ABBYY®

MOBILE OCR ENGINE 4 RELEASE 15 UPDATE

RELEASE NOTES

1/27/2016

1 CONTENTS

2 About This Release

2.1 Purpose

Purposes of the release:

1) Compile Android library for ARM-v7 and x86 processor architectures
2) Implement some improvements in Japanese OCR
3) Critical bug fix

2.2 Build Information

Part Number:	889/53
Build Number:	4.7.0.42

2.3 Additional Information

Knowledgebase: http://knowledgebase.abbyy.com/?searchText=&ProductId=375
Forum: http://forum.ocrsdk.com/

3 Software and Hardware Requirements

3.1 Supported Operating Systems

The following platforms are supported:

1) iOS 7.1 and higher
2) Android 2.2 and higher for ARM processors

iOS and Android native versions of the library are supplied as static libraries.

Note: for each supported platform we deliver a separate distributive (for more information see Distribution Package section)

3.2 Memory Requirements

Exact memory requirements vary depending on the operating system and specific recognition tasks (e.g. multilingual recognition requires more memory).

	European languages	Korean	Chinese + Japanese
Library, MB	5 – 7 (depends on the platform)		
Patterns, MB	1.64	1.37 – 5.23	11.5
Keywords for BCR, MB	1.56	0.113	0.869
Dictionaries, MB	14.4	NA	NA

In average recognition of an A4-sized color image (10-30 MB, BMP format) using iOS/Android Wrapper requires 80 MB RAM or less.

3.3 Compatibility with Previous Versions

The Backward Compatibility section can be found in the Help file included in the distribution pack

4 Distribution Package

We deliver three separate distributives:

- for development for supported platforms **MobileOCR.Android** and/or **MobileOCR.iOS**
- the utility pack **Windows.Tools**

Each distributive for development (**MobileOCR.Android, MobileOCR.iOS**) includes:

- **Native library**, corresponding wrapper and necessary data (dictionaries, recognition patterns, keywords for BCR)
- **Sample code** illustrating the use of Mobile OCR Engine library
- **Sample images** which can be used to test the product functionality
- **Documentation** providing description of native API and corresponding wrapper

The utility pack **Windows.Tools** contains test files required for library testing on Windows PC platform (x32):

- **Native library**
- **Sample code** illustrating the use of Mobile OCR Engine library
- **Demo tool** for Windows platform (TestShell application)
- **Documentation**

5 New Features and Improvements

5.1 (Android Wrapper) Library version for ARM-v7 processors

Starting from this release special library version for ARM-v7 processors were added to the **MobileOCR.Android** distribution. The corresponding **libMobileOcrEngine.so** files can be found in the following folders:

- lib\AndroidWrapper\armeabi-v7a
- lib\AndroidWrapper_NoCjk\armeabi-v7a – library version which is intended to be distributed in apps which do not work with Chinese (simplified or Traditional), Japanese and Korean recognition languages

This library version is optimized for modern devices with ARM-v7 architecture. In comparison with library version for ARM processor architecture it demonstrates 10-20% higher recognition speed on the devices with ARM-v7 processors. For more details, please, see the performance test results in the appendix.

5.2 (Android Wrapper) Library version for x86 architecture

Starting from this release special library version for x86 processor architecture were added to the **MobileOCR.Android** distribution. The corresponding **libMobileOcrEngine.so** files can be found in the following folders:

- lib\AndroidWrapper\x86
- lib\ AndroidWrapper_NoCjk\x86 – in case Chinese (simplified or Traditional), Japanese and Korean recognition languages aren't used

This library version is intended to be used on **Android Emulator**.

5.3 Japanese OCR Improvements

5.3.1.1 Post-processing for Japanese OCR results

From this version special post-processing is implemented for Japanese OCR results.

The Engine processes OCR results and replaces old Kanji characters which are no longer used in modern Japanese language with modern characters.

This post-processing takes place in case the Japanese is selected as the only recognition language. Whether other languages are added to the recognition set (for example, Japanese + English), the post-processing doesn't take place.

5.3.1.2 New characters in Japanese recognition set

New characters were added to Japanese recognition set (the list can be found in the appendix).

This changes also affect the accuracy of Chinese Simplified BCR

The changes listed in chapters 4.1.3.1 and 4.1.3.2 lead to a little improvement in recognition accuracy – percentage of character errors on test Japanese batches (archive scenario) decreased for 0.7% in Normal mode (from 16.09% to 15.97%) and 1.5% in Fast mode (from 16.34% to 16.09%).

5.4 (iOS Wrapper) New default values of recognition parameters

In iOS wrapper default values of recognition parameters were updated. Now they are consistent with the default values for Android wrapper:

- _barcodeTypes = 1048575
- _confidenceLevel = LEVEL3
- _defaultCodePage = UTF8
- _imageProcessingOptions = 0
- _imageResolution = 0
- _isMicrModeEnabled = false
- _recognitionMode = FULL

6 Resolved Issues

This section contains a list of reported bugs that have been fixed.

This four-point scale will help you to evaluate the severity of each issue, enabling you to make informed decisions on how important updates are for your system.

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 a disparity between feature functionality and internal specifications, memory leaks and data corruption.
Minor	Minor bugs can include recognition errors, missing or lost objects, incorrect 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 following table contains bugs fixed in this release, listed in descending order of severity. If the bugs have workarounds, root causes or side effects, they will be mentioned in the Description section.

Severity	Description	Platform
Critical	Working on the iOS Simulator iOS sample crashes when calling methods FineRecognizeImage or FineRecognizeBusinessCard	iOS (iPhone Sample)
Critical	Memory leak takes place while recognizing with the help of Mobile OCR Engine iOS wrapper methods (no more than 150 Kb per image). As a result the app crashes after recognition of a lot of images (about 3000-4000) in a row	iOS Wrapper
Critical	Error while building the app with bicode enabled (iOS 9)	iOS
Major	The call of isLanguageAvailableForOcr method leads to app failure	iOS
Major	The sample can't open images	iOS (iPhone Sample)
Major	The sample can't open images	iOS (Native Sample)

7 CUSTOMER SUPPORT

The ABBYY SDK Support team is ready to help you. Please refer to the contact information and hours below.

CONTACTS

Customer Support Management (CSM) Portal: www.abbyy.com/csm
Developer Support Email: Dev_support@abbyyusa.com
Office Hours: Monday – Friday from 9AM to 6PM PST

INFORMATION REQUIRED

When opening a support case or contacting support, please be prepared to provide the following information:

- Description of the issue
- Sample images for testing
- Full error messages that have occurred
- Any additional information you feel may be helpful for the investigation

The information above will assist the ABBYY Support team in investigating your issue and providing a prompt response.

8 APPENDIX 1. PROCESSING SPEED

The table below shows average processing time (**recognizeText** method of RecognitionManager interface) **in ms** for images of **A4** documents in different recognition languages.

Devices:

- LG Nexus 5, Android 6
- Samsung Galaxy S4, Android 5

	LG (ARM-v7 processor)			Samsung (ARM-v7 processor)		
	armeabi	armeabi-v7a	Difference	armeabi	armeabi-v7a	Difference
English	*2492*	***2102***	*-15.65%*	*2238*	***1772***	*-20.82%*
Chinese	*7103*	***5605***	*-21.08%*	*5697*	***4420***	*-22.41%*
Japanese	*9079*	***7701***	*-15.17%*	*4646*	***3501***	*-24.64%*
Korean	*4032*	***3265***	*-19.02%*	-	-	-

9 APPENDIX 2. CHARACTERS ADDED TO THE JAPANESE RECOGNITION SET

葵茜渥葦斡飴絢綾磯胤吋迂鵜嘘蔚厩噂洩瑛穎堰襖荻珂茄蝦嘩迦峨凱馨蛙鈎廓樫樺兜姦柑澗莞舘
巌祇蟻誼掬鞠砧杵黍笈鋸僑兇匡彊芹喰釧沓窪隈袈珪畦硯絃諺袴瑚醐鯉佼倖垢糠肱閤壕轟漉裟犀
堺鮭笹燦讃獅痔篠縞杓蒐酋駿渚薯樟樵蕉鞘蝕榛疹笥錐雛摺棲蹟蝉舛噌匝漕聡揃舵岱苔醍鷹啄琢
巽竪辿狸樽湛檀弛蜘喋諜槌槻佃柘辻椿剃悌禎轍顛澱兎宕禿噸惇沌呑捺馴楠迩賑濡膿覗蚤芭秤萩
箔粕曝駁肇畠筏蛤隼樋簸毘髭謬瓢瀕埠冨蕪葺糞瞥娩庖鵬俣巳箕稔姪牝緬籾貰宥柚猷輿熔慾稜諒

鱗 苓 煉 榔 倭 鷲 亘 詫 藁 椀 俐 凛 奎 巌 惺 昊 暉 檜 椰 橙 毬 洵 燎 珀 琥 皓 眸 稟 穹 笙 絆 綸 繁 繍 羚 脩 蕾 蟬 蠟 逞 醤 釉 顛 鷗 麒 櫻 ヅ