ABBYY®

ABBYY® Real-Time Recognition SDK®

Developer's Guide

Table of Contents

Introducing ABBYY Real-Time Recognition SDK 1

ABBYY Real-Time Recognition SDK provides a technology for recognizing text directly on the smartphone's camera preview screen.

Key features

Real-time OCR

Real-Time Recognition SDK does not require snapping a photo but offers the possibility to capture text on-the-fly using a series of images from the smartphone's camera preview screen. Combining several images enables Real-Time Recognition SDK to recognize text even in situation when it is hard to obtain a still photo of suitable quality for recognition. This makes real-time recognition more convenient and in many cases significantly faster than taking a picture of the text followed by OCR, or entering the text manually. On top of that, Real-Time Recognition SDK also supports recognizing an image from a file that allows it to process existing photos, working in the same way as traditional OCR.

Translation

Real-Time Recognition SDK provides built-in translation dictionaries for word-by-word and phrase-by-phrase translation. The dictionaries contain words and some common phrases for the main European languages as well as for Chinese and Japanese. Translation dictionaries are optimized to work on mobile devices.

Recognition of text from real-world objects

Real-Time Recognition SDK can locate texts in real-world scenes and extract meaningful text of any color from most backgrounds. This feature enables you to extract information from street signs, menus, etc.

Merging the recognition results

Images obtained from camera video stream often have noticeable defects (such as motion blur) which may lead to OCR errors. To increase recognition accuracy and eliminate random recognition errors, Real-Time Recognition SDK uses an intelligent aggregation mechanism, which combines recognition results from several video frames.

Data capture

Real-Time Recognition SDK can extract data from a document (for example, date, total amount, e-mails, codes, and other). All you have to do is set a regular expression that describes the required content, and the data capture engine will do the rest. If necessary, you can also specify validation rules to make sure that the information being extracted is the right one — when it does not satisfy validation rules, no data will be extracted at all. It is even much easier when it comes to processing machine-readable zones in documents (MRZ) or international bank account numbers (IBAN) — what you have to do is select a corresponding profile, and the data capture engine will extract all the necessary data.

Out-of-the-box document capture support

With Real-Time Recognition SDK you can easily add functionality to extract important fields from specific documents: passports, IDs, bank cards and others. If you would like to add support for these documents or a different kind of document to your app, please contact rtrsdk@abbyy.com.

For example, when reading a bank card, Real-Time Recognition SDK will automatically detect and extract the date of expiry, cardholder name, and card number, without requiring you to set specific rules or regular expressions.

Using the documentation

This Developer's Guide contains all the necessary information about ABBYY Real-Time Recognition SDK.

- **Guided Tour**
  This section will help you get started with ABBYY Real-Time Recognition SDK.
- **API Reference**
  The complete description of ABBYY Real-Time Recognition SDK Java API.
- **Specifications**
  The list of supported operating systems, hardware requirements, files necessary for distribution, acknowledgements of third-party solutions, etc.

See also the **help/javadoc** folder for the reference in Javadoc-generated HTML format.

Guided Tour

This section will help you to get started using ABBYY Real-Time Recognition SDK.

- How to Add the Library to Your Android Studio Project
- Step-by-step guides to the simple scenarios:
  - 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 Android Studio Project

To create an application which uses ABBYY Real-Time Recognition SDK to capture text from the camera preview, you will need to add the library and its assets to your project. This is required for new projects only — packaged code samples work out of the box.

1. If you are using a Maven or Ivy repository, add the ABBYY Real-Time Recognition SDK package there. If not, you can copy the library file (**abbyy-rtr-sdk-1.0.aar**) to your project or another folder and add this location as a flat repository. For example, add the following to the top-level **build.gradle** file in your project:

```
repositories {
    flatDir {
        dirs '<path to folder with the .AAR file>'
    }
}
```

2. Add the library dependency to the module-level **build.gradle** file. For example:

```
dependencies {
    compile(name:'abbyy-rtr-sdk-1.0', ext:'aar')
}
```

3. Copy the assets you need from the distribution to your project's assets (by default, **app/src/main/assets**). There are three types of resources used by the library: dictionaries, patterns, and translation dictionaries. See Distribution Kit for a detailed description of the necessary resources.

Important! *Your application needs an Internet connection to gather the information about the current state of the library. Include the following line into your* ***AndroidManifest.xml****:*

```
<uses-permission android:name="android.permission.INTERNET" />
```

How to Capture Text from Camera with Android

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 Real-Time Recognition SDK for Android 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 Real-Time Recognition SDK engine will automatically request 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 Real-Time Recognition SDK to recognize text even in situation when it is hard to obtain a still photo of suitable quality for recognition.

Note that Real-Time Recognition 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 Android Studio Project.*

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

1. Begin with the **Callback** interface implementation. Its methods will be used to pass the data to and from the recognition service. Here are the brief recommendations on what the methods should do:
   - The **onRequestLatestFrame** method should retrieve the image from the camera and pass it on to the **ITextCaptureService.submitRequestedFrame** method.
   - The **onFrameProcessed** method is where you work with the results, display them to the user, etc.
   - The **onError** method is for handling processing errors.
2. Call the **Engine.load** method on the UI thread to create an engine object via which all other objects may be created. This object should be reused for every new operation and should not be created again in the same activity.
3. Use the **createTextCaptureService** method of the **Engine** object to create a background recognition service (implementing the **ITextCaptureService** interface) on the UI thread. Only one instance of the service per application is necessary: multiple threads will be started internally.
4. Set up the processing parameters, according to the kind of text you expect to capture.
   The default text language is English; if you need other languages, specify them using the **setRecognitionLanguage** method.
5. When the camera is ready, call the **start** method of the **ITextCaptureService** interface. Its required input parameters are the size and orientation of the video frame and the rectangular area where to search for the text (e.g. if your application displays a highlighted rectangle in the center of the image, this rectangle should be specified as the "area of interest").
   The service will then start up several working threads and continue interacting with your application via the **Callback** interface.
6. Whenever the **Callback.onRequestLatestFrame** method is called, provide the current video frame from the camera by calling **ITextCaptureService.submitRequestedFrame**.

7. The **Callback.onFrameProcessed** method will be called on the UI thread to return the result when the frame is recognized.
   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 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 Available.
   The result consists of one or more text lines represented by objects of the **TextLine** class. Each **TextLine** contains information about the enclosing quadrangle for a single line of text and the recognized text as a string.
   Work with the results on your side.

8. When pausing or quitting the application, call the **ITextCaptureService.stop** method to terminate the processing threads.

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

How to Recognize Text on Photos

This guide explains how Real-Time Recognition SDK can be used as a common OCR solution, recognizing text on existing images.

How it Works

Real-Time Recognition 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 Android Studio Project.*

To implement the image recognition scenario, follow these steps:

1. Begin with the **TextRecognitionCallback** interface implementation. Its methods will be used to get status information and control the recognition process. Here are the brief recommendations on what the methods should do:
   - The **onProgress** method is used to report recognition status. It also allows you to interrupt the recognition process.
   - The **onTextOrientationDetected** is called when image orientation (or change in orientation) is detected.
   - The **onError** method is for handling processing errors.
2. Call the **Engine.load** method to create an engine object via which all other objects may be created. This object should be reused for every new operation and should not be created again in the same activity.

3. Use the **createRecognitionCoreAPI** method of the **Engine** object to create a recognizer object (implementing the **IRecognitionCoreAPI** interface). Use this object on the thread on which it was created; you may also create several objects on different threads and use them concurrently. All **IRecognitionCoreAPI** interface method calls are synchronous (will not return until the operation is completed), so the recognizer should not be used on the UI thread.

4. If you want to change recognition settings, use **IRecognitionCoreAPI.getTextRecognitionSettings** to get a **TextRecognitionSettings** object, then use its methods to set the recognition area and text language.
   - If you are using a recognition language different from English, specify it using the **TextRecognitionSettings.setRecognitionLanguage** method. Multiple languages are also supported, although setting too many languages may decrease recognition performance.
   - It is also recommended to call the **TextRecognitionSettings.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.
5. You can also set the number of processing threads using the object returned by **IRecognitionCoreAPI.getProcessingSettings** (**ProcessingSettings** interface).
6. To start recognition, call the **recognizeText** method of the **IRecognitionCoreAPI** interface. Its required input parameters are the bitmap to process and your **TextRecognitionCallback** object. The recognizer will start up several working threads and continue interacting with your application via the **TextRecognitionCallback** interface.
7. When finished, the **recognizeText** method will return an array of **TextBlock** objects which contain the results of recognition for the text areas found on the image. Each **TextBlock** contains one or more text lines represented by **TextLine** objects. Each **TextLine** contains information about the enclosing quadrangle for a single line of text and the recognized text as a string.
   Work with the results on your side.
8. When pausing or quitting the application, call the **IRecognitionCoreAPI.close** method to release resources.

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 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 Real-Time Recognition SDK engine will automatically request 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 Real-Time Recognition library are detailed in Data Capture Profiles; see also the summary below in Supported ID Documents.

Note that Real-Time Recognition 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

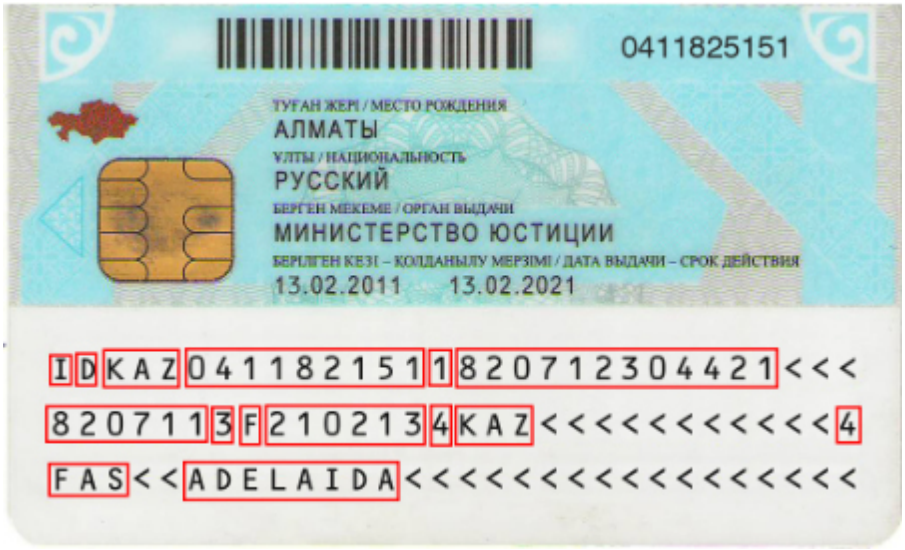
Real-Time Recognition SDK provides predefined data capture profiles for many types of data, including:

- machine-readable zone (MRZ) in various documents,
- international bank account numbers (IBAN),
- 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

Real-Time Recognition 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, Real-Time Recognition 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

Real-Time Recognition 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.

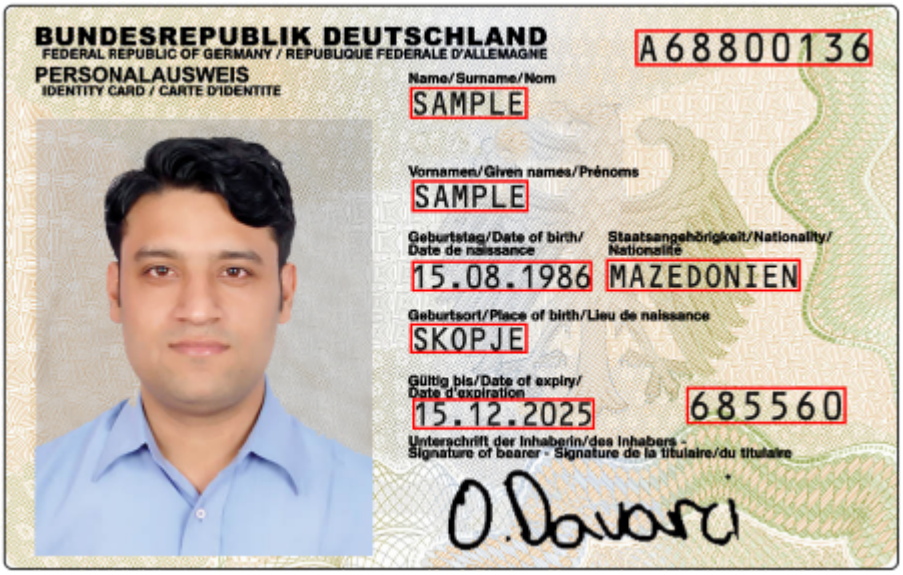
Bank card

Real-Time Recognition SDK can capture data from debit and credit cards, embossed and unembossed.

When recognizing a bank card, Real-Time Recognition SDK will detect and extract the card number, cardholder's full name, and date of expiry.

ID documents

Real-Time Recognition 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, Real-Time Recognition 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.

Implementation

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

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

1. Begin with the **Callback** interface implementation. Its methods will be used to pass the data to and from the recognition service. Here are the brief recommendations on what the methods should do:
   - The **onRequestLatestFrame** method should retrieve the image from the camera and pass it on to the **IDataCaptureService.submitRequestedFrame** method.
   - The **onFrameProcessed** method is where you work with the results, display them to the user, etc.
   - The **onError** method is for handling processing errors.
2. Call the **Engine.load** method on the UI thread to create an engine object via which all other objects may be created. This object should be reused for every new operation and should not be created again in the same activity.
3. Use the **createDataCaptureService** method of the **Engine** object to create a background recognition service (implementing the **IDataCaptureService** interface). Set the type of document you are going to capture using the *profileName* 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. When the camera is ready, call the **start** method of the **IDataCaptureService** interface. Its required input parameters are the size and orientation of the video frame and the rectangular area where to search for the text (e.g. if your application displays a highlighted rectangle in the center of the image, this rectangle should be specified as the "area of interest").

The service will then start up several working threads and continue interacting with your application via the **Callback** interface.

5. Whenever the **Callback.onRequestLatestFrame** method is called, provide the current video frame from the camera by calling **IDataCaptureService.submitRequestedFrame**.

6. The **Callback.onFrameProcessed** method will be called on the UI thread to return the result. Its parameters are:

   - a **DataScheme** 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 **null** is passed instead of a valid **DataScheme** 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 **DataField** objects, each representing one of the fields found and recognized. A **DataField** 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 Available and the data scheme has been matched.

7. Save the results for the recognized page. Call the **IDataCaptureService.stop** method to terminate the processing threads and clean up image buffers.

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

How to Capture a Custom Data Field with Android

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

How it Works

With Real-Time Recognition SDK you can create custom data capture profiles for documents that are not supported out-of-the-box. In 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 Android Studio Project.*

To implement the custom data field capture scenario, follow these steps:

1. Begin with the **Callback** interface implementation. Its methods will be used to pass the data to and from the recognition service. Here are the brief recommendations on what the methods should do:
   - The **onRequestLatestFrame** method should retrieve the image from the camera and pass it on to the **IDataCaptureService.submitRequestedFrame** method.
   - The **onFrameProcessed** method is where you work with the results, display them to the user, etc.
   - The **onError** method is for handling processing errors.
2. Call the **Engine.load** method on the UI thread to create an engine object via which all other objects may be created. This object should be reused for every new operation and should not be created again in the same activity.

3. Use the **createDataCaptureService** method of the **Engine** object to create a background recognition service (implementing the **IDataCaptureService** interface). The *profileName* should be an empty string (or **null**): you are going to add your custom profile and then apply it to the service.
   Only one instance of the service per application is necessary: multiple threads will be started internally.

4. Call the **configureDataCaptureProfile** method of the **IDataCaptureService** object to create an **IDataCaptureProfileBuilder** object. Then use its **addScheme** method to create an **ISchemeBuilder** object. The scheme builder allows you to set a human-readable name to the scheme (for example, it can be used for UI labels) and to add field definitions.

5. Call **ISchemeBuilder.addField** to create an **IFieldBuilder** object. The field builder is used to configure field's properties — its human-readable name and recognition rules.

6. Call **IFieldBuilder.setRegEx** to set the regular expression that should match the field text. The *regEx* parameter is `"[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 Real-Time Recognition SDK, see the Regular Expressions section.*
Also you can implement any string predicate and use it for additional validation after the data has passed the regular expression check — for example, calculate the field's checksum. To do so, implement **Predicate<T>** for the String type and set it as the additional validation callback using **setOnValidate**. An alphanumeric code needs no additional checks, so this step is skipped here.

7. After you have configured the field and scheme builders, call **IDataCaptureProfileBuilder.checkAndApply** 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.

Note: *The methods of builder objects return these objects, so in your code the steps above can be shortened as follows:*

```
IDataCaptureService dataCaptureService =
engine.createDataCaptureService( "", callback );
IDataCaptureProfileBuilder profileBuilder =
dataCaptureService.configureDataCaptureProfile()
    .setRecognitionLanguage( "English" );

profileBuilder.addScheme( "sampleScheme" )
    .setName( "Sample Profile" )
    .addField( "sampleField" )
        .setName( "Some Alphanumeric Code" )
        .setRegEx( "[0-9]{2}[0-9A-Z]{13}" );
```

```
profileBuilder.checkAndApply();
```

8. When the camera is ready, call the **start** method of the **IDataCaptureService** interface. Its required input parameters are the size and orientation of the video frame and the rectangular area where to search for the text (e.g. if your application displays a highlighted rectangle in the center of the image, this rectangle should be specified as the "area of interest").
   The service will then start up several working threads and continue interacting with your application via the **Callback** interface.

9. Whenever the **Callback.onRequestLatestFrame** method is called, provide the current video frame from the camera by calling **IDataCaptureService.submitRequestedFrame**.

10. The **Callback.onFrameProcessed** method will be called on the UI thread to return the result. Its parameters are:

    - A **DataScheme** object; its **Id** property should return the same identifier that you have specified when adding the scheme (the *id* argument to **addScheme**).

Important! *If **null** is passed instead of a valid **DataScheme** 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 of **DataField** objects, each representing one of the fields found and recognized. A **DataField** 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 Available and the data scheme has been matched.

11. Save the results. Call the **IDataCaptureService.stop** method to terminate the processing threads and clean up image buffers.

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

Code Samples

The ABBYY Real-Time Recognition 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 Java.

Sample scenario	Folder name	Description
Text Capture	**sample-textcapture**	A simple text capture scenario. The only setting available to the user is the text language.

Sample scenario		Folder name	Description
Data Capture	General	**sample-datacapture**	The general data capture scenario showing how to capture a predefined document and a custom data field.
	Russian Passport	**scenarios-datacapture/ru-passport/sample-passport**	The data capture scenario for the Russian passport.

Configuring the code samples

The samples should be open and built from the same folder where they are located in the distribution package. To work with any of the code samples you need to do only a little configuring first.

1. Please change the application ID before building, modifying or otherwise using any of the samples.
2. All samples expect that the license file (named **AbbyyRtrSdk.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 in the sample code.
   a. For the *Text Capture* and *Data Capture: General* scenarios:

```
public class MainActivity extends Activity {

    // Licensing
    private static final String licenseFileName = "AbbyyRtrSdk.license";
```

   b. For the *Data Capture: Russian Passport* scenario:

```
public class CameraActivity extends AppCompatActivity implements
ICameraActivity {

    // Licensing
    private static final String licenseFileName = "AbbyyRtrSdk.license";
```

API Reference

This section describes the Java API of ABBYY Real-Time Recognition SDK.

Classes

- Engine
- IDataCaptureService.CharInfo
- IDataCaptureService.DataField
- IDataCaptureService.DataScheme
- IDataCaptureService.TextLine
- ITextCaptureService.CharInfo
- ITextCaptureService.TextLine
- IRecognitionCoreAPI.CharInfo
- IRecognitionCoreAPI.TextBlock
- IRecognitionCoreAPI.TextLine

Interfaces

- Engine.EngineSettings
- IDataCaptureProfileBuilder
- IDataCaptureProfileBuilder.IFieldBuilder
- IDataCaptureProfileBuilder.ISchemeBuilder
- IDataCaptureProfileBuilder.Predicate<T>
- IRecognitionService
  - IDataCaptureService
  - ITextCaptureService
- IRecognitionService.Callback
  - IDataCaptureService.Callback
  - ITextCaptureService.Callback
- IRecognitionService.DebugLog
  - IDataCaptureService.DebugLog
  - ITextCaptureService.DebugLog
- IRecognitionService.ExtendedSettings
  - IDataCaptureService.ExtendedSettings
  - ITextCaptureService.ExtendedSettings
- IRecognitionCoreAPI
  - IRecognitionCoreAPI.ProcessingSettings
  - IRecognitionCoreAPI.TextRecognitionCallback
  - IRecognitionCoreAPI.TextRecognitionSettings

Enumerations

- IRecognitionService.ResultStabilityStatus
  - IDataCaptureService.ResultStabilityStatus
  - ITextCaptureService.ResultStabilityStatus
- IRecognitionService.Warning
  - IDataCaptureService.Warning
  - ITextCaptureService.Warning
- IRecognitionCoreAPI.Warning
- Language

Exceptions

- **Engine.LicenseException**
- **IDataCaptureProfileBuilder.ProfileCheckException**

Engine class

ABBYY Real-Time Recognition SDK engine via which all other objects may be created.

Creating the **Engine** and initializing the library may take up a lot of time, since all the resources have to be loaded. Therefore this object should only be created once (using the **load** method), when initializing the main activity of your application, and you should reuse it every time you need to start a new recognition operation.

```
public abstract class Engine
```

Methods

Name	Description
createDataCaptureService	Creates a background recognition service to run in data capture mode.
createRecognitionCoreAPI	Creates a core API object which provides access to low-level single image recognition functions.
createTextCaptureService	Creates a background recognition service to run in text capture mode.
getExtendedSettings	Provides access to the **EngineSettings** object via which you may specify additional settings for all scenarios.
load	Loads the ABBYY Real-Time Recognition SDK engine.
unload	***Important!*** *Using this method is not recommended.* Unloads the ABBYY Real-Time Recognition SDK engine.

Nested classes

Name	Description

LicenseException	The exception thrown when an invalid license is loaded.
EngineSettings	Additional settings for ABBYY Real-Time Recognition SDK engine which apply to all processing scenarios.

load method of the Engine class

Loads the ABBYY Real-Time Recognition SDK engine.

Creating the **Engine** and initializing the library may take up a lot of time, because all the resources need to be loaded. Therefore you should call this method only once, when initializing the main activity of your application, and reuse the **Engine** object every time you need to start a new recognition operation.

```
public static Engine load( Context context, String licenseFilePath ) throws
IOException, LicenseException
```

Parameters

context

The application context.

licenseFilePath

The path to the license file relative to the **assets** directory.

Return values

The method returns an instance of the **Engine** object.

Exceptions

Throws **java.io.IOException** if a required library or resource is not found or could not be loaded.

Throws **Engine.LicenseException** if the specified license is invalid.

createDataCaptureService method of the Engine class

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

```
public abstract IDataCaptureService createDataCaptureService( String
profileName, IDataCaptureService.Callback callback );
```

Parameters

profileName

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 **null** to configure your own profile for custom data field capture with the help of the **IDataCaptureService.configureDataCaptureProfile** method.

callback

An object implementing the **IDataCaptureService.Callback** interface, which will handle requests from the service.

Return values

The method returns a data capture service object implementing the **IDataCaptureService** interface.

createRecognitionCoreAPI method of the Engine class

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

```
public abstract IRecognitionCoreAPI createRecognitionCoreAPI();
```

Return values

The method returns an object implementing the **IRecognitionCoreAPI** interface.

createTextCaptureService method of the Engine class

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

```
public abstract ITextCaptureService
createTextCaptureService( ITextCaptureService.Callback callback );
```

Parameters

callback

An object implementing the **ITextCaptureService.Callback** interface, which will handle requests from the service.

Return values

The method returns a text capture service object implementing the **ITextCaptureService** interface.

getEngineSettings method of the Engine class

Provides access to the **EngineSettings** object via which you may specify additional settings for all scenarios.

```
EngineSettings getExtendedSettings();
```

Return values

This method returns an object implementing the **EngineSettings** interface, which allows you to change the additional engine settings.

unload method of the Engine class

Important! *Using this method is not recommended.*

Unloads the ABBYY Real-Time Recognition SDK engine.

Explicitly unloading the engine is not required and **not recommended** for most applications. Use this method only if in your application the engine is used in a separate activity, which is not likely to be used repeatedly, and you absolutely must reclaim the memory. If this is the case, the most appropriate place to unload the engine is the **onDestroy** method of the activity.

```
public abstract void unload();
```

LicenseException class

The exception thrown when an invalid license is loaded.

```
public static final class LicenseException extends Exception
```

EngineSettings interface

Additional settings for ABBYY Real-Time Recognition SDK engine. They apply to all processing scenarios.

```
public interface EngineSettings
```

Methods

Name	Description
getExternalAssetsPath	Returns the path to the custom directory with the necessary resources.

Name	Description
setExternalAssetsPath	Sets the path to the custom directory with the necessary resources. By default, patterns and dictionaries which ABBYY Real-Time Recognition SDK needs are located in the **assets** folder. This setting allows you to store the resource files in another location, so that your application folder takes up less memory. The subfolder structure should be maintained. ***Important!*** *The license file should still be placed in* ***assets****.*

getExternalAssetsPath method of the EngineSettings interface

Returns the path to the custom directory with the necessary resources.

The program will search for any resource file it needs first in **assets**, 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 **assets/patterns**
2) in **<custom search path>/patterns**
3) if the file is not found, an error will be returned

```
String getExternalAssetsPath();
```

Return values

The method returns the full path to the custom resource files folder.

setExternalAssetsPath method of the EngineSettings interface

Sets the path to the custom directory with the necessary resources.

The program will search for any resource file it needs first in **assets**, 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 **assets/patterns**
2) in **<custom search path>/patterns**
3) if the file is not found, an error will be returned

```
void getExternalAssetsPath( String path );
```

Parameters

path

The path to the custom resources folder. Pass **null** for this parameter to search only in **assets**.

IDataCaptureProfileBuilder interface

The data capture profile builder interface. The profile builder allows you to configure the data capture service to recognize custom documents.

To define the custom document scheme, use **addScheme** and configure the builder with the **ISchemeBuilder** and **IFieldBuilder** interface methods. Then call **checkAndApply** to create the data capture profile with this scheme and apply it to the existing data capture service. Note that the service must not be running at the time of the **checkAndApply** call (use **IDataCaptureService.stop** if necessary).

```
public interface IDataCaptureProfileBuilder
```

Methods

Name	Description
addScheme	Adds a new empty scheme configuration to the builder.
checkAndApply	Checks builder settings, creates the profile and applies it to the data capture service.
setRecognitionLanguage	Sets the languages to use for recognition.

Nested classes

Name	Description
IFieldBuilder	The interface for a field builder used to define custom field properties.
ISchemeBuilder	The interface for a custom document scheme builder used to add fields to the scheme.
Predicate<T>	Represents a predicate (boolean-valued function) of one argument. Used in IFieldBuilder.setOnValidate.

Name	Description
ProfileCheckException	The exception thrown when invalid settings are found by the checkAndApply method.

IFieldBuilder interface

The interface for a field builder used to define the name and recognition rules for a custom field. The rules include regular expression matching and optional custom validation.

```
public static interface IDataCaptureProfileBuilder.IFieldBuilder
```

Methods

Name	Description
setName	Sets the human-readable name for the field.
setOnValidate	Sets the validation callback for additional checks not covered by regular expression matching.
setRegEx	Sets the regular expression that should match the field's text.

setName method of the IFieldBuilder interface

Sets the human-readable name for the field. This name corresponds to the **Name** property of the DataField class.

```
IFieldBuilder setName( String name );
```

Parameters

name

The field name.

Return values

The method returns the IFieldBuilder instance to which it belongs.

setOnValidate method of the IFieldBuilder interface

Sets the callback for additional validation performed after regular expression matching. This callback can be used for custom checks or tests that are not covered by regular expressions, for example, to calculate the field's checksum. If you do not perform such tests, there is no need to call this method.

```
IFieldBuilder setOnValidate( Predicate<String> onValidate );
```

Parameters

`onValidate`

An object implementing the **Predicate<T>** interface for the String type. Implemented by user.

Return values

The method returns the **IFieldBuilder** instance to which it belongs.

setRegEx method of the IFieldBuilder interface

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

Note: *For details on regular expression syntax supported in ABBYY Real-Time Recognition 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.*

```
IFieldBuilder setRegEx( String regEx );
```

Parameters

regEx

A string describing the regular expression.

Return values

The method returns the **IFieldBuilder** instance to which it belongs.

ISchemeBuilder interface

The interface for a custom document scheme builder. Used to add fields to the scheme and set the name of the custom data capture profile.

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

```
public static interface IDataCaptureProfileBuilder.ISchemeBuilder
```

Methods

Name	Description
addField	Adds a new field to the scheme.
setName	Sets the human-readable name for the scheme.

addField method of the ISchemeBuilder interface

Adds a new field to the scheme. Field properties are configured using the field builder returned by this method.

```
IFieldBuilder addField( String id );
```

Parameters

id

Internal field identifier, corresponds to the **Id** property of the DataField class. For a human-readable field name which you can display to the user, see IFieldBuilder.setName.

Return values

The method returns an IFieldBuilder instance.

setName method of the ISchemeBuilder interface

Sets the human-readable name for the scheme. This name corresponds to the **Name** property of the DataScheme class.

```
ISchemeBuilder setName( String name );
```

Parameters

name

The scheme name.

Return values

The method returns the ISchemeBuilder instance to which it belongs.

Predicate<T> interface

Represents a predicate (boolean-valued function) of one argument. Mimics the standard **java.util.function.Predicate<T>** interface defined in the Java API as of Java 8.

This interface and its **test** method are to be implemented on the client side.

```
public static interface IDataCaptureProfileBuilder.Predicate<T>
```

Methods

Name	Description
test	Evaluates this predicate on the given argument.

test method of the Predicate<T> interface

Evaluates the predicate on the given argument.

This method is to be implemented on the client side.

```
boolean test( T value );
```

Parameters

`value`

The input argument.

Return values

The method returns **true** if the input argument passes validation and **false** otherwise.

addScheme method of the IDataCaptureProfileBuilder interface

Adds a new empty scheme configuration to the profile builder. The scheme is then configured using the **ISchemeBuilder** interface methods.

```
ISchemeBuilder addScheme( String id );
```

Parameters

id

Internal scheme identifier, corresponds to the **Id** property of the **DataScheme** class. For a human-readable scheme name which you can display to the user, see **ISchemeBuilder.setName**.

Return values

The method returns an **ISchemeBuilder** instance.

checkAndApply method of the IDataCaptureProfileBuilder interface

Checks the profile builder's settings, creates the profile and applies it to the data capture service. This new profile replaces any previous profile, including a predefined profile if the latter was specified in the **createDataCaptureService** call.

Note that the data capture service must not be running when calling this method (use **IDataCaptureService.stop** if necessary).

```
void checkAndApply() throws ProfileCheckException;
```

Exceptions

Throws **IDataCaptureProfileBuilder.ProfileCheckException** if any of the profile settings are invalid.

setRecognitionLanguage method of the IDataCaptureProfileBuilder interface

Sets the languages to be used for field recognition.

By default, no language is set. Setting the correct languages for your document will improve recognition accuracy. However, setting too many languages may decrease performance.

```
IDataCaptureProfileBuilder setRecognitionLanguage( Language... languages );
```

Parameters

languages

One or more languages to be used for recognition, each represented by a constant of the **Language** enumeration.

Return values

The method returns the **IDataCaptureProfileBuilder** instance to which it belongs.

ProfileCheckException class

The exception thrown when a custom data capture profile cannot be applied due to invalid settings.

```
final class ProfileCheckException extends RuntimeException
```

IDataCaptureService interface

A background data capture service interface.

This interface provides methods to tune the processing settings, start and stop the work, and pass the video frames from the camera to the background processing engine.

Extends the **IRecognitionService** interface.

```
public interface IDataCaptureService extends IRecognitionService
```

Methods

Name	Description
configureDataCaptureProfile	Creates a profile builder object, which allows you to configure the data capture service to recognize custom documents.
getExtendedSettings	Provides access to extended service configuration settings. Inherited from **IRecognitionService**.
setAreaOfInterest	Sets the area on the frame where the text is to be found. Inherited from **IRecognitionService**.
setDebugLog	Attaches a callback for collecting debug data. Inherited from **IRecognitionService**.
start	Starts processing. Inherited from **IRecognitionService**.
stop	Stops processing and releases the resources used by the recognition service. Inherited from **IRecognitionService**.
submitRequestedFrame	Submits the video frame requested through the **Callback.onRequestLatestFrame** method. Inherited from **IRecognitionService**.

Nested classes

Name	Description
Callback	A callback interface to interact with the recognition service:

Name	Description
	input the data and obtain the results. Extends **IRecognitionService.Callback**.
CharInfo	Extended information about character formatting. ***Important!*** *This class is reserved for future use.*
DataField	A recognized data field. Provides field contents, location and data scheme information.
DataScheme	Information on the data scheme applied to the recognized frame.
DebugLog	A callback interface for collecting debug data. Inherited from the **IRecognitionService.DebugLog** interface without any modification.
ExtendedSettings	Extended service configuration settings. Inherited without any modification from **IRecognitionService.ExtendedSettings**.
TextLine	A line of recognized text; the location and additional information are also available.

Enumerations

The enumerations are inherited from **IRecognitionService** without any modifications.

Name	Description
ResultStabilityStatus	Result stability status: the estimate of how stable the result is, and whether it is likely to be improved by adding new frames.
Warning	A warning that occurred during processing.

Callback interface

A callback interface to interact with the recognition service: input the data and obtain the results. This interface and its methods are to be implemented on the client side.

Extends the **IRecognitionService.Callback** interface.

```
interface Callback extends IRecognitionService.Callback
```

Note: *While the service is being stopped, frames continue to be requested and calls to this callback continue to be queued, so this callback can be called after the service has been stopped.*

Methods

Name	Description
onError	Called to report an error. Inherited from **IRecognitionService.Callback**.
onFrameProcessed	Called to deliver the result after recognizing the frames that were provided.
onRequestLatestFrame	Called to request the latest video frame. Inherited from **IRecognitionService.Callback**.

onRequestLatestFrame method of the Callback interface

Called by the service when it needs the latest video frame. The frame should be provided through a call to the **IDataCaptureService.submitRequestedFrame** method.

This method is to be implemented on the client side.

```
void onRequestLatestFrame( byte[] buffer )
```

Parameters

buffer

The buffer to be filled with image data for latest frame. Only NV21 format is currently supported.

Can be passed directly to **Camera.addCallbackBuffer**. When the buffer is filled with data, it should be passed back to the service by calling **submitRequestedFrame**.

onFrameProcessed method of the Callback interface

Called by the service to deliver the result after recognizing the frames that were supplied.

The result stability status is also provided and 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 stability level has reached Available 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 **IDataCaptureService.stop** method.

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 onFrameProcessed( DataScheme scheme, DataField[] fields,
ResultStabilityStatus resultStatus, Warning warning )
```

Parameters

scheme

A **DataScheme** object with information on the data scheme which was applied to the recognized frame.

Important! *If* ***null*** *is passed instead of a valid* ***DataScheme*** *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.*

fields

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

resultStatus

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

warning

The warning which occurred, if any; represented by a **Warning** enumeration constant.

onError method of the Callback interface

Called by the service when an error occurs.

This method is to be implemented on the client side, which may include displaying the error description to the user or handling it otherwise.

```
void onError( Exception error )
```

Parameters

error

The **Exception** object for the error that has occurred.

DebugLog interface

A callback interface for collecting debug data. This interface and its methods are to be implemented on the client side.

Inherited from the **IRecognitionService.DebugLog** interface without any modification.

```
interface IDataCaptureService.DebugLog
```

Methods

Name	Description
onBeginSeries	Called when a series of video frames begins.
onEndSeries	Called when a series of video frames ends.
onSaveImageBufferNV21	Called to log an image in the NV21 format.
onAttachDebugInfo	Called to deliver debug information associated with a logged image.

onBeginSeries method of the DebugLog interface

Called by the service when a series of video frames begins. This method is to be implemented on the client side.

```
void onBeginSeries();
```

onEndSeries method of the DebugLog interface

Called by the service when a series of video frames ends. This method is to be implemented on the client side.

```
void onEndSeries();
```

onSaveImageBufferNV21 method of the DebugLog interface

Called by the service to log an image in the NV21 format. This method is to be implemented on the client side.

```
String onSaveImageBufferNV21( int width, int height, int orientation,
                              Rect areaOfInterest, byte[] buffer,
                              int dataSize );
```

Parameters

width

The image width.

height

The image height.

orientation

The image orientation in degrees, a multiple of 90.

areaOfInterest

The rectangular area of interest on the image.

buffer

The buffer with image data in NV21 format. Only *dataSize* bytes in the buffer contain valid image data.

dataSize

The number of bytes in the *buffer* containing valid image data.

Return values

A string identifier of the image to which detailed debug information may be attached. If **null** is returned, the **onAttachDebugInfo** method will not be called and no detailed information will be reported.

onAttachDebugInfo method of the DebugLog interface

Called by the service to deliver detailed debug information associated with a logged image. This method is only called if the **onSaveImageBufferNV21** method returned a non-null identifier.

This method is to be implemented on the client side.

```
void onAttachDebugInfo( String imageId, String debugInfo );
```

Parameters

imageId

The identifier of the image to which the debug information corresponds.

debugInfo

A string containing the detailed debug information.

ExtendedSettings interface

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

Inherited from the **IRecognitionService.ExtendedSettings** interface without any modification.

Important! *Any modifications of these settings should be made before the call to the **start** method.*

```
interface ExtendedSettings extends IRecognitionService.ExtendedSettings
```

Methods

Name	Description
getProcessingThreadsCount	Gets the number of processing threads to be used by the service.
setProcessingThreadsCount	Sets the number of processing threads to be used by the service.

getProcessingThreadsCount method of the ExtendedSettings interface

Gets the number of processing threads to be used by the service.

```
int getProcessingThreadsCount();
```

Return values

The method returns the number of threads. Returns 0 if the number of threads is to be determined automatically, which is the default setting.

setProcessingThreadsCount method of the ExtendedSettings interface

Sets the number of processing threads to be used by the service.

```
void setProcessingThreadsCount( int ThreadsCount );
```

Parameters

ThreadsCount

The new number of threads. Up to 16 threads are allowed. Set to 0 to determine the number of threads automatically.

DataScheme class

Information on the data scheme applied to the recognized frame.

```
final class DataScheme {
    public final String Id;
    public final String Name;
}
```

Properties

Name	Type	Description
Id	**String**	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 in the **IDataCaptureProfileBuilder.addScheme** call.
Name	**String**	The human-readable name of the data scheme. If you are using a custom scheme, this is the name you set with the **ISchemeBuilder.setName** method.

DataField class

A recognized data field. Provides field contents, location and data scheme information.

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 a **TextLine** 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.

```
final class DataField {
    public final String Id;
    public final String Name;
    public final String Text;
    public final Point[] Quadrangle;
    public final TextLine[] Components;
}
```

Properties

Name	Type	Description
Components	**TextLine**[]	An array of text fragments representing field components, that is, the fragments found on the image, which constitute the field.
Id	**String**	The internal field identifier. Can be one of the predefined fields listed in Data Capture Profiles or the custom field identifier that you specified in the **ISchemeBuilder.addField** call.
Name	**String**	The human-readable name of the field. If you are using a custom data capture profile, this is the name you set with the **IFieldBuilder.setName** method.
Quadrangle	**Point**[]	The four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left. ***Note:*** *Before recognition, the service rotates the image obtained from camera in order to bring text orientation to normal (horizontal). The vertex coordinates are specified for this rotated image and may require coordinate conversion if you display the quadrangle on the video frame.*
Text	**String**	The text of the field.

TextLine class

A fragment of recognized text, with additional information about characters.

```
final class TextLine {
    public final String Text;
    public final Point[] Quadrangle;
    public final CharInfo[] CharInfo;
}
```

Properties

Name	Type	Description
CharInfo	CharInfo[]	Additional information about the characters. ***Important!*** *This property is reserved for future use.*
Quadrangle	**Point**[]	The four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left. ***Note:*** *Before recognition, the service rotates the image obtained from camera in order to bring text orientation to normal (horizontal). The vertex coordinates are specified for this rotated image and may require coordinate conversion if you display the quadrangle on the video frame.*
Text	**String**	The recognized text.

CharInfo class

Extended information about character formatting.

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

```
final class CharInfo {
    public final Point[] Quadrangle;
}
```

Properties

Name	Type	Description
Quadrangle	**Point**[]	The four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left.

configureDataCaptureProfile method of the IDataCaptureService interface

Creates a profile builder object, which allows you to configure the data capture service to recognize custom documents. The service should be created without a profile. If a predefined profile is specified when creating the service, later it will be completely replaced by the new custom profile when you call **IDataCaptureProfileBuilder.checkAndApply**.

```
IDataCaptureProfileBuilder configureDataCaptureProfile();
```

Return values

The method returns an object implementing the **IDataCaptureProfileBuilder** interface.

getExtendedSettings method of the IDataCaptureService interface

Provides access to extended service configuration settings. Intended for advanced users: most common scenarios will work with the default settings.

```
ExtendedSettings getExtendedSettings();
```

Return values

This method returns an object implementing the **ExtendedSettings** interface, which allows you to change the advanced configuration settings.

setAreaOfInterest method of the IDataCaptureService interface

Sets the area on the frame where the text is to be found.

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( Rect areaOfInterest );
```

Parameters

areaOfInterest

The rectangle specifying the area of interest.

setDebugLog method of the IDataCaptureService interface

Attaches a callback which collects image data for debugging and tuning the ABBYY Real-Time Recognition SDK library. The callback and its methods should be implemented on the client side.

```
void setDebugLog( DebugLog debugLog );
```

Parameters

debugLog

An object implementing the **DebugLog** interface, which will be used to process the debug data.

start method of the IDataCaptureService interface

Starts processing. The service will automatically create several processing threads, request video frames and return the results via the **Callback** interface.

```
void start( int width, int height, int orientation,
            Rect areaOfInterest );
```

Parameters

width

The width of the video frame.

height

The height of the video frame.

orientation

The orientation of the video frame in degrees. Should be a multiple of 90.

areaOfInterest

The rectangular area of the frame where the text is expected to be. For example, it may be selected by the user or highlighted in your application interface.

Note: *You can also change the area of interest while the service is running by calling the* ***setAreaOfInterest*** *method.*

Note: *Before recognition, the service rotates the image obtained from camera in order to bring text orientation to normal (horizontal). The area of interest is specified in the coordinates on this rotated image, which are different from the coordinates on the video frame except the case when the frame orientation is 0.*

stop method of the IDataCaptureService interface

Stops processing and releases the resources used by the service.

```
void stop();
```

submitRequestedFrame method of the IDataCaptureService interface

Submits the video frame obtained from the camera after it is requested through the **Callback.onRequestLatestFrame** method.

```
void submitRequestedFrame( byte[] buffer );
```

Parameters

buffer

The buffer filled with image data for the latest frame. Only NV21 format is currently supported. This should be the same buffer which has been passed via the call to the **Callback.onRequestLatestFrame** method.

ResultStabilityStatus enum

Result stability status: an 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 Available.

```
enum ResultStabilityStatus {
    NotReady,
    Tentative,
    Verified,
    Available,
    TentativelyStable,
    Stable
};
```

Constants

Name	Description
NotReady	No content available.
Tentative	Content detected on a single frame.
Verified	Content verified: matching content found in at least two frames.
Available	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.
TentativelyStable	The result has been stable in the last two frames.
Stable	The result has been stable in the last three or more frames.

Warning enum

A warning that occurred during processing.

```
enum Warning {
    TextTooSmall
}
```

Constants

Name	Description
TextTooSmall	The text is too small. Advise the end user to move the camera closer or zoom in.

ITextCaptureService interface

A background text capture service interface.

This interface provides methods to tune the processing settings, start and stop the work, and pass the video frames from the camera to the background processing engine.

Extends the **IRecognitionService** interface.

```
public interface ITextCaptureService extends IRecognitionService
```

Methods

Name	Description
getExtendedSettings	Provides access to extended service configuration settings. Inherited from **IRecognitionService**.
setAreaOfInterest	Sets the area on the frame where the text is to be found. Inherited from **IRecognitionService**.
setDebugLog	Attaches a callback for collecting debug data. Inherited from **IRecognitionService**.
setRecognitionLanguage	Sets the languages to be used for recognition.
setTranslationDictionary	Sets the name of the translation dictionary.
start	Starts processing. Inherited from **IRecognitionService**.
stop	Stops processing and releases the resources used by the recognition service. Inherited from **IRecognitionService**.
submitRequestedFrame	Submits the video frame requested through the **Callback.onRequestLatestFrame** method. Inherited from **IRecognitionService**.

Nested classes

Name	Description
Callback	A callback interface to interact with the recognition service: input the data and obtain the results. Extends IRecognitionService.Callback.
CharInfo	Extended information about the characters' formatting. ***Important!*** *This class is reserved for future use.*
DebugLog	A callback interface for collecting debug data. Inherited from the IRecognitionService.DebugLog interface without any modification.
ExtendedSettings	Extended service configuration settings. Extends IRecognitionService.ExtendedSettings.
TextLine	A line of recognized text; the location and additional information are also available.

Enumerations

The enumerations are inherited from IRecognitionService without any modifications.

Name	Description
ResultStabilityStatus	Result stability status: the estimate of how stable the result is, and whether it is likely to be improved by adding new frames.
Warning	A warning that occurred during processing.

Callback interface

A callback interface to interact with the recognition service: input the data and obtain the results. This interface and its methods are to be implemented on the client side.

Extends the IRecognitionService.Callback interface.

```
interface Callback extends IRecognitionService.Callback
```

Note: *While the service is being stopped, frames continue to be requested and calls to this callback continue to be queued, so this callback can be called after the service has been stopped.*

Methods

Name	Description
onError	Called to report an error. Inherited from **IRecognitionService.Callback**.
onFrameProcessed	Called to deliver the result after recognizing the frames that were provided.
onRequestLatestFrame	Called to request the latest video frame. Inherited from **IRecognitionService.Callback**.

onRequestLatestFrame method of the Callback interface

Called by the service when it needs the latest video frame. The frame should be provided through a call to the **ITextCaptureService.submitRequestedFrame** method.

This method is to be implemented on the client side.

```
void onRequestLatestFrame( byte[] buffer )
```

Parameters

buffer

The buffer to be filled with image data for latest frame. Only NV21 format is currently supported.

Can be passed directly to **Camera.addCallbackBuffer**. When the buffer is filled with data, it should be passed back to the service by calling **submitRequestedFrame**.

onFrameProcessed method of the Callback interface

Called by the service to deliver the result after recognizing the frames that were supplied.

The result stability status is also provided and 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 stability level has reached Available. When stability of the result has reached the desired level, the service may be stopped by calling the **ITextCaptureService.stop** method.

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 onFrameProcessed( TextLine[] lines, ResultStabilityStatus
resultStatus, Warning warning )
```

Parameters

lines

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

resultStatus

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

warning

The warning which occurred, if any; represented by a **Warning** enumeration constant.

onError method of the Callback interface

Called by the service when an error occurs.

This method is to be implemented on the client side, which may include displaying the error description to the user or handling it otherwise.

```
void onError( Exception error )
```

Parameters

error

The **Exception** object for the error that has occurred.

DebugLog interface

A callback interface for collecting debug data. This interface and its methods are to be implemented on the client side.

Inherited from the **IRecognitionService.DebugLog** interface without any modification.

```
interface ITextCaptureService.DebugLog
```

Methods

Name	Description
onBeginSeries	Called when a series of video frames begins.

Name	Description
onEndSeries	Called when a series of video frames ends.
onSaveImageBufferNV21	Called to log an image in the NV21 format.
onAttachDebugInfo	Called to deliver debug information associated with a logged image.

onBeginSeries method of the DebugLog interface

Called by the service when a series of video frames begins. This method is to be implemented on the client side.

```
void onBeginSeries();
```

onEndSeries method of the DebugLog interface

Called by the service when a series of video frames ends. This method is to be implemented on the client side.

```
void onEndSeries();
```

onSaveImageBufferNV21 method of the DebugLog interface

Called by the service to log an image in the NV21 format. This method is to be implemented on the client side.

```
String onSaveImageBufferNV21( int width, int height, int orientation,
                              Rect areaOfInterest, byte[] buffer,
                              int dataSize );
```

Parameters

width

The image width.

height

The image height.

orientation

The image orientation in degrees, a multiple of 90.

areaOfInterest

The rectangular area of interest on the image.

buffer

The buffer with image data in NV21 format. Only *dataSize* bytes in the buffer contain valid image data.

dataSize

The number of bytes in the *buffer* containing valid image data.

Return values

A string identifier of the image to which detailed debug information may be attached. If **null** is returned, the **onAttachDebugInfo** method will not be called and no detailed information will be reported.

onAttachDebugInfo method of the DebugLog interface

Called by the service to deliver detailed debug information associated with a logged image. This method is only called if the **onSaveImageBufferNV21** method returned a non-null identifier.

This method is to be implemented on the client side.

```
void onAttachDebugInfo( String imageId, String debugInfo );
```

Parameters

imageId

The identifier of the image to which the debug information corresponds.

debugInfo

A string containing the detailed debug information.

ExtendedSettings interface

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

Extends the **IRecognitionService.ExtendedSettings** interface.

Important! *Any modifications of these settings should be made before the call to the* ***start*** *method.*

```
interface ExtendedSettings extends IRecognitionService.ExtendedSettings
```

Methods

Name	Description
getProcessingThreadsCount	Gets the number of processing threads to be used by the service. Inherited from IRecognitionService.ExtendedSettings.
isCJKVerticalTextEnabled	**Deprecated**. Checks if vertical writing direction is enabled for Chinese, Japanese, and Korean languages.
isFrameMergingEnabled	Checks if frame merging is enabled.
isRecognitionEnabled	Checks if recognition is enabled.
setCJKVerticalTextEnabled	**Deprecated**. Enables or disables vertical writing direction for Chinese, Japanese, and Korean languages.
setFrameMergingEnabled	Enables or disables frame merging.
setProcessingThreadsCount	Sets the number of processing threads to be used by the service. Inherited from IRecognitionService.ExtendedSettings.
setRecognitionEnabled	Enables or disables recognition.

isCJKVerticalTextEnabled method of the ExtendedSettings interface

Checks if vertical writing direction for Chinese, Japanese, and Korean languages is enabled.

Important! *This method is deprecated and will be revised in future releases.*

```
boolean isCJKVerticalTextEnabled();
```

Return values

Returns **true** if vertical writing direction is enabled, **false** otherwise.

setCJKVerticalTextEnabled method of the ExtendedSettings interface

Enables or disables vertical writing direction for Chinese, Japanese, and Korean languages.

Important! *This method is deprecated and will be revised in future releases.*

```
void setCJKVerticalTextEnabled( boolean enable );
```

Parameters

enable

Set this parameter to **true** to enable vertical writing direction for Chinese, Japanese, and Korean languages, or to **false** to disable it.

isFrameMergingEnabled method of the ExtendedSettings interface

Checks if frame merging is enabled.

```
boolean isFrameMergingEnabled();
```

Return values

The method returns **true** if frame merging is enabled (the default setting), **false** if disabled.

setFrameMergingEnabled method of the ExtendedSettings interface

Enables or disables frame merging.

```
void setFrameMergingEnabled( boolean enable );
```

Parameters

enable

Set this parameter to **true** to enable frame merging, to **false** to disable it. By default frame merging is enabled.

getProcessingThreadsCount method of the ExtendedSettings interface

Gets the number of processing threads to be used by the service.

```
int getProcessingThreadsCount();
```

Return values

The method returns the number of threads. Returns 0 if the number of threads is to be determined automatically, which is the default setting.

setProcessingThreadsCount method of the ExtendedSettings interface

Sets the number of processing threads to be used by the service.

```
void setProcessingThreadsCount( int ThreadsCount );
```

Parameters

ThreadsCount

The new number of threads. Up to 16 threads are allowed. Set to 0 to determine the number of threads automatically.

isRecognitionEnabled method of the ExtendedSettings interface

Checks if recognition is enabled.

```
boolean isRecognitionEnabled();
```

Return values

The method returns **true** if recognition is enabled (the default setting).

setRecognitionEnabled method of the ExtendedSettings interface

Enables or disables recognition.

```
void setRecognitionEnabled( boolean enable );
```

Parameters

enable

Set this parameter to **true** to enable recognition, to **false** to disable it. By default recognition is enabled.

TextLine class

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

```
final class TextLine {
    public final String Text;
    public final Point[] Quadrangle;
    public final CharInfo[] CharInfo;
}
```

Properties

Name	Type	Description
CharInfo	CharInfo[]	Additional information about the characters. ***Important!*** *This property is reserved for future use.*
Quadrangle	**Point**[]	The four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left. ***Note:*** *Before recognition, the service rotates the image obtained from camera in order to bring text orientation to normal (horizontal). The vertex coordinates are specified for this rotated image and may require coordinate conversion if you display the quadrangle on the video frame.*
Text	**String**	The recognized text.

CharInfo class

Extended information about the character formatting.

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

```
final class CharInfo {
    public final int ForegroundColor;
    public final int BackgroundColor;
    public final Point[] Quadrangle;
}
```

Properties

Name	Type	Description
BackgroundColor	**int**	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	**int**	The color of the symbol.
Quadrangle	**Point[]**	The four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left.

getExtendedSettings method of the ITextCaptureService interface

Provides access to extended service configuration settings. Intended for advanced users: most common scenarios will work with the default settings.

```
ExtendedSettings getExtendedSettings();
```

Return values

This method returns an object implementing the **ExtendedSettings** interface, which allows you to change the advanced configuration settings.

setAreaOfInterest method of the ITextCaptureService interface

Sets the area on the frame where the text is to be found.

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( Rect areaOfInterest );
```

Parameters

areaOfInterest

The rectangle specifying the area of interest.

setDebugLog method of the ITextCaptureService interface

Attaches a callback which collects image data for debugging and tuning the ABBYY Real-Time Recognition SDK library. The callback and its methods should be implemented on the client side.

```
void setDebugLog( DebugLog debugLog );
```

Parameters

debugLog

An object implementing the **DebugLog** interface, which will be used to process the debug data.

setRecognitionLanguage method of the ITextCaptureService interface

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.

```
void setRecognitionLanguage( Language... languages );
```

Parameters

languages

One or more languages to be used for recognition, each represented by a constant of the **Language** enumeration.

setTranslationDictionary method of the ITextCaptureService interface

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 **assets/translation** folder. 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* ***onFrameProcessed*** *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* ***null*** *argument.*

```
void setTranslationDictionary( String dictionaryName );
```

Parameters

dictionaryName

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

start method of the ITextCaptureService interface

Starts processing. The service will automatically create several processing threads, request video frames and return the results via the **Callback** interface.

```
void start( int width, int height, int orientation,
            Rect areaOfInterest );
```

Parameters

width

The width of the video frame.

height

The height of the video frame.

orientation

The orientation of the video frame in degrees. Should be a multiple of 90.

areaOfInterest

The rectangular area of the frame where the text is expected to be. For example, it may be selected by the user or highlighted in your application interface.

Note: *You can also change the area of interest while the service is running by calling the*

setAreaOfInterest *method.*

Note: *Before recognition, the service rotates the image obtained from camera in order to bring text orientation to normal (horizontal). The area of interest is specified in the coordinates on this rotated image, which are different from the coordinates on the video frame except the case when the frame orientation is 0.*

stop method of the ITextCaptureService interface

Stops processing and releases the resources used by the service.

```
void stop();
```

submitRequestedFrame method of the ITextCaptureService interface

Submits the video frame obtained from the camera after it is requested through the **Callback.onRequestLatestFrame** method.

```
void submitRequestedFrame( byte[] buffer );
```

Parameters

buffer

The buffer filled with image data for the latest frame. Only NV21 format is currently supported. This should be the same buffer which has been passed via the call to the **Callback.onRequestLatestFrame** method.

ResultStabilityStatus enum

Result stability status: an 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 Available.

```
enum ResultStabilityStatus {
    NotReady,
    Tentative,
    Verified,
    Available,
    TentativelyStable,
    Stable
};
```

Constants

Name	Description
NotReady	No content available.
Tentative	Content detected on a single frame.
Verified	Content verified: matching content found in at least two frames.
Available	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.
TentativelyStable	The result has been stable in the last two frames.
Stable	The result has been stable in the last three or more frames.

Warning enum

A warning that occurred during processing.

```
enum Warning {
    TextTooSmall
}
```

Constants

Name	Description
TextTooSmall	The text is too small. Advise the end user to move the camera closer or zoom in.

IRecognitionService interface

The base background recognition service interface, extended by the **IDataCaptureService** and **ITextCaptureService** scenario-specific interfaces.

```
public interface IRecognitionService
```

Methods

Name	Description
getExtendedSettings	Provides access to extended service configuration settings.
setAreaOfInterest	Sets the area on the frame where the text is to be found.
setDebugLog	Attaches a callback to collect debug data.
start	Starts processing.
stop	Stops processing and releases the resources used by the recognition service.
submitRequestedFrame	Submits the video frame requested through the **Callback.onRequestLatestFrame** method.

Nested classes

Name	Description
Callback	A callback interface to interact with the recognition service: input the data and obtain the results.
DebugLog	A callback interface for collecting debug data.
ExtendedSettings	Extended service configuration settings.

Enumerations

Name	Description
ResultStabilityStatus	Result stability status: the estimate of how stable the result is, and whether it is likely to be improved by adding new frames.
Warning	A warning that occurred during processing.

Callback interface

The base callback interface for interacting with the recognition service, extended by the IDataCaptureService.Callback and ITextCaptureService.Callback interfaces.

```
public static interface IRecognitionService.Callback
```

Note: *While the service is being stopped, frames continue to be requested and calls to this callback continue to be queued, so this callback can be called after the service has been stopped.*

Methods

Name	Description
onError	Called to report an error.
onRequestLatestFrame	Called to request the latest video frame.

onRequestLatestFrame method of the Callback interface

Called by the service when it needs the latest video frame. The frame should be provided through a call to the IRecognitionService.submitRequestedFrame method.

This method is to be implemented on the client side.

```
void onRequestLatestFrame( byte[] buffer )
```

Parameters

buffer

The buffer to be filled with image data for latest frame. Only NV21 format is currently supported.

Can be passed directly to **Camera.addCallbackBuffer**. When the buffer is filled with data, it should be passed back to the service by calling **submitRequestedFrame**.

onError method of the Callback interface

Called by the service when an error occurs.

This method is to be implemented on the client side, which may include displaying the error description to the user or handling it otherwise.

```
void onError( Exception error )
```

Parameters

error

The **Exception** object for the error that has occurred.

DebugLog interface

A callback interface for collecting debug data. This interface and its methods are to be implemented on the client side.

```
interface IRecognitionService.DebugLog
```

Methods

Name	Description
onBeginSeries	Begins a series of video frames.
onEndSeries	Ends the series of video frames.
onSaveImageBufferNV21	Logs the image in NV21 format.
onAttachDebugInfo	Attaches debug info associated with the image.

onBeginSeries method of the DebugLog interface

Called by the service when a series of video frames begins. This method is to be implemented on the client side.

```
void onBeginSeries();
```

onEndSeries method of the DebugLog interface

Called by the service when a series of video frames ends. This method is to be implemented on the client side.

```
void onEndSeries();
```

onSaveImageBufferNV21 method of the DebugLog interface

Called by the service to log an image in the NV21 format. This method is to be implemented on the client side.

```
String onSaveImageBufferNV21( int width, int height, int orientation,
                              Rect areaOfInterest, byte[] buffer,
                              int dataSize );
```

Parameters

width

The image width.

height

The image height.

orientation

The image orientation in degrees, a multiple of 90.

areaOfInterest

The rectangular area of interest on the image.

buffer

The buffer with image data in NV21 format. Only *dataSize* bytes in the buffer contain valid image data.

dataSize

The number of bytes in the *buffer* containing valid image data.

Return values

A string identifier of the image to which detailed debug information may be attached. If **null** is returned, the **onAttachDebugInfo** method will not be called and no detailed information will be reported.

onAttachDebugInfo method of the DebugLog interface

Reports the detailed debug information associated with the image. This method is only called if the **onSaveImageBufferNV21** method returned a non-null identifier.

This method is to be implemented on the client side.

```
void onAttachDebugInfo( String imageId, String debugInfo );
```

Parameters

imageId

The identifier of the image to which the debug information corresponds.

debugInfo

A string containing the detailed debug information.

ExtendedSettings interface

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

Important! *Any modifications of these settings should be made before the call to the* ***start*** *method.*

```
interface ExtendedSettings
```

Methods

Name	Description
getProcessingThreadsCount	Gets the number of processing threads to be used by the service.
setProcessingThreadsCount	Sets the number of processing threads to be used by the service.

getProcessingThreadsCount method of the ExtendedSettings interface

Gets the number of processing threads to be used by the service.

```
int getProcessingThreadsCount();
```

Return values

The method returns the number of threads. Returns 0 if the number of threads is to be determined automatically, which is the default setting.

setProcessingThreadsCount method of the ExtendedSettings interface

Sets the number of processing threads to be used by the service.

```
void setProcessingThreadsCount( int ThreadsCount );
```

Parameters

ThreadsCount

The new number of threads. Up to 16 threads are allowed. Set to 0 to determine the number of threads automatically.

getExtendedSettings method of the IRecognitionService interface

Provides access to extended service configuration settings. Intended for advanced users: most common scenarios will work with the default settings.

```
ExtendedSettings getExtendedSettings();
```

Return values

This method returns an object implementing the **ExtendedSettings** interface, which allows you to change the advanced configuration settings.

setAreaOfInterest method of the IRecognitionService interface

Sets the area on the frame where the text is to be found.

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( Rect areaOfInterest );
```

Parameters

areaOfInterest

The rectangle specifying the area of interest.

setDebugLog method of the IRecognitionService interface

Attaches a callback which collects image data for debugging and tuning the ABBYY Real-Time Recognition SDK library. The callback and its methods should be implemented on the client side.

```
void setDebugLog( DebugLog debugLog );
```

Parameters

debugLog

An object implementing the **DebugLog** interface, which will be used to process the debug data.

start method of the IRecognitionService interface

Starts processing. The service will automatically create several processing threads, request video frames and return the results via the **Callback** interface.

```
void start( int width, int height, int orientation,
            Rect areaOfInterest );
```

Parameters

width

The width of the video frame.

height

The height of the video frame.

orientation

The orientation of the video frame in degrees. Should be a multiple of 90.

areaOfInterest

The rectangular area of the frame where the text is expected to be. For example, it may be selected by the user or highlighted in your application interface.

Note: *You can also change the area of interest while the service is running by calling the* ***setAreaOfInterest*** *method.*

Note: *Before recognition, the service rotates the image obtained from camera in order to bring text orientation to normal (horizontal). The area of interest is specified in the coordinates on this rotated image, which are different from the coordinates on the video frame except the case when the frame orientation is 0.*

stop method of the IRecognitionService interface

Stops processing and releases the resources used by the service.

```
void stop();
```

submitRequestedFrame method of the IRecognitionService interface

Submits the video frame obtained from the camera after it is requested through the **Callback.onRequestLatestFrame** method.

```
void submitRequestedFrame( byte[] buffer );
```

Parameters

buffer

The buffer filled with image data for the latest frame. Only NV21 format is currently supported. This should be the same buffer which has been passed via the call to the **Callback.onRequestLatestFrame** method.

ResultStabilityStatus enum

Result stability status: an 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 Available.

```
enum ResultStabilityStatus {
    NotReady,
    Tentative,
    Verified,
    Available,
    TentativelyStable,
    Stable
};
```

Constants

Name	Description
NotReady	No content available.
Tentative	Content detected on a single frame.
Verified	Content verified: matching content found in at least two frames.
Available	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.
TentativelyStable	The result has been stable in the last two frames.
Stable	The result has been stable in the last three or more frames.

Warning enum

A warning that occurred during processing.

```
enum Warning {
    TextTooSmall
}
```

Constants

Name	Description
TextTooSmall	The text is too small. Advise the end user to move the camera closer or zoom in.

IRecognitionCoreAPI interface

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.

Use the object on the thread on which it was created; you may also create several objects on different threads and use them concurrently. All method calls are synchronous (will not return until the operation is completed), so should not be used on the UI thread.

```
public interface IRecognitionCoreAPI
```

Methods

Name	Description
close	Releases the resources.
getProcessingSettings	Provides access to the general processing settings.
getTextRecognitionSettings	Provides access to the settings of text recognition.
recognizeText	Performs recognition of an image.

Nested classes

Name	Description
CharInfo	Extended information about the character formatting. ***Important!*** *This class is reserved for future use.*
ProcessingSettings	The general settings which are the same for different processing scenarios.
TextBlock	A collection of recognized text lines found in a text area (block) on the image.
TextLine	A line of recognized text; the location and additional information are also available.
TextRecognitionCallback	A callback interface to manage the processing: obtain information about progress and errors, terminate the operation if necessary.

Name	Description
TextRecognitionSettings	The settings for text recognition.

Enumerations

Name	Description
Warning	A warning that occurred during processing.

ProcessingSettings interface

The general settings which are the same for different processing scenarios.

```
interface ProcessingSettings
```

Methods

Name	Description
getProcessingThreadsCount	Gets the number of processing threads to be used.
setProcessingThreadsCount	Sets the number of processing threads to be used.

getProcessingThreadsCount method of the ProcessingSettings interface

Gets the number of processing threads to be used by the service.

```
int getProcessingThreadsCount();
```

Return values

The method returns the number of threads. Returns 0 if the number of threads is to be determined automatically, which is the default setting.

setProcessingThreadsCount method of the ProcessingSettings interface

Sets the number of processing threads to be used by the service.

```
void setProcessingThreadsCount( int ThreadsCount );
```

Parameters

ThreadsCount

The new number of threads. Up to 16 threads are allowed. Set to 0 to determine the number of threads automatically.

TextRecognitionCallback interface

A callback interface to manage the processing: obtain information about progress and errors, terminate the operation if necessary. This interface and its methods are to be implemented on the client side.

```
interface TextRecognitionCallback
```

Methods

Name	Description
onError	Reports an error.
onProgress	Reports the approximate percentage of operation completed and delivers the warnings that occurred during processing. Allows you to cancel processing.
onTextOrientationDetected	Informs the client application about the orientation of the image.

onError method of the TextRecognitionCallback interface

Reports an error.

This method is to be implemented on the client side, which may include displaying the error description to the user or handling it otherwise.

```
void onError( Exception error )
```

Parameters

error

The **Exception** object for the error that has occurred.

onProgress method of the TextRecognitionCallback interface

Reports the approximate percentage of operation completed and delivers the warnings that occurred during processing. Allows you to cancel processing.

This method is to be implemented on the client side, which may include a progress indicator and/or a message to the user about the warnings.

```
boolean onProgress( int percentage, Warning warning )
```

Parameters

percentage

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

warning

The warning which occurred, if any; represented by a constant of the **Warning** enumeration.

Return values

The method should return **true** if you wish to terminate the current operation, **false** otherwise.

onTextOrientationDetected method of the TextRecognitionCallback interface

Informs the client application about the orientation of the image. This may be useful if you wish to rotate the view for the user.

Note that the coordinates of the text, after the **recognizeText** method call, will be returned on the image rotated to normal orientation, so you will need to take the rotation angle into account if you intend to use those coordinates.

This method is to be implemented on the client side, which may include displaying the error description to the user or handling it otherwise.

```
void onTextOrientationDetected( int angle )
```

Parameters

angle

The angle on which the image should be rotated to get normal orientation. Possible values are: 0, 90, 180, 270.

TextRecognitionSettings interface

The settings for text recognition scenario.

```
interface TextRecognitionSettings
```

Methods

Name	Description
setAreaOfInterest	Sets the area on the image where the text is to be found.
setRecognitionLanguage	Sets the languages to be used for recognition.

setRecognitionLanguage method of the TextRecognitionSettings interface

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 slow down performance.

```
void setRecognitionLanguage( Language... languages );
```

Parameters

languages

One or more languages to be used for recognition, represented each by a constant of the Language enumeration.

setAreaOfInterest method of the TextRecognitionSettings interface

Sets the area on the image where the text is to be found. By default, no area of interest is selected, and the whole image is considered to contain text blocks.

```
void setAreaOfInterest( Rect areaOfInterest );
```

Parameters

areaOfInterest

The rectangle specifying the area of interest.

CharInfo class

Extended information about the character formatting.

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

```
final class CharInfo {
    public final Rect Rect;
    public final Point[] Quadrangle;
    public final int ForegroundColor;
    public final int BackgroundColor;
    public final int Attributes;
}
```

Properties

Name	Type	Description

Attributes	**int**	Character attributes as the OR combination of the following flags:
		`int CHAR_ATTRIBUTE_ITALIC = 0x0;`
		`int CHAR_ATTRIBUTE_BOLD = 0x1;`
		`int CHAR_ATTRIBUTE_UNDERLINED = 0x2;`
		`int CHAR_ATTRIBUTE_STRIKETHROUGH = 0x4;`
		`int CHAR_ATTRIBUTE_SMALLCAPS = 0x8;`
		`int CHAR_ATTRIBUTE_SUPERSCRIPT = 0x10;`
		`int CHAR_ATTRIBUTE_UNCERTAIN = 0x10000;`

BackgroundColor	**int**	The color of the background.

Name	Type	Description
		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	**int**	The color of the symbol.
Quadrangle	**Point[]**	The four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left.
Rect	**Rect**	The bounding rectangle of the symbol.

TextBlock class

A collection of recognized text lines found in a text area (block) on the image.

```
final class TextBlock {
    public final TextLine[] TextLines;
}
```

Properties

Name	Type	Description
TextLines	TextLine[]	The lines of recognized text.

TextLine class

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

```
final class TextLine {
    public final String Text;
    public final Rect Rect;
    public final Point[] Quadrangle;
```

```
    public final CharInfo[] CharInfo;
}
```

Properties

Name	Type	Description
CharInfo	CharInfo[]	Additional information about the characters. ***Important!*** *This property is reserved for future use.*
Quadrangle	**Point**[]	The four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left. ***Note:*** *Before recognition, the service rotates the image obtained from camera in order to bring text orientation to normal (horizontal). The vertex coordinates are specified for this rotated image and may require coordinate conversion if you display the quadrangle on the video frame.*
Rect	**Rect**	The bounding rectangle of the text line.
Text	**String**	The recognized text.

close method of the IRecognitionCoreAPI interface

Releases the resources used by the object.

```
void close();
```

getTextRecognitionSettings method of the IRecognitionCoreAPI interface

Provides access to the settings for text recognition.

```
TextRecognitionSettings getTextRecognitionSettings();
```

Return values

This method returns an object implementing the **TextRecognitionSettings** interface, which allows you to change the settings for text recognition scenario.

getProcessingSettings method of the IRecognitionCoreAPI interface

Provides access to general processing settings common to all scenarios.

```
ProcessingSettings getProcessingSettings();
```

Return values

This method returns an object implementing the **ProcessingSettings** interface, which allows you to change the general processing settings.

recognizeText method of the IRecognitionCoreAPI interface

Performs recognition of a single image.

```
TextBlock[] recognizeText( Bitmap image, TextRecognitionCallback callback )
```

Parameters

image

The image to be recognized.

callback

An object implementing the **TextRecognitionCallback** interface which will be used to report progress and terminate the processing if required.

Return values

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

Warning enum

A warning that occurred during processing.

```
enum Warning {
    RecognitionIsSlow,
    ProbablyLowQualityImage,
    ProbablyWrongLanguage,
    WrongLanguage,
    TextTooSmall
}
```

Constants

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

Language enum

The language of the text. See Available Languages for a full list with information on features support for each language.

```
public enum Language {
    Afrikaans,
    Albanian,
    Basque,
    Belarusian,
    Breton,
    Bulgarian,
    Catalan,
    Chechen,
    ChineseSimplified,
    ChineseTraditional,
    CrimeanTatar,
    Croatian,
    Czech,
    Danish,
    Dutch,
    DutchBelgian,
    English,
    Estonian,
    Fijian,
    Finnish,
    French,
    German,
    GermanNewSpelling,
    Greek,
```

```
    Hawaiian,
    Hungarian,
    Icelandic,
    Indonesian,
    Irish,
    Italian,
    Japanese,
    Kabardian,
    Korean,
    Latin,
    Latvian,
    Lithuanian,
    NorwegianBokmal,
    NorwegianNynorsk,
    Macedonian,
    Malay,
    Maori,
    Moldavian,
    Mongol,
    Ossetic,
    Polish,
    Portuguese,
    PortugueseBrazilian,
    Provencal,
    RhaetoRomanic,
    Romanian,
    Russian,
    Samoan,
    Serbian,
    Slovak,
    Slovenian,
    Spanish,
    Swahili,
    Swedish,
    Tagalog,
    Tatar,
    Turkish,
    Ukrainian,
    Welsh
}
```

Specifications

This section describes the technical requirements and capabilities of ABBYY Real-Time Recognition SDK.

Device Requirements

Android version: 4.4 or later for ARMv7 (armeabi-v7a) and ARMv8 (arm64-v8a) processors

Processor:

- Arm NEON support
- multi-core (augmented reality scenarios require **4 cores**)

Camera:

- autofocus lens
- HD preview: generally recommended frame size is **720×1280**, but it can vary depending on the scenario and processing speed

Memory requirements

Library operation in the text capture scenario 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 **170** MB RAM.

Distribution Kit

ABBYY Real-Time Recognition 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	RtrSdkDevelopersGuide.pdf	This Developer's Guide.
help/javadoc	All files in this folder.	The library API Reference in Javadoc-generated HTML format.
sample-datacapture	All files in this folder.	The sample code implementing a data capture scenario where the

Folder	File name	Description
		capture rule is specified by a regular expression.
sample-textcapture	All files in this folder.	The sample code implementing a simple text capture scenario.
scenarios-datacapture/ru-passport/sample-passport	All files in this folder. Note that the **scenarios-datacapture** folder also contains some redistributable assets.	The sample code implementing a data capture scenario for Russian internal passport.

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	abbyy-rtr-sdk-1.0.aar	The ABBYY Real-Time Recognition SDK library file.	Always required.
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.	

Folder	File name	Description	Distribution
	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) 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	

Folder	File name	Description	Distribution
		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.	
	Ukrain.edc	Ukrainian language recognition dictionary.	
scenarios-datacapture/ru-passport/assets/dictionaries	Passport_RU.edc Passport_RU_Numbers.edc	Dictionaries for Russian passport recognition.	Required for Russian passport recognition.
assets/patterns	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 except Chinese, Japanese, and Korean.	Required for all recognition languages except Chinese, Japanese and Korean.

Folder	File name	Description	Distribution
	FindText.rom	Recognition database for all languages.	Always required.
	KoreanSpecific.rom	Recognition database for Korean language.	Required for recognition of texts in Korean language.
scenarios-datacapture/assets/patterns	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). Note that this file may be not available in your distribution, depending on the type of your license.
	BankCards_EDC.rom	Bank card recognition database.	Required for bank card recognition. ***Note:*** *This file may be not available in your distribution, depending on the type of your license.*
	ID_AE_EDC.rom	Recognition database for UAE documents.	Only the databases for the countries you are going to support are required. ***Note:*** *These files may be not available in your distribution, depending on the type of your license.*
	ID_AT_EDC.rom	Recognition database for Austrian documents.	

Folder	File name	Description	Distribution
	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_BY_EDC.rom	Recognition database for Belarusian documents.	
	ID_CH_EDC.rom	Recognition database for Swiss documents.	
	ID_CN_EDC.rom	Recognition database for Chinese documents.	
	ID_CZ_EDC.rom	Recognition database for Czech documents.	
	ID_DE_EDC.rom	Recognition database for German documents.	
	ID_EE_EDC.rom	Recognition database for Estonian 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_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.	
	ID_KW_EDC.rom	Recognition database for Kuwait documents.	
	ID_KZ_EDC.rom	Recognition database for Kazakhstan documents.	
	ID_LV_EDC.rom	Recognition database for Latvian documents.	
	ID_MY_EDC.rom	Recognition database for Malaysian documents.	
	ID_PH_EDC.rom	Recognition database for Philippine documents.	
	ID_PL_EDC.rom	Recognition database for Polish documents.	

Folder	File name	Description	Distribution
	ID_PT_EDC.rom	Recognition database for Portuguese documents.	
	ID_RO_EDC.rom	Recognition database for Romanian 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_SY_EDC.rom	Recognition database for Syrian documents.	
	ID_TR_EDC.rom	Recognition database for Turkish documents.	
	ID_UK_EDC.rom	Recognition database for British documents.	
	ID_US_EDC.rom	Recognition database for USA documents.	
scenarios-datacapture/ru-passport/assets/patterns	Passport_RU.rom	Recognition database for Russian passports.	Required for Russian passport recognition.
assets/translation	Menu_CH-EN.trdic	Dictionary for translating menus	The files contain translation dictionaries.

Folder	File name	Description	Distribution
		from Chinese to English.	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 from English to Polish.	

Folder	File name	Description	Distribution
	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 translating menus	

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

Available OCR Languages

This section lists the languages available for text processing with ABBYY Real-Time Recognition 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 (Language enum constant)	Recognition language	Can be used for OCR	Full dictionary support
Afrikaans	Afrikaans	+	
Albanian	Albanian	+	
Basque	Basque	+	
Belarusian	Belarusian	+	
Breton	Breton	+	
Bulgarian	Bulgarian	+	+
Catalan	Catalan	+	
Chechen	Chechen	+	

Internal name (Language enum constant)	Recognition language	Can be used for OCR	Full dictionary support
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)	+	+
GermanNewSpelling	German (new spelling)	+	+
Greek	Greek	+	+
Hawaiian	Hawaiian	+	

Internal name (Language enum constant)	Recognition language	Can be used for OCR	Full dictionary support
Hungarian	Hungarian	+	
Icelandic	Icelandic	+	
Indonesian	Indonesian	+	+
Irish	Irish	+	
Italian	Italian	+	+
Japanese	Japanese	+	
Kabardian	Kabardian	+	
Korean	Korean	+	
Latin	Latin	+	
Latvian	Latvian	+	
Lithuanian	Lithuanian	+	
Macedonian	Macedonian	+	
Malay	Malay	+	
Maori	Maori	+	
Moldavian	Moldavian	+	
Mongol	Mongol	+	
NorwegianBokmal	Norwegian (Bokmal)	+	+

Internal name (Language enum constant)	Recognition language	Can be used for OCR	Full dictionary support
NorwegianNynorsk	Norwegian (Nynorsk)	+	+
Ossetic	Ossetic	+	
Polish	Polish	+	+
PortugueseBrazilian	Portuguese (Brazil)	+	+
Portuguese	Portuguese (Portugal)	+	+
Provencal	Provencal	+	
RhaetoRomanic	Rhaeto-Romanic	+	
Romanian	Romanian	+	
Russian	Russian	+	+
Samoan	Samoan	+	
Serbian	Serbian	+	
Slovak	Slovak	+	
Slovenian	Slovenian	+	
Spanish	Spanish	+	+
Swahili	Swahili	+	
Swedish	Swedish	+	+
Tagalog	Tagalog	+	

Internal name (Language enum constant)	Recognition language	Can be used for OCR	Full dictionary support
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 Real-Time Recognition 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 and indent	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, Moldova, New Zealand, Norway, Poland, Portugal, Russian Federation, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey, UK, Ukraine, USA (Alabama, Arizona, California, Florida, Washington DC, Massachusetts, Maine, Michigan, New Mexico, Texas, Washington), Uzbekistan, Vietnam
International Passport	Albania, Algeria, Armenia, Austria, Brazil, China, Croatia, Czech Republic, Georgia, Germany, Greece, Hungary, India, Israel, Italy, Japan, Kazakhstan, Kyrgyzstan, 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, Croatia, Cyprus, Czech Republic, Egypt, Estonia, Finland, France, Georgia, Germany, Hong Kong, Hungary, Israel, Italy, Kazakhstan, Kuwait, Kyrgyzstan, Latvia, Lithuania, Macedonia, Malaysia, Mexico, Moldova, Nigeria, Norway, Poland, Portugal, Romania, El Salvador, Serbia, Slovakia, Slovenia, Singapore, South Africa, Spain, Switzerland, Turkey, UAE, Ukraine
National passport	Belarus, Russian Federation
INN	Russian Federation

Document	Supported in
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, El Salvador, Kazakhstan, Russian Federation, Ukraine
Vehicle Passport - PTS	Russian Federation
VISA	Russian Federation, USA
Border Crossing Card	USA
Passport Card	USA
Health insurance card	Japan
Work permit	Russian Federation, Singapore
Residence permit	Austria, Germany, Russian Federation, Slovakia, Slovenia, Spain
Asylum Residence Permit	Austria
Migration Card	Russian Federation

Document	Supported in
Permanent residency card	USA
Residence License	Brazil
Crew Member Certificate	South Africa
Military, Police and Soldier ID	Russian Federation

The countries and documents mentioned are supported in technical preview mode. We are working to improve our technologies.

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.

Document type	Profile name	Result scheme	Result description
Bank card	BankCards	BankCardEmbossed	Bank cards with embossed fields (front side)
		BankCardUnembossed	Bank cards with indent-printed fields (front side)
International bank account number	IBAN	IBAN	International bank account number
Machine-readable document zone	MRZ	MRZ	MRZ of a passport
		MRZ_MRP	ICAO Doc 9303 machine-readable passports (2 lines, 44 characters each)

Document type	Profile name	Result scheme	Result description
		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)
		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)
		MRZ_BG	MRZ-like zone of the Bulgarian vehicle registration document (3 lines, 30 characters each)
		MRZ_CH	MRZ-like zone of the Swiss driver's license (3 lines, 9, 30 and 30 characters)
		MRZ_FR	MRZ-like zone of the French national identity card (2 lines, 36 characters each)
		MRZ_RU	MRZ-like zone of the Russian passport (2 lines, 44 characters each)

Document type	Profile name	Result scheme	Result description
		MRZ_MRV_RU	MRZ-like zone of the Russian visa (2 lines, 44 characters each)
Armenian ID card	ID_AM	ID_AM_TYPE1	Armenian ID card (front)
Austrian passport	InternationalPassport_AT	InternationalPassport_AT	Austrian international passport (main page)
Austrian ID card	ID_AT	ID_AT	Austrian ID card (front)
Austrian driver's license	DriverLicense_AT	DriverLicense_AT_TYPE1	Austrian driver's license with the title at the top (front side)
		DriverLicense_AT_TYPE2	Austrian driver's license with the title in the top-right corner (front side)
Bahrain ID card	ID_BH	ID_BH_TYPE1	Bahrain ID card (front)
Belarusian passport	Passport_BY	Passport_BY	Belarusian passport (main page)
		Passport_BY_PAGE31	Belarusian passport (page 31)
Belarusian driver's license	DriverLicense_BY	DriverLicense_BY	Belarusian driver's license (front side)
Belgium ID card	ID_BE	ID_BE	Belgium ID card (front side)
Belgium driving license	DriverLicense_BE	DriverLicense_BE_TYPE1	Belgium driving license, in English (front side)

Document type	Profile name	Result scheme	Result description
British passport	InternationalPassport_UK	InternationalPassport_UK_TYPE1	Non-biometric British passport (main page)
		InternationalPassport_UK_TYPE2	Biometric British passport (main page)
British driver's license	DriverLicense_UK	DriverLicense_UK_TYPE1	British driver's license (front side)
		DriverLicense_UK_TYPE2	British driver's license with a logo on the right (front side)
		DriverLicense_UK_PROVISIONAL	British driver's license, provisional (front side)
Bulgarian ID card	ID_BG	ID_BG_TYPE1	Bulgarian ID card (new type, front side)
		ID_BG_TYPE2	Bulgarian ID card (old type, front side)
Chile ID card	ID_CL	ID_CL_TYPE1	Chile identity card (front side)
Chinese passport	InternationalPassport_CN	InternationalPassport_CN_TYPE1	Chinese passport (old type, main page)
		InternationalPassport_CN_TYPE2	Chinese passport (new type, main page)
Croatian ID card	ID_HR	ID_HR_TYPE1	Croatian identity card (older type, front side)
		ID_HR_TYPE2	Croatian identity card (newer type, front side)

Document type	Profile name	Result scheme	Result description
Czech ID card	ID_CZ	ID_CZ	Czech ID card (front side)
Czech driver's license	DriverLicense_CZ	DriverLicense_CZ	Czech driver's license card (front side)
Estonian ID card	ID_EE	ID_EE	Estonian ID card (front side)
Finnish ID card	ID_FI	ID_FI_TYPE1	Finnish identity card (older type, front side)
		ID_FI_TYPE2	Finnish identity card (newer type, front side)
Finnish driver's license	DriverLicense_FI	DriverLicense_FI	Finnish driver's license (front side)
French ID card	ID_FR	ID_FR_TYPE1	French identity card (front side)
Georgian ID card	ID_GE	ID_GE_TYPE1	Georgian identity card (front side)
Georgian passport	InternationalPassport_GE	InternationalPassport_GE_TYPE1	Georgian passport (newer type, main page)
		InternationalPassport_GE_TYPE2	Georgian passport (older type, main page)
German passport	InternationalPassport_DE	InternationalPassport_DE_TYPE1	German passport (main page)
		InternationalPassport_DE_TYPE2	German passport (main page), with two lines for the first name

Document type	Profile name	Result scheme	Result description
German ID card	ID_DE	ID_DE_TYPE1	German ID card (front side)
		ID_DE_TYPE2	German ID card with TD-1 MRZ on the front (front side)
German driver's license	DriverLicense_DE	DriverLicense_DE	German driver's license (front side)
Greek driving license	DriverLicense_GR	DriverLicense_GR_TYPE1	Greek driving license (front side)
Hong Kong ID card	ID_HK	ID_HK_TYPE1	Hong Kong identity card (front side)
Hungarian driver's license	DriverLicense_HU	DriverLicense_HU_TYPE1	Hungarian driver's license (front side)
Hungarian ID card	ID_HU	ID_HU_TYPE1	Hungarian identity card (older type, front side)
		ID_HU_TYPE2	Hungarian identity card (newer type, front side)
Indian Aadhaar card	Aadhaar_IN	Aadhaar_IN_TYPE1	Indian card with Aadhaar number
Indian passport	InternationalPassport_IN	InternationalPassport_IN	Indian passport (main page)
Israeli driver's license	DriverLicense_IL	DriverLicense_IL	Israeli driver's license (front side)
Italian passport	InternationalPassport_IT	InternationalPassport_IT	Italian passport (main page)

Document type	Profile name	Result scheme	Result description
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)
Japanese passport	InternationalPassport_JP	InternationalPassport_JP	Japanese passport (main page)
Japanese driver's license	DriverLicense_JP	DriverLicense_JP	Japanese driver's license (front side)
Japanese health insurance	HealthInsuranceCard_JP	HealthInsuranceCard_JP	Japanese health insurance card (front side)
Kazakhstan ID card	ID_KZ	ID_KZ_TYPE1	Kazakhstan ID card with 2-line MRZ (front and back sides)
		ID_KZ_TYPE2	Kazakhstan ID card with 3-line MRZ (front and back sides)
Kuwait ID card	ID_KW	ID_KW_TYPE1	Kuwait ID card (card-size, front side)
Kyrgyzstani ID card	ID_KG	ID_KG	Kyrgyzstani ID card (front and back)
Latvian ID card	ID_LV	ID_LV	Latvian ID card (front side)
Lithuanian ID card	ID_LT	ID_LT_TYPE1	Lithuanian ID card (front side)
Malaysian ID card	ID_MY	ID_MY	Malaysian ID card (front side)

Document type	Profile name	Result scheme	Result description
New Zealand driving license	DriverLicense_NZ	DriverLicense_NZ_TYPE1	New Zealand driving license (new type, front side)
Nigerian ID card	ID_NG	ID_NG_TYPE1	Nigerian ID card (front side)
Norwegian driving license	DriverLicense_NO	DriverLicense_NO_TYPE1	Norwegian driving license (front side)
Norwegian ID card	ID_NO	ID_NO_TYPE1	Norwegian ID card (front side)
Philippine passport	InternationalPassport_PH	InternationalPassport_PH_TYPE1	Non-biometric Philippine passport (main page)
		InternationalPassport_PH_TYPE2	Biometric Philippine passport (main page)
Polish ID card	ID_PL	ID_PL_TYPE1	Polish ID card, older type (front and back)
		ID_PL_TYPE2	Polish ID card, newer type (front and back)
Polish driver's license	DriverLicense_PL	DriverLicense_PL_TYPE1	Polish driver's license, old type (front side)
Portuguese ID card	ID_PT	ID_PT_TYPE1	Portuguese ID card (front side)
Portuguese driver's license	DriverLicense_PT	DriverLicense_PT_TYPE1	Portuguese driver's license (front side)
Romanian ID card	ID_RO	ID_RO	Romanian ID card (front side)

Document type	Profile name	Result scheme	Result description
Russian international biometric passport	InternationalPassport_RU	InternationalPassport_RU	Russian international biometric passport (main page)
Russian visa	Visa_RU	Visa_RU_TYPE1	Russian visa
Russian passport	Passport_RU	Passport_RU	Russian passport (pages 2 and 3)
		Passport_RU_Top	Russian passport (page 2, with signatures)
		Passport_RU_Bottom	Russian passport (page 3, with a photo)
Russian birth certificate	BirthCertificate_RU	BirthCertificate_RU_TYPE1	Russian birth certificate
Russian insurance individual account number (SNILS)	SocialSecurityNumber_RU	SocialSecurityNumber_RU_TYPE1	Laminated SNILS (front side)
		SocialSecurityNumber_RU_TYPE2	Card-size SNILS (front side)
Russian driver's license	DriverLicense_RU	DriverLicense_RU	Russian driver's license, new type (front side)
		DriverLicense_RU_TYPE1	Russian driver's license, old type (front side)
		DriverLicense_RU_TYPE2	Russian driver's license, old type, vertical (front side)
		DriverLicense_RU_TYPE3	Russian driver's license, new type (front

Document type	Profile name	Result scheme	Result description
			side)
Russian vehicle registration certificate	VehicleRegistration_RU	VehicleRegistration_RU_TYPE1	Russian vehicle registration certificate, old type (front and back sides)
		VehicleRegistration_RU_TYPE2	Russian vehicle registration certificate, new type (front and back sides)
Salvadorean ID card	ID_SV	ID_SV_TYPE1	Salvadorean ID card (front side)
Serbian driving license	DriverLicense_RS	DriverLicense_RS_TYPE1	Serbian driving license (front side)
Serbian ID card	ID_RS	ID_RS_TYPE1	Serbian ID card (front side)
Singapore ID card	ID_SG	ID_SG	Singapore ID card (front side)
Singapore work permit	WorkPermit_SG	WorkPermit_SG_TYPE1	Singapore work permit (front side)
South African Republic ID card	ID_ZA	ID_ZA_TYPE1	South African Republic ID card (front side)
Spanish ID card	ID_ES	ID_ES_TYPE1	Spanish ID card (old type, front side)
		ID_ES_TYPE2	Spanish ID card (new type, front side)
Spanish driving license	DriverLicense_ES	DriverLicense_ES_TYPE1	Spanish driving license (older type, front side)

Document type	Profile name	Result scheme	Result description
		DriverLicense_ES_TYPE2	Spanish driving license (newer type, front side)
Spanish residence permit	ResidencePermit_ES	ResidencePermit_ES_TYPE1	Spanish residence permit, blue color (front side)
		ResidencePermit_ES_TYPE2	Spanish residence permit, pink color (front side)
Swedish driver's license	DriverLicense_SE	DriverLicense_SE_TYPE1	Swedish driver's license with a photo near the signature (front side)
		DriverLicense_SE_TYPE2	Swedish driver's license with a logo on the right (front side)
Swiss ID card	ID_CH	ID_CH_TYPE1	Swiss ID card (front side)
Swiss driver's license	DriverLicense_CH	DriverLicense_CH_TYPE1	Swiss driver's license (front side)
Syrian passport	InternationalPassport_SY	InternationalPassport_SY	Syrian passport (main page)
Tajikistani passport	InternationalPassport_TJ	InternationalPassport_TJ_TYPE1	Tajikistani passport in Latin only (main page)
Turkish ID card	ID_TR	ID_TR_TYPE1	Turkish ID card (front side)
Turkish driver's license	DriverLicense_TR	DriverLicense_TR_TYPE1	Turkish driver's license (front side)

Document type	Profile name	Result scheme	Result description
UAE ID card	ID_AE	ID_AE_TYPE1	UAE ID card (front side)
USA passport	InternationalPassport_US	InternationalPassport_US_TYPE1	USA passport, old type (main page)
		InternationalPassport_US_TYPE2	USA passport, new type (main page)
USA driver's license	DriverLicense_US	DriverLicense_US_AL_TYPE1	USA Alabama driver's license (front side)
		DriverLicense_US_AZ_TYPE1	USA Arizona driver's license, card size (front side)
		DriverLicense_US_CA_TYPE1	USA California driver's license, card size (front side)
		DriverLicense_US_DC_TYPE1	USA Washington DC driver's license, older type (front side)
		DriverLicense_US_DC_TYPE2	USA Washington DC driver's license, newer type (front side)
		DriverLicense_US_FL_TYPE1	USA Florida driver's license (front side)
		DriverLicense_US_MA_TYPE1	USA Massachusetts driver's license, card size (front side)
		DriverLicense_US_ME_TYPE1	USA Maine driver's license (front side)

Document type	Profile name	Result scheme	Result description
		DriverLicense_US_MI_TYPE1	USA Michigan driver's license (front side)
		DriverLicense_US_NM_TYPE1	USA New Mexico driver's license, card size (front side)
		DriverLicense_US_TX_TYPE1	USA Texas driver's license (front side)
		DriverLicense_US_WA_TYPE1	USA Washington driver's license (front side)
USA permanent residency card (Green card)	GreenCard_US	GreenCard_US_TYPE1	USA permanent residency card, also known as Green card (front side)
Uzbek passport	InternationalPassport_UZ	InternationalPassport_UZ_TYPE1	Uzbek passport (main page)

The following table lists field identifiers used in result data schemes returned by the Data Capture service.

Scheme	Field	Field description	Comments
Aadhaar_IN_TYPE1	Number	Aadhaar number	
BankCardEmbossed BankCardUnembossed	Number	Card number	
	FullName	Cardholder's full name	
	DateOfExpiry	Card expiry date	
BirthCertificate_RU_TYPE1	FullNumber	Full document number (series and number, including the number sign)	

Scheme	Field	Field description	Comments
	Series	Document series (two Latin and two Cyrillic letters, separated by a hyphen)	
	Number	Document number (not including the number sign)	
	DateOfIssue	Document issue date	
	DayOfIssue	The day of issue date	
	MonthOfIssue	The month of issue date	
	YearOfIssue	The year of issue date	
	Sex	Document holder's sex	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	MiddleName	Document holder's patronymic name	
	FirstNameMiddleName	Document holder's first name and patronymic	
	DateOfBirth	Document holder's birth date	
DriverLicense_AT_TYPE1	Number	License number	

Scheme	Field	Field description	Comments
DriverLicense_AT_TYPE 2	LastName	Driver's last name	
	FirstName	Driver's first name	
	DateOfBirthPlaceOfBirth	Driver's date and place of birth	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	In the DriverLicense_AT_TYPE 1 scheme only
	PlaceOfIssue	Region where the license was issued	
DriverLicense_BE_TYPE 1	DateOfIssue	License issue date	
	LastName	Driver's last name	
	FirstName	Driver's first name	
	Number	License number	
	DateOfExpiry	License expiry date	
	DateOfBirth	Driver's date of birth	
DriverLicense_BY	Number	License number	
	LastName	Driver's last name	
	FirstName	Driver's first name	

Scheme	Field	Field description	Comments
	MiddleName	Driver's patronymic name	
	FirstNameMiddleName	Driver's first and patronymic names	
	DateOfBirth	Driver's date of birth	
	PlaceOfBirth	Driver's place of birth	
	DateOfIssue	License issue date	
	DateOfExpiry	License expiry date	
DriverLicense_CH_TYPE1	Number	License number	
	LastName	Driver's last name	
	FirstName	Driver's first name	
	DateOfBirth	Driver's date of birth	
	PlaceOfBirth	Driver's place of birth	
	PlaceOfIssue	Region where the license was issued	
	DateOfIssue	License issue date	
	DateOfExpiry	License expiry date	
DriverLicense_CZ	Number	License number	
	DriverID	Driver's personal identifier	

Scheme	Field	Field description	Comments
	LastName	Driver's last name	
	FirstName	Driver's first name	
	DateOfBirth	Driver's date of birth	
	DateOfIssue	License issue date	
	DateOfExpiry	License expiry date	
DriverLicense_DE	Number	License number	
	LastName	Driver's last name	
	LastName_LINE2	Second line of the driver's last name	
	FirstName	Driver's first name	
	DateOfIssue	License issue date	
	PlaceOfBirth	Driver's place of birth	
	DateOfBirth	Driver's date of birth	
	PlaceOfIssue	Region where the license was issued	
	PlaceOfIssue_LINE2	Region where the license was issued, continued	
DriverLicense_ES_TYPE 1 DriverLicense_ES_TYPE 2	DateOfIssue	License issue date	
	DateOfBirth	Driver's date of birth	

Scheme	Field	Field description	Comments
	FirstName	Driver's first name	
	LastName	Driver's last name	
	LastName_LINE2	Driver's last name, continued	
	Number	License number	
	DateOfExpiry	License expiry date	
DriverLicense_FI	Number	License number	
	DriverID	Driver's personal identifier	
	LastName	Driver's last name	
	FirstName	Driver's first name	
	Nationality	Driver's nationality	
	DateOfBirth	Driver's date of birth	
	DateOfIssue	License issue date	
	DateOfExpiry	License expiry date	
	PlaceOfIssue	Region where the license was issued	
DriverLicense_GR_TYPE1	LastName	Driver's last name	
	FirstName	Driver's first name	

Scheme	Field	Field description	Comments
	DateOfBirth	Driver's birth date	
	DateOfIssue	License issue date	
	DateOfExpiry	License expiry date	
	IssuedBy	The authority that issued the license	
	PersonalCode	Driver's personal code	
	Number	License number	
DriverLicense_HU_TYPE1	FirstName	Driver's first name	
	LastName	Driver's last name	
	DateOfIssue	License issue date	
	DateOfExpiry	License expiry date	
	DateOfBirth	Driver's birth date	
	Number	License number	
DriverLicense_IL	Number	License number	
	DriverID	Driver's personal identifier	
	LastName	Driver's last name	
	FirstName	Driver's first name	
	DateOfBirth	Driver's date of birth	

Scheme	Field	Field description	Comments
	DateOfIssue	License issue date	
	DateOfExpiry	License expiry date	
DriverLicense_IT_TYPE1 DriverLicense_IT_TYPE2	Number	License number	
	LastName	Driver's last name	
	FirstName	Driver's first name	
	DateOfBirth	Driver's date of birth	
	DateOfIssue	License issue date	
	DateOfExpiry	License expiry date	
DriverLicense_JP	Number	License number	
DriverLicense_NO_TYPE1	FirstName	Driver's first name	
	LastName	Driver's last name	
	DateOfIssue	License issue date	
	DateOfExpiry	License expiry date	
	DateOfBirth	Driver's birth date	
	Number	License number	
	IssuedBy	The authority that issued the license	
	ReferenceNumber	Reference number	

Scheme	Field	Field description	Comments
DriverLicense_NZ_TYPE1	Number	License number	
	Version	License version	
	DateOfBirth	Driver's birth date	
	LastName	Driver's last name	
	FirstName	Driver's first name	
	FullName_LINE2	Additional line for driver's name	
DriverLicense_PL_TYPE1	LastName	Driver's last name	
	FirstName	Driver's first name	
	Number	License number	
	DateOfBirth	License birth date	
	DateOfIssue	License issue date	
DriverLicense_PT_TYPE1	Number	License number	
	DriverID	Driver's personal identifier	
	LastName	Driver's last name	
	FirstName	Driver's first name	
	DateOfBirth	Driver's date of birth	
	DateOfIssue	License issue date	

Scheme	Field	Field description	Comments
	DateOfExpiry	License expiry date	
DriverLicense_RS_TYPE 1	FirstName	Driver's first name	
	LastName	Driver's last name	
	DateOfExpiry	License expiry date	
	DateOfIssue	License issue date	
	Number	License number	
	IssuedBy	The authority that issued the license	
DriverLicense_RU DriverLicense_RU_TYPE 1 DriverLicense_RU_TYPE 2 DriverLicense_RU_TYPE 3	Number	License number	
	Number_EX	License number	In the DriverLicense_RU_TYPE 2 scheme only. Some licenses of this type contain an additional field that repeats the license number. The numbers recognized from the Number and Number_EX fields should be the same.
	LastName	Driver's last name	
	FirstName	Driver's first name	Except the DriverLicense_RU scheme
	MiddleName	Driver's patronymic name	Except the DriverLicense_RU scheme

Scheme	Field	Field description	Comments
	FirstNameMiddleName	Driver's first and patronymic names	In the DriverLicense_RU, DriverLicense_RU_TYPE1 schemes only
	Sex	Driver's sex	
	DateOfBirth	Driver's date of birth	
	PlaceOfBirth	Driver's place of birth	In the DriverLicense_RU scheme only
	RegionOfResidence	Driver's region of residence	In the DriverLicense_RU scheme only
	IssuedBy	The authority that issued the license	In the DriverLicense_RU scheme only
	DateOfIssue	License issue date	
	DateOfExpiry	License expiry date	
DriverLicense_SE_TYPE1 DriverLicense_SE_TYPE2	Number	License number	
	DriverID	Driver's personal identifier	
	LastName	Driver's last name	
	FirstName	Driver's first name	
	DateOfBirth	Driver's date of birth	
	DateOfIssue	License issue date	
	DateOfExpiry	License expiry date	

Scheme	Field	Field description	Comments
DriverLicense_TR_TYPE1	Number	License number	
	DriverID	Driver's personal identifier	
	LastName	Driver's last name	
	FirstName	Driver's first name	
	DateOfBirth	Driver's date of birth	
	DateOfIssue	License issue date	
	DateOfExpiry	License expiry date	
DriverLicense_UK_TYPE1 DriverLicense_UK_TYPE2 DriverLicense_UK_PROVISIONAL	Number	License number	
	LastName	Driver's last name	
	FirstName	Driver's first name	
	DateOfBirth	Driver's date of birth	
	PlaceOfBirth	Driver's place of birth	
	DateOfBirthPlaceOfBirth	Driver's date and place of birth	
	DateOfIssue	License issue date	
	DateOfExpiry	License expiry date	
	IssuedBy	The authority that issued the license	

Scheme	Field	Field description	Comments
DriverLicense_US_AL_TYPE1 DriverLicense_US_AZ_TYPE1 DriverLicense_US_CA_TYPE1 DriverLicense_US_DC_TYPE1 DriverLicense_US_DC_TYPE2 DriverLicense_US_FL_TYPE1 DriverLicense_US_MA_TYPE1 DriverLicense_US_ME_TYPE1 DriverLicense_US_MI_TYPE1 DriverLicense_US_NM_TYPE1 DriverLicense_US_TX_TYPE1 DriverLicense_US_WA_TYPE1	Number	License number	
	LastName	Driver's last name	Except the DriverLicense_US_DC_TYPE1 and DriverLicense_US_MI_TYPE1 schemes
	FirstName	Driver's first name	Except the DriverLicense_US_DC_TYPE1 and DriverLicense_US_MI_TYPE1 schemes
	FirstName_LINE2	Second line of the driver's first name	In the DriverLicense_US_DC_TYPE2 scheme only
	FullName	Driver's full name	In the DriverLicense_US_DC_TYPE1 and DriverLicense_US_MI_TYPE1 schemes only
	DateOfBirth	Driver's date of birth	
	DateOfIssue	License issue date	
	DateOfExpiry	License expiry date	
HealthInsuranceCard_JP	SerialNumber	Insurance number	
	InsuranceType	Insurance type	
	InsurerNumber	Insurer number	
	OrganizationCode	Insurer code	

Scheme	Field	Field description	Comments
GreenCard_US_TYPE1	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	Number	Document number	
	Category	Category of residency	
	DateOfBirth	Document holder's birth date	
	Sex	Document holder's sex	
	DateOfExpiry	Document expiry date	
	ResidentSince	The residency start date	
IBAN	IBAN	International bank account number	
ID_AE_TYPE1	Number	Document number	
ID_AM_TYPE1	LastName	Document holder's last name	
	LastName_EX	Document holder's last name in English	
	FirstName	Document holder's first name	
	FirstName_EX	Document holder's first name in English	

Scheme	Field	Field description	Comments
	MiddleName	Document holder's patronymic name	
	Sex	Document holder's sex	
	DateOfBirth	Document holder's date of birth	
	DateOfExpiry	Document expiry date	
	Number	Document number	
ID_AT	Number	Document number	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	Sex	Document holder's sex	
	DateOfBirth	Document holder's date of birth	
ID_BE	Number	Document number	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	Sex	Document holder's sex	

Scheme	Field	Field description	Comments
	Nationality	Nationality of the document holder	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
ID_BG_TYPE1 ID_BG_TYPE2	Number	Document number	
	PersonalCode	Document holder's personal code	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	MiddleName	Document holder's patronymic name	
	Sex	Document holder's sex	
	DateOfBirth	Document holder's date of birth	
	Address	Document holder's address	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
	City	The city where the document was issued	From the back side; in the ID_BG_TYPE2 scheme only

Scheme	Field	Field description	Comments
	RegionOfResidence	Document holder's region of residence	From the back side
	Municipality	Municipal district	From the back side; in the ID_BG_TYPE2 scheme only
	MRZ	Full contents of the machine-readable zone	
	Number_MRZ	Document number from MRZ	
	PersonalCode_MRZ	Document holder's personal code from MRZ	
	LastName_MRZ	Document holder's last name from MRZ	
	FirstName_MRZ	Document holder's first name from MRZ	
	Sex_MRZ	Document holder's sex from MRZ	
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted document holder's date of birth from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	

Scheme	Field	Field description	Comments
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
ID_BH_TYPE1	Number	Document number	
	FullName	Document holder's full name	
	DateOfExpiry	Document expiry date	
ID_CH_TYPE1	Number	Document number	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	DateOfBirth	Document holder's date of birth	
ID_CL_TYPE1	LastName	Document holder's last name	
	LastName_LINE2	Second line of the Document holder's last name	
	FirstName	Document holder's first name	
	Nationality	Document holder's nationality	
	Sex	Document holder's sex	

Scheme	Field	Field description	Comments
	DateOfBirth	Document holder's date of birth	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
	Number	Document number	
ID_CZ	Number	Document number	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	Sex	Document holder's sex	
	DateOfBirth	Document holder's date of birth	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
ID_DE_TYPE1 ID_DE_TYPE2	Number	Document number	
	LastName	Document holder's last name	
	LastName_LINE2	Second line of the Document holder's last name	

Scheme	Field	Field description	Comments
	FirstName	Document holder's first name	
	DateOfBirth	Document holder's date of birth	
	DateOfBirthNationality	Document holder's date of birth and nationality	In the ID_DE_TYPE1 scheme only
	DateOfBirthPlaceOfBirth	Document holder's date and place of birth	In the ID_DE_TYPE2 scheme only
	PlaceOfBirth	Document holder's place of birth	
	Address_LINE1	Document holder's address	
	Address_LINE2	Document holder's address, continued	
	Address_LINE3	Document holder's address, continued	
	Nationality	Nationality of the document holder	In the ID_DE_TYPE2 scheme only
	Height	Document holder's height	In the ID_DE_TYPE1 scheme only
	EyeColor	Document holder's eye color	In the ID_DE_TYPE1 scheme only
	IssuedBy	The authority that issued the license	In the ID_DE_TYPE1 scheme only

Scheme	Field	Field description	Comments
	DateOfIssue	Document issue date	In the ID_DE_TYPE1 scheme only
	DateOfExpiry	Document expiry date	
	RFID	RFID number	In the ID_DE_TYPE1 scheme only
	MRZ	Full contents of the machine-readable zone	
	Number_MRZ	Document number from MRZ	
	LastName_MRZ	Document holder's last name from MRZ	
	FirstName_MRZ	Document holder's first name from MRZ	
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
ID_EE	Number	Document number	
	PersonalCode	Document holder's personal code	

Scheme	Field	Field description	Comments
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	FirstName_EX	Document holder's first name, continued	
	Sex	Document holder's sex	
	Nationality	Nationality of the document holder	
	DateOfBirth	Document holder's date of birth	
	DateOfExpiry	Document expiry date	
ID_ES_TYPE1 ID_ES_TYPE2	Number	Document number	
	IDESP	Identity card serial number	
	LastName	Document holder's last name	
	LastName_LINE2	Second line of the Document holder's last name	
	FirstName	Document holder's first name	
	Sex	Document holder's sex	

Scheme	Field	Field description	Comments
	Nationality	Nationality of the document holder	
	DateOfBirth	Document holder's date of birth	
	DateOfExpiry	Document expiry date	
ID_FI_TYPE1 ID_FI_TYPE2	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
	DateOfBirth	Document holder's date of birth	
	Number	Document number	
	Sex	Document holder's sex	In the ID_FI_TYPE2 scheme only
ID_FR_TYPE1	Number	Document number	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	DateOfBirth	Document holder's date of birth	

Scheme	Field	Field description	Comments
	Sex	Document holder's sex	
	MRZ	Full contents of the machine-readable zone	
	FirstName_MRZ	Document holder's first name from MRZ	
	LastName_MRZ	Document holder's last name from MRZ	
	Number_MRZ	Document number from MRZ	
	Sex_MRZ	Document holder's sex from MRZ	
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	IssuingStateCode	Code of the country that issued the document	
	DocumentType_MRZ	Document type from MRZ	
ID_GE_TYPE1	FirstName	Document holder's first name	
	FirstName_EX	Document holder's first name in English	

Scheme	Field	Field description	Comments
	LastName	Document holder's last name	
	LastName_EX	Document holder's last name in English	
	Number	Document number	
	PersonalCode	Document holder's personal code	
	DateOfBirth	Document holder's birth date	
	DateOfExpiry	Document expiry date	
ID_HK_TYPE1	Number	Document number	
	Code	Document code	
	FullName	Document holder's full name	
	DateOfBirth	Document holder's birth date	
	DateOfIssue	Document issue date	
ID_HR_TYPE1 ID_HR_TYPE2	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	DateOfBirth	Document holder's birth date	

Scheme	Field	Field description	Comments
	DateOfExpiry	Document expiry date	
	Number	Document number	
	Sex	Document holder's sex	
	Nationality	Nationality of the document holder	
ID_HU_TYPE1 ID_HU_TYPE2	FullName	Document holder's full name	
	Number	Document number	
	DateOfBirth	Document holder's birth date	In ID_HU_TYPE2 scheme only
	DateOfExpiry	Document expiry date	In ID_HU_TYPE2 scheme only
	Sex	Document holder's sex	In ID_HU_TYPE2 scheme only
	CardAccessNumber	Card access number	In ID_HU_TYPE2 scheme only
ID_KG	Number	Document number	
	PersonalCode	Document holder's personal code	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	

Scheme	Field	Field description	Comments
	MiddleName	Document holder's patronymic name	
	Sex	Document holder's sex	
	Nationality	Nationality of the document holder	
	DateOfBirth	Document holder's date of birth	
	PlaceOfBirth	Document holder's place of birth	
	MaritalStatus	Document holder's marital status	
	Address	Document holder's address	
	Address_LINE2	Second line of the Document holder's address	
	IssuedBy	The authority that issued the document	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
	MRZ	Full contents of the machine-readable zone	
	Number_MRZ	Document number from MRZ	

Scheme	Field	Field description	Comments
	PersonalCode_MRZ	Document holder's personal code from MRZ	
	LastName_MRZ	Document holder's last name from MRZ	
	FirstName_MRZ	Document holder's first name from MRZ	
	Sex_MRZ	Document holder's sex from MRZ	
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
ID_KW_TYPE1	Number	Document number	
	SerialNumber	Serial number	
	FullName	Document holder's full name	
	FullName_LINE2	Second line of the Document holder's full name	
	DateOfExpiry	Document expiry date	

Scheme	Field	Field description	Comments
	MRZ	Full contents of the machine-readable zone	
ID_KZ_TYPE1 ID_KZ_TYPE2	Number	Document number	
	PIN	Personal PIN (VIZ)	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	MiddleName	Document holder's patronymic name	
	DateOfBirth	Document holder's date of birth	
	MRZ	Full contents of the machine-readable zone	
	Number_MRZ	Document number from MRZ	
	PIN_MRZ	Personal PIN (VIZ) from MRZ	In the ID_KZ_TYPE2 scheme only
	LastName_MRZ	Document holder's last name from MRZ	
	FirstName_MRZ	Document holder's first name from MRZ	
	Sex_MRZ	Document holder's sex from MRZ	

Scheme	Field	Field description	Comments
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
ID_LT_TYPE1	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	DateOfBirth	Document holder's birth date	
	PersonalCode	Document holder's personal number	
	Number	Document number	
	DateOfExpiry	Document expiry date	
	Sex	Document holder's sex	
ID_LV	Number	Document number	
	PersonalCode	Document holder's personal code	

Scheme	Field	Field description	Comments
	LastName	Document holder's primary last name	
	LastName_EX	Document holder's secondary last name	
	FirstName	Document holder's first name	
	Nationality	Nationality of the document holder	
	DateOfBirth	Document holder's date of birth	
	DateOfExpiry	Document expiry date	
ID_MY	Number	Document number	
	FullName	Document holder's full name	
	Sex	Document holder's sex	
	Nationality	Nationality of the document holder	
	Address_LINE1	Document holder's address	
	Address_LINE2	Document holder's address, continued	
	Address_LINE3	Document holder's address, continued	

Scheme	Field	Field description	Comments
	Address_LINE4	Document holder's address, continued	
	Address_LINE5	Document holder's address, continued	
ID_NG_TYPE1	FirstName	Document holder's first name	
	LastName	Document holder's last name	
	Sex	Document holder's sex	
	DateOfBirth	Document holder's birth date	
	Height	Document holder's height	
ID_NO_TYPE1	FirstName	Document holder's first name	
	LastName	Document holder's last name	
	Sex	Document holder's sex	
	Number	Document number	
	DateOfBirth	Document holder's birth date	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	

Scheme	Field	Field description	Comments
ID_PL_TYPE1 ID_PL_TYPE2	Number	Document number	In the ID_PL_TYPE1 scheme only
	PersonalNumber	PESEL number from the back side	In the ID_PL_TYPE1 scheme only
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	FamilyName	Document holder's family name (last name at birth)	
	ParentsFirstNames	First names of document holder's parents	
	Sex	Document holder's sex	
	DateOfBirth	Document holder's date of birth	
	DateOfExpiry	Document expiry date	In the ID_PL_TYPE1 scheme only
	MRZ	Full contents of the machine-readable zone	
	Number_MRZ	Document number from MRZ	
	PersonalNumber_MRZ	PESEL number from MRZ	In the ID_PL_TYPE2 scheme only

Scheme	Field	Field description	Comments
	LastName_MRZ	Document holder's last name from MRZ	
	FirstName_MRZ	Document holder's first name from MRZ	
	Sex_MRZ	Document holder's sex from MRZ	
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
ID_PT_TYPE1	Number	Document number	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	Sex	Document holder's sex	
	DateOfBirth	Document holder's date of birth	
	DateOfExpiry	Document expiry date	

Scheme	Field	Field description	Comments
ID_RO	CNP	CNP Number	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	MRZ	Full contents of the machine-readable zone	
	Number_MRZ	Document number from MRZ	
	LastName_MRZ	Document holder's last name from MRZ	
	FirstName_MRZ	Document holder's first name from MRZ	
	Sex_MRZ	Document holder's sex from MRZ	
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	

Scheme	Field	Field description	Comments
ID_RS_TYPE1	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	DateOfBirth	Document holder's birth date	
	Sex	Document holder's sex	
	Number	Document number	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
ID_SG	Number	Document number	
	FullName	Document holder's full name	
	FullName_EX	Document holder's full name, continued	
	FullName_EX2	Document holder's full name, continued	
	Sex	Document holder's sex	
	DateOfBirth	Document holder's date of birth	
	CountryOfBirth	Document holder's country of birth	

Scheme	Field	Field description	Comments
	Nationality	Nationality of the document holder	
ID_SV_TYPE1	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	Number	Document number	
	Sex	Document holder's sex	
	DateOfBirth	Document holder's birth date	
ID_TR_TYPE1	Number	Document number	
	PersonalCode	Document holder's personal code	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	Sex	Document holder's sex	
	DateOfBirth	Document holder's date of birth	
	DateOfExpiry	Document expiry date	
ID_ZA_TYPE1	LastName	Document holder's last name	

Scheme	Field	Field description	Comments
	FirstName	Document holder's first name	
	DateOfBirth	Document holder's birth date	
	Sex	Document holder's sex	
	Number	Document number	
	Nationality	Document holder's nationality	
InternationalPassport_AT	Number	Document number	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	DateOfBirth	Document holder's date of birth	
	PlaceOfBirth	Document holder's place of birth	
	Height	Document holder's height	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
	MRZ	Full contents of the machine-readable zone	

Scheme	Field	Field description	Comments
	Number_MRZ	Document number from MRZ	
	DVRNumber	DVR number	
	LastName_MRZ	Document holder's last name from MRZ	
	FirstName_MRZ	Document holder's first name from MRZ	
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
InternationalPassport_CN_TYPE1 InternationalPassport_CN_TYPE2	Number	Document number	
	LastName	Document holder's last name	In the InternationalPassport_CN_TYPE1 scheme only
	FirstName	Document holder's first name	In the InternationalPassport_CN_TYPE1 scheme only
	LastNameFirstName	Document holder's full name	In the InternationalPassport_CN_TYPE2 scheme only

Scheme	Field	Field description	Comments
	Sex	Document holder's sex	
	DateOfBirth	Document holder's date of birth	
	PlaceOfBirth	Document holder's place of birth	
	Nationality	Nationality of the document holder	In the InternationalPassport_CN_TYPE2 scheme only
	PlaceOfIssue	Region where the document was issued	
	IssuingStateCode	Code of the authority that issued the document	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
	MRZ	Full contents of the machine-readable zone	
	Number_MRZ	Document number from MRZ	
	FirstName_MRZ	Document holder's first name from MRZ	
	LastName_MRZ	Document holder's last name from MRZ	
	Sex_MRZ	Document holder's sex from MRZ	

Scheme	Field	Field description	Comments
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	Nationality_MRZ	Document holder's nationality from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
	Optional_MRZ_LINE2	Optional second line of MRZ	
InternationalPassport_DE_TYPE1 InternationalPassport_DE_TYPE2	Number	Document number	
	LastName	Document holder's last name	
	LastName_LINE2	Second line of the Document holder's last name	
	FirstName	Document holder's first name	
	FirstName_LINE2	Second line of the Document holder's first name	In the InternationalPassport_DE_TYPE2 scheme only
	Sex	Document holder's sex	

Scheme	Field	Field description	Comments
	DateOfBirth	Document holder's date of birth	
	PlaceOfBirth	Document holder's place of birth	
	Nationality	Nationality of the document holder	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
	IssuedBy	The authority that issued the document	
	MRZ	Full contents of the machine-readable zone	
	Number_MRZ	Document number from MRZ	
	LastName_MRZ	Document holder's last name from MRZ	
	FirstName_MRZ	Document holder's first name from MRZ	
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	

Scheme	Field	Field description	Comments
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
InternationalPassport_GE_TYPE1 InternationalPassport_GE_TYPE2	PersonalCode	Document holder's personal code	In the InternationalPassport_GE_TYPE1 scheme only
	PersonalCode_MRZ	Document holder's personal code from MRZ	In the InternationalPassport_GE_TYPE2 scheme only
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
	Sex_MRZ	Document holder's sex from MRZ	
	FirstName_MRZ	Document holder's first name from MRZ	
	LastName_MRZ	Document holder's last name from MRZ	
	Nationality_MRZ	Document holder's nationality from MRZ	
	Number_MRZ	Document number from MRZ	

Scheme	Field	Field description	Comments
	MRZ	Full contents of the machine-readable zone	
InternationalPassport_IN	Number	Document number	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	Sex	Document holder's sex	
	DateOfBirth	Document holder's date of birth	
	PlaceOfBirth	Document holder's place of birth	
	Nationality	Nationality of the document holder	
	PlaceOfIssue	The region where the document was issued	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
	MRZ	Full contents of the machine-readable zone	
	Number_MRZ	Document number from MRZ	

Scheme	Field	Field description	Comments
	LastName_MRZ	Document holder's last name from MRZ	
	FirstName_MRZ	Document holder's first name from MRZ	
	Sex_MRZ	Document holder's sex from MRZ	
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
InternationalPassport_IT	Number	Document number	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	Sex	Document holder's sex	
	DateOfBirth	Document holder's date of birth	
	PlaceOfBirth	Document holder's place of birth	

Scheme	Field	Field description	Comments
	Nationality	Nationality of the document holder	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
	MRZ	Full contents of the machine-readable zone	
	Number_MRZ	Document number from MRZ	
	FirstName_MRZ	Document holder's first name from MRZ	
	LastName_MRZ	Document holder's last name from MRZ	
	Sex_MRZ	Document holder's sex from MRZ	
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
InternationalPassport_JP	Number	Document number	

Scheme	Field	Field description	Comments
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	Sex	Document holder's sex	
	DateOfBirth	Document holder's date of birth	
	PlaceOfBirth	Document holder's place of birth	
	Nationality	Nationality of the document holder	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
	MRZ	Full contents of the machine-readable zone	
	Number_MRZ	Document number from MRZ	
	FirstName_MRZ	Document holder's first name from MRZ	
	LastName_MRZ	Document holder's last name from MRZ	
	Sex_MRZ	Document holder's sex from MRZ	

Scheme	Field	Field description	Comments
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
InternationalPassport_PH_TYPE1 InternationalPassport_PH_TYPE2	Number	Document number	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	MiddleName	Document holder's patronymic name	
	Sex	Document holder's sex	
	DateOfBirth	Document holder's date of birth	
	PlaceOfBirth	Document holder's place of birth	
	Nationality	Nationality of the document holder	In the InternationalPassport_PH_TYPE2 scheme only
	DateOfIssue	Document issue date	

Scheme	Field	Field description	Comments
	DateOfExpiry	Document expiry date	
	MRZ	Full contents of the machine-readable zone	
	Number_MRZ	Document number from MRZ	
	LastName_MRZ	Document holder's last name from MRZ	
	FirstName_MRZ	Document holder's first name from MRZ	
	Sex_MRZ	Document holder's sex from MRZ	
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	Nationality_MRZ	Nationality of the document holder from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
InternationalPassport_RU	Number	Document number	

Scheme	Field	Field description	Comments
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	MiddleName	Document holder's patronymic name	
	FirstNameMiddleName	Document holder's first and patronymic names	
	Sex	Document holder's sex	
	DateOfBirth	Document holder's date of birth	
	PlaceOfBirth	Document holder's place of birth	
	IssuedBy	The authority that issued the document	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
	MRZ	Full contents of the machine-readable zone	
	Number_MRZ	Document number from MRZ	
	LastName_MRZ	Document holder's last name from MRZ	

Scheme	Field	Field description	Comments
	FirstName_MRZ	Document holder's first name from MRZ	
	Sex_MRZ	Document holder's sex from MRZ	
	Nationality_MRZ	Nationality of the document holder from MRZ	
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
InternationalPassport_SY	DateOfBirth	Document holder's date of birth	
	MRZ	Full contents of the machine-readable zone	
	Number_MRZ	Document number from MRZ	
	LastName_MRZ	Document holder's last name from MRZ	
	FirstName_MRZ	Document holder's first name from MRZ	

Scheme	Field	Field description	Comments
	Sex_MRZ	Document holder's sex from MRZ	
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
InternationalPassport_TJ_TYPE1	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	Nationality	Document holder's nationality	
	DateOfBirth	Document holder's birth date	
	Sex	Document holder's sex	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
	Number	Document number	

Scheme	Field	Field description	Comments
	MRZ	Full contents of the machine-readable zone	
	FirstName_MRZ	Document holder's first name from MRZ	
	LastName_MRZ	Document holder's last name from MRZ	
	Number_MRZ	Document number from MRZ	
	Nationality_MRZ	Nationality of the document holder from MRZ	
	Sex_MRZ	Document holder's sex from MRZ	
	DateOfBirth_MRZ	Document holder's birth date from MRZ	
	DateOfBirth_FORMATTED	Formatted birth date from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
InternationalPassport_UK_TYPE1 InternationalPassport_UK_TYPE2	Number	Document number	
	LastName	Document holder's last name	

Scheme	Field	Field description	Comments
	FirstName	Document holder's first name	
	Sex	Document holder's sex	
	Nationality	Nationality of the document holder	
	DateOfBirth	Document holder's date of birth	
	PlaceOfBirth	Document holder's place of birth	
	IssuedBy	The authority that issued the document	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
	MRZ	Full contents of the machine-readable zone	
	Number_MRZ	Document number from MRZ	
	LastName_MRZ	Document holder's last name from MRZ	
	FirstName_MRZ	Document holder's first name from MRZ	
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	

Scheme	Field	Field description	Comments
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
InternationalPassport_US_TYPE1 InternationalPassport_US_TYPE2	Number	Document number	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	Sex	Document holder's sex	
	DateOfBirth	Document holder's date of birth	
	PlaceOfBirth	Document holder's place of birth	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
	MRZ	Full contents of the machine-readable zone	
	Number_MRZ	Document number from MRZ	

Scheme	Field	Field description	Comments
	LastName_MRZ	Document holder's last name from MRZ	
	FirstName_MRZ	Document holder's first name from MRZ	
	Sex_MRZ	Document holder's sex from MRZ	
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	Nationality_MRZ	Nationality of the document holder from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
	Optional_MRZ_LINE2	Optional second line of MRZ	
InternationalPassport_UZ_TYPE1	Number	Document number	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	

Scheme	Field	Field description	Comments
	DateOfBirth	Document holder's birth date	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
	Sex	Document holder's sex	
	MRZ	Full contents of the machine-readable zone	
	FirstName_MRZ	Document holder's first name from MRZ	
	LastName_MRZ	Document holder's last name from MRZ	
	Number_MRZ	Document number from MRZ	
	Nationality_MRZ	Document holder's nationality from MRZ	
	Sex_MRZ	Document holder's sex from MRZ	
	DateOfBirth_MRZ	Document holder's birth date from MRZ	
	DateOfBirth_FORMATTED	Formatted birth date from MRZ	
	DateOfExpiry_MRZ	Document expiry date from MRZ	

Scheme	Field	Field description	Comments
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	
MRZ	Number	Document number	
	DocumentType	Document type	
	DocumentSubtype	Document subtype	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	DateOfBirth	Document holder's date of birth	
	Sex	Document holder's sex	
	Nationality	Nationality of the document holder	
	PersonalNumber	Document holder's personal number	
	IssuingCountry	The country where the document was issued	
	DateOfExpiry	Document expiry date	
	OptionalData	Optional MRZ data	
MRZ_CH MRZ_FR MRZ_MRP MRZ_MRV_A	MRZ	Full contents of the MRZ	

Scheme	Field	Field description	Comments
MRZ_MRV_B MRZ_MRV_RU MRZ_RU MRZ_TD1 MRZ_TD2	MRZ_LINE1	The first line of MRZ	
	MRZ_LINE2	The second line of MRZ	
	MRZ_LINE3	The third line of MRZ	In the MRZ_TD1 scheme only
	Number	Document number	
	Number_FORMATTED	Formatted document number	
	DocumentType	Document type	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	DateOfBirth	Document holder's date of birth	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	Sex	Document holder's sex	Except the MRZ_CH scheme
	Nationality	Nationality of the document holder	Except the MRZ_CH scheme
	IssuingStateCode	Code of the country that issued the document	

Scheme	Field	Field description	Comments
	DepartmentCode	Code of the authority that issued the document	In the MRZ_RU scheme only
	DateOfIssue	Document issue date	In the MRZ_RU scheme only
	DateOfIssue_FORMATTED	Formatted document issue date	In the MRZ_RU scheme only
	DateOfExpiry	Document expiry date	Except the MRZ_CH, MRZ_FR, MRZ_RU schemes
	DateOfExpiry_FORMATTED	Formatted document expiry date	Except the MRZ_CH, MRZ_FR, MRZ_RU schemes
	OptionalData_LINE1	Optional MRZ line	
	OptionalData_LINE2	Optional MRZ line	
MRZ_BG	MRZ	Full contents of the MRZ	
	MRZ_LINE1	The first line of MRZ	
	MRZ_LINE2	The second line of MRZ	
	Number	Document number	
	Number_FORMATTED	Formatted document number	
	DocumentType	Document type	

Scheme	Field	Field description	Comments
	IssuingStateCode	Code of the country that issued the document	
	VehicleNumber	Vehicle license number	
	Owner	Vehicle owner's full name for personal vehicles, or a company name if the vehicle is owned by a legal entity	
	PersonalCode	Owner identification number	
	VIN	Vehicle identification number (VIN)	
Passport_BY Passport_BY_PAGE31	Number	Document number	In the Passport_BY scheme only
	ID	Document holder's personal identifier	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	Sex	Document holder's sex	In the Passport_BY scheme only
	DateOfBirth	Document holder's date of birth	
	PlaceOfBirth	Document holder's place of birth	

Scheme	Field	Field description	Comments
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	
	MRZ	Full contents of the machine-readable zone	In the Passport_BY scheme only
	Number_MRZ	Document number from MRZ	In the Passport_BY scheme only
	LastName_MRZ	Document holder's last name from MRZ	In the Passport_BY scheme only
	FirstName_MRZ	Document holder's first name from MRZ	In the Passport_BY scheme only
	Sex_MRZ	Document holder's sex from MRZ	In the Passport_BY scheme only
	Nationality_MRZ	Nationality of the document holder from MRZ	In the Passport_BY scheme only
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	In the Passport_BY scheme only
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	In the Passport_BY scheme only
	DateOfExpiry_MRZ	Document expiry date from MRZ	In the Passport_BY scheme only
	DateOfExpiry_FORMATTED	Formatted document expiry date from MRZ	In the Passport_BY scheme only

Scheme	Field	Field description	Comments
Passport_RU Passport_RU_Top Passport_RU_Bottom	Series	Document series	In the Passport_RU scheme only
	Number	Document number	In Passport_RU_Top and Passport_RU_Bottom, this field contains both series and number
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	MiddleName	Document holder's patronymic name	
	Sex	Document holder's sex	
	DateOfBirth	Document holder's date of birth	
	PlaceOfBirth	Document holder's place of birth	
	IssuedBy	The authority that issued the document	
	DepartmentCode	The code of the authority that issued the document	
	DateOfIssue	Document issue date	
	MRZ	Full contents of the machine-readable zone	In the Passport_RU scheme only

Scheme	Field	Field description	Comments
	MRZ_LINE1	The first line of MRZ	In the Passport_RU scheme only
	MRZ_LINE2	The second line of MRZ	In the Passport_RU scheme only
ResidencePermit_ES_TYPE1 ResidencePermit_ES_TYPE2	Number	Document number	
	NIENumber	NIE number	
	FullName	Document holder's full name	
	FullName_LINE2	Second line of the document holder's full name	
	DateOfBirth	Document holder's date of birth	
	Nationality	Nationality of the document holder	
	Address	Document holder's address	
	Address_LINE2	Second line of the document holder's address	
	DateOfRegistration	Date of resident registration	
	ProvinceOfIssue	Province of issue	
	PlaceOfIssue	Place of issue	

Scheme	Field	Field description	Comments
SocialSecurityNumber_RU_TYPE1 SocialSecurityNumber_RU_TYPE2	Number	Document number	
	LastName	Document holder's last name	
	FirstName	Document holder's first name	
	MiddleName	Document holder's patronymic name	
	DateOfBirth	Document holder's date of birth	
VehicleRegistration_RU_TYPE1 VehicleRegistration_RU_TYPE2	Number	Document number	
	LicensePlate	License plate number	
	VIN	VIN number	
Visa_RU_TYPE1	DocumentType	Type of visa	
	Number	Visa number	
	PassportNumber	Document holder's passport number	
	FullName	Document holder's full name	
	FullName_EX	Document holder's full name in English	
	Sex	Document holder's sex	

Scheme	Field	Field description	Comments
	DateOfBirth	Document holder's date of birth	
	Nationality	Nationality of the document holder	
	InvitationNumber	The number of invitation	
	VisaId	Visa ID	
	FromTo	Duration	
	DateOfIssue	Visa issue date	
	EntryFromDate	Entry from date	
	StayUntilDate	Stay until date	
	Duration	Duration	
	MRZ	Full contents of the machine-readable zone	
	PassportNumber_MRZ	Document holder's passport number from MRZ	
	LastName_MRZ	Document holder's last name from MRZ	
	LastName_FORMATTED	Formatted Document holder's last name from MRZ	

Scheme	Field	Field description	Comments
	FirstName_MRZ	Document holder's first name from MRZ	
	FirstName_FORMATTED	Formatted Document holder's first name from MRZ	
	Sex_MRZ	Document holder's sex from MRZ	
	DateOfBirth_MRZ	Document holder's date of birth from MRZ	
	DateOfBirth_FORMATTED	Formatted date of birth from MRZ	
	Nationality_MRZ	Nationality of the document holder from MRZ	
	DocumentType_MRZ	Document type from MRZ	
	OptionalData_MRZ_LINE1	Optional MRZ line	
	OptionalData_MRZ_LINE2	Optional MRZ line	
WorkPermit_SG_TYPE1	DocumentType	Type of permit	
	Number	Primary document number	
	Number_EX	Secondary document number	

Scheme	Field	Field description	Comments
	FullName	Document holder's full name	
	Sector	Occupation sector	
	Employer	Employer company name	
	Occupation	Occupational title	
	DateOfApplication	Work application date	
	DateOfIssue	Document issue date	
	DateOfExpiry	Document expiry date	

Regular Expressions

This section describes the regular expression syntax supported by the ABBYY Real-Time Recognition 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;

Pattern	Syntax	Examples and comments
		to match these literally, you have to escape them with a 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	

Pattern	Syntax	Examples and comments
Non-printable characters	\n — line feed LF \r — carriage return CR \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".

Pattern	Syntax	Examples and comments
Grouping	()	*(word)*+ matches "word", "wordword" and so on

Unsupported syntax

The following regular expression syntax features are not yet supported in ABBYY Real-Time Recognition 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

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 HOLDER 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.

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.

Contact ABBYY

In this section you can find the contacts of ABBYY sales offices and technical support.

How to Buy

You can order ABBYY Real-Time Recognition SDK or other ABBYY products by contacting our offices at the following addresses:

- ABBYY Russia: engine@abbyy.com
- ABBYY Northern American Headquarters: sales@abbyyusa.com
- ABBYY European Headquarters: engine_eu@abbyy.com
- ABBYY Eastern European Headquarters: sdk@abbyy.ua
- ABBYY 3A (Asia, Africa and South America): sales_3A@abbyy.com

Technical Support

If you have any questions regarding the use of ABBYY Real-Time Recognition SDK, first of all consult this Developer's Guide. Useful information can also be found in the technical support section of the ABBYY website.

If you cannot find the answer to your question, please contact the ABBYY office serving your region by e-mail. Please provide the following information when contacting technical support:

- your first and last name
- the name of your organization
- your phone number (or fax, or e-mail)
- the serial number of your license
- the ABBYY Real-Time Recognition SDK build number
- a description of the problem
- a project that demonstrates the problem (with the necessary data files). This may be a slightly modified ABBYY Real-Time Recognition SDK sample. We recommend that you compress the files using any popular archiving program (WinZIP, WinRAR, etc.)
- the name of your development tool
- the type of your device and processor
- the version of your operating system

You can also provide any additional information you consider important.

Support contacts

North/Central Americas	Customers from USA, Canada, Japan, Mexico or other Central American countries, please contact ABBYY North American Headquarters at dev_support@abbyyusa.com
Western Europe	Customers from Austria, Benelux, Denmark, France, Germany, Italy, Ireland, Norway, Portugal, Spain, Sweden, Switzerland, the United Kingdom or other Western European countries, please contact ABBYY European Headquarters at TechSupport_eu@abbyy.com

Eastern Europe and the Mediterranean	Customers from Ukraine, Moldova, Turkey, Israel or Eastern European countries, please contact ABBYY Eastern European Headquarters at engine_support@abbyy.ua
All other regions	Customers from the countries not mentioned above, please contact ABBYY International Headquarters at SDK_Support@abbyy.com