWithDelegate:** method. Its methods are used to tune the processing settings, pass video frames from the camera to the background processing engine, and release the resources afterwards.

The image capture service requires a delegate that conforms to the **RTRImageCaptureServiceDelegate** protocol. The service informs the delegate when the result is ready, sends progress information, warnings and errors.

Properties

Name	Type	Description
aspectRatioMin	CGFloat	Lower limit of the document's aspect ratio. This property is used in pair with the **aspectRatioMax**, defining an interval of acceptable aspect ratio values of the document to be captured. Setting aspect ratio will help to improve boundary detection accuracy. If only **aspectRatioMax** is set, **aspectRatioMin** will be set to 1. By default the property is set to 0 (aspect ratio is not set).
aspectRatioMax	CGFloat	Upper limit of document's aspect ratio. This property is used in pair with the **aspectRatioMin**, defining an interval of acceptable aspect ratio values of the document to be captured. Setting aspect

Name	Type	Description
		ratio will help to improve boundary detection accuracy. If only **aspectRatioMin** is set, **aspectRatioMax** will be set to CGFLOAT_MAX. By default the property is set to 0 (aspect ratio is not set).

Document aspect ratio setting is intended to specify the exact proportions of the target document, which will increase capture accuracy.

Notes:

- *Aspect ratio detection requires the mobile device to be placed strictly horizontally over the document during image capture. In case the mobile device is tilted, camera will capture distorted image and the document aspect ratio may be detected incorrectly.*
- *The value of aspect ratio is calculated by division of the longer side to the shorter side and is expected to be greater than or equal to 1 (or 0 if not set). If neither **aspectRatioMin** nor **aspectRatioMin** are set, the values will be calculated from the **documentSize** setting.*

Methods

Name	Description
- **addSampleBuffer:**	Sends the video frame obtained from camera to the service. Inherited from the **RTRRecognitionService** protocol.
- **setAreaOfInterest:**	Sets the search area on the frame. Inherited from the **RTRRecognitionService** protocol.
- **setDocumentSize:**	Sets the physical size of the document to be captured.
- **stopTasks**	Stops processing and releases the resources used by the recognition service. Inherited from the **RTRRecognitionService** protocol.

addSampleBuffer: method of the RTRImageCaptureService protocol

Sends the video frame obtained from camera to the service.

Call this method to pass on the video sample buffer received by an **AVCaptureVideoDataOutputSampleBufferDelegate** object via the **captureOutput:didOutputSampleBuffer:fromConnection:** method. The service will pick the frames it needs from the sequence of the frames you supply.

Note: *The video output must be configured to use the kCVPixelFormatType_32BGRA video pixel format. Other pixel formats are currently not supported.*

```
- (void)addSampleBuffer:(CMSampleBufferRef)sampleBuffer;
```

Parameters

sampleBuffer

A **CMSampleBuffer** object containing the video frame data.

setAreaOfInterest: method of the RTRImageCaptureService protocol

Sets the search area on the frame.

The size of the area of interest affects performance and the speed of convergence of the result. The best result is achieved when the area of interest does not touch the boundaries of the frame but has a margin of at least half the size of a typical printed character.

```
- (void)setAreaOfInterest:(CGRect)areaOfInterest;
```

Parameters

areaOfInterest

The rectangle specifying the area of interest in the image coordinates. Pass CGRectZero as this parameter to select the default area of interest that covers the whole frame (not recommended).

setDocumentSize: method of the RTRImageCaptureService protocol

Sets the physical size of the document to be captured.

The values set by this method are used in various purposes. Setting this parameter will help to improve document boundary detection accuracy and preserve aspect ratio after crop. Known physical size of the document is used for document orientation detection during capture. The image resolution is automatically calculated to the physical size before export.

```
- (void)setDocumentSize:(CGSize)size;
```

Parameters

size

The size of the document in millimeters.

stopTasks method of the RTRImageCaptureService protocol

Stops processing and cleans up image buffers. The service keeps its configuration settings and necessary resources, so the processing will start automatically when the service receives a new frame.

```
- (void)stopTasks;
```

RTRImageCaptureServiceDelegate protocol

The protocol for a delegate object to receive results, status information, warnings and errors from the text capture service. Inherits from the **RTRRecognitionServiceDelegate** protocol. The methods are to be implemented on the client side.

Methods

Name	Description
- **onBufferProcessedWithStatus:result:**	Notifies the delegate that the image was captured and delivers the result.
- **onError:**	Notifies the delegate about an error. Inherited from the **RTRRecognitionServiceDelegate** protocol.
- **onWarning:**	Optional method. Informs the delegate about warnings from the service. Inherited from the **RTRRecognitionServiceDelegate** protocol.

onBufferProcessedWithStatus:result: method of the RTRImageCaptureServiceDelegate protocol

Notifies the delegate that an image was captured and delivers the result.

When stability of the result has reached the desired level, the service may be stopped by calling the **stopTasks** method of the **RTRImageCaptureService** protocol.

This method is to be implemented on the client side. The implementation of this method will probably contain assessing the result plausibility, displaying the results to the user or using them in any way you need.

```
- (void)onBufferProcessedWithStatus:(RTRImageCaptureStatus*)status result:
(RTRImageCaptureResult*)result;
```

Parameters

status

The status of the image capture.

result

The result of the image capture.

onError: method of the RTRImageCaptureServiceDelegate protocol

Notifies the delegate about an error.

```
- (void)onError:(NSError*)error;
```

Parameters

error

The error that has occurred.

onWarning: method of the RTRImageCaptureServiceDelegate protocol

Informs the delegate about warnings. This method is optional.

```
- (void)onWarning:(RTRCallbackWarningCode)warningCode;
```

Parameters

warningCode

The warning that has occurred, represented by an **RTRCallbackWarningCode** enumeration constant.

RTRTextCaptureService protocol

A background text capture service protocol. Inherits from the **RTRRecognitionService** protocol.

This protocol is adopted by the text capture service object returned by the **createTextCaptureServiceWithDelegate:** method. Its methods are used to tune the processing settings, pass video frames from the camera to the background processing engine, and release the resources afterwards.

The text capture service requires a delegate that conforms to the **RTRTextCaptureServiceDelegate** protocol. The service informs the delegate when the result is ready, sends progress information, warnings and errors.

Methods

Name	Description
- **addSampleBuffer:**	Sends the video frame obtained from camera to the service. Inherited from the **RTRRecognitionService** protocol.
- **setAreaOfInterest:**	Sets the search area on the frame. Inherited from the **RTRRecognitionService** protocol.
- **setRecognitionLanguages:**	Sets the languages to be used for recognition.
- **setTranslationDictionary:**	Sets the name of the translation dictionary.
- **stopTasks**	Stops processing and releases the resources used by the recognition service. Inherited from the **RTRRecognitionService** protocol.

addSampleBuffer: method of the RTRTextCaptureService protocol

Sends the video frame obtained from camera to the service.

Call this method to pass on the video sample buffer received by an **AVCaptureVideoDataOutputSampleBufferDelegate** object via the **captureOutput:didOutputSampleBuffer:fromConnection:** method. The service will pick the frames it needs from the sequence of the frames you supply.

Note: *The video output must be configured to use the kCVPixelFormatType_32BGRA video pixel format. Other pixel formats are currently not supported.*

```
- (void)addSampleBuffer:(CMSampleBufferRef)sampleBuffer;
```

Parameters

sampleBuffer

A **CMSampleBuffer** object containing the video frame data.

setAreaOfInterest: method of the RTRTextCaptureService protocol

Sets the search area on the frame.

The size of the area of interest affects performance and the speed of convergence of the result. The best result is achieved when the area of interest does not touch the boundaries of the frame but has a margin of at least half the size of a typical printed character.

```
- (void)setAreaOfInterest:(CGRect)areaOfInterest;
```

Parameters

areaOfInterest

The rectangle specifying the area of interest in the image coordinates. Pass CGRectZero as this parameter to select the default area of interest that covers the whole frame (not recommended).

setRecognitionLanguages: method of the RTRTextCaptureService protocol

Sets the languages to be used for recognition.

By default, only the English language is set. Setting the correct languages for your text will improve recognition accuracy. However, setting too many languages may decrease performance.

Use the **languagesAvailableForOCR** method to check which languages are supported in your application.

```
- (void)setRecognitionLanguages:(NSSet*)recognitionLanguages;
```

Parameters

recognitionLanguages

The set of languages to be used for recognition, each language represented by its internal name as a string. See Available Languages for the list of languages and corresponding internal names.

setTranslationDictionary: method of the RTRTextCaptureService protocol

Sets current translation dictionary, attaches or detaches a dictionary to enable or disable translation. By default, translation is disabled and no translation dictionary is used.

Translation dictionaries should be put in the **Translation** subfolder of the application bundle. Some dictionaries are supplied with the distribution. See Available Translation Dictionaries for a full list.

Important! *Calling this method with a dictionary name attaches this translation dictionary (or changes the one currently attached). With a dictionary attached, the recognized text will be translated automatically, and the* ***onBufferProcessedWithTextLines:resultStatus:*** *method will return the result in the target language. The result of recognition in the source language will be unavailable. To detach a dictionary, pass a* ***nil*** *argument.*

```
- (void)setTranslationDictionary:(NSString*)dictionaryName;
```

Parameters

dictionaryName

The name of the translation dictionary file, without extension. Can also be **nil** to detach the current dictionary.

stopTasks method of the RTRTextCaptureService protocol

Stops processing and cleans up image buffers. The service keeps its configuration settings and necessary resources, so the processing will start automatically when the service receives a new frame.

```
- (void)stopTasks;
```

RTRTextCaptureServiceDelegate protocol

The protocol for a delegate object to receive results, status information, warnings and errors from the text capture service. Inherits from the **RTRRecognitionServiceDelegate** protocol. The methods are to be implemented on the client side.

Methods

Name	Description
- **onBufferProcessedWithTextLines:resultStatus:**	Notifies the delegate that a frame was recognized, delivers the result and status information.
- **onError:**	Notifies the delegate about an error. Inherited from the **RTRRecognitionServiceDelegate**

Name	Description
	protocol.
- **onWarning:**	Optional method. Informs the delegate about warnings from the service. Inherited from the **RTRRecognitionServiceDelegate** protocol.

onBufferProcessedWithTextLines:resultStatus: method of the RTRTextCaptureServiceDelegate protocol

Notifies the delegate that a frame was recognized, delivers the result and its stability status.

The result stability status should be used to determine if the accuracy is high enough for the result to be used for any practical purposes. We recommend not to use the data in any way until the stability level has reached at least RTRResultStabilityAvailable. When stability of the result has reached the desired level, the service may be stopped by calling the **stopTasks** method of the **RTRTextCaptureService** protocol.

This method is to be implemented on the client side. The implementation of this method will probably contain assessing the result plausibility, displaying the results to the user or using them in any way you need.

```
- (void)onBufferProcessedWithTextLines:(NSArray*)textLines resultStatus:
(RTRResultStabilityStatus)resultStatus;
```

Parameters

textLines

The result as an array of text lines, represented by **RTRTextLine** objects.

resultStatus

The estimate of how stable the result is, represented by an **RTRResultStabilityStatus** enumeration constant. It is not guaranteed that it ever reaches the desired level for a particular scene.

onError: method of the RTRTextCaptureServiceDelegate protocol

Notifies the delegate about an error.

```
- (void)onError:(NSError*)error;
```

Parameters

error

The error that has occurred.

onWarning: method of the RTRTextCaptureServiceDelegate protocol

Informs the delegate about warnings. This method is optional.

```
- (void)onWarning:(RTRCallbackWarningCode)warningCode;
```

Parameters

warningCode

The warning that has occurred, represented by an **RTRCallbackWarningCode** enumeration constant.

RTRRecognitionService protocol

The base background recognition service protocol, inherited by the **RTRDataCaptureService**, **RTRTextCaptureService** and **RTRImageCaptureService** protocols.

Requires a delegate that conforms to the **RTRRecognitionServiceDelegate** protocol.

Methods

Name	Description
- **addSampleBuffer:**	Sends the video frame obtained from camera to the service.
- **setAreaOfInterest:**	Sets the search area on the frame.
- **stopTasks**	Stops processing and releases the resources used by the service.

addSampleBuffer: method of the RTRRecognitionService protocol

Sends the video frame obtained from camera to the service.

Call this method to pass on the video sample buffer received by an **AVCaptureVideoDataOutputSampleBufferDelegate** object via the **captureOutput:didOutputSampleBuffer:fromConnection:** method. The service will pick the frames it needs from the sequence of the frames you supply.

Note: *The video output must be configured to use the kCVPixelFormatType_32BGRA video pixel format. Other pixel formats are currently not supported.*

```
- (void)addSampleBuffer:(CMSampleBufferRef)sampleBuffer;
```

Parameters

sampleBuffer

A **CMSampleBuffer** object containing the video frame data.

setAreaOfInterest: method of the RTRRecognitionService protocol

Sets the search area on the frame.

The size of the area of interest affects performance and the speed of convergence of the result. The best result is achieved when the area of interest does not touch the boundaries of the frame but has a margin of at least half the size of a typical printed character.

```
- (void)setAreaOfInterest:(CGRect)areaOfInterest;
```

Parameters

areaOfInterest

The rectangle specifying the area of interest in the image coordinates. Pass CGRectZero as this parameter to select the default area of interest that covers the whole frame (not recommended).

stopTasks method of the RTRRecognitionService protocol

Stops processing and cleans up image buffers. The service keeps its configuration settings and necessary resources, so the processing will start automatically when the service receives a new frame.

```
- (void)stopTasks;
```

RTRRecognitionServiceDelegate protocol

The base protocol for a recognition service delegate, inherited by the **RTRDataCaptureServiceDelegate**, **RTRTextCaptureServiceDelegate** and **RTRImageCaptureServiceDelegate** protocols. The methods are to be implemented on the client side.

Methods

Name	Description
- **onError:**	Notifies the delegate about an error.

Name	Description
- **onWarning:**	Optional method. Informs the delegate about warnings from the service.

onError: method of the RTRRecognitionServiceDelegate protocol

Notifies the delegate about an error.

```
- (void)onError:(NSError*)error;
```

Parameters

error

The error that has occurred.

onWarning: method of the RTRRecognitionServiceDelegate protocol

Informs the delegate about warnings. This method is optional.

```
- (void)onWarning:(RTRCallbackWarningCode)warningCode;
```

Parameters

warningCode

The warning that has occurred, represented by an **RTRCallbackWarningCode** enumeration constant.

RTRResultStabilityStatus enumeration

Result stability status: the estimate of how stable the result is, and whether it is likely to be improved by adding new frames. We do not recommend using the results in any way while stability is below RTRResultStabilityAvailable.

```
typedef NS_ENUM(NSInteger, RTRResultStabilityStatus) {
   RTRResultStabilityNotReady,
   RTRResultStabilityTentative,
   RTRResultStabilityVerified,
   RTRResultStabilityAvailable,
   RTRResultStabilityTentativelyStable,
   RTRResultStabilityStable
};
```

Constants

Name	Description
RTRResultStabilityNotReady	No content available.
RTRResultStabilityTentative	Content detected on a single frame.
RTRResultStabilityVerified	Content verified: matching content found in at least two frames.
RTRResultStabilityAvailable	Matching content found in three or more frames. The content is recognized and the result is available, though the result can still vary with the addition of new frames.
RTRResultStabilityTentativelyStable	The result has been stable in the last two frames.
RTRResultStabilityStable	The result has been stable in the last three or more frames.

Core API

RTRCoreAPI protocol

Provides access to low-level functions for single image processing. Useful when you need to recognize an image that was not taken by the camera of the device on which the application operates — for example, scans sent by email.

Inherits from **RTRRecognitionCoreAPI**, **RTRDataCaptureCoreAPI** and **RTRImagingCoreAPI** protocols.

Properties

Name	Type	Description
processingSettings	RTRCoreAPIProcessingSettings	Provides access to the general processing settings common for different scenarios.
textRecognitionSettings	RTRCoreAPITextRecognitionSettings	Provides access to the settings of text recognition.

Methods

Name	Description
- recognizeText:onProgress:onTextOrientationDetected:error:	Perfroms recognition of a single image.

recognizeText:onProgress:onTextOrientationDetected:error: method of the RTRCoreAPI protocol

Performs recognition of a single image.

```
- (NSArray*)recognizeText:(UIImage*)image
       onProgress:(BOOL(^)(int percentage, RTRCallbackWarningCode
warningCode))progressCallback
       onTextOrientationDetected:(void(^)(int angle))
textOrientationDetectedCallback
       error:(__autoreleasing NSError**)error;
```

Parameters

image

The image to be recognized.

onProgress

The callback informing you of approximate percentage of operation completed, and any warning that occurred (represented by an **RTRCallbackWarningCode** constant). This callback can also be used to interrupt processing: return TRUE if you wish to terminate the current operation, FALSE to continue.

onTextOrientationDetected

The callback informing you when the image orientation is detected. The *angle* parameter can take values of 0, 90, 180, and 270, and means the angle on which the image should be rotated to get normal orientation.

error

The error callback.

Return values

The method returns an array of **RTRTextBlock** objects which contain the results of recognition for the text areas found on the image.

RTRCoreAPIDataCaptureSettings protocol

Settings specific for data capture scenario.

Properties

Name	Type	Description
profile	NSString*	The name of a data capture profile (data scheme) to use. For the available predefined profiles see Data Capture Profiles. ***Note:*** Currently only the BusinessCards profile is supported in the data capture scenario with the core API usage.

Methods

Name	Description
- **setAreaOfInterest:**	Sets the search area on the image.
- **configureDataCaptureProfile**	Sets the languages to be used for recognition.

setAreaOfInterest: method of the RTRCoreAPIDataCaptureSettings protocol

Sets the search area on the image.

```
- (void)setAreaOfInterest:(CGRect)areaOfInterest;
```

Parameters

areaOfInterest

The rectangle specifying the area of interest in the image coordinates. Pass CGRectZero as this parameter to select the default area of interest that covers the whole frame.

configureDataCaptureProfile method of the RTRCoreAPIDataCaptureSettings protocol

Creates a profile builder object with which you will be able to configure the data capture core API to recognize a custom field.

```
- (id<RTRDataCaptureProfileBuilder>)configureDataCaptureProfile;
```

Return values

The method returns an instance implementing the **RTRDataCaptureProfileBuilder** protocol, or **nil** if a profile may not be configured (e.g. you have already specified a profile name on creating the data capture service).

RTRCoreAPIProcessingSettings protocol

General processing settings common for different scenarios.

Properties

Name	Type	Description
processingThreadsCount	NSInteger	Specifies the number of threads to be used for processing. The default value of this property is 0, which means that the number of processing threads will be determined automatically.

RTRCoreAPITextRecognitionSettings protocol

Settings specific for text recognition scenario.

Properties

Name	Type	Description
textOrientationDetectionEnabled	BOOL	Enables or disables detection of the image orientation while preprocessing. If the property is set to YES, the image top is detected and correct orientation can be used for image rotation. You can set this property to NO for

Name	Type	Description
		speeding the process up. ***Note:*** *Disable the image detection only if you can be sure that the captured image has correct orientation. Otherwise the text on image will not be detected and recognized.* The default value of this property is YES (enabled).

Methods

Name	Description
- **setAreaOfInterest:**	Sets the search area on the image.
- **setRecognitionLanguages:**	Sets the languages to be used for recognition.

setAreaOfInterest: method of the RTRCoreAPITextRecognitionSettings protocol

Sets the search area on the image.

```
- (void)setAreaOfInterest:(CGRect)areaOfInterest;
```

Parameters

areaOfInterest

The rectangle specifying the area of interest in the image coordinates. Pass CGRectZero as this parameter to select the default area of interest that covers the whole frame.

setRecognitionLanguages: method of the RTRCoreAPITextRecognitionSettings protocol

Sets the languages to be used for recognition.

By default, only the English language is set. Setting the correct languages for your text will improve recognition accuracy. However, setting too many languages may decrease performance.

Use the **languagesAvailableForOCR** method to check which languages are supported in your application.

```
- (void)setRecognitionLanguages:(NSSet*)recognitionLanguages;
```

Parameters

recognitionLanguages

The set of languages to be used for recognition, each language represented by its internal name as a string. See Available Languages for the list of languages and corresponding internal names.

RTRDataCaptureCoreAPI protocol

Provides access to low-level single image core API functions for current thread, that are intended for capturing data. The protocol should be used in the same thread in which it was created. Multiple objects can be created on different threads and used concurrently. All methods are synchronous (blocking) and should not be used on UI thread. Intended for advanced users.

Properties

Name	Type	Description
processingSettings	**RTRCoreAPIProcessingSettings**	Provides access to the general processing settings common for different scenarios.
dataCaptureSettings	**RTRCoreAPIDataCaptureSettings**	Provides access to the settings of data capture.
extendedSettings	**RTRExtendedSettings**	Extended service configuration settings. Intended for advanced users: most common scenarios will work with the default settings.
textOrientationDetectionEnabled	BOOL	Enables or disables detection of the image orientation while preprocessing. If the property is set to YES, the image top is detected and correct orientation can be used for image rotation. You can set this property to NO for speeding the process up. ***Note:*** *Disable the image detection only if you can be sure that the captured image has correct orientation. Otherwise the text on image will not be detected and*

Name	Type	Description
		recognized. The default value of this property is YES (enabled).

Methods

Name	Description
- extractDataFromImage:onProgress:onTextOrientationDetected:dataScheme:error:	Extracts data from a still image.

extractDataFromImage:onProgress:onTextOrientationDetected:dataScheme:error: method of the RTRDataCaptureCoreAPI protocol

Extracts data from a still image.

```
- (NSArray<RTRDataField*>*)extractDataFromImage:(UIImage*)image onProgress:
(RTRProgressCallbackBlock)progressCallback
           onTextOrientationDetected:(RTRTextOrientationDetectedBlock)
textOrientationDetectedCallback
           error:(__autoreleasing NSError**)error;
```

Parameters

image

The image to be recognized..

progressCallback

The callback informing you of approximate percentage of operation completed, and any warning that occurred (represented by an **RTRCallbackWarningCode** constant). This callback can also be used to interrupt processing: return TRUE if you wish to terminate the current operation, FALSE to continue.

textOrientationDetectedCallback

The callback informing you when the image orientation is detected. The *angle* parameter can take values of 0, 90, 180, and 270, and means the angle on which the image should be rotated to get normal orientation.

error

The error callback.

Return values

The method returns an array of data fields, represented by **RTRDataField** objects.

RTRRecognitionCoreAPI protocol

Provides access to low-level single image core API functions for current thread, that are intended for text recognition on photos. The protocol should be used in the same thread in which it was created. Multiple objects can be created on different threads and used concurrently. All methods are synchronous (blocking) and should not be used on UI thread. Intended for advanced users.

Properties

Name	Type	Description
processingSettings	**RTRCoreAPIProcessingSettings**	Provides access to the general processing settings common for different scenarios.
textRecognitionSettings	**RTRCoreAPITextRecognitionSettings**	Provides access to the settings of text recognition scenario.
extendedSettings	**RTRExtendedSettings**	Extended service configuration settings. Intended for advanced users: most common scenarios will work with the default settings.

Methods

Name	Description
- recognizeTextOnImage:onProgress:onTextOrientationDetectedCallback:error:	Recognizes text on a still image.

recognizeTextOnImage:onProgress:onTextOrientationDetectedCallback:error: method of the RTRRecognitionCoreAPI protocol

Recognizes text on a still image.

```
- (NSArray<RTRTextBlock*>*)recognizeTextOnImage:(UIImage*)image onProgress:
(RTRProgressCallbackBlock)progressCallback
           onTextOrientationDetected:(RTRTextOrientationDetectedBlock)
textOrientationDetectedCallback
           error:(__autoreleasing NSError**)error;
```

Parameters

image

The image to be recognized..

progressCallback

The callback informing you of approximate percentage of operation completed, and any warning that occurred (represented by an **RTRCallbackWarningCode** constant). This callback can also be used to interrupt processing: return TRUE if you wish to terminate the current operation, FALSE to continue.

textOrientationDetectedCallback

The callback informing you when the image orientation is detected. The *angle* parameter can take values of 0, 90, 180, and 270, and means the angle on which the image should be rotated to get normal orientation.

error

The error callback.

Return values

The method returns an array of data fields, represented by **RTRTextBlock** objects.

MobileImaging Core API

RTRCoreAPICropOperation protocol

An operation for image crop. The crop is performed on the image taking into account the orientation that is stored in certain metadata of the corresponding **UIImage** object (see the **UIImageOrientation** for more information).

This operation not only crops the image but also applies perspective distortion if needed.

Use the **applyToImage:** method of the **RTRCoreAPIImageOperation** protocol to apply the operation to the image.

Properties

Name	Type	Description
documentBoundary	NSArray<NSValue*>*, read-only	[in] The detected document boundary. Currently the result is always returned as the four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left. Get the **NSValue.CGPointValue** property to obtain point coordinates as a **CGPoint**.
documentSize	CGSize	[in, optional] the document size in millimeters.
imageResolution	CGSize	[out] Image resolution as calculated from image size and physical page size.

RTRCoreAPIDetectDocumentBoundaryOperation protocol

An operation for image boundaries detection. Use the **applyToImage** method of the **RTRCoreAPIImageOperation** protocol to apply the operation to the image.

Properties

Name	Type	Description
areaOfInterest	CGPoint	Area of interest for the operation.
mode	**RTRDetectDocumentBoundaryMode**	Document boundary detection mode. The mode influences on the crop speed and accuracy.
documentSize	CGSize	[in, out] The document size in millimeters.
documentBoundary	NSArray<NSValue*>*, read-only	[out] The detected document boundary. Currently the result is always returned as the four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left.

Name	Type	Description
		Get the **NSValue.CGPointValue** property to obtain point coordinates as a **CGPoint**.

RTRCoreAPIExportOperation protocol

Export operation protocol.

Images are added to the export operation as pages. Some export operations support adding multiple pages and some do not. Export operations should be properly closed with the **close** method to ensure that all required content has been written to the output stream.

Methods

Name	Description
- **addPageWithImage:**	Adds the page to the export target.
- **close:**	Call this method to ensure that all required content has been written to the output stream.

addPageWithImage: method of the RTRCoreAPIExportOperation protocol

Adds the page to the export target.

```
- (BOOL)addPageWithImage:(UIImage*)page;
```

Parameters

page

The page that must be added to the export target.

close: method of the RTRCoreAPIExportOperation protocol

This method is called to ensure that all required content has been written to the output stream.

```
- (BOOL)close;
```

RTRCoreAPIExportToPngOperation protocol

An operation for image export into PNG format. Inherits from the **RTRCoreAPIExportOperation** protocol. Use the **addPageWithImage:** method of the **RTRCoreAPIExportOperation** protocol to export the image.

Properties

Name	Type	Description
imageResolution	CGSize	Image resolution in EXIF. The default value of this property is 0.

RTRCoreAPIExportToJpgOperation protocol

An operation for image export into JPG format. Inherits from the **RTRCoreAPIExportOperation** protocol. Use the **addPageWithImage:** method of the **RTRCoreAPIExportOperation** protocol to export the image.

Properties

Name	Type	Description
imageResolution	CGSize	Image resolution in EXIF. The default value of this property is 0.
compression	**RTRCoreAPIExportCompressionLevel**	Page compression rate. This property should be tuned depending on the image resolution and the font size.

RTRCoreAPIExportToJpeg2000Operation protocol

An operation for image export into JPG 2000 format. Inherits from the **RTRCoreAPIExportOperation** protocol. Use the **addPageWithImage:** method of the **RTRCoreAPIExportOperation** protocol to export the image.

Properties

Name	Type	Description
imageResolution	CGSize	Image resolution in EXIF. The default value of this property is 0.
compression	RTRCoreAPIExportCompressionLevel	Page compression rate. This property should be tuned depending on the image resolution and the font size.

RTRCoreAPIExportToPdfOperation protocol

An operation for image export into PDF format. Inherits from the **RTRCoreAPIExportOperation** protocol. Use the **addPageWithImage:** method of the **RTRCoreAPIExportOperation** protocol to export the image.

Properties

Name	Type	Description
compressionType	RTRCoreAPIPdfExportCompressionType	The type of compression for PDF files depending on the base image format. The default value of this property is RTRCoreAPIPdfExportJpgCompression.
compression	RTRCoreAPIExportCompressionLevel	Page compression rate. This property should be tuned depending on the image resolution and the font size.
pageSize	CGSize	Page width in points (1/72 per inch). The page size of A4 is 595x842. If the value of this property is CGSizeZero, the page size is the same as the size of the image in pixels. The default value of this property is CGSizeZero.

RTRCoreAPIImage protocol

The protocol stores the captured image. All operations are applied to the image loaded to this format.

```
- (UIImage*)UIImage;
```

RTRCoreAPIImageOperation protocol

The image operation protocol. Inherits from the **RTRCoreAPIOperation** protocol.

After the image is captured a sequence of specified operations is applied to it. The operations can modify the image, then the returned result is the modified image, fill the [out] parameters of the operation or combine these two behavior types.

Methods

Name	Description
- **applyToImage:**	Applies chosen operation to the image.

applyToImage: method of the RTRCoreAPIOperation protocol

Applies chosen operation to the image.

```
- (BOOL)applyToImage:(id<RTRCoreAPIImage>)image;
```

Parameters

image

The image represented by an **RTRCoreAPIImage** object to which the operation is to by applied.

RTRCoreAPIOperation protocol

The base background operation protocol, inherited by the **RTRCoreAPIImageOperation** protocol.

Properties

Name	Type	Description
error	NSError*	The error that has occurred.

RTRCoreAPIQualityAssessmentForOCROperation protocol

Note: *This is a technology preview feature. The functionality will be improved and completed in future versions.*

An operation for image quality assessment for OCR. Use the **applyToImage:** method of the **RTRCoreAPIImageOperation** protocol to apply the operation to the image.

Properties

Name	Type	Description
documentBoundary	NSArray<NSValue*>*, read-only	[out] The detected document boundary. Currently the result is always returned as the four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left. Get the **NSValue.CGPointValue** property to obtain point coordinates as a **CGPoint**.
qualityAssessmentBlocks	NSArray<**RTRQualityAssessmentForOCRBlock***>*	[out] The quality assessment blocks.

RTRCoreAPIRotateOperation protocol

An operation for rotating the image to a specified angle. Use the **applyToImage:** method of the **RTRCoreAPIImageOperation** protocol to apply the operation to the image.

Properties

Name	Type	Description
angle	NSInteger	The angle in degrees. Available values of the angle: 0, 90, 180, 270.

RTRImagingCoreAPI protocol

Provides access to low-level single image core API functions for current thread. Inherits from the **RTRCoreAPI** protocol. Should be used on the same thread on which it was created. Multiple objects can be created on different threads and used concurrently. All methods are synchronous (blocking) and should not be used on UI thread. Intended for advanced users.

Methods

Name	Description
- **loadImage:error:**	This method loads the image object into the internal format.
- **createDetectDocumentBoundaryOperation**	Creates an operation for document boundary detection.
- **createQualityAssessmentForOCROperation**	Creates an operation for image quality assessment for OCR.
- **createCropOperation**	Creates an operation for image crop.
- **createRotateOperation**	Creates an operation for rotating the image.
- **createExportToPngOperation:**	Creates an operation for exporting image to PNG format.
- **createExportToJpgOperation:**	Creates an operation for exporting image to JPG format.
- **createExportToJpg2000Operation:**	Creates an operation for exporting image to JPG 2000 format.
- **createExportToPdfOperation:**	Creates an operation for exporting image to PDF format.

createCropOperation method of the RTRImagingCoreAPI protocol

Creates an operation for image crop.

```
- (id<RTRCoreAPICropOperation>)createCropOperation;
```

Return values

The method returns an instance implementing the **RTRCoreAPICropOperation** protocol. Use the **applyToImage:** method of the **RTRCoreAPIImageOperation** protocol to apply the operation to the image.

createDetectDocumentBoundaryOperation method of the RTRImagingCoreAPI protocol

Creates an operation for document boundary detection.

```
- (id<RTRCoreAPIDetectDocumentBoundaryOperation>)
createDetectDocumentBoundaryOperation;
```

Return values

The method returns an instance implementing the **RTRCoreAPIDetectDocumentBoundaryOperation** protocol. Use the **applyToImage:** method of the **RTRCoreAPIImageOperation** protocol to apply the operation to the image.

createExportToPngOperation: method of the RTRImagingCoreAPI protocol

Creates an operation for exporting image to PNG format.

```
- (id<RTRCoreAPIExportToPngOperation>)createExportToPngOperation:
(id<RTROutputStream>)outputStream;
```

Parameters

outputStream

The output stream for export.

Return values

The method returns an instance implementing the **RTRCoreAPIExportToPngOperation** protocol. Use the **addPageWithImage:** method of the **RTRCoreAPIExportOperation** protocol to export the image.

createExportToJpgOperation: method of the RTRImagingCoreAPI protocol

Creates an operation for exporting image to JPG format.

```
- (id<RTRCoreAPIExportToJpgOperation>)createExportToJpgOperation:
(id<RTROutputStream>)outputStream;
```

Parameters

outputStream

The output stream for export.

Return values

The method returns an instance implementing the **RTRCoreAPIExportToJpgOperation** protocol. Use the **addPageWithImage:** method of the **RTRCoreAPIExportOperation** protocol to export the image.

createExportToJpeg2000Operation: method of the RTRImagingCoreAPI protocol

Creates an operation for exporting image to JPG 2000 format.

```
- (id<RTRCoreAPIExportToJpg2000Operation>)createExportToJpg2000Operation:
(id<RTROutputStream>)outputStream;
```

Parameters

outputStream

The output stream for export.

Return values

The method returns an instance implementing the **RTRCoreAPIExportToJpeg2000Operation** protocol. Use the **addPageWithImage:** method of the **RTRCoreAPIExportOperation** protocol to export the image.

createExportToPdfOperation: method of the RTRImagingCoreAPI protocol

Creates an operation for exporting image to PDF format.

```
- (id<RTRCoreAPIExportToPdfOperation>)createExportToPdfOperation:
(id<RTROutputStream>)outputStream;
```

Parameters

outputStream

The output stream for export.

Return values

The method returns an instance implementing the **RTRCoreAPIExportToPdfOperation** protocol. Use the **addPageWithImage:** method of the **RTRCoreAPIExportOperation** protocol to export the image.

createQualityAssessmentForOCROperation method of the RTRImagingCoreAPI protocol

Note: *This is a technology preview feature. The functionality will be improved and completed in future versions.*

Creates an operation for image quality assessment for OCR.

```
- (id<RTRCoreAPIQualityAssessmentForOCROperation>)
createQualityAssessmentForOCROperation;
```

Return values

The method returns an instance implementing the **RTRCoreAPIQualityAssessmentForOCROperation** protocol. Use the **applyToImage:** method of the **RTRCoreAPIImageOperation** protocol to apply the operation to the image.

createRotateOperation method of the RTRImagingCoreAPI protocol

Creates an operation for rotating the image.

```
- (id<RTRCoreAPIRotateOperation>)createRotateOperation;
```

Return values

The method returns an instance implementing the **RTRCoreAPIRotateOperation** protocol. Use the **applyToImage:** method of the **RTRCoreAPIImageOperation** protocol to apply the operation to the image.

loadImage:error: method of the RTRImagingCoreAPI protocol

This method loads the image object into the internal format.

```
- (id<RTRCoreAPIImage>)loadImage:(UIImage*)image;
                        error: (NSError**)error;
```

Parameters

image

The image to be loaded.

error

The error that has occurred.

Return values

The method returns an instance implementing the **RTRCoreAPIImage** protocol.

RTROutputStream protocol

This output stream protocol is used for exporting the processed image. You can use one of three interfaces implementing the protocol. The protocol and its methods can also be implemented on the client side.

Methods

Name	Description
- **writeData:**	Writes the provided data to the specified destination.

writeData: method of the RTROutputStream protocol

Writes the provided data to the specified destination.

```
- (BOOL)writeData:(NSData*)data;
```

Parameters

data

The data that must be written to stream.

RTRQualityAssessmentForOCRBlock protocol

Note: *This is a technology preview feature. The functionality will be improved and completed in future versions.*

The block for the quality assessment for OCR.

Properties

Name	Type	Description
type	**RTRQualityAssessmentForOCRBlockType**	Type of quality assessment block.

Name	Type	Description
quality	NSInteger	Value from 0 to 100 that indicates suitability of the text for OCR.
rect	CGRect	Block rectangle.

RTRMemoryOutputStream class

This output stream class conforms to the **RTROutputStream** protocol and is used for exporting the processed image as data to memory.

Properties

Name	Type	Description
data	NSData*	The data exported to memory.

Methods

Name	Description
- **writeData:**	Writes the provided data to the specified destination.

writeData: method of the RTRMemoryOutputStream class

Writes the provided data to the specified destination.

```
- (BOOL)writeData:(NSData*)data;
```

Parameters

data

The data that must be written to stream.

RTRFileOutputStream class

This output stream class conforms to the **RTROutputStream** protocol and is used for exporting the processed image as data to a specified file.

Properties

Name	Type	Description
filePath	NSString*	The address of the file to which the exported data will be written.
error	NSError*	The error that has occurred. In case no errors occurred, this property is **nil**.

Methods

Name	Description
- **writeData:**	Writes the provided data to the specified destination.
- **initWithFilePath:**	Returns an initialized output stream for writing to the specified 'filePath'.

initWithFilePath: method of the RTRFileOutputStream class

Returns an initialized output stream for writing to the specified 'filePath'.

```
- (instancetype)initWithFilePath:(NSString*)filePath;
```

Parameters

filePath

The address of the file to which the exported data will be written.

Return values

The method returns an instance of the **RTRFileOutputStream** object, or **nil** if object creation failed.

writeData: method of the RTRFileOutputStream class

Writes the provided data to the specified destination.

```
- (BOOL)writeData:(NSData*)data;
```

Parameters

data

The data that must be written to stream.

RTROutputStream class

This output stream class conforms to the **RTROutputStream** protocol and is used for exporting the processed image as data to the stream.

Properties

Name	Type	Description
outputStream	NSOutputStream*	The output stream to which the exported data will be written.
error	NSError*	The error that has occurred. In case no errors occurred, this property is **nil**.

Methods

Name	Description
- **writeData:**	Writes the provided data to the specified destination.
- **initWithOutputStream:**	Returns an initialized output stream for writing to the specified output stream.

initWithOutputStream: method of the RTROutputStream class

Returns an initialized output stream for writing to the specified output stream. This output stream must be opened before using.

```
- (instancetype)initWithOutputStream:(NSOutputStream*)outputStream;
```

Parameters

outputStream

The output stream to which the exported data will be written.

Return values

The method returns an instance of the **RTROutputStream** object, or **nil** if object creation failed.

writeData: method of the RTROutputStream class

Writes the provided data to the specified destination.

```
- (BOOL)writeData:(NSData*)data;
```

Parameters

data

The data that must be written to stream.

RTRDetectDocumentBoundaryMode enumeration

The type of document boundary detection and crop.

```
typedef NS_ENUM(NSUInteger, RTRDetectDocumentBoundaryMode) {
    RTRDetectDocumentBoundaryModeDefault,
    RTRDetectDocumentBoundaryModeFast,
};
```

Constants

Name	Description
RTRDetectDocumentBoundaryModeDefault	[Default] Balanced mode, that combines optimal processing speed and high quality.
RTRDetectDocumentBoundaryModeFast	Fast mode, that signifies processing speed.

RTRProgressCallbackBlock

This is a type definition for approximate progress of the operation block that will be passed to the methods of data capturing or text recognition as an **onProgress** parameter.

```
typedef BOOL (RTRProgressCallbackBlock)(NSInteger percentage,
RTRCallbackWarningCode warningCode);
```

Parameters

percentage

The approximate percentage of the work currently done. This parameter is in the range from 0 to 100.

warningCode

A warning that occurred during processing represented by an **RTRCallbackWarningCode** constant.

Return values

The block must return NO if the recognition process should be terminated, TRUE otherwise.

RTRTextOrientationDetectedBlock

This is a type definition for a block informing if the image orientation is detected. The block is passed to the methods of data capturing or text recognition as an **onTextOrientationDetected** parameter.

```
typedef void (RTRTextOrientationDetectedBlock)(NSInteger angle);
```

Parameters

angle

The angle on which the image should be rotated to get normal orientation. The angle parameter can take values of 0, 90, 180, and 270.

Supplementary API

RTRImageCaptureStatus

The current status of the image capture. Returned value is intended for the UI feedback.

Properties

Name	Type	Description
motionVector	CGVector	The vector that indicates the image position shifting in comparison to the previous state.
relativeQuality	NSInteger	The value from internal image quality scale. Larger value means better image quality. The minimum value is 0. ***Note:*** *This API is available only in the extended version of the library. For correct quality comparison the image should represent the document at the same scene. If the background changes at some images, parameter values will not represent appropriate for comparison data.*
documentBoundary	NSArray<NSValue*>*	The detected document boundary. Currently the result is always returned as the four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left. Get the **NSValue.CGPointValue** property to obtain point coordinates as a **CGPoint**.
qualityAssessmentForOCRBlocks	NSArray<**RTRQualityAssessmentForOCRBlock***>*	The quality assessment blocks.

RTRImageCaptureResult

The result of the image capture.

Properties

Name	Type	Description
image	UIImage*	The captured image.
documentBoundary	NSArray<NSValue*>*	The detected document boundary. Currently the result is

Name	Type	Description
		always returned as the four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left. Get the **NSValue.CGPointValue** property to obtain point coordinates as a **CGPoint**.
documentSize	CGSize	The document size (as specified in **setDocumentSize:**).
qualityAssessmentForOCRBlocks	NSArray<**RTRQualityAssessmentForOCRBlock***>*	The quality assessment blocks.

RTREngineSettings protocol

Additional settings for ABBYY Mobile Capture SDK engine. They apply to all processing scenarios.

Properties

Name	Type	Description
externalAssetsPath	NSString*	The additional path to search for framework data. The program will search for any resource file it needs first in the bundle root, then in the specified custom folder, each time looking in the corresponding subfolder. For example, it will try to locate a pattern file (*.rom) like this: 1) in **<bundle path>/Patterns** 2) in **<custom search path>/Patterns** 3) if the file is not found, an error will be returned

RTRCharInfo class

Extended information about the character formatting.

Important! *This class is reserved for future use.*

Properties

Name	Type	Description
backgroundColor	NSInteger, read-only	The color of the background. ***Note:*** *The **int** value is calculated from the RGB triplet using the formula: (**red value**) + (256 x **green value**) + (65536 x **blue value**), where **red value** is the first triplet component, **green value** is the second triplet component, **blue value** is the third triplet component. For example, the **int** value of the color white equals 16777215.*
foregroundColor	NSInteger, read-only	The color of the symbol.
quadrangle	NSArray<NSValue*>*, read-only	The four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left. Get the **NSValue.CGPointValue** property to obtain point coordinates as a **CGPoint**.
rect	CGRect, read-only	The bounding rectangle.

RTRDataField class

A recognized data field. Provides field contents, location and included data fields, if applicable.

The field can be compound, which means that it consists of several parts.

Note that a field may have several components — for example, it can contain two or more words. Component details are available from the **components** array. Each element of this array is an **RTRDataField** object with its own **text** property (for example, a word) and **quadrangle** property (the bounding quadrangle of this component). The field's **text** property contains its entire text, and the field's

quadrangle property represents the whole area of a field: this quadrangle encloses the quadrangles of all components.

The **components** array always contains at least one element. When a field contains only one component, the **text** and **quadrangle** properties of the field and this component are identical.

Properties

Name	Type	Description
id	NSString*, read-only	The internal field identifier. Can be one of the predefined fields listed in Data Capture Profiles or the custom field identifier that you specified when adding the field in **RTRDataSchemeBuilder**. May be **nil** in case the **component** is a part of a compound component.
name	NSString*, read-only	The name of the field as seen in the document or specified in the custom data capture profile.
quadrangle	NSArray<NSValue*>*, read-only	The four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left. Get the **NSValue.CGPointValue** property to obtain point coordinates as a **CGPoint**.
text	NSString*, read-only	The recognized field contents.
components	NSArray<RTRDataField*>*, read-only	Field components represented by **RTRDataField** objects. If the field has only one component, this array contains one element.

RTRDataFieldInfo class

A data field, that can be detected on the document, corresponding to current data scheme. Provides field id, name and included data fields, if applicable.

The field can be compound, which means that it consists of several parts, i.e. *Address* field can consist of such fields as *City*, *Street*, etc. Component details are available from the **components** array.

Properties

Name	Type	Description
id	NSString*, read-only	The internal field identifier.
name	NSString*, read-only	The name of the field as seen in the document or specified in the custom data capture profile.
components	NSArray<RTRDataFieldInfo*>*, read-only	Field components represented by **RTRDataFieldInfo** objects. If the field has only one component, this array contains one element. This property may be **nil**, if the field is not compound.

RTRDataScheme class

Information on the data scheme applied to the recognized frame. This class provides access to the identifier and human-readable name of the scheme, as well as a full list of fields, which can be detected on a document applying this data scheme.

Properties

Name	Type	Description
id	NSString, read-only	The internal scheme identifier. Can be one of the predefined data schemes listed in Data Capture Profiles or the custom scheme identifier that you specified when creating the scheme in the **RTRDataCaptureProfileBuilder**.
name	NSString, read-only	The name of the scheme. If you are using a custom data capture profile, this is the same name you specified when creating the scheme in the **RTRDataCaptureProfileBuilder**.
fields	NSArray<RTRDataFieldInfo*>*, read-only	Array of **RTRDataFieldInfo** objects, representing all fields that can be detected for this data scheme.

RTRExtendedSettings class

Extended service configuration settings. Intended for advanced users: most common scenarios will work with the default settings.

Properties

Name	Type	Description
CJKVerticalTextEnabled	BOOL	Enables or disables vertical writing direction for Chinese, Japanese, and Korean languages. The default value of this property is NO (disabled).
frameMergingEnabled	BOOL	Enables or disables merging of recognition results. Frame merging is one of the key features of Mobile Capture SDK, which improves recognition accuracy. The default value of this property is YES (enabled).
processingThreadsCount	NSInteger	The number of processing threads to be used by the service. Up to 16 threads are allowed. Set to 0 to determine the number of threads automatically. The default value of this property is 0.

RTRTextLine class

A line of recognized text; the location and additional information are also available.

Properties

Name	Type	Description
charsInfo	NSArray*, read-only	Extended characters' information as an array of RTRCharInfo objects. ***Important!*** *This property is reserved for future use.*
quadrangle	NSArray<NSValue*>*, read-only	The four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left.

Name	Type	Description
		Get the **NSValue.CGPointValue** property to obtain point coordinates as a **CGPoint**.
rect	CGRect, read-only	The bounding rectangle.
text	NSString*, read-only	The recognized text.

RTRTextBlock class

A block of recognized text, containing an array of text lines found in one text area on the image.

Properties

Name	Type	Description
textLines	NSArray*, read-only	The array of **RTRTextLine** objects representing the lines of recognized text.

RTRQualityAssessmentForOCRBlockType enumeration

Note: *This is a technology preview feature. The functionality will be improved and completed in future versions.*

Type of the block for quality assessment for OCR.

```
typedef NS_ENUM(NSUInteger, RTRQualityAssessmentForOCRBlockType) {
    RTRQualityAssessmentForOCRTextBlock,
    RTRQualityAssessmentForOCRUnknownBlock,
};
```

Constants

Name	Description
RTRQualityAssessmentForOCRTextBlock	The text detected.

Name	Description
RTRQualityAssessmentForOCRUnknownBlock	The unknown type.

RTRCoreAPIExportCompressionLevel enumeration

The uniform image compression scale for lossy formats.

```
typedef NS_ENUM(NSUInteger, RTRCoreAPIExportCompressionLevel) {
    RTRCoreAPIExportCompressionLowLevel,
    RTRCoreAPIExportCompressionNormalLevel,
    RTRCoreAPIExportCompressionHighLevel,
    RTRCoreAPIExportCompressionExtraHighLevel,
};
```

Constants

Name	Description
RTRCoreAPIExportCompressionLowLevel	The lowest compression rate, that still might have any noticeable effect on recognition of small text.
RTRCoreAPIExportCompressionNormalLevel	[Default] Balanced trade-off between compression and quality. Good safety margin.
RTRCoreAPIExportCompressionHighLevel	More compression, less safety margin. Might perform poorly with small text, but generally still ok.
RTRCoreAPIExportCompressionExtraHighLevel	The maximum recommended compression rate. Will perform poorly with small text. Advisable only for relatively large text and very slow networks.

RTRCoreAPIPdfExportCompressionType enumeration

The type of compression for PDF files depending on the base image format.

```
typedef NS_ENUM(NSUInteger, RTRCoreAPIPdfExportCompressionType) {
    RTRCoreAPIPdfExportJpgCompression,
    RTRCoreAPIPdfExportJpeg2000Compression,
};
```

Constants

Name	Description
RTRCoreAPIPdfExportJpgCompression	The image compression type for PDF files based on the JPG format.
RTRCoreAPIPdfExportJpeg2000Compression	The image compression type for PDF files based on the JPG 2000 format.

RTRCallbackWarningCode enumeration

A warning that occurred during processing.

```
typedef NS_ENUM(NSInteger, RTRCallbackWarningCode) {
    RTRCallbackWarningNoWarning,
    RTRCallbackWarningRecognitionIsSlow,
    RTRCallbackWarningProbablyLowQualityImage,
    RTRCallbackWarningProbablyWrongLanguage,
    RTRCallbackWarningWrongLanguage,
    RTRCallbackWarningTextTooSmall
};
```

Constants

Name	Description
RTRCallbackWarningProbablyLowQualityImage	The image quality (contrast, resolution) may not be good enough for accurate results.
RTRCallbackWarningProbablyWrongLanguage	The recognition language may be specified incorrectly.
RTRCallbackWarningRecognitionIsSlow	Recognition takes too much time. Check if there is some problem.
RTRCallbackWarningTextTooSmall	The text is too small. Advise the end user to move the camera closer or zoom in.
RTRCallbackWarningWrongLanguage	The recognition language is specified incorrectly.

User Interface API Reference

This section describes provided Objective-C API for user interface implementation.

AUICaptureScenario interface

Provides access to the capture scenario management. Inherited by the **AUIImageCaptureScenario** and **AUIMultiPageImageCaptureScenario** interfaces.

Requires a delegate that conforms to the **AUICaptureScenarioDelegate** protocol.

Properties

Name	Type	Description
delegate	**AUICaptureScenarioDelegate**	Delegate object for receiving results, status information, warnings and errors.

Methods

Name	Description
+ **supportedCameraResolutionsForDevice**	Returns the camera resolutions from the **AUICameraResolution** enumeration, supported by the device.
- **cancel:**	Stops the scenario.

cancel method of the AUICaptureScenario interface

Stops the scenario.

```
- (void)cancel;
```

supportedCameraResolutionsForDevice

Returns the camera resolutions from the **AUICameraResolution** enumeration, supported by the device.

```
+ (NSArray<NSNumber*>*)supportedCameraResolutionsForDevice;
```

Return values

The method returns an array storing the resolutions supported by the device camera.

AUICaptureScenarioDelegate protocol

Delegate object for receiving results, status information, warnings and errors during the capture scenario. Inherited by the **AUIImageCaptureScenarioDelegate** and **AUIMultiPageImageCaptureScenarioDelegate** protocols.

The method is to be implemented on the client side.

Methods

Name	Description
- **captureScenarioDidCancel:**	Notifies the delegate that performing scenario was canceled.

captureScenarioDidCancel: method

Notifies the delegate that the capture scenario was canceled.

```
- (void)captureScenarioDidCancel:(AUICaptureScenario)scenario;
```

Parameters

scenario

Scenario that was canceled.

AUICaptureController interface

Configures the work with camera preview. Provides properties for specifying capture scenario and customizing the user interface appearance.

If created without initial view controller, pushes camera controller automatically.

Properties

Name	Type	Description
localizedStringsBundle	NSBundle*	Bundle storing the localized source strings for the interface. ***Note:*** *By default all the strings are in English. To change the interface language according to your needs, please localize the source strings in the* ***AbbyyUI.strings*** *file located in the* ***AbbyyUI.framework****.*
captureScenario	AUICaptureScenario*	Capture scenario that will be performed on the camera preview. ***Note:*** *Only image capture scenario is currently supported.*
cameraOverlayView	UIView*	Camera overlay view. Shows over camera view. To define in storyboard it must be placed outside **AUICaptureController**.
flashButton	UIButton*	Camera torch toggle button. Selected state is stored in **NSUserDefaults**.
captureButton	UIButton*	Button for capturing image immediately.
closeButton	UIButton*	Button for closing the camera preview.
cameraSettings	AUICameraSettings	Camera settings for current capture session. If the capture session is not started, this property is **nil**.
theme	AUITheme	Color theme of the user interface, defining colors of the text, buttons' background, etc. The theme can be light or dark.

Name	Type	Description
customColor	UIColor*	Custom color of the button for manual taking photo.
paused	BOOL	Indicates if the scenario is in a paused state.

Methods

Name	Description
- **setPaused:animated:**	Sets or turns off the paused state.
- **pushCameraControllerAnimated:animated:**	Pushes the camera controller. ***Note:*** *This method is used in case camera controller is not the only controller inside the* ***AUICaptureController*** *object and so is not pushed automatically.*

setPaused:animated: method of the AUICaptureController interface

Sets or turns off the paused state. During the pause state the interface elements disappear and the background is turning blurred.

```
- (void)setPaused:(BOOL)paused animated:(BOOL)animated;
```

Parameters

paused

Defines if the scenario is set to the paused state. Set YES for pausing, NO otherwise.

animated

Defines if the interface elements vanishing and background blurring will be performed slower when the pause state is set. Pass YES to show the animation, NO otherwise.

pushCameraControllerAnimated: method of the AUICaptureController interface

Pushes the camera controller.

```
- (void)pushCameraController:(BOOL)animated;
```

Parameters

animated

Defines if the interface elements vanishing and background blurring will be performed slower on the start. Pass YES to show the animation, NO otherwise.

AUIImageCaptureResult interface

Captured image with detected document boundaries. This result can be received in both real-time capturing from camera preview and from an immediate manual capture scenario.

Properties

Name	Type	Description
image	UIImage*	The captured image. ***Note:*** *In case* ***cropEnabled*** *property of the* ***AUIImageCaptureScenario*** *interface is set to TRUE, the property stores the result of the crop operation. Otherwise the full captured image is saved to this property.*
documentBoundary	NSArray<NSValue*>*	The detected document boundary, returned as the four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left. Get the **NSValue.CGPointValue** property to obtain point coordinates as a **CGPoint**. You can edit document boundaries, changing values of the points' coordinates. ***Important!*** *In case* ***cropEnabled*** *property of the* ***AUIImageCaptureScenario*** *interface is set to TRUE, the property is* ***nil****.*

AUIImageCaptureScenario interface

Provides access to the image capture scenario settings. Inherits from the **AUICaptureScenario** interface.

Requires a delegate that conforms to the **AUIImageCaptureScenarioDelegate** protocol.

Properties

Name	Type	Description
delegate	**AUIImageCaptureScenarioDelegate**	Delegate object for receiving results, status information, warnings and errors.
active	BOOL	Denotes if the automatic capturing from the video stream is enabled. If FALSE, only manual capturing is available during the scenario. This property is TRUE by default and is set to FALSE automatically after image is captured.
minimumDocumentToViewRatio	CGFloat	The [0..1] ratio of the minimum document area relative to the whole frame area. The default value is 0.15.
aspectRatioMin*	CGFloat	Lower limit of the document's aspect ratio. This property is used in pair with the **aspectRatioMax**, defining an interval of acceptable aspect ratio values of the document to be captured. Setting aspect ratio will help to improve boundary detection accuracy. If only **aspectRatioMax** is set, **aspectRatioMin** will be set to 1. By default the property is set to 0 (aspect ratio is not set).
aspectRatioMax*	CGFloat	Upper limit of document's aspect ratio. This property is used in pair with the **aspectRatioMin**, defining an interval of acceptable aspect ratio values of the

Name	Type	Description
		document to be captured. Setting aspect ratio will help to improve boundary detection accuracy. If only **aspectRatioMin** is set, **aspectRatioMax** will be set to CGFLOAT_MAX. By default the property is set to 0 (aspect ratio is not set).
documentSize	AUIDocumentSize	Document physical size. The default value is AUIDocumentSizeAny.
cropEnabled	BOOL	Indicates if the captured image should be cropped. If this parameter is set to TRUE, AUIImageCaptureResult.image parameter will contain cropped image and the AUIImageCaptureResult.documentBoundary parameter will be **nil**. The default value is FALSE.

Document aspect ratio setting is intended to specify the exact proportions of the target document, which will increase capture accuracy.

Notes:

- *Aspect ratio detection requires the mobile device to be placed strictly horizontally over the document during image capture. In case the mobile device is tilted, camera will capture distorted image and the document aspect ratio may be detected incorrectly.*
- *The value of aspect ratio is calculated by division of the longer side to the shorter side and is expected to be greater than or equal to 1 (or 0 if not set). If neither **aspectRatioMin** nor **aspectRatioMin** are set, the values will be calculated from the **documentSize** setting.*

Methods

Name	Description
- initWithEngine:	Connects the **AUIImageCaptureScenario** object with the RTREngine object that implements

Name	Description
	recognition.
- captureImageManually	Captures image immediately.

initWithEngine: method of the AUIImageCaptureScenario interface

Connects the **AUIImageCaptureScenario** object with the **RTREngine** object that implements capturing and recognition.

```
- (instancetype)initWithEngine:(RTREngine*)engine;
```

Parameters

engine

An instance of the corresponding to the scenario **RTREngine** object.

Return values

The method returns an instance of the **AUIImageCaptureScenario** interface initialized with the **RTREngine** object.

captureImageManually

Captures image immediately.

```
- (void)captureImageManually;
```

AUIImageCaptureScenarioDelegate protocol

Delegate object for receiving results, status information, warnings and errors during the image capture scenario.

Inherits from the **AUICaptureScenarioDelegate** protocol. The methods are to be implemented on the client side.

Methods

Name	Description
- captureScenario:didCaptureImageWithResult:	Notifies the delegate that an image was captured, delivers the result object.
- captureScenario:didFailWithError:	Notifies the delegate that an error has occurred while capturing.
- captureScenarioDidCancel:	Notifies the delegate that the capture scenario was canceled.

captureScenario:didCaptureImageWithResult: method

Notifies the delegate that an image was captured, delivers the result object.

```
- (void)captureScenario:(AUIImageCaptureScenario*)captureScenario
didCaptureImageWithResult:(AUIImageCaptureResult*)result;
```

Parameters

captureScenario

AUIImageCaptureScenario object, corresponding to the AUIImageCaptureScenarioDelegate delegate that should be notified.

result

The captured image as an AUIImageCaptureResult object with detected document boundaries.

captureScenario:didFailWithError: method

Notifies the delegate that an error has occurred while capturing.

```
- (void)captureScenario:(AUICaptureScenario*)scenario didFailWithError:
(NSError*)error;
```

Parameters

captureScenario

AUIImageCaptureScenario object, corresponding to the AUIImageCaptureScenarioDelegate delegate that should be notified about the error.

error

The error that has occurred.

captureScenarioDidCancel: method

Notifies the delegate that the capture scenario was canceled.

```
- (void)captureScenarioDidCancel:(AUICaptureScenario)scenario;
```

Parameters

scenario

Scenario that was canceled.

AUIImageCaptureSettings protocol

Provides access to the settings of the multipage image capture scenario for the certain page.

An instance of this protocol is passed as a parameter to the **captureScenario:onConfigureImageCaptureSettings:** method of the **AUIMultiPageCaptureSettings** protocol.

Properties

Name	Type	Description
minimumDocumentToViewRatio	CGFloat	The [0..1] ratio of the minimum document area relative to the whole frame area. The default value is 0.15.
aspectRatioMin*	CGFloat	Lower limit of the document's aspect ratio. This property is used in pair with the **aspectRatioMax**, defining an interval of acceptable aspect ratio values of the document to be captured. Setting aspect ratio will help to improve boundary detection accuracy. If only **aspectRatioMax** is set, **aspectRatioMin** will be set to 1. By default the property is set to 0 (aspect ratio is not set).

Name	Type	Description
aspectRatioMax*	CGFloat	Upper limit of document's aspect ratio. This property is used in pair with the **aspectRatioMin**, defining an interval of acceptable aspect ratio values of the document to be captured. Setting aspect ratio will help to improve boundary detection accuracy. If only **aspectRatioMin** is set, **aspectRatioMax** will be set to CGFLOAT_MAX. By default the property is set to 0 (aspect ratio is not set).
documentSize	**AUIDocumentSize**	Document physical size. The default value is **AUIDocumentSizeAny**.
imageFromGalleryMaxSize	**NSInteger**	Maximum available size of an image, loaded from the gallery. The size is defined as the length of the largest side of an image (in pixels). The default value is 4096.

Document aspect ratio setting is intended to specify the exact proportions of the target document, which will increase capture accuracy.

Notes:

- *Aspect ratio detection requires the mobile device to be placed strictly horizontally over the document during image capture. In case the mobile device is tilted, camera will capture distorted image and the document aspect ratio may be detected incorrectly.*
- *The value of aspect ratio is calculated by division of the longer side to the shorter side and is expected to be greater than or equal to 1 (or 0 if not set). If neither **aspectRatioMin** nor **aspectRatioMin** are set, the values will be calculated from the **documentSize** setting.*

AUIMultiPageCaptureSettings protocol

Manages the image capture process of a certain page.

Methods

Name	Description
- captureScenario:onConfigureImageCaptureSettings:	Provides image capture processing settings to the page during capture. The page with specified index will be captured according to these settings. This method should be called from the image capture scenario with settings, that will be applied to the specified page capture process, represented by an AUIImageCaptureSettings object.

captureScenario:onConfigureImageCaptureSettings: method

Provides image capture processing settings to the page during capture. The page with specified index will be captured according to these settings.

```
- (void)captureScenario:(AUIMultiPageImageCaptureScenario*)captureScenario
    onConfigureImageCaptureSettings:(id<AUIImageCaptureSettings>)settings
    forPageAtIndex:(NSUInteger)index;
```

Parameters

captureScenario

The multipage image capture scenario, represented by the AUIMultiPageImageCaptureScenario object. The image capture will be performed according to the settings during this scenario.

settings

The AUIImageCaptureSettings object storing all the setting to be applied.

index

The number of the processing page.

AUIMultiPageImageCaptureResult protocol

Provides access to the result of a multipage document capture.

Methods

Name	Description
- clearWithError	Removes all the pages from the storage.
- deleteWithId:	Removes the page with specified identifier.
- loadBoundaryWithId:	Returns the document boundary for the specified page.
- loadImageWithId:	Returns the captured and cropped image for the specified page.
- loadOriginalImageWithId:	Returns the original captured image for the specified page.
- loadThumbnailWithId:	Returns the captured and cropped image's thumbnail for the specified page.
- pagesWithError:	Returns all the pages identifiers in the storage.

clearWithError: method

Removes all the pages from the storage.

```
- (BOOL)clearWithError:(NSError**)error;
```

Parameters

error

The error that has occurred. In case no errors occurred, this parameter is **nil**.

Return values

The method returns returns YES, if the pages were successfully deleted, and NO otherwise.

deleteWithId:error: method

Removes the page with specified identifier.

```
- (BOOL)deleteWithId:(AUIPageId)identifier error:(NSError**)error;
```

Parameters

identifier

AUIPageId object, storing the identifier of the page to be removed.

error

The error that has occurred. In case no errors occurred, this parameter is **nil**.

Return values

The method returns YES, if the page folder was successfully deleted, and NO otherwise.

loadBoundaryWithId:error: method

Returns the document boundary for the specified page.

```
- (nullable NSArray<NSValue*>*)loadBoundaryWithId:(AUIPageId)identifier
error:(NSError**)error;
```

Parameters

identifier

AUIPageId object, storing the identifier of the required page.

error

The error that has occurred. In case no errors occurred, this parameter is **nil**.

Return values

The method returns the four vertex points of the image bounding quadrangle. The vertices are indexed clockwise starting from the bottom left.

loadImageWithId:error: method

Returns the captured and cropped image for the specified page.

```
- (nullable UIImage*)loadImageWithId:(AUIPageId)identifier error:
(NSError**)error;
```

Parameters

identifier

AUIPageId object, storing the identifier of the required page.

error

The error that has occurred. In case no errors occurred, this parameter is **nil**.

Return values

The method returns the **UIImage** object, that represents the captured and cropped image or **nil** in case an error occurred.

loadOriginalImageWithId:error: method

Returns the original captured image for the specified page.

```
- (nullable UIImage*)loadOriginalImageWithId:(AUIPageId)identifier error:
(NSError**)error;
```

Parameters

identifier

AUIPageId object, storing the identifier of the required page.

error

The error that has occurred. In case no errors occurred, this parameter is **nil**.

Return values

The method returns the **UIImage** object, that represents the captured original image or **nil** in case an error occurred.

loadThumbnailWithId:error: method

Returns the captured image's thumbnail for the specified page.

```
- (nullable UIImage*)loadThumbnailWithId:(AUIPageId)identifier error:
(NSError**)error;
```

Parameters

identifier

AUIPageId object, storing the identifier of the required page.

error

The error that has occurred. In case no errors occurred, this parameter is **nil**.

Return values

The method returns the **UIImage** object, that represents the captured image's thumbnail or **nil** in case an error occurred.

pagesWithError:error: method

Returns all the pages identifiers in the storage.

```
- (nullable NSArray<AUIPageId>*)pagesWithError:(NSError**)error;
```

Parameters

error

The error that has occurred. In case no errors occurred, this parameter is **nil**.

Return values

The method returns an array of **AUIPageId** objects, storing the identifiers of all pages in the storage.

AUIMultiPageImageCaptureScenario interface

Provides access to the multipage image capture scenario management.

This interface helps to easily integrate the image capture technology into your application and manage the scenario when more than one image should be captured.

Inherits from the **AUICaptureScenario** interface.

Requires a delegate that conforms to the **AUIMultiPageImageCaptureScenarioDelegate** protocol.

Properties

Name	Type	Description
uiSettings	**AUIMultiPageUISettings**	Appearance settings of pages.
captureSettings	**AUIMultiPageCaptureSettings**	Additional multipage image capture scenario processing settings.

Name	Type	Description
active	BOOL	Defines if the automatic capture is enabled. if YES, the image will be captured automatically from the video stream. If NO, only manual capture will be available. After each capture this parameter is set to NO. This property is passed to the **setActive** method as a parameter. The default value is YES.
isShowPreviewEnabled	BOOL	Specifies if a preview of an image should be shown after capture. If YES, a preview is shown. If NO, the camera view is shown instead. The default value is NO.
requiredPageCount	NSUInteger	Total number of pages to be captured. Use this property to set the page-limitation mode of the image capture: • set to 0 to allow unlimited image capture. The batch of the result images can be saved or edited at any time • set to a positive value to set the exact number of images that should be captured. Images saving is enabled only when this number of images are been captured. If the **uiSettings** property is set to the default value, required number of images is shown as a tip at the bottom of the camera view and in the top left corner of the gallery view, i.e. "1 of 3", where "3" is the required number of images. The value of this property should be set before the start of the capture scenario. The default value is 0 (no pages limit).

Name	Type	Description
startAsEditorAtPageId	AUIPageId	Identifier of the page with previously captured image that will be displayed as a start page at the scenario beginning. This property may be **nil**. In this case at the scenario beginning the camera view will be shown. ***Note:*** *This property becomes available after some images are been captured and saved to the* ***AUIPageStorage*** *with corresponding* ***AUIPageId****. Before the first image capture scenario start this property is* ***nil****.* The default value is **nil**.
result	AUIMultiPageImageCaptureResult	Result of the multipage image capture scenario. Captured images can be retrieved by this property methods and used in further scenario, i.e. images demonstration in a view with all captured documents.

Methods

Name	Description
- initWithEngine:	Connects the **AUIMultiPageImageCaptureScenario** object with the RTREngine object that implements recognition.
- setActive:animated:	Sets on and off automatic image capture mode.

setActive:animated: method

Sets on and off the automatic capture mode.

```
- (void)setActive:(BOOL)active animated:(BOOL)animated;
```

Parameters

active

Defines if the automatic capture should be enabled. if YES, the image will be captured automatically from the video stream. If NO, only manual capture will be available.

animated

Defines if the capture area corners should appear with zoom animation on the automatic capture enabling. Pass YES to show the animation, NO otherwise.

initWithEngine: method

Connects the **AUIMultiPageImageCaptureScenario** object with the **RTREngine** object that implements capturing and recognition.

```
- (instancetype)initWithEngine:(RTREngine*)engine pageStorage:
(id<AUIPageStorage>)storage NS_DESIGNATED_INITIALIZER;


- (instancetype)initWithEngine:(RTREngine*)engine storagePath:(NSString*)
path error:(NSError**)error;
```

Parameters

engine

An instance of the corresponding to the scenario **RTREngine** object.

storage

Previously created **AUIPageStorage** object with images.

path

Absolute path to the folder with images that will be converted into the **AUIPageStorage** object.

error

The error that has occurred. In case no errors occurred, this parameter is **nil**.

Return values

The method returns an instance of the **AUIMultiPageImageCaptureScenario** interface initialized with the **RTREngine** object.

AUIMultiPageImageCaptureScenarioDelegate protocol

Delegate object for receiving results, status information, warnings and errors during the multipage image capture scenario.

Inherits from the **AUICaptureScenarioDelegate** protocol. The methods are to be implemented on the client side.

Methods

Name	Description
- captureScenario:didFinishWithResult:	Notifies the delegate that the images were captured successfully and user pressed "Done" button.
- captureScenario:onCloseWithResult:	Notifies the delegate that the capture process was interrupted by user.
- captureScenario:didFailWithError:	Notifies the delegate that an error has occurred.

captureScenario:didFinishWithResult: method

Notifies the delegate that the images were captured successfully and user pressed "Done" button.

```
- (void)captureScenario:(AUIMultiPageImageCaptureScenario*)captureScenario
    didFinishWithResult:(id<AUIMultiImageCaptureResult>)result;
```

Parameters

captureScenario

> **AUIMultiPageImageCaptureScenario** object, corresponding to the **AUIMultiPageImageCaptureScenarioDelegate** delegate that should be notified.

result

> The captured images as an **AUIMultiPageImageCaptureResult** object.

Note: *In case you do not want to save images in the storage after stopped or failed scenario, you can remove the images from the storage, using* ***clearWithError:*** *method.*

captureScenario:onCloseWithResult: method

Notifies the delegate that the capture process was interrupted by user.

```
- (void)captureScenario:(AUIMultiPageImageCaptureScenario*)captureScenario
    onCloseWithResult:(id<AUIMultiImageCaptureResult>)result;
```

Parameters

captureScenario

> **AUIMultiPageImageCaptureScenario** object, corresponding to the **AUIMultiPageImageCaptureScenarioDelegate** delegate that should be notified.

result

> If any images were captured, they are stored in the **AUIMultiPageImageCaptureResult** result object.

Note: *In case you do not want to save images in the storage after stopped or failed scenario, you can remove the images from the storage, using **clearWithError:** method.*

captureScenario:didFailWithError:result:

Notifies the delegate that an error has occurred.

```
- (void)captureScenario:(AUIMultiPageImageCaptureScenario*)captureScenario
    didFailWithError:(NSError*)error
    result:(id<AUIMultiImageCaptureResult>)result;
```

Parameters

captureScenario

> **AUIMultiPageImageCaptureScenario** object, corresponding to the **AUIMultiPageImageCaptureScenarioDelegate** delegate that should be notified.

error

> The error that has occurred.

result

> If any images were captured, they are stored in the **AUIMultiPageImageCaptureResult** result object.

Note: *In case you do not want to save images in the storage after stopped or failed scenario, you can remove the images from the storage, using **clearWithError:** method.*

AUIPageStorage protocol

API for implementing a custom image storage.

Most common scenarios will work with the default page storage. **AUIPageStorage** protocol is intended for advanced users and specific scenarios.

PageStorage can be represented as a collection of pages, where each page is assigned with a unique identifier (GUID) and is itself a key-value collection with string keys and arbitrary data values (**NSData**). Key-value relationship is to be maintained on the client side. An example of a key is an image or a thumbnail.

The default implementation of the **PageStorage** is file-based. By default it has a root folder located in application's internal storage. Each page is stored in a subfolder of the root folder named with page's identifier. All page-related data, such as captured image and its properties, is stored in files inside the corresponding subfolder.

Methods can be used for creation and managing pages of the image storage.

This logic is also used in the custom image storage implementation with the **AUIPageStorage** protocol and its methods.

To use the implemented custom storage instead of the default one during the **AUIMultiPageImageCaptureScenario** object initialization, use the corresponding **initWithEngine:** method signature.

This protocol and its methods are to be implemented on the client side.

Methods

Name	Description
- **clearWithError**	Removes all the pages from the storage.
- **createWithError**	Creates an empty page and adds it to the storage.
- **deleteWithId:**	Removes the page with specified identifier.
- **loadDataForPage:key:**	Returns the data, corresponding to a certain page and associated with the specified key.
- **pagesWithError**	Returns all the pages identifiers in the storage.
- **storeData:page:key:**	Adds, removes or edits data, associated with the key of the page with the specified identifier.

clearWithError: method

Removes all the pages from the storage.

```
- (BOOL)clearWithError:(NSError**)error;
```

Parameters

error

The error that has occurred. In case no errors occurred, this parameter is **nil**.

Return values

The method returns returns YES, if the pages were successfully deleted, and NO otherwise.

createWithError: method

Creates an empty page with a string identifier and adds it to the storage.

```
- (nullable AUIPageId)createWithError:(NSError**)error;
```

Parameters

error

The error that has occurred. In case no errors occurred, this parameter is **nil**.

Return values

The method returns an **AUIPageId** object, storing the identifier of the created page.

deleteWithId:error: method

Removes the page with specified identifier.

```
- (BOOL)deleteWithId:(AUIPageId)identifier error:(NSError**)error;
```

Parameters

identifier

AUIPageId object, storing the identifier of the page to be removed.

error

The error that has occurred. In case no errors occurred, this parameter is **nil**.

Return values

The method returns YES, if the page folder was successfully deleted, and NO otherwise.

loadDataForPage:key:error: method

Returns the data of the certain page, associated with the specified key.

```
- (nullable NSData*)loadDataForPage:(AUIPageId)identifier key:(NSString*)
key error:(NSError**)error;
```

Parameters

identifier

AUIPageId object, storing the identifier of the page to be removed.

key

An identifier, associated with required page data.

error

The error that has occurred. In case no errors occurred, this parameter is **nil**.

Return values

The method returns required data about the page.

pagesWithError: method

Returns all the pages identifiers in the storage.

```
- (nullable NSArray<AUIPageId>*)pagesWithError:(NSError**)error;
```

Parameters

error

The error that has occurred. In case no errors occurred, this parameter is **nil**.

Return values

The method returns an array of **AUIPageId** objects, storing the identifiers of all pages in the storage.

storeData:page:key:error: method

Adds, removes or edits data, associated with the key of the page with the specified identifier.

```
- (BOOL)storeData:(nullable NSData*)data page:(AUIPageId)identifier key:
(NSString*)key error:(NSError**)error;
```

Parameters

data

NSData object, storing the exact data to be added to the page by the key.

identifier

AUIPageId object, storing the identifier of the page with certain data.

key

The key, associated with the data to be edited.

error

The error that has occurred. In case no errors occurred, this parameter is **nil**.

Return values

The method returns YES, if the data of the page was successfully edited, and NO otherwise.

AUIMultiPageUISettings protocol

Provides access to extra user interface setting of current view control.

Methods

Name	Description
- captureScenario:stringForResourceType:	Provides string resources for the view control, depending on the scenario step.

captureScenario:stringForResourceType:forPageAtIndex: method

Applies the appearance settings to the specified page.

Use this method to customize string resources, that will be shown as tips for a user.

```
- (void)captureScenario:(AUIMultiPageImageCaptureScenario*)captureScenario
    stringForResourceType:(id<AUIMultiPageResourceType>)event
    forPageAtIndex:(NSUInteger)index;
```

Parameters

captureScenario

The multipage image capture scenario, represented by the **AUIMultiPageImageCaptureScenario** object. The page to which the appearance setting should be applied is processed during this scenario.

event

The **AUIMultiPageResourceType** object, defining the source string that will be shown at the current view.

index

The number of the page to which the settings should be applied.

AUIThemeButton interface

Provides settings for button with borders and blurred background appearance. The color of the borders depends on the color theme.

Properties

Name	Type	Description
selectedColor	UIColor*	The color of button borders.

AUICameraSettings protocol

Manages camera flashlight and resolution, defined for the capture scenario.

Properties

Name	Type	Description
preferredResolution	**AUICameraResolution**	Defines the camera resolution.
hasFlashlight	BOOL	Defines if the device has a flashlight.

Name	Type	Description
flashlightEnabled	BOOL	Defines if the flashlight is currently enabled in a torch mode.

AUIDocumentSize

The size of captured document in millimeters. You can define the size of the document to be captured so that the capturing mechanism will search the exact document on frames of the video stream. In case document size is unknown the mechanism will search for any document.

```
typedef CGSize AUIDocumentSize NS_TYPED_EXTENSIBLE_ENUM;{
    extern AUIDocumentSize const AUIDocumentSizeAny;
    extern AUIDocumentSize const AUIDocumentSizeA4;
    extern AUIDocumentSize const AUIDocumentSizeBusinessCard;
    extern AUIDocumentSize const AUIDocumentSizeLetter;
};
```

Constants

Name	Description
AUIDocumentSizeAny	The document can be of any size.
AUIDocumentSizeA4	The size of the captured document is 210 x 297 mm.
AUIDocumentSizeBusinessCard	The size of the captured document is 53.98 x 85.6 mm. ***Note:*** *It is recommended to define aspect ratio using* ***aspectRatioMin*** *and* ***aspectRatioMax*** *for business cards capture. Otherwise only business cards with 53.98x85.6 aspect ratio will be captured.*
AUIDocumentSizeLetter	The size of the captured document is 215.9 x 279.4 mm.

AUIPageId

Page identifier in the images result collection, represented by string.

```
typedef NSString* AUIPageId;
```

UIView (AUIRotation)

The category AUIRotation extends the UIView class. Add the method of this category to the custom control and return YES to rotate according to device orientation on iPhone.

Methods

Name	Description
- aui_canRotate	Defines if the custom control should be rotated according to the device orientation.

aui_canRotate method

Add this method to custom control and return YES to rotate the control according to device orientation on iPhone.

```
- (BOOL)aui_canRotate;
```

Return values

The method returns YES if the control should be rotated, NO otherwise.

AUICameraResolution enumeration

The resolution of the images captured from the camera preview.

```
typedef NS_ENUM(NSUInteger, AUICameraResolution) {
    AUICameraResolutionHD,
    AUICameraResolutionFullHD
    AUICameraResolution4K
};
```

Constants

Name	Description
AUICameraResolutionHD	Captured image will have: • 1280x720 px resolution on iPhone • 1024x768 px resolution on iPad
AUICameraResolutionFullHD	Captured image will have: • 1920x1080 px resolution on iPhone • 1920x1440 px resolution on iPad This is the default value.
AUICameraResolution4K	Captured image will have • 3840x2160 px resolution on iPhone • 3264x2448 px resolution on iPad

AUIMultiPageResourceType enumeration

String resources, that should be shown according to the corresponding view control.

```
typedef NS_ENUM(NSUInteger, AUIMultiPageResourceType) {
    AUIMultiPageResourceTypeLookingForPageTip,
    AUIMultiPageResourceTypePageTitle,
    AUIMultiPageResourceTypeAddPageTip,
};
```

Constants

Name	Description
AUIMultiPageResourceTypeLookingForPageTip	Screen message for automatic capture cases, shown when the camera should be brought closer to the document. I.e., "Looking for document."
	Screen message for manual capture cases, shown when the camera is turned on and ready for manual capture. I.e., "Point camera at a document and make a photo."

Name	Description
AUIMultiPageResourceTypePageTitle	Interface source string, storing the page name to display in preview navigation bar. I.e., "Passport".
AUIMultiPageResourceTypeAddPageTip	Interface source string, storing text to display on a button in case more images should be captured according to the scenario settings. I.e.: "Tap to add image of the third document page."

AUITheme enumeration

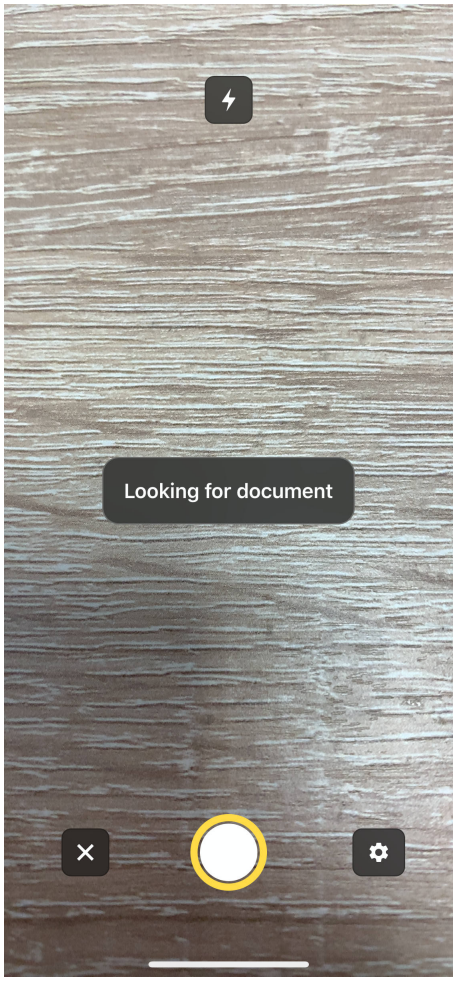
The user interface theme. Chosen theme defines the following:

- background color
- color of the text
- buttons color
- document tracking frame color
- capture area corners color

```
typedef NS_ENUM(NSUInteger, AUITheme) {
    AUIThemeLight,
    AUIThemeDark
};
```

Constants

Name	Description
AUIThemeLight	

Name	Description
AUIThemeDark	

Specifications

This section describes the technical requirements and capabilities of ABBYY Mobile Capture SDK.

Device Requirements

iOS version: 11.x or later

Supported devices:

- iPhone 5S or newer
- iPad Pro
- iPad Air or newer
- iPad mini 2 or newer

Memory requirements

Library operation takes up to:

- for texts in alphabetic languages — **40** MB RAM
- for texts in Chinese, Japanese, or Korean languages — **70** MB RAM

Library operation in the data capture scenario (for example, passport recognition) takes up to **105** MB RAM.

Library operation in the image capture scenario takes up to **35** MB RAM.

Please note, that your device may require more memory for certain processing scenario than specified in this section. For example, the next parameters may increase required RAM:

- recognition threads number
- device speed
- camera resolution
- recognition complexity

The higher are these indices, the more RAM is required.

Distribution Kit

ABBYY Mobile Capture SDK distribution pack includes the library, various resource files, samples and documentation. This section will help you determine which of the files to include when distributing your own application, and minimize the size of the final package.

The following folders contain files for development purposes only, not to be distributed:

Folder	File name	Description
	Readme.html	Readme file.
help	MobileCaptureDevelopersGuide.pdf	This Developer's Guide.
	DataCaptureProfilesReference.html	Document, containing full list of predefined capture profiles, corresponding result data schemes and fields.
sample-ui-imagecapture **sample-ui-imagecapture-swift**	All files in this folder.	This sample illustrates the steps you need to perform to create a simple mobile application for image capture.
sample-ui-imagecapture-multipage **sample-ui-imagecapture-multipage-swift**	All files in this folder.	The sample code in Objective-C and Swift programming languages implementing a multipage image capture scenario with tuned user interface.
sample-datacapture **sample-datacapture-swift**	All files in this folder.	The sample code in Objective-C and Swift programming languages implementing a data capture scenario where the capture rule is specified by a regular expression.
sample-textcapture **sample-textcapture-swift**	All files in this folder.	The sample code in Objective-C and Swift programming languages implementing a simple text capture scenario.
sample-imagecapture **sample-imagecapture-swift**	All files in this folder.	The sample code in Objective-C and Swift programming languages implementing an image capture scenario.
sample-coreapi	All files in this folder.	The sample code in Objective-C and Swift

Folder	File name	Description
sample-coreapi-swift		programming languages demonstrating the core API usage in a simple scenario of capturing data from an image.

The files in the **libs**, **assets**, and **notice** folders are intended for the final distribution of your application. The table below shows what files you should distribute depending on your needs.

Folder	File name	Description	Distribution
libs	MobileOCR.framework	The ABBYY Mobile Capture SDK framework.	Required for the main capture scenario: text capture, data capture, image capture and core API.
	MobileImaging.framework	The ABBYY Mobile Capture SDK framework for image capture.	Required only for the image capture scenario.
	AbbyyUI.framework	The ABBYY Mobile Capture SDK framework for developing mobile application with user interface.	Required for creating mobile application with native interface for image capture scenario.
	AbbyyRtrSDK.framework AbbyyZlib.framework CustomAllocator.framework FineMachineLearning.framework FineMachineLearningExt.framework FineObj.framework Image.Mobile.framework copy_frameworks.sh	The ABBYY Mobile Capture SDK frameworks.	Always required.

Folder	File name	Description	Distribution
assets/dictionaries	Brazil.edc	Portuguese (Brazil) language recognition dictionary.	Only those dictionaries that correspond to the languages you will work with.
	Bulgar.edc	Bulgarian language recognition dictionary.	
	Czech.edc	Czech language recognition dictionary.	
	Danish.edc	Danish language recognition dictionary.	
	Dutch.edc	Dutch (Netherlands) language recognition dictionary.	
	English.edc	English language recognition dictionary.	
	Eston.edc	Estonian language recognition dictionary.	
	Finnish.edc	Finnish language recognition dictionary.	
	Flemmish.edc	Dutch (Belgium) language recognition dictionary.	
	French.edc	French language recognition dictionary.	
	German.edc	German (old spelling) language recognition dictionary.	
	GermanNS.edc	German (new spelling)	

Folder	File name	Description	Distribution
		language recognition dictionary.	
	Greek.edc	Greek language recognition dictionary.	
	Indones.edc	Indonesian language recognition dictionary.	
	Italian.edc	Italian language recognition dictionary.	
	NorwBok.edc	Norwegian (Bokmal) language recognition dictionary.	
	NorwNyn.edc	Norwegian (Nynorsk) language recognition dictionary.	
	Polish.edc	Polish language recognition dictionary.	
	Portug.edc	Portuguese (Portugal) language recognition dictionary.	
	Russian.edc	Russian language recognition dictionary.	
	Spanish.edc	Spanish language recognition dictionary.	
	Swedish.edc	Swedish language recognition dictionary.	
	Turkish.edc	Turkish language recognition dictionary.	

Folder	File name	Description	Distribution
	Ukrain.edc	Ukrainian language recognition dictionary.	
assets/patterns	DIQBlockClassifier.imodel CropClassifierPhoto.imodel DIQClassifier.imodel	The ABBYY Mobile Imaging SDK II resource files	Required for image capture scenario.
	ChineseJapanese.rom	Recognition database for Chinese, Japanese, and Korean languages.	Required for recognition of texts in Chinese, Japanese and Korean languages.
	European.rom	Recognition database for all supported recognition languages.	Required for all recognition languages.
	FindText.rom	Recognition database for all languages.	Always required.
	KoreanSpecific.rom	Recognition database for Korean language.	Required for recognition of texts in Korean language.
assets/bcr	Brazil.akw	Source file for Brazilian business cards recognition.	Required for business cards recognition scenario.
	ChineseSimplified.akw	Source file for Chinese Simplified business cards recognition.	
	ChineseTraditional.akw	Source file for Chinese Traditional business cards recognition.	
	Czech.akw	Source file for Czech business cards recognition.	

Folder	File name	Description	Distribution
	Danish.akw	Source file for Danish business cards recognition.	
	Dutch.akw	Source file for Dutch business cards recognition.	
	English.akw	Source file for English business cards recognition.	
	Eston.akw	Source file for Estonian business cards recognition.	
	Finnish.akw	Source file for Finnish business cards recognition.	
	French.akw	Source file for French business cards recognition.	
	German.akw	Source file for German business cards recognition.	
	Greek.akw	Source file for Greek business cards recognition.	
	Indones.akw	Source file for Indonesian business cards recognition.	
	Italian.akw	Source file for Italian business cards recognition.	

Folder	File name	Description	Distribution
	Japanese.akw	Source file for Japanese business cards recognition.	
	Korean.akw	Source file for Korean business cards recognition.	
	NorwBok.akw	Source file for Norwegian (Bokmal) business cards recognition.	
	NorwNyn.akw	Source file for Norwegian (Nynorsk) business cards recognition.	
	Polish.akw	Source file for Polish business cards recognition.	
	Portug.akw	Source file for Portuguese business cards recognition.	
	Russian.akw	Source file for Russian business cards recognition.	
	Spanish.akw	Source file for Spanish business cards recognition.	
	Swedish.akw	Source file for Swedish business cards recognition.	
	Turkish.akw	Source file for Turkish business cards recognition.	

Folder	File name	Description	Distribution
	Ukrain.akw	Source file for Ukrainian business cards recognition.	
	WestEuropean.akw	Source file for recognition of English, French, German, Portuguese, Spanish and Italian business cards.	
assets/	copy_assets.py	Script for automatic copying resource files to corresponding destinations and adding necessary dictionaries to the Xcode project.	
scenarios-datacapture/assets/patterns	All_EDC.rom	All recognition databases from this directory.	Required if all *.rom files from this directory will be used.
	MRZ.rom	Recognition database for MRZ of the passport.	Required for MRZ data recognition.
	MRZ_EDC.rom	Extended MRZ recognition database for various document types.	Required for recognizing MRZ and MRZ-like zone data on supported documents (see Data Capture Profiles for details).
	BankCards_EDC.rom	Bank card recognition database.	Required for bank card recognition.
	ID_AE_EDC.rom	Recognition database for UAE documents.	Only the databases for the countries you are going to support are required.

Folder	File name	Description	Distribution
	ID_AL_EDC.rom	Recognition database for Albanian documents.	
	ID_AM_EDC.rom	Recognition database for Armenian documents.	
	ID_AT_EDC.rom	Recognition database for Austrian documents.	
	ID_AZ_EDC.rom	Recognition database for Azerbaijani documents.	
	ID_BE_EDC.rom	Recognition database for Belgium documents.	
	ID_BG_EDC.rom	Recognition database for Bulgarian documents.	
	ID_BH_EDC.rom	Recognition database for Bahrain documents.	
	ID_BR_EDC.rom	Recognition database for Brazilian documents.	
	ID_BY_EDC.rom	Recognition database for Belarusian documents.	
	ID_CA_EDC.rom	Recognition database for Canadian documents.	

Folder	File name	Description	Distribution
	ID_CH_EDC.rom	Recognition database for Swiss documents.	
	ID_CL_EDC.rom	Recognition database for Chile documents.	
	ID_CN_EDC.rom	Recognition database for Chinese documents.	
	ID_CY_EDC.rom	Recognition database for Cyprus documents.	
	ID_CZ_EDC.rom	Recognition database for Czech documents.	
	ID_DE_EDC.rom	Recognition database for German documents.	
	ID_DZ_EDC.rom	Recognition database for Algerian documents.	
	ID_EE_EDC.rom	Recognition database for Estonian documents.	
	ID_EG_EDC.rom	Recognition database for Egyptian documents.	
	ID_ES_EDC.rom	Recognition database for Spanish documents.	
	ID_FI_EDC.rom	Recognition database for Finnish documents.	

Folder	File name	Description	Distribution
	ID_FR_EDC.rom	Recognition database for French documents.	
	ID_GE_EDC.rom	Recognition database for Georgian documents.	
	ID_GR_EDC.rom	Recognition database for Greek documents.	
	ID_HK_EDC.rom	Recognition database for Hong Kong documents.	
	ID_HR_EDC.rom	Recognition database for Croatian documents.	
	ID_HU_EDC.rom	Recognition database for Hungarian documents.	
	ID_IL_EDC.rom	Recognition database for Israeli documents.	
	ID_IN_EDC.rom	Recognition database for Indian documents.	
	ID_IT_EDC.rom	Recognition database for Italian documents.	
	ID_JP_EDC.rom	Recognition database for Japanese documents.	
	ID_KG_EDC.rom	Recognition database for Kyrgyzstani documents.	

Folder	File name	Description	Distribution
	ID_KW_EDC.rom	Recognition database for Kuwait documents.	
	ID_KZ_EDC.rom	Recognition database for Kazakhstan documents.	
	ID_LT_EDC.rom	Recognition database for Lithuanian documents.	
	ID_LV_EDC.rom	Recognition database for Latvian documents.	
	ID_MD_EDC.rom	Recognition database for documents of Republic of Moldova.	
	ID_MK_EDC.rom	Recognition database for Macedonian documents.	
	ID_MX_EDC.rom	Recognition database for Mexican documents.	
	ID_MY_EDC.rom	Recognition database for Malaysian documents.	
	ID_NG_EDC.rom	Recognition database for Nigerian documents.	
	ID_NO_EDC.rom	Recognition database for Norwegian documents.	
	ID_NZ_EDC.rom	Recognition database for New Zealand	

Folder	File name	Description	Distribution
		documents.	
	ID_PH_EDC.rom	Recognition database for Philippine documents.	
	ID_PL_EDC.rom	Recognition database for Polish documents.	
	ID_PT_EDC.rom	Recognition database for Portuguese documents.	
	ID_RO_EDC.rom	Recognition database for Romanian documents.	
	ID_RS_EDC.rom	Recognition database for Serbian documents.	
	ID_RU_EDC.rom	Extended recognition database for Russian documents.	
	ID_SE_EDC.rom	Recognition database for Swedish documents.	
	ID_SG_EDC.rom	Recognition database for Singapore documents.	
	ID_SI_EDC.rom	Recognition database for Slovenian documents.	
	ID_SK_EDC.rom	Recognition database for Slovak documents.	

Folder	File name	Description	Distribution
	ID_SV_EDC.rom	Recognition database for Salvadorean documents.	
	ID_SY_EDC.rom	Recognition database for Syrian documents.	
	ID_TJ_EDC.rom	Recognition database for Tajikistan documents.	
	ID_TR_EDC.rom	Recognition database for Turkish documents.	
	ID_UA_EDC.rom	Recognition database for Ukrainian documents.	
	ID_UK_EDC.rom	Recognition database for British documents.	
	ID_US_EDC.rom	Recognition database for USA documents.	
	ID_UY_EDC.rom	Recognition database for Uruguayn documents.	
	ID_UZ_EDC.rom	Recognition database for Uzbekistan documents.	
	ID_VN_EDC.rom	Recognition database for Vietnamese documents.	
	ID_ZA_EDC.rom	Recognition database for South African Republic documents.	

Folder	File name	Description	Distribution
assets/translation	Menu_CH-EN.trdic	Dictionary for translating menus from Chinese to English.	The files contain translation dictionaries. You need only the files for the language pairs you use.
	Menu_DE-EN.trdic	Dictionary for translating menus from German to English.	
	Menu_EN-CH.trdic	Dictionary for translating menus from Chinese to English.	
	Menu_EN-DE.trdic	Dictionary for translating menus from English to German.	
	Menu_EN-ES.trdic	Dictionary for translating menus from English to Spanish.	
	Menu_EN-FR.trdic	Dictionary for translating menus from English to French.	
	Menu_EN-ID.trdic	Dictionary for translating menus from English to Indonesian.	
	Menu_EN-JP.trdic	Dictionary for translating menus from English to Japanese.	
	Menu_EN-PL.trdic	Dictionary for translating menus	

Folder	File name	Description	Distribution
		from English to Polish.	
	Menu_EN-PTBR.trdic	Dictionary for translating menus from English to Portuguese (Brazil).	
	Menu_EN-RU.trdic	Dictionary for translating menus from English to Russian.	
	Menu_ES-EN.trdic	Dictionary for translating menus from Spanish to English.	
	Menu_FR-EN.trdic	Dictionary for translating menus from French to English.	
	Menu_ID-EN.trdic	Dictionary for translating menus from Indonesian to English.	
	Menu_JP-EN.trdic	Dictionary for translating menus from Japanese to English.	
	Menu_PL-EN.trdic	Dictionary for translating menus from Polish to English.	
	Menu_PTBR-EN.trdic	Dictionary for translating menus from Portuguese (Brazil) to English.	
	Menu_RU-EN.trdic	Dictionary for	

Folder	File name	Description	Distribution
		translating menus from Russian to English.	
notice	All files in this folder.	Third party software components information and licenses.	These files have to be redistributed.

Available Recognition Languages

This section lists the languages available for text processing with ABBYY Mobile Capture SDK. Some of the languages have built-in dictionary support, which improves recognition quality but takes up additional memory.

See also Available Translation Dictionaries.

Internal name	Recognition language	Can be used for OCR	Can be used for BCR	Full dictionary support	Available in free version
Afrikaans	Afrikaans	+			+
Albanian	Albanian	+			+
Basque	Basque	+			+
Belarusian	Belarusian	+			
Breton	Breton	+			+
Bulgarian	Bulgarian	+		+	
Catalan	Catalan	+			+
Chechen	Chechen	+			

Internal name	Recognition language	Can be used for OCR	Can be used for BCR	Full dictionary support	Available in free version
ChineseSimplified	Chinese Simplified	+	+		
ChineseTraditional	Chinese Traditional	+	+		
CrimeanTatar	Crimean Tatar	+			
Croatian	Croatian	+			+
Czech	Czech	+	+	+	+
Danish	Danish	+	+	+	+
DutchBelgian	Dutch (Belgium)	+	+	+	+
Dutch	Dutch (Netherlands)	+	+	+	+
English	English	+	+	+	+
Estonian	Estonian	+	+	+	+
Fijian	Fijian	+			+
Finnish	Finnish	+	+	+	+
French	French	+	+	+	+
German	German (old spelling)	+	+	+	+

Internal name	Recognition language	Can be used for OCR	Can be used for BCR	Full dictionary support	Available in free version
GermanNewSpelling	German (new spelling)	+	+	+	+
Greek	Greek	+	+	+	+
Hawaiian	Hawaiian	+			+
Hungarian	Hungarian	+			+
Icelandic	Icelandic	+			+
Indonesian	Indonesian	+	+	+	+
Irish	Irish	+			+
Italian	Italian	+	+	+	+
Japanese	Japanese	+	+		
Kabardian	Kabardian	+			
Korean	Korean	+	+		
KoreanHangul	Korean (Hangul)	+	+		
Latin	Latin	+			+
Latvian	Latvian	+			+
Lithuanian	Lithuanian	+			+
Macedonian	Macedonian	+			

Internal name	Recognition language	Can be used for OCR	Can be used for BCR	Full dictionary support	Available in free version
Malay	Malay	+			+
Maori	Maori	+			+
Moldavian	Moldavian	+			+
Mongol	Mongol	+			
NorwegianBokmal	Norwegian (Bokmal)	+	+	+	+
NorwegianNynorsk	Norwegian (Nynorsk)	+	+	+	+
Ossetic	Ossetic	+			
Polish	Polish	+	+	+	+
PortugueseBrazilian	Portuguese (Brazil)	+	+	+	+
Portuguese	Portuguese (Portugal)	+	+	+	+
Provencal	Provencal	+			+
RhaetoRomanic	Rhaeto-Romanic	+			+
Romanian	Romanian	+			+
Russian	Russian	+	+	+	
Samoan	Samoan	+			+

Internal name	Recognition language	Can be used for OCR	Can be used for BCR	Full dictionary support	Available in free version
Serbian	Serbian	+			
Slovak	Slovak	+			+
Slovenian	Slovenian	+			+
Spanish	Spanish	+	+	+	+
Swahili	Swahili	+			+
Swedish	Swedish	+	+	+	+
Tagalog	Tagalog	+			+
Tatar	Tatar	+			
Turkish	Turkish	+	+	+	+
Ukrainian	Ukrainian	+	+	+	
Welsh	Welsh	+			+

Translation Dictionaries

In the distribution pack you can find several translation dictionaries. Currently all the dictionaries are intended for translating restaurant menus and may not work in other contexts. The following language pairs are available:

English to Chinese	Chinese to English
English to French	French to English
English to German	German to English
English to Indonesian	Indonesian to English
English to Japanese	Japanese to English

English to Polish	Polish to English
English to Portuguese (Brazil)	Portuguese (Brazil) to English
English to Russian	Russian to English
English to Spanish	Spanish to English

You can also create your own dictionary and use it for translation. Contact our technical support for advice on the required format.

Supported ID Documents

ABBYY Mobile Capture SDK supports a whole range of identity documents out of the box. Consult the table below for a full list. For the detailed profile specifications, see Data Capture Profiles.

Document	Supported in
All documents with Machine Readable Zone (MRZ)	All Countries
Bank cards: embossed, indent, freeform	All Countries
Driver's license	Albania, Armenia, Austria, Belarus, Belgium, Brazil, Bulgaria, Canada, Croatia, Czech Republic, Finland, Germany, Greece, Hungary, Israel, Italy, Japan, Kazakhstan, Kyrgyzstan, Luxembourg, Moldova, New Zealand, Norway, Poland, Portugal, Romania, Russian Federation, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey, UK, Ukraine, USA, Uzbekistan, Vietnam
International Passport	Albania, Algeria, Armenia, Austria, Brazil, Canada, China, Croatia, Czech Republic, Georgia, Germany, Greece, Hungary, India, Israel, Italy, Japan, Kazakhstan, Kyrgyzstan, Luxembourg, Moldova, Philippines, Poland, Russian Federation, Slovakia, Slovenia, Spain, Sweden, Syria, Tajikistan, Turkey, UK, Ukraine, Uruguay, USA, Uzbekistan
National ID card	Albania, Armenia, Austria, Bahrain, Belgium, Brazil, Bulgaria, Chile, China, Croatia, Cyprus, Czech Republic, Egypt, Estonia, Finland, France, Georgia, Germany, Hong Kong, Hungary, Israel, Italy, Kazakhstan, Kuwait, Kyrgyzstan, Latvia, Lithuania, Luxembourg, Macedonia, Malaysia, Mexico, Moldova, Nigeria, Norway, Poland,

Document	Supported in
	Portugal, Romania, El Salvador, Serbia, Slovakia, Slovenia, Singapore, South Africa, Spain, Switzerland, Turkey, UAE, Ukraine
National passport	Belarus, Russian Federation
INN	Russian Federation
Aadhaar card	India
Birth certificate	Russian Federation
Death Certificate	Russian Federation
Marriage Certificate	Russian Federation
Divorce Certificate	Russian Federation
Compulsory Health Insurance Certificate – OMS	Russian Federation
Personal insurance policy number	Russian Federation
Vehicle Registration Certificate (STS)	Azerbaijan, Belarus, Czech Republic, El Salvador, Kazakhstan, Russian Federation, Slovakia, Ukraine
Vehicle Passport - PTS	Russian Federation
VISA	Russian Federation, USA, Czech Republic, Slovakia
Border Crossing Card	USA
Passport Card	USA
Health insurance card	Japan

Document	Supported in
Work permit	Russian Federation, Singapore
Residence permit	Austria, Czech Republic, Germany, Luxembourg, Russian Federation, Slovakia, Slovenia, Spain
Asylum Residence Permit	Austria
Migration Card	Russian Federation
Permanent residency card (Green card)	USA
Residence License	Brazil
Crew Member Certificate	South Africa
Military, Police and Soldier ID	Russian Federation

The list of supported documents and captured fields for each document differ depending on the country. You can find the detailed information in the Data Capture Profiles table.

Data Capture Profiles

The following table lists predefined capture profiles and corresponding result data schemes. Profile name is specified when creating a Data Capture service, and result scheme identifiers are returned by the service. Note that in some cases the result scheme depends on the type of your license. If you are not sure which profiles are enabled by your license, please contact support.

You can investigate the full list of the result data schemes returned by the Data Capture service in corresponding **DataCaptureProfilesReference.html** file in the **/help** directory of your distribution.

Document type	Profile name	Result scheme	Result description
Indian Aadhaar card	Aadhaar_IN	Aadhaar_IN_TYPE1	Indian Aadhaar card (square card; flag on the top center; emblem on the top left, front)

Document type	Profile name	Result scheme	Result description
Austrian asylum residence permit	AsylumResidencePermit_AT	AsylumResidencePermit_AT_RP_TYPE1	Austrian asylum residence permit (white background, front)
Bank card	BankCards	BankCardEmbossed	Bank cards with embossed fields (front side)
		BankCardFreeform	Bank cards of all types, i.e. bank cards with all data on one side (back side)
		BankCardUnembossed	Bank cards with indent-printed fields (front side)
Russian birth certificate	BirthCertificate_RU	BirthCertificate_RU_TYPE1	Russian birth certificate (Russian emblem on the top, main page)
USA border crossing	BorderCrossing_US	BorderCrossing_US_TYPE1	USA border crossing (a horizontal card; rhombuses and building on the background, front)
		BorderCrossing_US_TYPE2	USA border crossing (a horizontal card; flag outline on the top left, front)
Business card	BusinessCards	BusinessCards	Business card of a person or a company
South African pilot's license	CrewMember_ZA	CrewMember_ZA_TYPE1	South African pilot's license (biometric symbol in the top-right corner, main page)

Document type	Profile name	Result scheme	Result description
Russian death certificate	DeathCertificate_RU	DeathCertificate_RU_TYPE1	Russian death certificate
Russian divorce certificate	DivorceCertificate_RU	DivorceCertificate_RU_TYPE1	Russian divorce certificate
Albanian driver's license	DriverLicense_AL	DriverLicense_AL_TYPE 1	Albanian driver's license (Albanian emblem in the background)
Armenian driver's license	DriverLicense_AM	DriverLicense_AM_TYPE 1	Armenian driver's license (stamp with Armenian emblem on the right, front)
		DriverLicense_AM_TYPE 2	Armenian driver's license (with authority name, place of birth and residence fields in English)
Austrian driver's license	DriverLicense_AT	DriverLicense_AT_TYPE 1	Austrian driver's license with the title at the top (front side)
		DriverLicense_AT_TYPE 2	Austrian driver's license with the title in the top-right corner (front side)
Belgian driver's license	DriverLicense_BE	DriverLicense_BE_TYPE 1	Belgian driver's license (the sign of the European Union with letter B in the top-left corner and the contour of country on the bottom-right corner, front side)
		DriverLicense_BE_TYPE 2	Belgian driver's license (the sign of

Document type	Profile name	Result scheme	Result description
			the European Union with letter B in the top-left corner and the contour of country on the bottom-right corner, front side)
		DriverLicense_BE_TYPE 3	Belgian driver's license (the sign of the European Union with letter B in the top-left corner and the contour of country on the bottom-right corner, front side)
Bulgarian driver's license	DriverLicense_BG	DriverLicense_BG_TYPE 1	Bulgarian driver's license (sign of the European Union with letters BG in the top-left corner and the countour of country on the top in the middle, front side)
		DriverLicense_BG_TYPE 2	Bulgarian driver's license (a round stamp with letters BG on the right side, front side)
Brazilian driver's license	DriverLicense_BR	DriverLicense_BR_TYPE 1	Brazilian driver's license (Brazilian emblem in the top-left corner, front; green background, back)
Belorussian driver's license	DriverLicense_BY	DriverLicense_BY_TYPE 1	Belorussian driver's license (card-size, front side, horizontal)
		DriverLicense_BY_TYPE 2	Belorussian driver's license (front side, vertical)

Document type	Profile name	Result scheme	Result description
Canadian driver's license	DriverLicense_CA	DriverLicense_CA_AB_TYPE1	Canadian driver's license -Alberta (a barcode in the bottom-right corner, front side)
		DriverLicense_CA_AB_TYPE2	Canadian driver's license - Alberta (a round stamp on the right, front side)
		DriverLicense_CA_BC_TYPE1	Canadian driver's license - British Columbia (flag of British Columbia on the background and the coat of arms of British Columbia on the right, front side)
		DriverLicense_CA_BC_TYPE3	Canadian driver's license
		DriverLicense_CA_MB_TYPE1	Canadian driver's license - Manitoba (a bull above the personal photo, front side)
		DriverLicense_CA_NB_TYPE1	Canadian driver's license - New Brunswick (flower on the background, front)
		DriverLicense_CA_NB_TYPE2	Canadian driver's license - New Brunswick (watermark in the form of a leaf, front)
		DriverLicense_CA_NL_TYPE1	Canadian driver's license -

Document type	Profile name	Result scheme	Result description
			Newfoundland and Labrador (a flower on the background on the right, front side)
		DriverLicense_CA_NL_TYPE2	
		DriverLicense_CA_NS_TYPE1	Canadian driver's license - Nova Scotia (the escutcheon of Nova Scotia in the top-right corner, front side)
		DriverLicense_CA_NS_TYPE2	Canadian driver's license - Nova Scotia (national flag of Canada in the bottom-right corner and a road on the background, front side)
		DriverLicense_CA_NS_TYPE3	Canadian driver's license - Nova Scotia
		DriverLicense_CA_NT_TYPE1	Canadian driver's license - Northwest Territories (a bear on the top-left and top-right corner, front side)
		DriverLicense_CA_NU_TYPE1	Canadian driver's license - Nunavut (flag of Nunavut on the top near the name of document, front side)
		DriverLicense_CA_ON_TYPE1	Canadian driver's license (the flower on the background and

Document type	Profile name	Result scheme	Result description
			sign of ON in the top-right corner, front side)
		DriverLicense_CA_ON_TYPE2	Canadian driver's license (inscription Canada in the down-right corner, front side)
		DriverLicense_CA_PE_TYPE1	Canadian driver's license - Prince Edward Island (the escutcheon of Prince Edward Island in the top-right corner, front side)
		DriverLicense_CA_QC_TYPE1	Canadian driver's license - Quebec (the flag of Quebec above the personal photo on the left, front side)
		DriverLicense_CA_SK_TYPE1	Canadian driver's license - Saskatchewan (flower in the top-right corner, front)
		DriverLicense_CA_TYPE1	Canadian driver's license (globe on the background, front)
		DriverLicense_CA_YT_TYPE1	Canadian driver's license - Youkon (the coat of arms of Youkon and a small personal photo on the right, front side)
Swiss driver's license	DriverLicense_CH	DriverLicense_CH_TYPE1	Swiss driver's license (swiss flag in the top-

Document type	Profile name	Result scheme	Result description
			left corner, CH sign in the top-right corner, front side)
Czech driver's license	DriverLicense_CZ	DriverLicense_CZ_TYPE 1	Czech driver's license card (front side)
		DriverLicense_CZ_TYPE 2	
German driver's license	DriverLicense_DE	DriverLicense_DE_TYPE 1	German driver's license (sign of the European Union with letter D in the top-left corner and three road signs in the bottom-right corner, front side)
		DriverLicense_DE_TYPE 2	German driver's license (sign of the European Union with letter D in the top-left corner and stamp with the silhouette of the letter D in the bottom-right corner, front side)
Spanish driver's license	DriverLicense_ES	DriverLicense_ES_TYPE 1	Spanish driver's license (card-size, number field below the photo)
		DriverLicense_ES_TYPE 2	Spanish driver's license (card-size, number field to the right of the photo)
Finnish driver's license	DriverLicense_FI	DriverLicense_FI_TYPE1	Finnish driver's license (number-field under the personal photo, front side)

Document type	Profile name	Result scheme	Result description
		DriverLicense_FI_TYPE2	Finnish driver's license (number-field in the right side of the photo, front side)
Greek driver's license	DriverLicense_GR	DriverLicense_GR_TYPE 1	Greek driver's license (the sign of the European Union with letter in the top-left corner, front)
Croatian driver's license	DriverLicense_HR	DriverLicense_HR_TYPE 1	Croatian driver's license (sign of the European Union with letters HR in the top-left corner, front side)
Hungarian driver's license	DriverLicense_HU	DriverLicense_HU_TYPE 1	Hungarian driver's license (the sign of the European Union with letter in the top-left corner, front)
Israel driver's license	DriverLicense_IL	DriverLicense_IL_TYPE1	Israel driver's license (the coat of arms of Israel in the top-right corner, front side)
		DriverLicense_IL_TYPE2	Israel driver's license (the coat of arms of Israel in the top-right corner, blue sign in the top-left corner, front side)
Italian driver's license	DriverLicense_IT	DriverLicense_IT_TYPE1	Italian driver's license (new type, front side)
		DriverLicense_IT_TYPE2	Italian driver's license (issued 2007-2013, front side)

Document type	Profile name	Result scheme	Result description
Japanese driver's license	DriverLicense_JP	DriverLicense_JP_TYPE1	Japanese driver's license (a personal photo in the bottom-right corner, front side)
Kyrgyz driver's license	DriverLicense_KG	DriverLicense_KG_TYPE 1	Kyrgyz driver's license (KS sign in the top-left corner and the flag of Kyrgyzstan in the top-right corner, front side)
Kazakh driver's license	DriverLicense_KZ	DriverLicense_KZ_TYPE 1	Kazakh driver's license (stamp with car on the middle, front)
		DriverLicense_KZ_TYPE 2	Kazakh driver's license (chip on the right and Kazakh flag in the top-left corner, front)
Luxembourgian driver's license	DriverLicense_LU	DriverLicense_LU_TYPE 1	Luxembourgian driver's license (flag in the upper left corner and text PERMIS DE CONDUIRE at the middle, front side)
Moldavian driver's license	DriverLicense_MD	DriverLicense_MD_TYPE1	Driver's license of Republic of Moldova (flag of Republic of Moldova in the top-left corner, front)
		DriverLicense_MD_TYPE2	Driver's license of Republic of Moldova (emblem of Republic of Moldova in the top-right corner, front)

Document type	Profile name	Result scheme	Result description
		DriverLicense_MD_TYPE3	Driver's license of Republic of Moldova (emblem of Republic of Moldova at the middle, front)
Norwegian driver's license	DriverLicense_NO	DriverLicense_NO_TYPE1	Norwegian driver's license (ninth point under the photo and a watermark with Norwegian coat of arms to the right of the photo, front side)
		DriverLicense_NO_TYPE2	Norwegian driver's license (N sign in the top-left corner and ninth point under the signature, main page)
New Zealand driver's license	DriverLicense_NZ	DriverLicense_NZ_TYPE1	New Zealand driver's license (flag of New Zealand above the personal photo, front side)
		DriverLicense_NZ_TYPE2	New Zealand driver's license (flag of New Zealand above the personal photo, front side)
Polish driver's license	DriverLicense_PL	DriverLicense_PL_TYPE1	Polish driver's license (eye-like sign in the bottom-right corner, "PRAWO JAZDY" in the top-right corner, front side)
		DriverLicense_PL_TYPE2	Polish driver's license (flower on the background, front side)

Document type	Profile name	Result scheme	Result description
		DriverLicense_PL_TYPE3	Polish driver's license (tape with text on the background, front side)
Portuguese driver's license	DriverLicense_PT	DriverLicense_PT_TYPE 1	Portuguese driver's license (sign of the European Union with letter P in the top-left corner, front side)
Romanian driver's license	DriverLicense_RO	DriverLicense_RO_TYPE 1	Romanian driver's license (sign of the European Union with letters RO in the top-left corner, front side)
Serbian driver's license	DriverLicense_RS	DriverLicense_RS_TYPE 1	Serbian driver's license (SRB sign in the top-left corner and the coat of arms of Serbia in the top-right corner, front side)
Russian driver's license	DriverLicense_RU	DriverLicense_RU_TYPE 1	Russian driver's license, old type (front side)
		DriverLicense_RU_TYPE 2	Russian driver's license, old type, vertical (front side)
		DriverLicense_RU_TYPE 3	Russian driver's license, new type (front side)
Swedish driver's license	DriverLicense_SE	DriverLicense_SE_TYPE 1	Swedish driver's license (a small personal photo on the right, front side)

Document type	Profile name	Result scheme	Result description
		DriverLicense_SE_TYPE 2	Swedish driver's license (a stamp under the signature, front side)
Slovenian driver's license	DriverLicense_SI	DriverLicense_SI_TYPE1	Slovenian driver's license (country's name is written in one line, front side)
		DriverLicense_SI_TYPE2	Slovenian driver's license (country's name is written in two lines, front side)
Slovakian driver's license	DriverLicense_SK	DriverLicense_SK_TYPE 1	Slovakian driver's license (the contour of country with letters SK on the right, main page)
		DriverLicense_SK_TYPE 2	Slovakian driver's license (the contour of country with letters SK in the bottom-right corner, main page)
		DriverLicense_SK_TYPE 3	Slovakian driver's license (leaves in the bottom-right corner, main page)
		DriverLicense_SK_TYPE 4	Slovakian driver's license
Turkish driver's license	DriverLicense_TR	DriverLicense_TR_TYPE 1	Turkish driver's license (TR sign in the top-left corner and a car in the bottom-right corner, front side)

Document type	Profile name	Result scheme	Result description
		DriverLicense_TR_TYPE 2	Turkish driver's license (T.C. sign in the top-left corner, front side)
Ukrainian driver's license	DriverLicense_UA	DriverLicense_UA_TYPE 1	Ukrainian driver's license (Blue background and flag in the upper left corner, front side)
		DriverLicense_UA_TYPE 2	Ukrainian driver's license (Pink background and flag in the upper right corner, front side)
		DriverLicense_UA_TYPE 3	Ukrainian driver's license (Yellow background and flag in the upper left corner, front side)
		DriverLicense_UA_TYPE 4	Ukrainian driver's license
British driver's license	DriverLicense_UK	DriverLicense_UK_PROVISIONAL_TYPE1	British driver's license, provisional (line of text Provisional on the top, front)
		DriverLicense_UK_PROVISIONAL_TYPE2	British driver's license, provisional (rubber stamp on the right, front)
		DriverLicense_UK_PROVISIONAL_TYPE3	British driver's license, provisional (round stamp on the left, front)
		DriverLicense_UK_TYPE	British driver's license

Document type	Profile name	Result scheme	Result description
		1	(line of text on the top, front)
		DriverLicense_UK_TYPE2	British driver's license (British flag on the right, front)
		DriverLicense_UK_TYPE3	British driver's license (round stamp on the left, front)
USA driver's license	DriverLicense_US	DriverLicense_US_AK_TYPE1	USA driver's license - Alaska (mountains on the background, front)
		DriverLicense_US_AK_TYPE2	USA driver's license - Alaska (flag of Alaska on the background, front)
		DriverLicense_US_AK_TYPE3	USA driver's license - Alaska (mountains on the background and photo on the right, front)
		DriverLicense_US_AK_TYPE4	USA driver's license - Alaska (mountains on the background and photo on the right, front)
		DriverLicense_US_AL_TYPE1	USA driver's license - Alabama (building on the background, front)
		DriverLicense_US_AL_TYPE2	USA driver's license - Alabama (a vertical card, building on the background, front)

Document type	Profile name	Result scheme	Result description
		DriverLicense_US_AR_TYPE1	USA driver's license - Arkansas (stamp with emblem of Arkansas on the background, front)
		DriverLicense_US_AR_TYPE2	USA driver's license - Arkansas (DL sign in the middle on the top, front)
		DriverLicense_US_AR_TYPE3	USA driver's license - Arkansas (stamp with emblem of Arkansas on the background and a copy of personal photo in the bottom-right corner, vertical type under 21)
		DriverLicense_US_AR_TYPE4	USA driver's license - Arkansas (diamonds on the background and a copy of personal photo in the bottom-right corner, front)
		DriverLicense_US_AZ_TYPE1	USA driver's license - Arizona (a horizontal card and a cactus silhouette on the right on the background, front side)
		DriverLicense_US_AZ_TYPE2	USA driver's license - Arizona (a vertical card and a cactus silhouette on the right on the background, front side)
		DriverLicense_US_AZ_TYPE3	USA driver's license - Arizona (the Grand

Document type	Profile name	Result scheme	Result description
			Canyon on the background and a personal photo in the bottom-right corner, front side)
		DriverLicense_US_CA_TYPE1	USA driver's license - California (a horizontal card, bears on the background and a small personal photo on the right, front side)
		DriverLicense_US_CA_TYPE2	USA driver's license - California (a vertical card, a small personal photo in the bottom-left corner, front side)
		DriverLicense_US_CA_TYPE3	USA driver's license - California (a horizontal card, a bear with a star above the man on the right on the background, front side)
		DriverLicense_US_CA_TYPE4	USA driver's license - California (a horizontal card, DMV sign in the top-right and in the top-left corner, front side)
		DriverLicense_US_CO_TYPE1	USA driver's license - Colorado (a horizontal card, a star in the top-right corner and curves on the bottom, front side)
		DriverLicense_US_CO_T	USA driver's license -

Document type	Profile name	Result scheme	Result description
		YPE2	Colorado (a horizontal card, DL sign and a star in a circle are near the state name, front side)
		DriverLicense_US_CO_TYPE3	USA driver's license - Colorado (a vertical card, DL sign and a star in a circle are under the state name, front side)
		DriverLicense_US_CO_TYPE4	USA driver's license - Colorado (a vertical card, a star in the top-right corner, front side)
		DriverLicense_US_CT_TYPE1	USA driver's license - Connecticut (a horizontal card, letters DL on the top, front side)
		DriverLicense_US_CT_TYPE2	USA driver's license - Connecticut (a vertical card, letters ALP in the top-left corner, front side)
		DriverLicense_US_CT_TYPE3	USA driver's license - Connecticut (a horizontal card, the document's name near the state name and a helicopter on the top, front side)
		DriverLicense_US_CT_TYPE4	USA driver's license - Connecticut (a vertical card, the document's name under the state

Document type	Profile name	Result scheme	Result description
			name and a helicopter in the middle, front side)
		DriverLicense_US_CT_TYPE5	USA driver's license - Connecticut (a horizontal card, a lighthouse on the left on the background and letters DL in the top-right corner, front side)
		DriverLicense_US_DC_TYPE1	USA driver's license - Columbia (heart in the top-left corner and flag of Washington in the down-right corner, front)
		DriverLicense_US_DC_TYPE2	USA driver's license - Columbia (stamp with emblem of district of Columbia on the right, front)
		DriverLicense_US_DC_TYPE3	USA driver's license - Columbia (a star in the top-right corner, front)
		DriverLicense_US_DE_TYPE1	USA driver's license - Delaware (blue rectangle on the top and star in the top-tight corner, front)
		DriverLicense_US_FL_TYPE1	USA driver's license - Florida (stamp with emblem of Florida on the left, front)
		DriverLicense_US_FL_TY	USA driver's license -

Document type	Profile name	Result scheme	Result description
		PE2	Florida (star in the circle in the top-right corner, front)
		DriverLicense_US_FL_TYPE3	USA driver's license - Florida (star in the circle in the top-right corner, front)
		DriverLicense_US_FL_TYPE4	USA driver's license - Florida (star in the circle in the top-right corner, front)
		DriverLicense_US_GA_TYPE1	USA driver's license - Georgia (copies of a personal photo in the bottom-right corner and a peach on the background, front)
		DriverLicense_US_GA_TYPE2	USA driver's license - Georgia (vertically oriented, front)
		DriverLicense_US_GA_TYPE3	USA driver's license - Georgia (a round stamp in the top-left corner of photo and three peaches on the top, front)
		DriverLicense_US_GA_TYPE4	USA driver's license - Georgia
		DriverLicense_US_HI_TYPE1	USA driver's license - Hawaii (a barcode under the photo, front)
		DriverLicense_US_HI_TYPE2	USA driver's license - Hawaii (the flag of

Document type	Profile name	Result scheme	Result description
			state in the top-right corner, front)
		DriverLicense_US_IA_TYPE1	USA driver's license - Hawaii (windmill on the background, front)
		DriverLicense_US_IA_TYPE2	USA driver's license - Hawaii (windmill on the background - vertical card, front)
		DriverLicense_US_IA_TYPE3	USA driver's license - Hawaii (coat of arms of Iowa on the background, front)
		DriverLicense_US_ID_TYPE1	USA driver's license - Idaho (the seal of Idaho in the top-right corner of the personal photo, front side)
		DriverLicense_US_ID_TYPE2	USA driver's license - Idaho (mountains on the background and copy of a personal photo in the bottom-right corner, front side)
		DriverLicense_US_ID_TYPE3	USA driver's license - Idaho (mountains on the background and mark in the bottom-right corner, front side)
		DriverLicense_US_IL_TYPE1	USA driver's license - Illinois (Abraham Lincoln on the

Document type	Profile name	Result scheme	Result description
			background, front side)
		DriverLicense_US_IL_TYPE2	USA driver's license - Illinois (the curves on the background, front side)
		DriverLicense_US_IL_TYPE3	USA driver's license - Illinois (a personal photo on the right, front side)
		DriverLicense_US_IL_TYPE4	USA driver's license - Illinois (a vertical card, the curves on the background, front side)
		DriverLicense_US_IL_TYPE5	USA driver's license - Illinois (a vertical card, the curves on the background, front side)
		DriverLicense_US_IN_TYPE1	USA driver's license - Indiana (a horizontal card; a small personal photo in the bottom-right corner, front side)
		DriverLicense_US_IN_TYPE2	USA driver's license - Indiana (a vertical card; a small personal photo on the right, front side)
		DriverLicense_US_IN_TYPE3	USA driver's license - Indiana (a horizontal card, the seal of Indiana in the top-left corner, front side)

Document type	Profile name	Result scheme	Result description
		DriverLicense_US_KS_TYPE1	USA driver's license - Kansas (ears of corn on the background, horizontal card, front)
		DriverLicense_US_KS_TYPE2	USA driver's license - Kansas (ears of corn on the background, vertical card, front)
		DriverLicense_US_KS_TYPE3	USA driver's license - Kansas (tractor and wagon on the background and star in the top-right corner, horizontal card, front)
		DriverLicense_US_KS_TYPE4	USA driver's license - Kansas (tractor and wagon on the background and star in the top-right corner, vertical card, front)
		DriverLicense_US_KS_TYPE5	USA driver's license - Kansas (patterns on the bottom and DL sign in the top-right corner, front)
		DriverLicense_US_KY_TYPE1	USA driver's license - Kentucky (horizontal card, fence on the background, front)
		DriverLicense_US_KY_TYPE2	USA driver's license - Kentucky (vertical card, fence on the background, front)
		DriverLicense_US_LA_T	USA driver's license -

Document type	Profile name	Result scheme	Result description
		YPE1	Louisiana (horizontal card, emblem of Louisiana in the top-right corner on the background, front)
		DriverLicense_US_LA_TYPE2	USA driver's license - Louisiana (vertical card, emblem of Louisiana in the down-right corner on the background, front)
		DriverLicense_US_LA_TYPE3	USA driver's license - Louisiana (horizontal card, photo on the right, front)
		DriverLicense_US_LA_TYPE4	USA driver's license - Louisiana
		DriverLicense_US_MA_TYPE1	USA driver's license - Massachusetts (a stamp with a bird in the center, front)
		DriverLicense_US_MA_TYPE2	USA driver's license - Massachusetts (a personal photo on the left and a round stamp in the top-left corner of the photo, front)
		DriverLicense_US_MA_TYPE3	USA driver's license - Massachusetts (a personal photo on the right and the contour of state on the background, front)

Document type	Profile name	Result scheme	Result description
		DriverLicense_US_MA_TYPE4	USA driver's license - Massachusetts (vertical card, front)
		DriverLicense_US_MA_TYPE5	USA driver's license - Massachusetts (front side)
		DriverLicense_US_MD_TYPE1	USA driver's license - Maryland (the flag of the state in the top-left corner and a star on the top, front)
		DriverLicense_US_MD_TYPE2	USA driver's license - Maryland (a crab in the top right corner and a coat of arms of the state on the background, front)
		DriverLicense_US_ME_TYPE1	USA driver's license - Maine (a moose on the background, front)
		DriverLicense_US_ME_TYPE3	USA driver's license - Maine (sunset view on the top, front)
		DriverLicense_US_MI_TYPE1	USA driver's license - Michigan (bridge on the top, horizontal card, front)
		DriverLicense_US_MI_TYPE2	USA driver's license - Michigan (bridge on the top, vertical card, front)
		DriverLicense_US_MI_TYPE3	USA Operator License - Michigan (bridge on

Document type	Profile name	Result scheme	Result description
			the top, horizontal card, front)
		DriverLicense_US_MN_TYPE1	USA driver's license - Minnesota (emblem of Minnesota on the background, front)
		DriverLicense_US_MO_TYPE1	USA driver's license - Missouri (emblem of Missouri on the background, front)
		DriverLicense_US_MO_TYPE2	USA driver's license - Missouri (building on the background, front)
		DriverLicense_US_MO_TYPE3	USA driver's license - Missouri (emblem of Missouri on the background, vertical orientation)
		DriverLicense_US_MS_TYPE1	USA driver's license - Mississippi (DL sign on the top, front)
		DriverLicense_US_MS_TYPE2	USA driver's license - Mississippi (building on the background, front)
		DriverLicense_US_MS_TYPE3	USA driver's license - Mississippi (building on the background, front)
		DriverLicense_US_MS_TYPE4	USA driver's license - Mississippi

Document type	Profile name	Result scheme	Result description
		DriverLicense_US_MT_TYPE1	USA driver's license - Montana (DL sign on the top and emblem of Montana on the background, front)
		DriverLicense_US_MT_TYPE2	USA driver's license - Montana (mountains and stars on the background, front)
		DriverLicense_US_NC_TYPE1	USA driver's license - North Carolina (lighthouse on the bottom on the background, front)
		DriverLicense_US_NC_TYPE2	USA driver's license - North Carolina (building on the middle on the background, front)
		DriverLicense_US_NC_TYPE3	USA driver's license - North Carolina (a vertical card, building on the middle on the background, front)
		DriverLicense_US_NC_TYPE4	USA driver's license - North Carolina (a vertical card, building on the middle on the background, front)
		DriverLicense_US_ND_TYPE1	USA driver's license - North Dakota (a horizontal card; letters DL in the top-right corner, front side)
		DriverLicense_US_ND_T	USA driver's license -

Document type	Profile name	Result scheme	Result description
		YPE2	North Dakota (a horizontal card; horses on the background, front side)
		DriverLicense_US_NE_TYPE1	USA driver's license - Nebraska (a horizontal card; the great seal of Nebraska in the top-left corner, front side)
		DriverLicense_US_NE_TYPE2	USA driver's license - Nebraska (a vertical card; a star in the circle in the top-right corner, front side)
		DriverLicense_US_NH_TYPE1	USA driver's license - New Hampshire (a horizontal card; the contour of New Hampshire in the top-right corner, front side)
		DriverLicense_US_NH_TYPE2	USA driver's license - New Hampshire (a horizontal card; a small personal photo in the middle on the background, front side)
		DriverLicense_US_NH_TYPE3	USA driver's license - New Hampshire (a horizontal card; the circle of New Hampshire in the upper-left corner, front side)
		DriverLicense_US_NJ_T	USA driver's license -

Document type	Profile name	Result scheme	Result description
		YPE1	New Jersey (a horizontal card; a small personal photo in the bottom-right corner on the background, front side)
		DriverLicense_US_NJ_TYPE2	USA driver's license - New Jersey (a vertical card; a small personal photo in the middle on the right, front side)
		DriverLicense_US_NM_TYPE1	USA driver's license - New Mexico (a horizontal card; flag of New Mexico in the top-left corner, front side)
		DriverLicense_US_NM_TYPE2	USA driver's license - New Mexico (a vertical card; flag of New Mexico in the top-left corner, front side)
		DriverLicense_US_NV_TYPE1	USA driver's license - Nevada (a horizontal card; an eagle in the bottom-right corner, front side)
		DriverLicense_US_NV_TYPE2	USA driver's license - Nevada (a vertical card; an eagle in the bottom-right corner, front side)
		DriverLicense_US_NV_TYPE3	USA driver's license - Nevada (a horizontal card; the great seal of

Document type	Profile name	Result scheme	Result description
			Nevada in the top-left corner of personal photo on the right, front side)
		DriverLicense_US_NY_TYPE1	USA driver's license - New York (a horizontal card, emblem of New York on the right and statue of Liberty on the left on the background, front)
		DriverLicense_US_NY_TYPE2	USA driver's license - New York (statue of Liberty on the right on the background, front)
		DriverLicense_US_NY_TYPE3	USA driver's license - New York (emblem of New York on the middle on the background and landscape on the top, front)
		DriverLicense_US_NY_TYPE4	USA driver's license - New York (a vertical card, emblem of New York on the bottom and statue of Liberty in the top-left corner on the background, front)
		DriverLicense_US_OH_TYPE1	USA driver's license - Ohio (horizontal card, flag of Ohio in the down-left corner, front)
		DriverLicense_US_OH_T	USA driver's license -

Document type	Profile name	Result scheme	Result description
		YPE2	Ohio (vertical card, flag of Ohio on the bottom, front)
		DriverLicense_US_OH_TYPE3	USA driver's license - Ohio (horizontal card, emblem of Ohio on the background and star in the top-right corner, front)
		DriverLicense_US_OH_TYPE4	USA driver's license - Ohio (vertical card, emblem of Ohio on the background and star in the top-right corner, front)
		DriverLicense_US_OK_TYPE1	USA driver's license - Oklahoma (horizontal card, photo on the left and right, front)
		DriverLicense_US_OK_TYPE2	USA driver's license - Oklahoma (vertical card, photo on the left and right, front)
		DriverLicense_US_OK_TYPE3	USA driver's license - Oklahoma (horizontal card, photo on the right and middle, front)
		DriverLicense_US_OR_TYPE1	USA driver's license - Oregon (a horizontal card; the seal of Oregon in the top-right corner of the personal photo, front side)
		DriverLicense_US_OR_T	USA driver's license -

Document type	Profile name	Result scheme	Result description
		YPE2	Oregon (a horizontal card; the seal of Oregon in the top-left corner of the personal photo, front side)
		DriverLicense_US_PA_TYPE1	USA driver's license - Pennsylvania (a horizontal card; the state's name is written vertically on the left, front side)
		DriverLicense_US_PA_TYPE2	USA driver's license - Pennsylvania (a vertical card; letters JR in the bottom-right corner and a personal photo in the bottom-left corner, front side)
		DriverLicense_US_PA_TYPE3	USA driver's license - Pennsylvania (a vertical card; a small personal photo with letters DL in the bottom-right corner, front side)
		DriverLicense_US_PA_TYPE4	USA driver's license - Pennsylvania (a horizontal card; a small personal photo with letters CDL in the bottom-right corner, front side)
		DriverLicense_US_RI_TYPE1	USA driver's license - Rhode Island (a horizontal card; letters DL in the bottom-right corner, front side)

Document type	Profile name	Result scheme	Result description
		DriverLicense_US_RI_TYPE2	USA driver's license - Rhode Island (a horizontal card; a bridge on the background and a small personal photo in the top-right corner, front side)
		DriverLicense_US_SC_TYPE1	USA driver's license - South Carolina (emblem of South Carolina on the background, front)
		DriverLicense_US_SC_TYPE2	USA driver's license - South Carolina (flag of South Carolina on the background, front)
		DriverLicense_US_SC_TYPE3	USA driver's license - South Carolina (a vertical card; a palm on the background, front)
		DriverLicense_US_SC_TYPE4	USA driver's license - South Carolina (a horizontal card, a palm on the background, front)
		DriverLicense_US_SC_TYPE5	USA driver's license - South Carolina (a horizontal card; South Carolina State House on the background, front)
		DriverLicense_US_SC_TYPE6	USA driver's license - South Carolina (a vertical card; South Carolina State House

Document type	Profile name	Result scheme	Result description
			on the background, front)
		DriverLicense_US_SD_TYPE1	USA driver's license - South Dakota (Rushmore on the background, front)
		DriverLicense_US_SD_TYPE2	USA driver's license - South Dakota (emblem of South Dakota on the background, front)
		DriverLicense_US_TN_TYPE1	USA driver's license - Tennessee (building on the background, front)
		DriverLicense_US_TN_TYPE2	USA driver's license - Tennessee (flag of Tennessee in the top-right corner, front)
		DriverLicense_US_TN_TYPE3	USA driver's license - Tennessee (flag of Tennessee in the top-right corner, front)
		DriverLicense_US_TX_TYPE1	USA driver's license - Texas (a horizontal card; United States Capitol on the background, front side)
		DriverLicense_US_TX_TYPE2	USA driver's license - Texas (a vertical card; United States Capitol on the background, front side)

Document type	Profile name	Result scheme	Result description
		DriverLicense_US_TX_TYPE3	USA driver's license - Texas (a horizontal card; the seal of Texas in the top-right corner and flag of Texas in the top-left corner, front side)
		DriverLicense_US_TX_TYPE4	USA driver's license - Texas (a horizontal card; the seal of Texas in the top-right corner and flag of Texas in the top-left corner, front side)
		DriverLicense_US_UT_TYPE1	USA driver's license - Utah (a horizontal card; United States Capitol on the background, front side)
		DriverLicense_US_UT_TYPE2	USA driver's license - Utah (a horizontal card; the great seal of Utah on the background, front side)
		DriverLicense_US_UT_TYPE3	USA driver's license - Utah (a vertical card; United States Capitol on the background, front side)
		DriverLicense_US_UT_TYPE4	USA driver's license - Utah (a vertical card; United States Capitol on the background, front side)
		DriverLicense_US_VA_TYPE1	USA driver's license - Virginia (a horizontal

Document type	Profile name	Result scheme	Result description
			card; a new sample with the seal of Virginia in the middle on the background, front side)
		DriverLicense_US_VA_TYPE2	USA driver's license - Virginia (a horizontal card; a star in the circle in the top-right corner and a small personal photo on the right, front side)
		DriverLicense_US_VA_TYPE3	USA driver's license - Virginia (a vertical card; the seal of Virginia in the middle on the background, front side)
		DriverLicense_US_VA_TYPE4	USA driver's license - Virginia (a horizontal card; an old sample with the seal of Virginia in the middle on the background, front side)
		DriverLicense_US_VA_TYPE5	USA driver's license - Virginia (a vertical card; flowers on the background, front side)
		DriverLicense_US_VI_TYPE1	USA driver's license - Virginia Island (emblem of Virginia Island on the background)
		DriverLicense_US_VT_TYPE1	USA operator's license - Vermont (flag of USA on the

Document type	Profile name	Result scheme	Result description
			left and name of state on the left, front)
		DriverLicense_US_VT_TYPE2	USA operator's license - Vermont (name of state on the top-middle, front)
		DriverLicense_US_VT_TYPE3	USA operator's license - Vermont (flag of USA on the left and name of state on the left, front)
		DriverLicense_US_WA_TYPE1	USA driver's license - Washington (George Washington in the top-left corner, front)
		DriverLicense_US_WA_TYPE2	USA driver's license - Washington (tree on the down-middle, front)
		DriverLicense_US_WA_TYPE3	USA driver's license - Washington (a national flag on the right side of a personal photo, front)
		DriverLicense_US_WI_TYPE1	USA driver's license - Wisconsin (house on the top and emblem of Wisconsin on the background, front)
		DriverLicense_US_WI_TYPE2	USA driver's license - Wisconsin (building on the background, front)
		DriverLicense_US_WI_T	USA driver's license -

Document type	Profile name	Result scheme	Result description
		YPE3	Wisconsin (flag of USA in the top-left corner, front)
		DriverLicense_US_WI_TYPE4	USA driver's license - Wisconsin (flag of USA in the top-left corner, front)
		DriverLicense_US_WI_TYPE5	USA driver's license - Wisconsin (star in circle in the top-right corner, front)
		DriverLicense_US_WI_TYPE6	USA driver's license - Wisconsin (star in circle in the top-right corner, front)
		DriverLicense_US_WV_TYPE1	USA driver's license - West Virginia (a coat of arms of a state in the top-left corner and the contour of state in the top-right corner, front)
		DriverLicense_US_WY_TYPE1	USA driver's license - Wyoming (a coat of arms of a state in the top-left corner and mountains on the background, front)
		DriverLicense_US_WY_TYPE2	USA driver's license - Wyoming (a mountain on the background on the top, front)
Uzbek driver's license	DriverLicense_UZ	DriverLicense_UZ_TYPE1	Uzbek driver's license (UZ sign on the right on the background, front side)

Document type	Profile name	Result scheme	Result description
		DriverLicense_UZ_TYPE 2	
		DriverLicense_UZ_TYPE 3	
Vietnamese driver's license	DriverLicense_VN	DriverLicense_VN_TYPE 1	Vietnamese driver's license (a circle watermark on the bottom of personal photo , front side)
USA permanent residency card (Green card)	GreenCard_US	GreenCard_US_TYPE1	USA permanent residency card, also known as Green card (The building with columns and the Statue of Liberty on the background, front side)
Japanese health insurance card	HealthInsuranceCard_JP	HealthInsurance_JP_TYPE1	Japanese health insurance card ((a square in the bottom-right
Russian health insurance	HealthInsurance_RU	HealthInsurance_RU_TYPE1	Russian health Insurance (round stamp in the down-left corner, front side)
		HealthInsurance_RU_TYPE2	Russian health Insurance (round stamp in the down-right corner, front side)
		HealthInsurance_RU_TYPE3	Russian health Insurance (national coat of arms in the top-left corner and chip in the middle on the left, front side)

Document type	Profile name	Result scheme	Result description
		HealthInsurance_RU_TYPE4	Russian Health Insurance (the coat of arms of Moscow in the top-right corner, front side)
International bank account number	IBAN	IBAN	International bank account number
UAE ID card	ID_AE	ID_AE_TYPE1	UAE ID card (the emblem of United Arab Emirates on the top in the middle, front side)
Albanian ID card	ID_AL	ID_AL_TYPE1	Albanian ID Card (Albanian emblem in the background, front side; Albanian emblem in the background and top-left corner, back side)
Armenian ID card	ID_AM	ID_AM_TYPE1	Armenian ID card (Armenian emblem on the background, front)
Austrian ID card	ID_AT	ID_AT_TYPE1	Austrian ID card (red stripes on the left, back; Austrian emblem on the left)
Belgian ID card	ID_BE	ID_BE_TYPE1	Belgian ID card (front side)
		ID_BE_TYPE2	
		ID_BE_TYPE3	
Bulgarian ID card	ID_BG	ID_BG_TYPE1	Bulgarian ID card

Document type	Profile name	Result scheme	Result description
			(Bulgarian coat of arms on the right side, front side)
		ID_BG_TYPE2	Bulgarian ID card (Bulgarian flag in the left-top corner, front side)
Bahrain ID card	ID_BH	ID_BH_TYPE1	Bahrain ID card (front)
Brazilian ID card	ID_BR	ID_BR_TYPE1	Brazilian ID card (Brazilian emblem on the background, new type)
		ID_BR_TYPE2	Brazilian ID card (Brazilian emblem on the background, old type, front)
Swiss ID card	ID_CH	ID_CH_TYPE1	Swiss ID card (swiss flag in the top-left corner, stamp with cross in the top-right corner, front side)
Chilean ID card	ID_CL	ID_CL_TYPE1	Chile ID card (front side)
Chinese ID card	ID_CN	ID_CN_TYPE1	Chinese ID card
Cyprus ID card	ID_CY	ID_CY_TYPE1	ID card of Cyprus (Cyprus coat of arms on the background, front side)
		ID_CY_TYPE2	ID card of Cyprus (Cyprus coat of arms on the top-left corner and biometric symbol,

Document type	Profile name	Result scheme	Result description
			front side)
Czech ID card	ID_CZ	ID_CZ_TYPE1	Czech ID card (CZ sign on the right and rings on the background, front)
		ID_CZ_TYPE2	Czech ID card (old type, Czech emblem in the down-left corner, back; stamp in the form of a tree, front)
		ID_CZ_TYPE3	Czech ID card (Czech emblem in the down-left corner, back; stamp in the form of a tree, front)
German ID card	ID_DE	ID_DE_TYPE1	German ID card (the coat of arms of Germany on the background, front side; three-line MRZ on the bottom, back side)
		ID_DE_TYPE2	German ID card (the coat of arms of Germany on the background and two-line MRZ on the bottom, front side)
Estonian ID card	ID_EE	ID_EE_TYPE1	Estonian ID card (a horizontal card; Estonian flag in the upper left corner; lion in the upper right, front)
		ID_EE_TYPE2	Estonian ID card (a

Document type	Profile name	Result scheme	Result description
			horizontal card; Estonian emblem in the upper left corner; flag on the background, front)
Egyptian ID card	ID_EG	ID_EG_TYPE1	Egyptian ID card (Egyptian pyramids on the background, front side)
Spanish ID card	ID_ES	ID_ES_TYPE1	Spanish ID card (card-size, old one; a personal photo on the right and biometric symbol in the top-left corner, front side)
		ID_ES_TYPE2	Spanish ID card (card-size, new one; a personal photo on the left and chip on the left, front side)
Finnish ID card	ID_FI	ID_FI_TYPE1	Finnish ID card (chip on the left side and a personal photo in the bottom-right corner, front side)
		ID_FI_TYPE2	Finnish ID card (a personal photo on the left side and stamp with photo in the middle, front side)
French ID card	ID_FR	ID_FR_TYPE1	French ID card (front side)
Georgian ID card	ID_GE	ID_GE_TYPE1	Georgian ID card (a horizontal card; country outline in the

Document type	Profile name	Result scheme	Result description
			upper right; dotted lines on the background, front)
Hong Kong ID card	ID_HK	ID_HK_TYPE1	Hong Kong ID card (stamp with a photo under the chip, front side)
Croatian ID card	ID_HR	ID_HR_TYPE1	Croatian ID card (older type, front side)
		ID_HR_TYPE2	Croatian ID card (newer type, front side)
Hungarian ID card	ID_HU	ID_HU_TYPE1	Hungarian ID card (Hungarian emblem in the top-left corner, front)
		ID_HU_TYPE2	Hungarian ID card (Hungarian emblem in the top-right corner and biometric symbol in the top, front)
Israel ID card	ID_IL	ID_IL_TYPE1	ID card of Israel (chip on the left and a personal photo on the right, front side)
		ID_IL_TYPE2	ID card of Israel (a personal photo in the top-left corner and the coat of arms of Israel on the background, front side)
Italian ID card	ID_IT	ID_IT_TYPE1	Italian ID card (Italian

Document type	Profile name	Result scheme	Result description
			emblem on the top, page 1)
		ID_IT_TYPE2	Italian ID card (Italian emblem in the top-left corner, front)
		ID_IT_TYPE3	Italian ID card (biometric symbol the top-left corner, page 1)
Kyrgyz ID card	ID_KG	ID_KG_TYPE1	Kyrgyz ID card (the coat of arms of Kyrgyzstan in the middle on the top, front side)
		ID_KG_TYPE2	Kyrgyz ID card (the coat of arms of Kyrgyzstan in the top-left corner and biometric symbol in the top-right corner, front side)
Kuwait ID card	ID_KW	ID_KW_TYPE1	Kuwait ID card (static fields in one language on both sides, front side)
		ID_KW_TYPE2	Kuwait ID card (static fields in one language on one side, front side)
Kazakh ID card	ID_KZ	ID_KZ_TYPE1	Kazakhstan ID card with 2-line MRZ (signature frame in the bottom, back)
		ID_KZ_TYPE2	Kazakhstan ID card

Document type	Profile name	Result scheme	Result description
			with 3-line MRZ (Kazakh flag in the down-right corner, front)
Lithuanian ID card	ID_LT	ID_LT_TYPE1	Lithuanian ID card (biometric symbol in the top-left corner and chip on the left, front side)
		ID_LT_TYPE2	Lithuanian ID card (star watermark on the background, front side)
Luxembourgian ID card	ID_LU	ID_LU_TYPE1	Luxembourgian ID card (biometric symbol in the top-left corner, front)
		ID_LU_TYPE2	Luxembourgian ID card (the contour of Luxembourg in the top-right corner, front, new type)
		ID_LU_TYPE3	Luxembourgian ID card (the contour of Luxembourg in the top- right corner, front, old type)
		ID_LU_TYPE4	Luxembourgian ID card (number field on the bottom, front)
Latvian ID card	ID_LV	ID_LV_TYPE1	Latvian ID card (the coat of arms of Latvia in the top-right corner, front side)

Document type	Profile name	Result scheme	Result description
Moldavian ID card	ID_MD	ID_MD_TYPE1	ID card of Republic of Moldova (emblem of Republic of Moldova on the background, front)
		ID_MD_TYPE2	ID card of Republic of Moldova (blue stamp with emblem of Republic of Moldova in the left and pink background, front)
		ID_MD_TYPE3	ID card of Republic of Moldova (biometric symbol in the top-right corner, front)
Macedonian ID card	ID_MK	ID_MK_TYPE1	Macedonian ID card (name of document in the top is written on two languages, front side)
		ID_MK_TYPE2	Macedonian ID card (name of document in the top is written on three languages, front side)
Mexican ID card	ID_MX	ID_MX_TYPE1	Mexican ID card (Mexican emblem in the top-left corner and MEXICO-description on the top, front)
		ID_MX_TYPE2	Mexican ID card (Mexican emblem in the top-left corner and a copy of personal photo on the right, front)

Document type	Profile name	Result scheme	Result description
		ID_MX_TYPE3	Mexican ID card (Mexican emblem in the top-left corner, front)
		ID_MX_TYPE4	Mexican ID card (Text Mexico at the middle top and green and red stripes under it, front)
Malaysian ID card	ID_MY	ID_MY_TYPE1	Malaysian ID card (Malaysian flag in the top-right corner and a chip on the left, main page)
Nigerian ID card	ID_NG	ID_NG_TYPE1	Nigerian ID card (emblem of Nigeria in the top-left corner, front side)
Norwegian ID card	ID_NO	ID_NO_TYPE1	Norwegian ID card (a chip on the left and Norwegian coat of arms on the top, front side)
Polish ID card	ID_PL	ID_PL_TYPE1	Polish ID card (emblem in the top-right corner, front side)
		ID_PL_TYPE2	Polish ID card (emblem in the top-left corner, two photos on the card, front side)
		ID_PL_TYPE3	Polish ID card (blue emblem in the top-left corner, only one photo on the card,

Document type	Profile name	Result scheme	Result description
			front side)
		ID_PL_TYPE4	Polish ID card (transparent emblem in the top-left corner, only one photo on the card, front side)
Portuguese ID card	ID_PT	ID_PT_TYPE1	Portuguese ID card (a chip on the left and a stamp with a cross of shields in the top-left corner, front side)
Romanian ID card	ID_RO	ID_RO_TYPE1	Romanian ID card (national flag on the top and the coat of arms on the background, front side)
Serbian ID card	ID_RS	ID_RS_TYPE1	Serbian ID card (the coat of arms of Serbia in the top-left corner and a shield on the right, front side)
Russian ID card	ID_RU	ID_RU_MILITARY_TYPE1	Russian military ID card (front side)
		ID_RU_MILITARY_TYPE2	Russian military ID card (front side)
		ID_RU_MILITARY_TYPE3	Russian military ID card (front side)
		ID_RU_MILITARY_TYPE4	Russian military ID card (front side)
		ID_RU_MILITARY_TYPE5	Russian military ID

Document type	Profile name	Result scheme	Result description
			card (front side)
		ID_RU_POLICE_TYPE1	Russian police ID card (front side)
		ID_RU_PROSECUTOR_TYPE1	Russian prosecutor ID card (name of document is written in two lines, page 2)
		ID_RU_PROSECUTOR_TYPE2	Russian prosecutor ID card (name of document is written in one line, page 2)
		ID_RU_SOLDIER_TYPE1	Russian soldier ID card (front side)
Singapore ID card	ID_SG	ID_SG_TYPE1	Singapore ID card (national coat of arms in the top-right corner, front side)
Slovenian ID card	ID_SI	ID_SI_TYPE1	Slovenian ID card (the Slovenian coat of arms on the top, front side; cavalryman motif in the middle above the MRZ zone, back side)
Slovakian ID card	ID_SK	ID_SK_TYPE1	ID card of Slovakia (a round stamp in the top-right corner of the photo and leaves in the top-right corner, front side)
		ID_SK_TYPE2	ID card of Slovakia (a round stamp under the photo and the national coat of arms

Document type	Profile name	Result scheme	Result description
			on the background, main page)
Salvadorean ID card	ID_SV	ID_SV_TYPE1	Salvadorean ID card (the national flag on the top-left corner and a coat of arms on the top-right corner, main page)
Turkish ID card	ID_TR	ID_TR_TYPE1	Turkish ID card (national emblem of the Republic of Turkey on the right and a personal photo on the left, front side)
		ID_TR_TYPE2	Turkish ID card (national emblem of the Republic of Turkey on the left and a personal photo on the right, front side)
Ukrainian ID card	ID_UA	ID_UA_TYPE1	Ukrainian ID card (Ukrainian flag in the top-right corner, card-sized)
South African Republic ID card	ID_ZA	ID_ZA_TYPE1	South African Republic ID card (the national flag in the top-left corner and a round stamp in the bottom-left corner of photo, main page)
Russian INN	INN_RU	INN_RU_CITIZEN_TYPE1	Russian INN for citizens (main page)
		INN_RU_CITIZEN_TYPE2	Russian INN for citizens (main page)

Document type	Profile name	Result scheme	Result description
		INN_RU_CITIZEN_TYPE3	Russian INN for citizens (main page)
		INN_RU_CITIZEN_TYPE4	Russian INN for citizens (main page)
		INN_RU_ENTITY_TYPE1	Russian entity INN (main page)
		INN_RU_ENTITY_TYPE2	Russian entity INN (main page)
Albanian passport	InternationalPassport_AL	InternationalPassport_AL_TYPE1	Albanian passport (Albanian emblem in the background)
		InternationalPassport_AL_TYPE2	Albanian passport (Albanian emblem in the background and top-left corner, red horizontal line along the entire document)
Armenian passport	InternationalPassport_AM	InternationalPassport_AM_TYPE2	Armenian passport (line of patterns on the top, main page)
		InternationalPassport_AM_TYPE3	Armenian passport (new type, main page)
Austrian passport	InternationalPassport_AT	InternationalPassport_AT_TYPE1	Austrian passport (eagle in the top-left corner, main page)
		InternationalPassport_AT_TYPE2	Austrian passport (eagle in the top-left corner, main page, with residence field)
Brazilian passport	InternationalPassport_B	InternationalPassport_B	Brazilian passport

Document type	Profile name	Result scheme	Result description
	R	R_TYPE1	(Brazil on the background, main page)
		InternationalPassport_BR_TYPE2	Brazilian passport (barcode on the bottom, main page)
Canadian passport	InternationalPassport_CA	InternationalPassport_CA_TYPE1	Canadian passport (Canadian national symbols in the top-right corner, main page)
		InternationalPassport_CA_TYPE2	Canadian passport (biometric symbol in the top-right corner and Canadian coat of arms on the background, main page)
Chinese passport	InternationalPassport_CN	InternationalPassport_CN_TYPE1	Chinese passport (China from the bottom in the background and barcode on the left, main page)
		InternationalPassport_CN_TYPE3	Chinese passport (biometric symbol in the top-right corner and a flower on the background, main page)
		InternationalPassport_CN_TYPE4	Chinese passportand CHN sign on the right, main page)
		InternationalPassport_CN_TYPE5	Chinese passport (biometric symbol in the top-left corner,

Document type	Profile name	Result scheme	Result description
			main page)
		InternationalPassport_CN_TYPE6	Chinese passport (green lotus on the left, main page)
Czech passport	InternationalPassport_CZ	InternationalPassport_CZ_TYPE1	Czech passport (stamp on the top, main page)
German passport	InternationalPassport_DE	InternationalPassport_DE_TYPE1	German passport (the coat of arms of Germany on the right side, main page)
		InternationalPassport_DE_TYPE2	German passport (the coat of arms of Germany in the top-left corner and in the middle on the background, main page)
		InternationalPassport_DE_TYPE3	German passport (stamp with the coat of arms of Germany under the photo, main page)
Algerian passport	InternationalPassport_DZ	InternationalPassport_DZ_TYPE1	Algerian passport (the contour of the country on the right and sun in the bottom-right of the photo, main page)
Spanish passport	InternationalPassport_ES	InternationalPassport_ES_TYPE1	Spanish passport (new biometric passport, biometric symbol on the top, main page)
		InternationalPassport_E	Spanish passport (old

Document type	Profile name	Result scheme	Result description
		S_TYPE2	biometric, biometric symbol in the top-left corner and a personal photo on the background, main page)
		InternationalPassport_ES_TYPE3	Spanish passport (one-line MRZ on the bottom, main page)
		InternationalPassport_ES_TYPE4	Spanish passport (diplomatic document, main page)
		InternationalPassport_ES_TYPE5	Spanish passport
Georgian passport	InternationalPassport_GE	InternationalPassport_GE_TYPE1	Georgian passport (country emblem in the upper left; has owner's signature, main page)
		InternationalPassport_GE_TYPE2	Georgian passport (country emblem in the upper left; has patterns on the corners of the photo, main page)
		InternationalPassport_GE_TYPE3	Georgian passport (circle with patterns in the center, main page)
		InternationalPassport_GE_TYPE4	Georgian passport
Greek passport	InternationalPassport_GR	InternationalPassport_GR_TYPE1	Greek passport (Greek emblem on the background, main

Document type	Profile name	Result scheme	Result description
			page)
Croatian passport	InternationalPassport_HR	InternationalPassport_HR_TYPE1	Croatian passport (RH sign on the left, main page)
Hungarian passport	InternationalPassport_HU	InternationalPassport_HU_TYPE1	Hungarian passport (stamp in the top-right corner, main page)
Israel passport	InternationalPassport_IL	InternationalPassport_IL_TYPE1	Passport of Israel (the coats of arms of Israel all over the background, main page)
		InternationalPassport_IL_TYPE2	Passport of Israel (the coat of arms of Israel in the middle of the background, main page)
Indian passport	InternationalPassport_IN	InternationalPassport_IN_TYPE1	Indian passport (lines on the background, main page)
Italian passport	InternationalPassport_IT	InternationalPassport_IT_TYPE1	Italian passport (Italian emblem on the background, main page)
Japanese passport	InternationalPassport_JP	InternationalPassport_JP_TYPE1	Japanese passport (Mount Fuji on the background and the Government Seal of Japan in the top-left and -right corner, main page)
Kyrgyz passport	InternationalPassport_KG	InternationalPassport_KG_TYPE1	Kyrgyz passport (the coat of arms of

Document type	Profile name	Result scheme	Result description
			Kyrgyzstan in the top-right corner near the small personal photo, main page)
Kazakh passport	InternationalPassport_KZ	InternationalPassport_KZ_TYPE1	Kazakh passport (stamp in the top-left corner on the photo, main page)
		InternationalPassport_KZ_TYPE2	Kazakh passport (Kazakh emblem on the bottom, main page)
Passport of Luxembourg	InternationalPassport_LU	InternationalPassport_LU_TYPE1	Passport of Luxembourg (line on the background, main page)
		InternationalPassport_LU_TYPE2	Passport of Luxembourg (tiger on the background, main page)
Moldavian passport	InternationalPassport_MD	InternationalPassport_MD_TYPE1	Passport of Republic of Moldova (biometric symbol in the top-left corner, main page)
		InternationalPassport_MD_TYPE2	Passport of Republic of Moldova (vertical field nationality on the right, main page)
Philippine passport	InternationalPassport_PH	InternationalPassport_PH_TYPE1	Philippine passport (the Philippine coat of arms on the background, main page)

Document type	Profile name	Result scheme	Result description
		InternationalPassport_PH_TYPE2	Philippine passport (the Philippine flag in the top-left corner and biometric symbol on the top-right corner, main page)
Polish passport	InternationalPassport_PL	InternationalPassport_PL_TYPE1	Polish passport (map in the top-left corner)
		InternationalPassport_PL_TYPE2	Polish passport (stamp with Polish emblem in the top-left corner, main page)
Russian international biometric passport	InternationalPassport_RU	InternationalPassport_RU_BIOMETRIC	Russian international biometric passport (main page)
Swedish passport	InternationalPassport_SE	InternationalPassport_SE_TYPE1	Swedish passport (a biometric symbol in the top-right corner, main page)
		InternationalPassport_SE_TYPE2	Swedish passport (a square stamp in the top-right corner, main page)
Slovenian passport	InternationalPassport_SI	InternationalPassport_SI_TYPE1	Slovenian passport (a leaf in the top-right corner and a small personal photo on the right , main page)
Slovakian passport	InternationalPassport_SK	InternationalPassport_SK_TYPE2	Passport of Slovakia (SVK sign in the left-bottom corner of photo, main page)
		InternationalPassport_SK_TYPE3	Passport of Slovakia (inscription with the

Document type	Profile name	Result scheme	Result description
			name of country above the MRZ zone, main page)
Syrian passport	InternationalPassport_SY	InternationalPassport_SY_TYPE1	Passport of Syrian Arab Republic (national coat of arms on the top, main page)
Tajikistani passport	InternationalPassport_TJ	InternationalPassport_TJ_TYPE1	Passport of Tajikistan (the national flag in the top-left corner and a copy of personal photo in the right, main page)
		InternationalPassport_TJ_TYPE2	Passport of Tajikistan (a round stamp in the bottom-right corner of the photo, main page)
		InternationalPassport_TJ_TYPE3	Passport of Tajikistan
Turkish passport	InternationalPassport_TR	InternationalPassport_TR_TYPE1	Turkish passport (TR watermark in the top-right corner and a small personal photo on the right, main page)
		InternationalPassport_TR_TYPE2	Turkish passport (Turkish emblem in the top-center, main page)
Ukrainian passport	InternationalPassport_UA	InternationalPassport_UA_TYPE1	Ukrainian passport (Building on the right and the camera symbol in the upper left corner, front side)

Document type	Profile name	Result scheme	Result description
		InternationalPassport_UA_TYPE2	Ukrainian passport (Pink, yellow and gray wave in the center, front side)
British passport	InternationalPassport_UK	InternationalPassport_UK_TYPE1	British passport (bird on the background, main page)
		InternationalPassport_UK_TYPE2	British passport (compass in the top-left corner, main page)
		InternationalPassport_UK_TYPE3	British passport (a measuring device in the top-left corner, main page)
USA passport	InternationalPassport_US	InternationalPassport_US_TYPE1	American passport (only for children, main page)
		InternationalPassport_US_TYPE2	American passport (a national flag and coat of arms on the background, main page)
Uruguayan passport	InternationalPassport_UY	InternationalPassport_UY_TYPE1	Uruguayan passport (Large coat of arms on the background, front side)
		InternationalPassport_UY_TYPE2	Uruguayan passport (the text REPUBLICA ORIENTAL DEL URUGUAY round on the background, front side)
Uzbek passport	InternationalPassport_	InternationalPassport_	Uzbek passport (UZB

Document type	Profile name	Result scheme	Result description
	UZ	UZ_TYPE1	sign on the right, main page)
		InternationalPassport_UZ_TYPE2	Uzbek passport (the coat of arms of Uzbekistan on the background, main page)
		InternationalPassport_UZ_TYPE3	Uzbek passport
Machine-readable document zone	MRZ	MRZ_BG_VEHICLEREGISTRATION	MRZ-like zone of the Bulgarian vehicle registration document (3 lines, 30 characters each)
		MRZ_CH_DRIVERLICENSE	MRZ-like zone of the Swiss driver's license (3 lines, 9, 30 and 30 characters)
		MRZ_FR_ID	MRZ-like zone of the French national ID card (2 lines, 36 characters each)
		MRZ_MRP	ICAO Doc 9303 machine-readable passports (2 lines, 44 characters each)
		MRZ_MRV_A	ICAO Doc 9303 machine-readable visa MRV-A (2 lines, 44 characters each)
		MRZ_MRV_B	ICAO Doc 9303 machine-readable visa MRV-B (2 lines, 36 characters each)

Document type	Profile name	Result scheme	Result description
		MRZ_RU_PASSPORT	Russian passport (page 2, with signatures)
		MRZ_RU_VISA	MRZ-like zone of the Russian visa (2 lines, 44 characters each)
		MRZ_TD1	ICAO Doc 9303 machine-readable travel document TD-1 (3 lines, 30 characters each)
		MRZ_TD2	ICAO Doc 9303 machine-readable travel document TD-2 (2 lines, 36 characters each)
Russian marriage certificate	MarriageCertificate_RU	MarriageCertificate_RU_TYPE1	Russian marriage certificate (main page)
Russian migration card	MigrationCard_RU	MigrationCard_RU_TYPE1	Russian migration card (front side)
USA passport card	PassportCard_US	PassportCard_US_TYPE1	USA passport card (Blue background with pink and dark blue honeycombs, front side)
		PassportCard_US_TYPE2	USA passport card (Flag, coat of arms and mountain in the background, front side)
Belorussian passport	Passport_BY	Passport_BY_PAGE31_TYPE1	Belorussian passport (internal page)

Document type	Profile name	Result scheme	Result description
		Passport_BY_TYPE1	Belorussian passport (biodata page)
Russian passport	Passport_RU	Passport_RU	Russian passport (pages 2 and 3)
Brazilian residence license	ResidenceLicense_BR	ResidenceLicense_BR_TYPE1	Brazilian real estate license (Brazilian emblem on the top-left corner, back; hummingbird in the middle, front)
Austrian residence permit	ResidencePermit_AT	ResidencePermit_AT_TYPE1	Austrian residence permit (Austrian emblem on the left, front; vertical red inscription on the left, back)
		ResidencePermit_AT_TYPE2	Austrian residence permit (Austrian emblem on the left and biometric symbol on the top, front)
		ResidencePermit_AT_TYPE3	Austrian residence permit (Austrian emblem left top, front)
		ResidencePermit_AT_TYPE4	Austrian residence permit (Austrian emblem left top and bottom, front)
German residence permit	ResidencePermit_DE	ResidencePermit_DE_TYPE1	German residence permit (biometric symbol in the top-left corner and a bull above the photo, front side)

Document type	Profile name	Result scheme	Result description
Spanish residence permit	ResidencePermit_ES	ResidencePermit_ES_TYPE1	Spanish residence permit, (EU card size, blue-purple)
		ResidencePermit_ES_TYPE2	Spanish residence permit, (old card size, different colors, large E on the background)
Luxembourgian residence permit	ResidencePermit_LU	ResidencePermit_LU_TYPE1	Luxembourgian residence permit (the contour of Luxembourg in the top-left corner, front)
		ResidencePermit_LU_TYPE2	Luxembourgian residence permit (biometric symbol in the top-left corner, front)
		ResidencePermit_LU_TYPE3	Luxembourgian residence permit (bull with stars on the middle, front)
Russian residence permit	ResidencePermit_RU	ResidencePermit_RU_TYPE2	Russian biometric residence permit (main page)
Russian residence permit (old)	ResidencePermit_RU_OLD	ResidencePermit_RU_TYPE1	Russian residence permit (main page)
Slovenian residence permit	ResidencePermit_SI	ResidencePermit_SI_TYPE1	Slovenian residence permit (a bull in the middle on the background and sign of the European Union in the top-left corner, front side)
		ResidencePermit_SI_TY	Slovenian residence

Document type	Profile name	Result scheme	Result description
		PE2	permit (a bull above the personal photo and a biometric symbol on the top, front side)
Slovakian residence permit	ResidencePermit_SK	ResidencePermit_SK_TYPE1	Residence permit of Slovakia (biometric symbol on the top, front side)
		ResidencePermit_SK_TYPE2	Residence permit of Slovakia (parallelogram on the top, front side)
Russian insurance individual account number (SNILS)	SocialSecurityNumber_RU	SocialSecurityNumber_RU_TYPE1	Laminated SNILS (patterns in the right on the background, front)
		SocialSecurityNumber_RU_TYPE2	Card-size SNILS (old type, front)
Russian vehicle passport	VehiclePassport_RU	VehiclePassport_RU_TYPE1	Russian vehicle passport (front side)
Azerbaijan vehicle registration certificate	VehicleRegistration_AZ	VehicleRegistration_AZ_TYPE1	Azerbaijan vehicle registration certificate (document without a personal photo, AZ sign in the top-left corner and the flag of Azerbaijan near it, main page)
Belorussian vehicle registration certificate	VehicleRegistration_BY	VehicleRegistration_BY_TYPE1	Belorussian vehicle registration certificate
Czech vehicle registration certificate	VehicleRegistration_CZ	VehicleRegistration_CZ_TYPE1	Czech vehicle registration certificate

Document type	Profile name	Result scheme	Result description
			(the sign of the European Union with letters CZ in the top-left corner and leaves in the bottom-left corner, front side)
Kazakh vehicle registration certificate	VehicleRegistration_KZ	VehicleRegistration_KZ_TYPE1	Kazakh vehicle registration certificate (KZ sign on the top, back)
Russian vehicle registration certificate	VehicleRegistration_RU	VehicleRegistration_RU_TYPE1	Russian vehicle registration certificate (new type, stamp with car in the top-right corner, front)
		VehicleRegistration_RU_TYPE2	Russian vehicle registration certificate (old type, RUS sign on the top, front)
Slovakian vehicle registration certificate	VehicleRegistration_SK	VehicleRegistration_SK_TYPE1	Vehicle registration certificate of Slovakia (card-sized, national symbol in the top-right corner and sign of the European Union with letter SK in the top-left corner, front side)
Salvadorean vehicle registration	VehicleRegistration_SV	VehicleRegistration_SV_TYPE1	Salvadorean vehicle registration (a chip on the left, main page)
Ukrainian vehicle registration certificate	VehicleRegistration_UA	VehicleRegistration_UA_TYPE1	Ukrainian vehicle registration certificate (card-sized, coat of arms of Ukraine on background)

Document type	Profile name	Result scheme	Result description
Russian visa	Visa_RU	Visa_RU_TYPE1	Russian visa
USA visa	Visa_US	Visa_US_TYPE1	USA visa (Lincoln Memorial, Washington State Capitol on the background, main page)
Russian work permit	WorkPermit_RU	WorkPermit_RU_TYPE1	Russian work permit (front side)
Singapore work permit	WorkPermit_SG	WorkPermit_SG_TYPE1	Singapore work permit (barcode on the bottom and a mark with lion on the right, front side)

Regular Expressions

This section describes the regular expression syntax supported by the ABBYY Mobile Capture SDK engine for capturing custom data fields (see How to Capture a Custom Data Field).

Note: *All matches are always greedy (match as much as possible). The search stops at the first match: if a string contains two or more substrings matching your regular expression, only the first one (closest to the beginning) is matched.*

Supported syntax

Pattern	Syntax	Examples and comments
Literal	any character or text, except metacharacters \^$.|?*+()[{	*pill* matches "pill" in "caterpillar" *a* matches the first "a" in "caterpillar" but not the second (the search stops at the first match) Metacharacters are part of regular expression syntax; to match these literally, you have to escape them with a

Pattern	Syntax	Examples and comments
		backslash. If you want to match *1+1*, the correct expression is *1\+1* — otherwise "+" has a special meaning.
Any character	. (dot)	*s.t* matches "sat", "sit" but not "seat"
Character set	[]	*gr[ae]y* matches both "gray" and "grey" but not "greay"
Character range in a set	- (minus)	*[0-9]* matches a single digit concatenation is allowed: *[a-zA-Z0-9]* matches an alphanumeric character
Negated character set	[^]	*[^0-9]* matches anything that is not a digit
Shorthand classes	\s — any whitespace \S — anything that is not a whitespace \d — any digit \D — anything that is not a digit \w — a word character, which includes alphanumerics and punctuation marks \W — a non-word character \R — a new line character or the CR LF sequence \v — a new line character but not the CR LF sequence \V — a non-new line character \h — a horizontal white space character \H — anything except horizontal white space	
Non-printable characters	\n — line feed LF \r — carriage return CR	

Pattern	Syntax	Examples and comments
	\t — tab character \f — form feed \a — bell character \u0007 \e — escape character	
Unicode character	\uFFFF \x{FFFF}	*\u20AC* or *\x{20AC}* matches the euro currency sign.
Character by its hexadecimal index	\xFF	*\xA9* matches the copyright character in the Latin-1 character set
Alternation	|	*abc|123* matches either "abc" or "123" *|word* matches either an empty string "" or "word"
Repetitions	+ * ? {n} {n,m} {n,} {,m}	+ matches once or more times * matches zero or more times *?* matches zero times or once (optional match) *{n}* matches exactly n times *{n,m}* matches n to m times times *{n,}* matches n or more times *{,m}* matches zero or more times up to m Note that all repetitions are greedy (prefer to match as much as possible): *c.+r* will match "caterpillar", not stopping with "cater". If you want to match up to the first occurence of a certain character, use its negation: *c[^r]+r* will match "cater" in "caterpillar".
Grouping	()	*(word)*+ matches "word", "wordword" and so on

Unsupported syntax

The following regular expression syntax features are not yet supported in ABBYY Mobile Capture SDK:

- Anchors: ^ (beginning of a line), $ (end of a line), \b (word boundary) and its negation \B, and other.
- Lazy quantifiers such as +? or {n,m}? that prefer to match as few times as possible.
- Concatenation with nested character sets such as [[a-z][0-9]].
- Advanced features such as lookarounds, backreferences, possessive matches, named groups, non-capturing and atomic match groups, evaluation flag settings and other.

Copyright and Trademark Notices

LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MD5 message digest algorithm reference implementation
This software is derived in part from the RSA Data
Security, Inc. MD5 Message-Digest Algorithm

Protobuf:

This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in
  src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
  This file is copyrighted by Red Hat Inc.
- Atomicops support for AIX/POWER, located in
  src/google/protobuf/stubs/atomicops_internals_power.h.
  This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer;
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

Libzip:

Copyright (C) 1999-2014 Dieter Baron and Thomas Klausner
The authors can be contacted at <libzip@nih.at>
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY DIRECT,

INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Eigen:

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, you can obtain one at https://mozilla.org/MPL/2.0/.

zlib

zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.3, July 18th, 2005
Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler
This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.
Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:
1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.
Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu

LZMA SDK

LZMA SDK is placed in the public domain.
Anyone is free to copy, modify, publish, use, compile, sell, or distribute the original LZMA SDK code, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

dlmalloc

This is a version (aka dlmalloc) of malloc/free/realloc written by Doug Lea and released to the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/ Send questions, comments, complaints, performance data, etc to dl@cs.oswego.edu

HTML help

All rights, title, and copyrights in and to the SOFTWARE PRODUCT (including, but not limited to, any images, photographs, animations, video, audio, music, text, and "applets" incorporated into the SOFTWARE PRODUCT) and any copies of the SOFTWARE PRODUCT are owned by Microsoft or its suppliers. You may not copy the printed materials, if any, accompanying the SOFTWARE PRODUCT.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "{}"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Copyright 2013 Square, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and
limitations under the License.

All other trademarks and copyrights are the property of their respective owners.

Contact ABBYY

In this section you can find the contacts of ABBYY sales offices and technical support.

How to Buy

You can order ABBYY Mobile Capture or other ABBYY products by contacting an ABBYY office in your region. You can find contact details of the ABBYY offices in the ABBYY Contacts web-page.

Technical Support

If you have questions regarding the use of ABBYY Mobile Capture, please visit the ABBYY Knowledgebase or Developer Forum, to find answers to your questions or post your own questions in the forum. If neither of the mentioned sources was helpful, please contact ABBYY Technical Support by submitting a request at global ABBYY Help Center.