ABBYY®

REAL-TIME RECOGNITION SDK RELEASE 7

RELEASE NOTES

11/7/2017

1 CONTENTS

2 About the Product

ABBYY Real-Time Recognition SDK (RTR SDK) is a software development kit which provides a capability to recognize text directly on the smartphone camera preview screen.

3 About This Release

3.1 Purpose

Purposes of the release:

- Support new IDs
- Protection update

3.2 Build Information

Part Number: 1305/25
Build Number: 1.0.9.52

3.3 Additional Information

ABBYY Technology Portal: https://abbyy.technology/en:products:rtrsdk:start
Forum: http://forum.ocrsdk.com/tags/rtr/
Knowledgebase: http://knowledgebase.ocrsdk.com/?searchText=&ProductId=2008

4 History of releases

Releases of the current product version in descending order (only releases of the same Part# type).

Part #	*Version*	*Release date*	*Build #*
1305/12	Release 1	2016-11-14	1.0.1.178
1305/14	Release 1 patch	2016-12-01	1.0.1.180
1305/15	Release 1 patch 2	2016-12-14	1.0.1.181

1305/16	Release 2	2017-03-13	1.0.3.190
1305/17	Release 3	2017-05-02	1.0.4.141
1305/18	Release 4	2017-06-23	1.0.5.61
1305/19	Release 5	2017-08-04	1.0.6.58
1305/20	Release with EDC	2017-09-14	1.0.7.56
1305/21	Release with EDC Update	2017-09-21	1.0.7.59
1305/22	Release 6	2017-10-20	1.0.8.21
1305/23	Release 6 Patch	2017-11-07	1.0.8.26

5 Product Specification

5.1 Hardware and software requirements

5.1.1 Supported Operating Systems

- Android 4.4 and later for ARMv7 processors and arm64-v8a processors
- iOS 8.x and later

5.1.2 Processor requirements

- A multi-core processor (augmented reality scenarios will require **4** cores or more)
- Advanced SIMD (NEON)

5.1.3 Memory requirements (RAM)

Library operation in text capture scenario takes up:

- for text in alphabetic languages - 40 MB RAM
- for text in CJK languages - 70 MB RAM

Library operation in data capture scenario takes up to 105 MB RAM for iOS and 170 MB RAM for Android

5.1.4 Memory requirements (ROM)

- Library — 3 to 19 MB RAM
- Resources for OCR — 1.6 to 18.4 MB ROM, depending on the recognition languages
- Optional resources for better OCR quality — 14.4 MB ROM, European languages only
- Resources for Data Capture scenarios:
    - bank card recognition – 1 MB ROM
    - MRZ recognition – 0.5 MB ROM
    - ID recognition – 0.5 to 16 MB ROM, depending on the documents to be recognized

5.1.5 Camera requirements

- Autofocus lens
- HD preview: generally recommended frame size is **720*1080**, but it can vary depending on the scenario and processing speed

5.2 Supported Languages, Types, and Formats

5.2.1 Supported OCR Languages

62 recognition languages are available for OCR:

- 23 languages with dictionary support:
  Czech, Bulgarian, Danish, Dutch (Belgian), Dutch (Netherlands), English, Estonian, Finnish, French, German (old and new spelling), Greek, Indonesian, Italian, Norwegian (Bokmål), Norwegian (Nynorsk), Polish, Portuguese, Portuguese (Brazilian), Russian, Spanish, Swedish, Turkish, and Ukrainian
- 35 additional languages with Latin, Cyrillic, or Greek characters:
  Afrikaans, Albanian, Basque, Breton, Belarusian, Catalan, Chechen, Crimean Tatar, Croatian, Fijian, Hawaiian, Hungarian, Icelandic, Irish, Kabardian, Latin, Latvian, Lithuanian, Macedonian, Malay, Maori, Mixed, Moldavian, Mongol, Ossetic, Provencal, Rhaeto-Romanic, Romanian, Samoan, Serbian, Slovak, Slovenian, Swahili, Tagalog, Tatar, and Welsh
- 4 Asian complex script languages:
  Chinese (Simplified and Traditional), Japanese and Korean

5.2.2 Supported Text Types

- Common typographic type of text is supported (Normal)

6 Key Features

6.1 Real-Time OCR

ABBYY Real-Time Recognition SDK enables mobile developers to create mobile apps with capability to recognize text **on video frames** from smartphone's camera.

In comparison with ordinary OCR technologies which work with photos RTR SDK doesn't require snapping a photo, but offers the possibility to automatically "snap text": recognizing video frames the engine automatically estimates whether the recognition result is stable enough and can be returned to user. Also recognition in real-time provides some additional advantages:

- provides better UX and helps to save user time for taking and retaking a photo;
- proposes additional advantages for some scenarios which require confidentiality of captured data (like ID capture), because RTR doesn't require to save image in the memory storage

6.2 Recognition of Text in Real-World Scenes

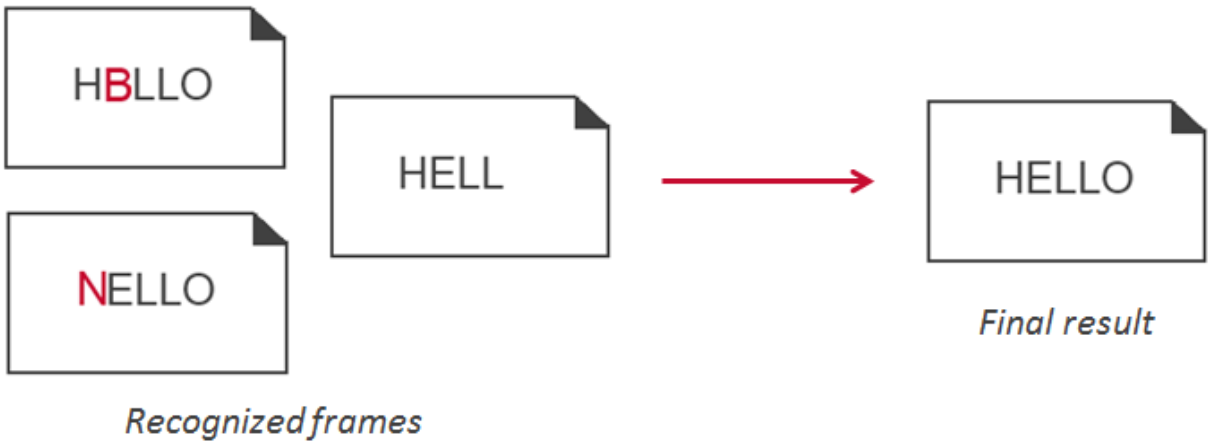
ABBYY Real-Time Recognition SDK enables localization and then recognition of text in real-world scenes, like:

Recognition methods can search for text of any color on any background.

In comparison with ordinary mobile OCR technologies, which use ordinary document analysis, ABBYY Real-Time Recognition SDK demonstrates better performance and recognition accuracy **on natural scenes** (like signs, etc.) – the recognition results contain less noise. Thus RTR SDK better suits for computer vision applications like searching images by their textual content, assisting visually impaired, reading labels and street signs in map applications and applications for tourists.

6.3 Merging the Recognition Results

Video frames and even photos often contain glare, motion blur and other defects which lead to recognition errors. To increase recognition quality and to eliminate such random recognition errors RTR SDK uses so-called Frame Merging, i.e. special statistical mechanism which aggregates, estimates and combines recognition results from several video frames.



For some contexts merging recognition results from several video frames RTR achieves even better recognition accuracy than recognizing a photo of the same context.

6.4 Translation

6.4.1 Real-Time Translation

ABBYY Real-Time Recognition SDK enables mobile developers to create apps capable of translating text **on video frames** from smartphone's camera on device without any Internet connection.

RTR SDK provides build-in translation dictionaries for word-by-word and phrase translation. The dictionaries contain words and some common phrases for the main European, Chinese and Japanese languages (please, see the full list of supported translation directions). Translation dictionaries optimized for work on mobile devices. Translation process requires few resources and takes a fraction of a second to complete.

6.4.2 Supported translation directions

ABBYY Real-Time Recognition SDK 1.0 Release 1 provides dictionaries for the following translation directions:

- English -> Russian; German; French; Spanish; Portuguese (Brazilian); Polish; Chinese (simplified); Japanese; Indonesian

- Russian; German; French; Spanish; Portuguese (Brazilian); Polish; Chinese (simplified); Japanese; Indonesian -> English

7 Upgrade from previous Versions and Releases

7.1.1 Compatibility of serial numbers/protection keys

Serial Numbers from the previous release (i.e. Release 5) can be used with this release.

7.1.2 Migrating code

The release has compatible Android API with the previous ones.

8 New Features and Improvements

8.1 Release 1

8.1.1 iOS Support

Support iOS platform (iOS 7.X and later) for ARMv7 and ARM64 processors.

8.1.2 Text Characteristics - Vertical CJK (available in Extended Version only)

Support recognition of vertical CJK text lines. This feature is useful for text capture and text translation apps, because vertical text is often used for guide signs, in book titles, in newspapers.

8.2 Release 1 Patch

8.2.1 Reporting Usage Statistics

RTR SDK's methods can report statistics about library usage to the ABBYY server.

This feature is controlled by license:

- reporting is always turned on for all Online licenses which are provided to customers within Online Distribution Kit

- reporting can be turned on for Extended licenses

When the reporting feature is turned on, the library tries to connect to the ABBYY server and to send the statistics before it starts running recognition. The following statistics data is collected:

- library copy identifier (aka download identifier) for Online distribution or extended license Serial Number for Offline distribution
- name and version of the OS on which the RTR SDK is running
- identifier of application which uses the RTR SDK
- manufacturer, model and identifier of the mobile device on which the RTR SDK is running
- IP address of the mobile device

This feature enables ABBYY to track the usage of Extended licenses (count the number of apps which use RTR SDK and the number of installations for each application).

8.3 Release 2

8.3.1 Text Analysis Enhancement

The FindText algorithm, which looks for text lines on the video frames, was updated.

The current method misses in average 5% characters less than the previous one. It leads to the recognition accuracy changes:

- for English (books) 3,5% increase (percentage of characters' recognition errors: 1,9% → 1,83%)
- for Korean (books) 2% increase (percentage of recognition errors: 6,51% → 6,35%)
- for other languages results vary

8.3.2 [iOS Wrapper] Changes in Processing Errors Handling

All errors are returned by onError method of the RTRRecognitionServiceDelegate. Error is returned each time when developer tries to transfer RTRRecognitionService to invalid state (for example, specify the non-existent recognition language), but the RTRRecognitionService state remains without changes (for example, the recognition language isn't changed).

The error messages were updated (more descriptive wording).

8.4 Release 3

8.4.1 Support of ARM64 Android processors

The Android library version compiled for ARM64 processors is added to the distribution kit.

8.4.2 Recognition of a Single Photo

Some scenarios require the ability to process documents, which aren't 'in place' right now, and recognition of video stream isn't applicable for them. For example, video stream processing can't be used to recognize an image of receipt, which you received by email, etc.

To support such scenarios, we extended API and added the possibility to process a single image (not video stream). The availability of this functionality is regulated by the license.

New methods support standard image formats for corresponding platforms:

- android.graphics.Bitmap – for Android
- UIImage – for iOS

New methods of Java and Objective C APIs are just wrappers of the FineRecognizeImage method of Mobile OCR Engine 4 with the following default settings:

- imageResolution = 0
- disableDeskew = false
- detectPageOrientation = true
- findAllText = false
- useOldBinarization = false
- recognitionConfidenceLevel = LEVEL 3

The developer can't change these default settings.

The recognition accuracy of new methods are the same as we have for Mobile OCR Engine 4 – the testing results will be listed in a separate document (will add the link as soon as the document will be available)

8.4.3 Working with Custom Dictionaries

Custom dictionaries are used during the recognition process to limit the number of recognition hypothesis. They help to increase recognition accuracy in case you need to capture some specific data, namely:

- a set of predefined words
- a field which can be described with regular expression
- a code which can contain only restricted set of character

To recognize with custom dictionaries developer needs to have special pattern and dictionary files, which he/she can get from the technical support. To do it developer needs to contact the support team via email (SDK_Support@abbyy.com) and provide them with the following data:

- description of the data to be recognized (depending on the scenario):
  - set of predefined words as TXT file
  - description of regular expression
  - the necessary alphabet
- photos which can be used for testing of created dictionaries

The technical support team will respond within 3 days and provide developer with necessary files and instruction how to use them. This feature isn't described in the documentation, because in the future versions we'd like to change the procedure for working with custom dictionaries (let developers generate all necessary data by themselves).

8.5 Release 4

8.5.1 Stop supporting iOS7

Starting from this release RTR SDK no longer supports iOS7[1]

The reasons:

- the percentage of the devices running this version of iOS is less than 1% of all iOS devices
- Apple itself doesn't support iOS7

8.5.2 Store resources (patterns and dictionaries) outside the application package/bundle

Applications' stores (App Store, Google Play) have a limit for app size under which the app can be downloaded using mobile data (not Wi-Fi). According to mobile developers, this has a huge impact on downloads (something in the range of 25-50%), so all developers try to maintain the size of their app under such limit.

Starting from this release, developers can store resources (patterns and dictionaries) outside the application package/bundle: we added a new Engine setting which specifies the path to a 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 (Android) or <bundle path>/Patterns (iOS)
2. in <custom search path>/patterns

[1] We didn't add to the library anything that shouldn't work for iOS7, but starting from this release QA team won't check whether the library works on this platform or not.

3. if the file is not found, an error will be returned

Important! For Android, the license file should still be placed in assets – but for iOS it can be placed to a custom folder as well as other resources

8.5.3 Technical Preview

8.5.3.1 Data Capture Interfaces

Starting from this release, we extended our API and added new interfaces, using which developers are able to implement simple Data Capture scenarios:

- extract **one field** which can be described by regular expression
- extract several regular fields one-by-one

In addition, the developer can specify the necessary algorithm for result validation via a specialized callback function

So, from the technical point of view, developer needs only to specify a regular expression, which describes the necessary field and, if necessary, the algorithm for result validation (it is useful if the necessary field has a checksum). Then when the end user points the camera at the text, containing necessary field, the Engine will extract only the field, which corresponds to the regular expression. In case the validation rule for result is also specified, the Engine will extract the text only if the result meets this rule.

Important! Whether two or more fields correspond to the specified regular expression, the Engine will extract only the first one.

The interfaces are available in both online and extended versions in the **beta mode**. It means:

- the current distribution kit doesn't include the code sample demonstrating the usage of these interfaces
- the new functionality isn't described in the Help file (the descriptions can be found only in Javadoc for Android and header files for iOS)
- the interfaces can be changed a little in next releases

The official release of Data Capture interfaces is scheduled for August 2017.

8.6 Release 5

8.6.1 Data Capture Interfaces

Starting from this release, new interfaces, using which developers are able to implement simple Data Capture scenarios, are officially available. The scenarios, which can be implemented:

4. extract **one field**, which can be described by a regular expression (for instance, some code, email, etc.)
5. extract several regular fields one-by-one (for example, total amount and date from receipt)

In addition, the developer can specify the necessary algorithm for result validation via a specialized callback function (it is useful if the necessary field has a checksum, like IBAN has).

Thus from the technical point of view, developer needs only to specify a regular expression, which describes the field to be extracted and, if necessary, the algorithm for result validation Then when the end user points the camera at the text, containing necessary field, the Engine will extract only the field, which corresponds to the regular expression. If the validation rule for result is also specified, the Engine will extract the text only if the result meets this rule.

Important! If two or more matches found for the specified regular expression, the Engine will extract only the one, which is the closest to the center of the area of interest.

The simplest example, which demonstrates how to use the new interfaces to extract some emails (.com, .ru or .ua domains):

Android

```
IDataCaptureService dataCaptureService =
engine.createDataCaptureService( "", dataCaptureCallback );
IDataCaptureProfileBuilder profileBuilder =
dataCaptureService.configureDataCaptureProfile();

profileBuilder.addScheme( "SAMPLE" ).addField( "EMAIL" ).setRegEx( "[
a-z_]+@[a-z\\x2D]+\\.(com|ru|ua)" );
profileBuilder.checkAndApply();
```

iOS

```
id<RTRDataCaptureService> service =
[engine createDataCaptureServiceWithDelegate:self profile:nil];

id<RTRDataCaptureProfileBuilder> builder = [service configureDataCaptureProfile];

[builder setRecognitionLanguages:[NSSet setWithObjects:@"English", nil]];

[[[builder addScheme:@ “ SAMPLE"] addField:@ “ EMAIL"] setRegEx:@"[a-z_]+@[a-
z\\x2D]+\\.(com|ru|ua)"];

[builder checkAndApply];
```

The interfaces are available in both online and extended versions. In comparison with the previous release (Release 4) the Help file contains the description of these new methods. Also special code samples for Android and iOS have been added to the distribution kits. The samples are located in the *'sample-datacapture'* folders.

8.6.2 IBAN CAPTURE

Starting from this release RTR SDK supports IBAN capture for Germany, France, Spain and the United Kingdom out-of-the-box.

Instead of creating complex regular expressions, developer just needs to add the following lines of code to recognize IBANs:

Android

```
IDataCaptureService dataCaptureService =
engine.createDataCaptureService( "IBAN", dataCaptureCallback );
```

iOS

```
id<RTRDataCaptureService> service = [engine createDataCaptureServiceWithDelegate:self
profile:@"IBAN"];
```

The availability of this feature is regulated by the license.

8.6.3 MRZ CAPTURE

Starting from this release RTR SDK supports MRZ capture out-of-the-box. We support recognition of 2 and 3 lines of machine-readable data written in accordance with ICAO Document 9303 (endorsed by the International Organization for Standardization and the International Electrotechnical Commission as ISO/IEC 7501-1)). Recognition of MRZ on VISAs isn't supported yet.

Instead of creating complex regular expressions, developer just needs to add the following lines of code to recognize MRZ:

Android

```
IDataCaptureService dataCaptureService =
engine.createDataCaptureService( "MRZ", dataCaptureCallback );
```

iOS

```
id<RTRDataCaptureService> service = [engine createDataCaptureServiceWithDelegate:self
profile:@"MRZ"];
```

In addition, MRZ.rom pattern file should be added to the assets.

The availability of this feature is regulated by the license

8.7 Release 5 with Extended Data Capture

8.7.1 ID Capture and Bank Card Recognition

Starting from this release RTR SDK supports ID capture and bank card recognition out-of-the-box.

Instead of creating complex regular expressions, developer just needs to add the following lines of code to recognize data from these documents:

Android

```
IDataCaptureService dataCaptureService = engine.createDataCaptureService(
"_profile_name_", dataCaptureCallback );
```

iOS

```
id<RTRDataCaptureService> service = [engine createDataCaptureServiceWithDelegate:self
profile:@"_profile_name_"];
```

For detailed information about supported ID documents and profile names, please, see the Help file.

8.7.2 Support of landscape mode in code samples

Starting from this release we added landscape mode support to TextCapture and DataCapture code samples, which are included into the distribution kit and published on github.

8.8 Release 6

8.8.1 Support of new type of Singapore ID

Starting from this release RTR SDK can capture the front side of the latest version of Singapore ID. For more information about supported fields, please, see the Help file

8.8.2 Swift4 code sample

Starting from this release all distribution kits of RTR SDK contain code samples written in Swift4 (previously, the Swift-code sample was written in Swift3).

Github repository was updated as well

8.9 Release 6 Patch

8.9.1 [iOS] Protection Update

Starting from this release the iOS protection (extended licenses) was updated.

The change concerns customers who create their own iOS framework, which can be used in different end-user applications. Previously, extended license was bound with the Bundle ID of such

framework, but starting from this release the license is bound with the Bundle ID of the end-user application, which uses the framework developed with the help of ABBYY RTR SDK.

Note. *For Android the situation is the same – we always bind the license with the ID of end-user application.*

8.10 RELEASE 7

8.10.1 SUPPORT OF NEW IDS

Starting from this release RTR SDK can capture the following documents:

- Armenia
  - National ID Card
- Belgium
  - Driver's License
- Chile
  - National ID Card
- Croatia
  - National ID Card
- El Salvador
  - National ID Card
- Finland
  - National ID Card
- France
  - National ID Card
- Georgia
  - International Passport
  - National ID Card
- Greece
  - Driver's License
- Hong Kong
  - National ID Card
- Hungary
  - National ID Card
  - Driver's License
- India
  - Aadhaar Card
- Lithuania
  - National ID Card
- New Zealand
  - Driver's License
- Nigeria
  - National ID Card

- Norway
  - National ID Card
  - Driver's License
- Poland
  - Driver's License
- Russian Federation
  - Birth Certificate
- Serbia
  - National ID Card
  - Driver's License
- South Africa
  - National ID Card
- Spain
  - Driver's License
- Tajikistan
  - International Passport
- USA
  - Green Card
- Uzbekistan
  - International Passport

In addition, we updated the template for Romanian ID card. Now we also capture an address field.

8.10.2 Protection Update

All new documents are protected with specified licensing parameters.

Also we added to the licensing scheme a special internal parameter, PrototypeID. In the future it will help development team to protect new documents, which aren't supported officially in SDK.

For example, when some office requests to support new ID for demonstration, development team will not need to release a new SDK version. We will need just to modify demo app and generate corresponding license. Thus, we will get working demo faster – 1-2 days instead of a week.

8.10.3 Documentation Update

Starting from this release the RTR SDK Help file contains additional How to… articles, which describe:

- How to recognize single photo
- How to capture data from documents

9 FIXED BUGS

Severity	Description
	No bug fixes

10 KNOWN ISSUES AND WORKAROUNDS

This section contains a list of known problems for this release. Four-point scale will help you to evaluate the severity of each issue.

Critical	A bug that causes crashes or hangings of software. Critical bugs can include access violations, internal program errors, stack overflow, out of memory or other exceptions that can lead to program failure.
Major	A bug that does not cause program failure but affects major functionality of a feature or impairs the system's performance. Major bugs can include disparity of the feature functionality to the internal specifications, memory leaks or data corruption.
Minor	A bug that leads to feature malfunctioning or affects minor functionality of the software. Minor bugs can include recognition errors, missing or lost objects, wrong color detection, incorrect document analysis, license counter errors, etc.
Trivial	A cosmetic issue that does not affect the functionality of the product but can cause inconveniences. Trivial bugs can include Help file errors, log errors, incomplete information in error messages, etc.

The table below contains the list of main issues sorted in descending order of the severity.

Severity	Description	Platform
Major	RTR SDK is not compatible with Mobile OCR Engine 4. **Workaround. A method for single image processing was added to RTR SDK**	Android

11 CUSTOMER SUPPORT

The ABBYY SDK Support team is ready to help you. Please use ABBYY Developers forum to contact us: http://forum.ocrsdk.com/tags/rtr/