ABBYY FineReader Engine 11 Technical Release Notes

Contents

[R1 GM – 9](#_Toc474162985)

[Part number, build number 9](#_Toc474162986)

[Product Description 9](#_Toc474162987)

[What is New 9](#_Toc474162988)

[New Usage Scenarios 9](#_Toc474162989)

[OCR/ICR Improvements 12](#_Toc474162990)

[New Image Preprocessing Tools 13](#_Toc474162991)

[Performance Improvements / Shortened Development Curve 15](#_Toc474162992)

[Export Improvements 16](#_Toc474162993)

[New Samples in Code Sample Library 19](#_Toc474162994)

[Other improvements 21](#_Toc474162995)

[Upgrade from Previous Version 22](#_Toc474162996)

[Installing on the Same Machine 22](#_Toc474162997)

[Compatibility of Protection Keys 22](#_Toc474162998)

[Using Source Code for Previous Version 22](#_Toc474162999)

[Components Delivery 25](#_Toc474163000)

[FTP Delivery 25](#_Toc474163001)

[DVD Box 25](#_Toc474163002)

[Compatibility Issues with Version 10 25](#_Toc474163003)

[New default installation folders 25](#_Toc474163004)

[Licensed 3rd-party software 26](#_Toc474163005)

[Distribution Components 26](#_Toc474163006)

[Documentation 26](#_Toc474163007)

[Sample images 27](#_Toc474163008)

[Code samples 27](#_Toc474163009)

[Licensing Model and Parameters 29](#_Toc474163010)

[Definitions 29](#_Toc474163011)

[License types 30](#_Toc474163012)

[Non-functional License Limitations 34](#_Toc474163013)

[Functional License Limitations 36](#_Toc474163014)

[Protection Key Types, Activation, Deactivation, Registration 37](#_Toc474163015)

[Supported Protection Types 37](#_Toc474163016)

[Supported Activation Types 37](#_Toc474163017)

[License Deactivation 38](#_Toc474163018)

[License Limitations 38](#_Toc474163019)

[Page Counter 38](#_Toc474163020)

[Installation 39](#_Toc474163021)

[Supported Languages, Types and Formats 39](#_Toc474163022)

[Supported Recognition Languages 39](#_Toc474163023)

[Supported barcode types 41](#_Toc474163024)

[Supported text types and field marking types 41](#_Toc474163025)

[Supported import and export formats 42](#_Toc474163026)

[GetEngineObject function changes 43](#_Toc474163027)

[Full native 64-bit support 43](#_Toc474163028)

[Classification 43](#_Toc474163029)

[Classification modes 44](#_Toc474163030)

[Classification confidence 44](#_Toc474163031)

[FlexiFormsDA and FullTextIndexDA options in API and licensing 44](#_Toc474163032)

[Installation on a virtual machine and activation 44](#_Toc474163033)

[CJK doesn’t Use User Patterns 45](#_Toc474163034)

[Software and Hardware Requirements 45](#_Toc474163035)

[ABBYY FineReader Engine 11 Requirements 45](#_Toc474163036)

[ABBYY SDK 11 License Server Requirements 46](#_Toc474163037)

[New Features and Improvements 47](#_Toc474163038)

[Pattern training for Vietnamese language 47](#_Toc474163039)

[Support for Russian with Accents language 47](#_Toc474163040)

[Support for Russian Old Spelling language 47](#_Toc474163041)

[Support for FullAscii and Code32 1D barcodes 47](#_Toc474163042)

[Backward compatibility with prior versions, new Help article 48](#_Toc474163043)

[Re-formatted Distribution.csv file for easier runtime file list composing 48](#_Toc474163044)

[New export formats are supported in CLI sample: ALTO, EBook 49](#_Toc474163045)

[JBIG2 lossless compression is supported 49](#_Toc474163046)

[Java interop throws more informative errors 49](#_Toc474163047)

[Improvements in MRC visual quality 49](#_Toc474163048)

[Method for checking if a page is empty 50](#_Toc474163049)

[New OnBlockAdded callback for ImageViewer and ZoomViewer interfaces 50](#_Toc474163050)

[Aggressive table detection mode 50](#_Toc474163051)

[Predefined processing profile for writing highly compressed image-only PDFs 50](#_Toc474163052)

[XPS export format 50](#_Toc474163053)

[Retaining document layout in textual export 51](#_Toc474163054)

[Fixed Bugs 51](#_Toc474163055)

[Known Issues and Workarounds 51](#_Toc474163056)

[CLI sample does not allow exporting if a license lacks of any exporting module 51](#_Toc474163057)

[VB6 samples are not compatible with 64-bit version of the Engine 52](#_Toc474163058)

[IFontSet::EnablePdfStandardFonts is non-functional 52](#_Toc474163059)

[FRE installer fails on localized Chinese Windows 7 x64 52](#_Toc474163060)

[OnPageProcessed comes once per a document 52](#_Toc474163061)

[Some API is not implemented 52](#_Toc474163062)

[Initialization of APDFL library fails if a path to the contains hieroglyphs 53](#_Toc474163063)

[Recognition of a PDF file with text in Vietnamese may fail 53](#_Toc474163064)

[Long font names in MS Word 53](#_Toc474163065)

[Out of memory exception during export very large image into PDF using LZW compression 53](#_Toc474163066)

[PDF/A validation report 53](#_Toc474163067)

[R2 GM – 54](#_Toc474163068)

[Part number, build number 54](#_Toc474163069)

[New Features and Improvements 54](#_Toc474163070)

[New samples for Java 54](#_Toc474163071)

[Improved Java wrapper 54](#_Toc474163072)

[IEngine::InjectTextLayer method was added 55](#_Toc474163073)

[Support for IntelligentMail barcodes 55](#_Toc474163074)

[Predefined languages OcrA and OcrB were added 55](#_Toc474163075)

[New special predefined languages for phone numbers and EIN 55](#_Toc474163076)

[Detection of vertical text for all languages 55](#_Toc474163077)

[New IClassificationTrainer::AddPage method 55](#_Toc474163078)

[The possibility to add custom features for classification 56](#_Toc474163079)

[New predefined profile EngineeringDrawingsProcessing 56](#_Toc474163080)

[Export with original layout to XLSX 56](#_Toc474163081)

[New properties for IBarcodeParams object 56](#_Toc474163082)

[Support for embedded files in PDF 56](#_Toc474163083)

[Enhanced JBIG2 lossless compression 57](#_Toc474163084)

[New property IEngine::Version 57](#_Toc474163085)

[Version for FREngine.jni.dll was added 57](#_Toc474163086)

[Screenshot detection 57](#_Toc474163087)

[Launch of AInfo utility from the command line 57](#_Toc474163088)

[Improved key value tables detection 57](#_Toc474163089)

[Serbian Cyrillic ICR support 57](#_Toc474163090)

[Improved memory allocation 57](#_Toc474163091)

[Fixed Bugs 57](#_Toc474163092)

[Known Issues and Workarounds 60](#_Toc474163093)

[Some API is not implemented 60](#_Toc474163094)

[OnPageProcessed comes once per a document 60](#_Toc474163095)

[Skew Correction works on the image without text 61](#_Toc474163096)

[In parallel mode IWords collection is empty after processing 61](#_Toc474163097)

[Skew is not corrected properly on the specific images 61](#_Toc474163098)

[Problem with number recognition 61](#_Toc474163099)

[Incorrect positioning of images in the results of export to HTML 61](#_Toc474163100)

[The results of export to XML in several cases don’t pass XML validation 61](#_Toc474163101)

[The text doesn’t fit the size of cells during export to XLSX. 61](#_Toc474163102)

[Wrong detection of footers and headers 61](#_Toc474163103)

[InjectTextLayer method works incorrectly on some Japanese images 61](#_Toc474163104)

[PDF/A validation report 61](#_Toc474163105)

[R2 GM Patch – 63](#_Toc474163106)

[Part number, build number 63](#_Toc474163107)

[Fixed Bugs 63](#_Toc474163108)

[R3 GM – 65](#_Toc474163109)

[Part number, build number 65](#_Toc474163110)

[New Features and Improvements 65](#_Toc474163111)

[Possibility to extract and to add attachments from PDF 65](#_Toc474163112)

[New property IPrepareImageMode::BackgroundFillingColor to fill areas added after skew correction 65](#_Toc474163113)

[Possibility to see PDF layers in PDF Viewers 66](#_Toc474163114)

[Parallel export to PDF and to PPTX 67](#_Toc474163115)

[New method IImageDocument::RemoveColorObjectsEx 67](#_Toc474163116)

[New advanced language detection mode 68](#_Toc474163117)

[New predefined filter FNF\_PDF in FontNamesFiltersEnum 69](#_Toc474163118)

[New property IFRDocument::PDFFontNames 69](#_Toc474163119)

[Skew correction in preprocess stage 69](#_Toc474163120)

[Advanced IsEmptyEx method for checking if the page is empty 69](#_Toc474163121)

[WibuKey support 70](#_Toc474163122)

[Possibility to export TIFF files with one strip 70](#_Toc474163123)

[New property Recognition Set 70](#_Toc474163124)

[Deactivation in silent mode 70](#_Toc474163125)

[Improved CRM\_ContentOnly mode 70](#_Toc474163126)

[Correct display of PDF files with PMingLIU/MingLIU fonts in PDF viewers 71](#_Toc474163127)

[Information about adding comments in profiles in Help file 71](#_Toc474163128)

[Performance results 71](#_Toc474163129)

[English 71](#_Toc474163130)

[Japanese 72](#_Toc474163131)

[Korean 73](#_Toc474163132)

[Chinese 73](#_Toc474163133)

[Fixed Bugs 74](#_Toc474163134)

[Known Issues and Workarounds 76](#_Toc474163135)

[Some API is not implemented 76](#_Toc474163136)

[IWords collection is empty and IWordRecognitionVariants and ICharacterRecognitionVariant return NULL after processing 76](#_Toc474163137)

[The text doesn’t fit the size of cells during export to XLSX. 77](#_Toc474163138)

[Wrong detection of footers and headers 77](#_Toc474163139)

[InjectTextLayer method works incorrectly on some Japanese images 77](#_Toc474163140)

[IPDFMRCParams::MonochromeText doesn’t work correctly 77](#_Toc474163141)

[Errors during opening of PDF file 77](#_Toc474163142)

[CSS files during export to HTML 77](#_Toc474163143)

[PDF/A validation report 77](#_Toc474163144)

[R4 GM – 78](#_Toc474163145)

[Part number, build number 78](#_Toc474163146)

[New Features and Improvements 78](#_Toc474163147)

[Protection redundancy 78](#_Toc474163148)

[Export to memory 78](#_Toc474163149)

[Load from memory method in outproc 78](#_Toc474163150)

[Possibility to enable and disable interpolation in PDF viewers 79](#_Toc474163151)

[New property IEngine::AvailablePredefinedLanguages 79](#_Toc474163152)

[New method OnChangeBlockType in Visual Components Sample 79](#_Toc474163153)

[New property for shadows and highlights correction in photographs 79](#_Toc474163154)

[Support of Farsi 80](#_Toc474163155)

[New property IHTMLExportParams::UseCss 80](#_Toc474163156)

[Possibility to improve recognition quality by removing color objects before recognition during preprocessing stage 80](#_Toc474163157)

[Possibility to replace black or white color of exported PNG images with transparent 81](#_Toc474163158)

[New method IEngine::CreateMultipageImageWriterEx 82](#_Toc474163159)

[New attribute ‘rotation’ in xml export scheme 82](#_Toc474163160)

[New property IRTFExportParams::KeepPageBreaks 82](#_Toc474163161)

[New profile for speed up of barcode recognition 82](#_Toc474163162)

[New section in Developer’s Help ”Predefined Profiles Specification” 83](#_Toc474163163)

[The changing of the GUI forms height of some samples 83](#_Toc474163164)

[Support of corrupted tiff files opening 83](#_Toc474163165)

[Performance results 84](#_Toc474163166)

[English 84](#_Toc474163167)

[Japanese 85](#_Toc474163168)

[Korean 86](#_Toc474163169)

[Chinese 86](#_Toc474163170)

[Fixed Bugs 87](#_Toc474163171)

[Known Issues and Workarounds 90](#_Toc474163172)

[Some API is not implemented 90](#_Toc474163173)

[The text doesn’t fit the size of cells during export to XLSX. 90](#_Toc474163174)

[Wrong detection of footers and headers 90](#_Toc474163175)

[InjectTextLayer method works incorrectly on some Japanese images 90](#_Toc474163176)

[IPDFMRCParams::MonochromeText doesn’t work correctly 91](#_Toc474163177)

[Errors during opening of PDF file 91](#_Toc474163178)

[Small displacement of the text rectangle in PDF files 91](#_Toc474163179)

[PDF/A validation report 91](#_Toc474163180)

[R5 GM – 92](#_Toc474163181)

[Part number, build number 92](#_Toc474163182)

[New Features and Improvements 92](#_Toc474163183)

[Exporting to ALTO up to version 3.0 92](#_Toc474163184)

[Memory saving in case of no need in coordinates on original image 92](#_Toc474163185)

[Speeding up multi-page image opening 92](#_Toc474163186)

[Crop function supports greyscale and b/w images 93](#_Toc474163187)

[New Java wrapper functions for loading Engine able to throw exceptions 93](#_Toc474163188)

[Fine tuning of paper size detection 94](#_Toc474163189)

[Improved MRC quality 94](#_Toc474163190)

[Performance results 99](#_Toc474163191)

[English 99](#_Toc474163192)

[Japanese 100](#_Toc474163193)

[Korean 101](#_Toc474163194)

[Chinese PRC 102](#_Toc474163195)

[Chinese Taiwan 103](#_Toc474163196)

[Arabic 104](#_Toc474163197)

[Fixed Bugs 105](#_Toc474163198)

[Known Issues and Workarounds 107](#_Toc474163199)

[Some API is not implemented 107](#_Toc474163200)

[The text doesn’t fit the size of cells during export to XLSX. 107](#_Toc474163201)

[Wrong detection of footers and headers 107](#_Toc474163202)

[InjectTextLayer method works incorrectly on some Japanese images 107](#_Toc474163203)

[IPDFMRCParams::MonochromeText doesn’t work correctly 107](#_Toc474163204)

[Errors during opening of PDF file 107](#_Toc474163205)

[Small displacement of the text rectangle in PDF files 108](#_Toc474163206)

[‘User Pattern Training Utility’ sample crashes for ‘FieldLevelRecognition’ profile 108](#_Toc474163207)

[PDF/A validation report 108](#_Toc474163208)

[R6 GM – 109](#_Toc474163209)

[Part number, build number 109](#_Toc474163210)

[New Features and Improvements 109](#_Toc474163211)

[Windows 10 support 109](#_Toc474163212)

[Large document conversion to searchable PDFs 109](#_Toc474163213)

[Japanese OCR improvements 110](#_Toc474163214)

[Ability to process single-page documents from memory in batch mode 110](#_Toc474163215)

[Arabic Fathatan (ً) diacritics support 111](#_Toc474163216)

[Thai OCR accuracy improvement 111](#_Toc474163217)

[Extended information on character position in a word 111](#_Toc474163218)

[Smaller PDF with pages of mixed colority 111](#_Toc474163219)

[‘BarcodeRecognition\_Accuracy’ became more accurate 112](#_Toc474163220)

[Callback from parallel processing 112](#_Toc474163221)

[Warnings are connected to particular pages 112](#_Toc474163222)

[Updated Admin’s Guide 112](#_Toc474163223)

[‘Classification’ sample became faster 113](#_Toc474163224)

[CodeMeter drivers update 113](#_Toc474163225)

[‘Is Redundancy’ option is listed in LM 113](#_Toc474163226)

[Performance results 113](#_Toc474163227)

[English 113](#_Toc474163228)

[Japanese 114](#_Toc474163229)

[Korean 115](#_Toc474163230)

[Chinese PRC 116](#_Toc474163231)

[Chinese Taiwan 117](#_Toc474163232)

[Arabic 118](#_Toc474163233)

[Fixed Bugs 119](#_Toc474163234)

[Known Issues and Workarounds 121](#_Toc474163235)

[Some API is not implemented 121](#_Toc474163236)

[Farsi language is based on Arabic language 121](#_Toc474163237)

[IPDFMRCParams::MonochromeText doesn’t work correctly 121](#_Toc474163238)

[‘User Pattern Training Utility’ sample crashes for ‘FieldLevelRecognition’ profile 121](#_Toc474163239)

[Infinite loop in Java crashes 121](#_Toc474163240)

[Password protected PDF files couldn’t be opened in parallel processing mode 122](#_Toc474163241)

[JapaneseModern OCR language is not available for BCR 122](#_Toc474163242)

[Language auto detection uses CJK resources though none is selected for OCR 122](#_Toc474163243)

[R6 GM Patch – 123](#_Toc474163244)

[Part number, build number 123](#_Toc474163245)

[New Features and Improvements 123](#_Toc474163246)

[Performance results 123](#_Toc474163247)

[Fixed Bugs 123](#_Toc474163248)

[Known Issues and Workarounds 124](#_Toc474163249)

[Some API is not implemented 124](#_Toc474163250)

[Farsi language is based on Arabic language 124](#_Toc474163251)

[IPDFMRCParams::MonochromeText doesn’t work correctly 124](#_Toc474163252)

[‘User Pattern Training Utility’ sample crashes for ‘FieldLevelRecognition’ profile 124](#_Toc474163253)

[Language auto detection uses CJK resources though none is selected for OCR 125](#_Toc474163254)

[PDF/A validation report 125](#_Toc474163255)

[R7 GM – 126](#_Toc474163256)

[Part number, build number 126](#_Toc474163257)

[New Features and Improvements 126](#_Toc474163258)

[Conversion documents to searchable PDFs at full throttle requires no page numbers setup any more 126](#_Toc474163259)

[Simultaneous network and standalone usage of one FRE installation 126](#_Toc474163260)

[Garbage removal from color images 126](#_Toc474163261)

[Possibility to inject text layer to selected pages of PDF documents 128](#_Toc474163262)

[A new article about working with screenshots in documentation 128](#_Toc474163263)

[Performance results 128](#_Toc474163264)

[English 128](#_Toc474163265)

[Japanese 129](#_Toc474163266)

[Korean 130](#_Toc474163267)

[Chinese PRC 131](#_Toc474163268)

[Chinese Taiwan 132](#_Toc474163269)

[Arabic 133](#_Toc474163270)

[Fixed Bugs 134](#_Toc474163271)

[Known Issues and Workarounds 137](#_Toc474163272)

[Some API is not implemented 137](#_Toc474163273)

[Farsi language is based on Arabic language 138](#_Toc474163274)

[IPDFMRCParams::MonochromeText doesn’t work correctly 138](#_Toc474163275)

[‘User Pattern Training Utility’ sample crashes for ‘FieldLevelRecognition’ profile 138](#_Toc474163276)

[The process hangs when the multiprocessing methods works on the machine with one CPU. 138](#_Toc474163277)

[PDF/A validation report 138](#_Toc474163278)

[R7 GM Patch – 139](#_Toc474163279)

[Part number, build number 139](#_Toc474163280)

[New Features and Improvements 139](#_Toc474163281)

[Performance results 139](#_Toc474163282)

[English 139](#_Toc474163283)

[Japanese 140](#_Toc474163284)

[Korean 141](#_Toc474163285)

[Chinese PRC 142](#_Toc474163286)

[Chinese Taiwan 143](#_Toc474163287)

[Arabic 144](#_Toc474163288)

[Fixed Bugs 145](#_Toc474163289)

[Known Issues and Workarounds 146](#_Toc474163290)

[R8 GM – 146](#_Toc474163291)

[Part number, build number 146](#_Toc474163292)

[Change in Behavior 146](#_Toc474163293)

[New Features and Improvements 146](#_Toc474163294)

[Orientation and skew correction using InjectTextLayer method 146](#_Toc474163295)

[An opportunity to check PDF files that placed in memory for text layer 147](#_Toc474163296)

[An opportunity not to write BOM on export to TXT 147](#_Toc474163297)

[Human-readable names of paragraph style on export to XML 147](#_Toc474163298)

[An opportunity to rasterize FreeText annotations 147](#_Toc474163299)

[Added ‘Not enough disk space’ error code 147](#_Toc474163300)

[New option for faster printing of PDF using MRC 148](#_Toc474163301)

[New article about OMR/Barcode licenses in documentation 148](#_Toc474163302)

[Performance results 148](#_Toc474163303)

[English 148](#_Toc474163304)

[Japanese 150](#_Toc474163305)

[Korean 151](#_Toc474163306)

[Chinese PRC 152](#_Toc474163307)

[Chinese Taiwan 153](#_Toc474163308)

[Arabic 154](#_Toc474163309)

[Fixed Bugs 155](#_Toc474163310)

[Known Issues and Workarounds 156](#_Toc474163311)

[R8 Patch GM – 157](#_Toc474163312)

[Part number, build number 157](#_Toc474163313)

[New Features and Improvements 157](#_Toc474163314)

[Performance results 157](#_Toc474163315)

[Fixed Bugs 157](#_Toc474163316)

[Known Issues and Workarounds 158](#_Toc474163317)

# R1 GM –

## Part number, build number

|  |  |
| --- | --- |
| Part# | 1041/17 |
| Build# | 11.1.4.118 |

## Product Description

ABBYY FineReader Engine 11 is a comprehensive software development kit (SDK) for integrating ABBYY's multilingual OCR, ICR, OMR, OBR, BCR, document classification, document imaging, document conversion and PDF conversion technologies into applications for Windows operating systems.

## What is New

### New Usage Scenarios

|  |  |  |
| --- | --- | --- |
| **Feature** | **Description** | **Benefits** |
| **Automatic document classification**  *The task of document classification is to assign a document to some category on base of its content.* | ABBYY FineReader Engine 11 provides an API for automatic document classification that enables applications to categorize and sort batches of documents by predefined document classes.    Classification in FineReader Engine 11 can be performed in two modes (classification profiles):   * **Maximum Speed.** This mode is useful for documents that contain not much text, and the difference between classes is visible in the appearance of the documents. It uses the following classification criteria:   + Image pattern (black pixels location template)   + OCRed text analysis: Title text * **Maximum accuracy.** This mode is useful for documents that contain a lot of text, and the difference between classes can be determined only when text content is taken into account. Classification criteria are:   + OCRed text analysis: Full-text   The “Maximum speed” mode provides classification speed from 3 to 10 times faster than “Maximum accuracy” mode.  The classification results are:   * Detected category of the document, * Probability that a document belongs to a category.   Classification probability may be used to determine how to further process classified documents, for example, whether to re-classify some of documents manually, or to be able to route documents to the right department.  ***Sample usage scenarios:***   * Archiving: Sorting documents by type for electronic archive creation. * Mailrooms and Workflow Automation: According to document class detected some further actions can be initiated. * Batch Processing: Document separation. * BPO: Pre-sort documents for further processing. * Banking/Insurance: Verification of document set completeness is applied to loan applications and insurance payouts. * OEM: Smart MFP/scanner interfaces suggesting typical actions for each document class. | Enables workflow automation  Boost productivity and reduce costs by eliminating manual pre-sorting  It’s easy, no specific knowledge needed. No templates required. Can be trained easily by non-technical end-users.  Easily adjustable. Whenever he needs anyone can train the engine to classify new types of documents.  Universal. Fits for all types of documents.  Easy to integrate. One page of code is required for basic scenario implementation. |
| **Business Card Recognition** | Business card recognition technology now is integrated in FineReader Engine 11. The API provides a full set of features for business cards processing: from special preprocessing features to the API that provides access to extracted data.  BusinessCard_pict.gif  The following fields can be extracted:   * Personal name * Company name * Position in the company * Company address * Phone number * Fax * Mobile phone number * E-mail * Web site   **Export to vCard format**   * Recognized data can be saved in vCard format, which is often used to pass business card by e-mail or networks.   **Business Card reading in 27 languages**   * ChinesePRC, ChineseTaiwan, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Greek, Hungarian, Indonesian, Italian, Japanese, Korean, Norwegian, NorwegianBokmal, NorwegianNynorsk, Polish, PortugueseBrazilian, PortugueseStandard, Russian, Spanish, Swedish, Turkish, Ukrainian.   **Auto-splitting of multiple business cards scanned as one page**   * Multiple business cards can be detected on a page during processing. * Multiple business cards scanned on one page can be split into several pages before processing. | Fast and easy way to transfer and put your business contacts to work  Superior accuracy of text and data processing and recognition  Add more value to your application  A must-have add-on for any CRM system |

### OCR/ICR Improvements

|  |  |  |
| --- | --- | --- |
| **Feature** | **Description** | **Benefits** |
| **Arabic OCR** | FineReader Engine 11 includes Arabic OCR technology. | Expand your business in Arabic markets  Put your finger on the pulse of Arabic world |
| **Improved Japanese and Chinese OCR**  http://img.scoop.it/fU1Hh-PG1UmWGgL12MCIljl72eJkfbmt4t8yenImKBVaiQDB_Rd1H6kmuBWtceBJ | * The speed of processing of **Japanese** documents with FineReader Engine 11 is for **90% faster**. * The speed of processing of **Chinese (Simplified)** documents with FineReader Engine 11 is for **30% faster**. * The speed of processing of **Chinese (Traditional)** documents with FineReader Engine 11 is for **45% faster**. * The speed of processing of **Chinese (Taiwan)** documents with FineReader Engine 11 is for **60% faster**. | Better competition on local Oriental markets with domestic OCR vendors |
| **New languages for OCR and ICR**  **Most Comprehensive Language Pack** | FineReader Engine 11 supports the following **new recognition languages:**   * Turkmen (Latin), Old Slavonic   **Full dictionary support** is now available for the following new languages:   * Latin, Azerbaijani (Latin), Russian (old spelling)   The following new languages are now available for **ICR**:   * Danish, Norwegian (Bokmal), Norwegian (Nynorsk), Old English, Serbian (Cyrillic), Tajik   **User dictionaries** can be created for Japanese and Korean languages. | Widest language support available in the industry – 202 languages for OCR and 126 for ICR |
| **Maxicode barcode support**  http://upload.wikimedia.org/wikipedia/commons/thumb/b/b4/MaxiCode.svg/200px-MaxiCode.svg.png | Maxicode barcode is used for tracking and managing the shipment of packages (i.e. by UPS company). |  |
| **USPS 4CB barcode type\***  *\* still under development* | USPS 4CB or IMB is a barcode used by USA post office. http://www.leadtools.com/help/leadtools/v18/dh/to/Planet.png |  |
| **Receipt recognition**  (new Receipt text type) | This type of text is designed for recognizing sales receipts, invoices, etc. Unlike the other types, it is not concerned with the actual font of the text. Rather, it tells the recognizer that there may be text of low quality, mostly in monospaced or normal font. The typical receipt text can look like this: |  |
| **New mode for low resolution scans** | The special new recognition mode for low quality documents – old faxes, low resolution scans provides 20% higher accuracy for such documents than standard Normal mode | Now you can recognize the low quality documents that caused too many OCR errors before |

### New Image Preprocessing Tools

|  |  |  |
| --- | --- | --- |
| **Feature** | **Description** | **Benefits** |
| **Improved**  **Camera OCR**  *Preprocessing features for photographed documents* | New version of FineReader Engine includes the following new and improved preprocessing features:   * **Geometrical distortions correction** (not only trapezium distortions as in previous version, but any type of geometrical distortions), * **Auto-cropping** * **Background lightening** * **Better ISO noise removal** (as compared to previous version) | Better recognition results  Allows to produce high quality searchable PDF with excellent appearance |
| **Auto-splitting of double-page spread** | Books are usually scanned as double-page spreads. This produces some difficulties for recognition (curved lines, scanning shadows). In the output document it is usually better to have the book page-by-page.  New version of FineReader Engine 11 can perform page splitting automatically. This means higher effectiveness of image preprocessing (curved lines correction, scanning shadows removal). | Better appearance of output document (page-by-page). |
| **New image preprocessing methods** | * **RemoveNoise Method.** This method reduces the noise on the image. * **EnhanceLocalContrast Method.** This method increases the local contrast of the image. Such preprocessing may increase recognition quality of low contrast images. | Better recognition results |

### Performance Improvements / Shortened Development Curve

|  |  |  |
| --- | --- | --- |
| **Feature** | **Description** | **Benefits** |
| **Native 64-bit support** | FineReader Engine 11 now provides native 64-bit support and can be used in 64-bit applications without any further development:   * FineReader Engine 11 provides C++ DLLs that could be linked in x64 applications directly without using COM proxy. * Neutral .NET interops that can be used for building .Net project for “any CPU” (once compiled an application can be used on 32-bit or 64-bit machine without recompilation).   On a 64-bit system both 32-bit and 64-bit FineReader Engine libraries can be installed together, or only one set of libraries can be selected for installation. | Eliminating difficulties with 64-bit applications development |
| **Java Wrapper \***    *\* still under development* | FineReader Engine 11 provides ready-to-use Java classes for the Engine library. These Java classes can be used directly from Java applications.  Because Java platform is OS-independent, it has no special tools for integration with Windows operating system. Creating wrappers using Java Native Interface (JNI) is not an easy task: developer should write huge amount of code to implement wrappers for all necessary interfaces and methods. Now there is no need to do it for FineReader Engine 11.  Due to the native 64-bit support, Engine can be used from Java on 64-bit systems either by loading into the current process (InprocLoader), or by loading into a separate process (OutprocLoader). | Easy access to an Enterprise market (JAVA is a mainstay of enterprise IT development)  Most of OCR SDK seekers in social media are Java developers. |
| **Scanning API \***  *\* still under development* | * An ability to run recognition of scanned pages before scanning of all pages is finished – by use of asynchronous scanning. * Extended access to scan settings, including access to scan source capabilities – which means that the customer will know allowed values of this or that property, even if he works through API (this is useful when creating user scan dialogs). * Filtration of scan sources by available user interfaces (FineReader, scanner UI, none) or scan API types (TWAIN, WIA), which allows limiting the number of scan sources in the user interface. * All limitations for implementing service-like scanning has been removed (writing log file can be canceled, scanning does not require Registry access). * An ability to specify compression type of scanned images. |  |
| **Opening images from memory\***  *\* still under development* | In FineReader Engine 10 the source images can be loaded into the Engine from files only. In the new version, it is possible to implement your own external image queue (custom image source), which will either return references to files on demand or provide the source images as a stream in memory. | Results in increased flexibility, security and performance |
| **Improved font management API** | Font management is much more easier with FineReader Engine 11 – it provides a variety of predefined font filters which save developer from manual font specifying:   * default set used by ABBYY FineReader * a set for European languages * a set for Chinese language * a set for Japanese language * a set for Korean language * a set for Arabic language * a set for Hebrew language * a set for Thai language * a set for Armenian language | Extended access to the fonts used during document synthesis. |

### Export Improvements

|  |  |  |
| --- | --- | --- |
| **Feature** | **Description** | **Benefits** |
| **Improved PDF Export** | | |
| **Faster PDF Export** | Export to PDF now is up to 12% faster than in previous version of FineReader Engine. | Converting images into searchable PDF is one of the most needed scenarios on the market.  Better appearance and minimum size of the document. |
| **Higher quality of PDF MRC** | PDF MRC improvements include:   * higher background image compression, * contrast elements stay in foreground.   Higher background image compression reduces the size of output PDF MRC file for up to 50%. |
| **Stamps and written notes processing for PDF MRC \***  *\* still under development* | Stamps and written notes can be placed to any part of documents and lay over sensitive information, which is needed to be extracted. Such marks are merged with the text during binarization and sensitive information can be lost.  FineReader Engine 11 provides special preprocessing mode for such cases. The idea of this preprocessing mode is that an image is split into two layers: color and black-and-white. Black-and-white layer is used for recognition, while the color layer is passed to export without modifications and it will stay in foreground of resulting PDF MRC file instead of appearing in background as blurred elements.  As a result, output PDF has high-level quality and compression. Text and separators are compressed with suitable black-and-white codec, while color layer - with color codec. |
| **Export to PDF/A-2** | Additionally to PDF, PDF/A-1a, PDF/A-1b FineReader Engine 11 supports export to PDF/A-2a and PDF/A-2u formats.  The main differences of the new format are:   * JPEG2000 compression available * version 1.7 of PDF * Conformity levels:   + ‘a’ – tagged PDF/A-2   + ‘u’ –not-tagged PDF/A-2 with an ability to extract text in Unicode.   **Please note:** *PDF/A-2b, which is a not-tagged PDF/A-2 without an ability to extract text in Unicode, is not created by FineReader Engine 11. This is due to the fact that our export technologies always work with text in Unicode. Therefore PDF/A-2b file generated by FineReader Engine is the same as PDF/A-2u.* |
| **Export to PDF/A-3\***    *\* still under development* | PDF/A-3 is an extension of the A-2 Standard and it allows including not only PDF/A files, but also all kind of other binary formats, like XML or Office formats.  Long-term archiving and readability of the PDF/A part is still guaranteed, and the binary attachments can deliver additional benefits.  The PDF/A-3 extended container capabilities will make this format attractive in new areas, for example when a graphical representation of a document should be combined with some source data. |
| **An ability to create linear or non-linear PDF files** | Linear PDF files have internal data arranged in a page order. A page of a linear PDF file can be read in a web browser plug-in without waiting for the whole file to be downloaded. Non-linear PDF have the data necessary to assemble a document page scattered through the whole file.  New PDF export parameters include new option which specifies whether a linear PDF file should be created. |
| **Keeping existing bookmarks in PDF** | In “PDF -> searchable PDF” scenario there are could be some source PDFs that contain bookmarks. They may appear in scanned PDF indicating beginning of a new source document. Up to the version 11 these bookmarks get lost during export.  In FineReader Engine 11 these bookmarks can be retained in the output PDF file. |
| **Detection and re-use of the an existing text layer** | FineReader Engine 11 includes a set of options, which allows one to omit OCR of text-based PDFs, or use text information from them:   * In the scenario “image PDF -> searchable PDF” one may do not perform processing of the text-based PDF at all, but simply copy it to the output folder.   In the scenarios when text-based PDF is converted to some other format, one can select whether the text information from the file should be used during processing. |
| **Other Export Improvements** | | |
| **Docx and Xlsx Export Improvements\***  *\* based on ABBYY internal test results* | Docx improvements:   * Better pages number detection * Better font detection: Font type , Bold, Italic, Header * Text boxes in “Editable copy” mode * Better text wrapping * Better page orientation detection   Xlsx improvements:   * Better cells detection * Better columns’ width detection |  |
| **Excel export improvements\***  *\* still under development* | FineReader Engine 11 allows retaining formatting of all data in the tables exported to Excel, including numbers:   * Bold font style * Font colors |  |
| **Export to XPS\***    *\* still under development* | XPS (XML Paper Specification) format is based on XML. As PDF format, it provides device-independent document appearance. XPS also comes handy when one doesn't have a printer installed and the XPS virtual printer allows to save the document in "ready-to-print" original format for later printing. |  |
| **Export to memory\***  *\* still under development* | FineReader Engine 11 will be able to save recognized documents not only on disk, but into a file stream. | Increased security for your confidential data |
| **Extended ABBYY XML** | Now there is an ability to save paragraph style and roles into output XML. This can be useful to identify the role of a paragraph, e.g. to detect running titles and footnotes. |  |
| **Other** | * Recreation of the logical structure of a document is an option during export to RTF, DOCX, and HTML formats. * An ability to save information of paragraph styles and roles in XML file. * New color settings for embedded pictures in RTF, DOCX, PPTX, HTML, EPUB, and FB2 formats. |  |

### New Samples in Code Sample Library

|  |  |  |
| --- | --- | --- |
| **Feature** | **Description** | **Benefits** |
| **New Technological Advantages Demonstrations Samples** | | |
| **Classification**  icon_Classification | This sample demonstrates how ABBYY FineReader Engine can be used for document classification.  It allows you to classify a batch of images into three types: receipts, invoices, business cards.  **Classification** code sample includes pre-trained classification data bases for the following document types:   * For English, French, German, Italian, Russian, Spanish languages: **Receipt, Invoice, Contract, Business card**. * For Chinese (PRC), Chinese (Taiwan): **Receipt, Invoice, Business card.** * For Japanese, Korean: **Receipt, Business card**.   As well you can train FineReader Engine to classify any types of documents. | * Ready-to-use UI for your application * Quick learning curve * No need in specialized training courses |
| [**Business Card Recognition**](file:///C:\Users\andreykov\Documents%20and%20Settings\All%20Users\Application%20Data\ABBYY\SDK\10\FlexiCapture%20Engine\Samples\SamplesBrowsingTools\pages\sample_ImagePreprocessing.html)  **icon_BCR** | This sample demonstrates how ABBYY FineReader Engine can be used for business card recognition.    The sample splits multiple business cards scanned on one page if necessary, recognizes each business card, and displays recognized data. You can view recognized contacts in the default mail application. |
| **Updated Technological Advantages Demonstrations Samples** | | |
| **Camera OCR**  **icon_CameraOCR** | This sample was updated with the following new options to play with:   * Correction of geometric distortions * Image crop * Remove noise * Image contrast | * Select the required image pre-processing technologies for your particular document type |
| **New How to…** | | |
| **Convert a batch of documents**  **icon_BatchProcessing** | This sample shows how to use Batch Processor for processing a large amount of one-page documents. It processes a batch of images from the specified folder and saves them in PDF format. | * Quick learning curve * No need in specialized training courses |
| **Effectively use resources of a high-performance computer**  icon_EnginesPool | This sample on the one hand provides a complete reusable solution for a pool of Engines in a multithreaded application, and on the other hand demonstrates the gain in speed when using multiprocessing.  The sample processes images from a predefined folder. It creates a pool of Engine objects, which recognize images from this folder in parallel. The default number of threads equals the number of CPU cores. User can reduce it to compare the difference in speed. |
| **Scan with FineReader Engine**  **icon_Scanning** | This sample provides an implementation of a "scanning computer". It lets the user select the scanner, set up scanning options and scan images into a specified folder.  The sample retrieves a list of scanners available on your workstation and allows you to filter the list by API type and scanning options dialog box type. |

### Other improvements

|  |  |  |
| --- | --- | --- |
| **Feature** | **Description** | **Benefits** |
| **Opening PDF files from memory** | FineReader Engine 10 can open image files in different formats from memory, but not PDF. PDF files must be saved to a disk before they can be processed with Engine. In FineReader Engine 11 this restriction is removed. | * Increased processing speed |
| **Other improvements** | * Added Japanese message language * New Receipt text type * Extended access to the fonts used during document synthesis * An ability to specify resolution for rasterization during PDF opening * Detection of PDF text layer * An ability to open the PDF, DjVu, WIC and WDP files from memory * One method for removing all blocks from layout (ILayout::Clean) * Training image can be loaded directly from an image document (ITrainingImage::SetImageData) * An ability to [cancel processing](mk:@MSITStore:C:\Program%20Files\ABBYY%20SDK\11\FineReader%20Engine\Help\FREngine11.chm::/APIReference/IAsyncProcessingCallback.htm) operation and [repeat processing](mk:@MSITStore:C:\Program%20Files\ABBYY%20SDK\11\FineReader%20Engine\Help\FREngine11.chm::/APIReference/BatchProcessor_ProcessPageAsync.htm) of a page with Batch Processor * FineReader Engine collections can be iterated using [the foreach statement](mk:@MSITStore:C:\Program%20Files\ABBYY%20SDK\11\FineReader%20Engine\Help\FREngine11.chm::/GuidedTour/Collections.htm#foreach) in .NET |  |

## Upgrade from Previous Version

### Installing on the Same Machine

ABBYY FineReader Engine 11 works with any previous ABBYY FineReader Engine major version installed on the same PC if products’ installation folders are different.

### Compatibility of Protection Keys

ABBYY FineReader Engine 11 requires valid serial number for functioning.

### Using Source Code for Previous Version

ABBYY FineReader Engine 11 has certain incompatibilities with API implemented in the previous versions described in “ABBYY FineReader Engine 11 and 10 compatibility” section of the product Help file. Every existing customer willing to upgrade his copy of ABBYY FineReader Engine should read the article first.

Below is short statistics of changes:

|  |  |  |  |
| --- | --- | --- | --- |
| # of API entries in FREngine 10 | # of API entries in FREngine 11 | # of changes in FREngine 11 | % of changes |
| 2788 | 2753 | 155 | **5,56%\*** |

\* “% of Changes” = “# of Changes” / “# of API entries in FREngine 10”. We do not count:

* Changes of default value
* Changes in names of the constants in enumerations

We treat interconnected changes as a single change:

* Renaming or removing of an object and subsequent changes in parameters of related methods
* Uniform renaming of methods, e.g. methods of collection objects.

Short summary of API changes

|  |  |  |
| --- | --- | --- |
| Scenario | Number of changes  (% in the  whole number  of changes) | Comment |
| Common API | **59 (42,14%)** | Mainly these are the changes in processing methods due to changes in parameters.  **Important!** Note that FineReader Engine includes the whole set of methods for each procesing stage and each customer uses only one or several from these methods, e.g. you can process a document using only one FRDocument.Process method, or using Preprocess, Analyze, Recognized, Synthesize methods. Therefore, only from **5% to 15%** of these changes really affect the customer.  The main of these changes are:   * In order to simplify the basic usage scenarios, the **GetEngineObject** function now takes only one parameter – Project ID, and returns a reference to the Engine object. On the other hand, the **GetEngineObjectEx** function provides the full range of parameters for advanced initialization. * As we consider the old “one-page API” as obsolete since version 10, we continue the replacement of the **processing methods** provided by the **Engine** object with the methods of **FRDocument** or **FRPage** objects. |
| Archiving | **14 (10,00%)** | The main changes in the Archiving scenario are due to the following reasons:   * Support for the new PDF/A-2 standard forces some changes in the API, e.g. some PDF export settings has changed their type (from Boolean to enumeration) in order to support automatic selection of value depending on the output standard. For example, **WriteTaggedPDF** property has been replaced with the **WriteTaggedPDFMode** property. * Some obsolete objects have been removed, e.g. **PDFExportParamsOld**, **PDFAExportParamsOld**. * Regroupping of parameters for more structured API. For example, for different export formats (including PDF) the paper size parameters are now specified through the same new object **PaperSizeParams**. |
| Document Conversion | **15 (10,71%)** | * To unify parameter setting for different export formats, there are some regroupping of parameters. For example, for different export formats the parameters of picture export are now specified through the same new object **PictureExportParams**, which provides additional options of picture compression. * To improve working with fonts, especially for languages with complex alphabets or hieroglyphic script, FREngine now provides wider set of predefined fonts and limits manual work for adding fonts. It automatically selects the fonts on the basis of recognition languages.This forces the changes in font management API: now all work with fonts is done via the new **FontSet** object. |
| Field-level Recognition | **13 (9,29%)** | * In field-level recognition scenario, customer usually works with collections of different objects (e.g. languages, words, characters, etc.). Therefore, unified interfaces of all collections are comfortable for developer. In FREngine 11 API we’ve done several unification of the collection objects so that the methods and properties of different collections have the same names. For example, in the **BaseLanguages** collection, the **Add** method has been replaced with **AddNew**; **Remove**, **RemoveAll** methods – with **DeleteAt**, **DeleteAll** methods, respectively. * Unified source for working with languages and dictionaries. In FRE10 some of the methods for working with languages and dictionaries were located in the **Engine** object, others in **LanuageDatabase** object, which may confuse a customer. Now we provide all methods in one place (**LanguageDatabase**). |
| Barcode&OMR | **4 (2,86%)** | * We’ve done several unification of the collection objects so that the methods and properties of different collections have the same names, e.g. in **CheckmarkGroup** object. * Unified names due to extended functionality, e.g. **PDF417CodePage** property of **BarcodeParams** has been renamed to **CodePage**, because this property can be used for barcodes of several types besides PDF417. |
| Image Preprocessing | **14 (10,00%)** | Methods for opening images have been renamed to make their function clear:   * **Prepare...** methods create a copy of image on disk in internal format, ready for loading as **ImageDocument**; * **Open...** methods create an **ImageDocument** of the image and load it into memory; * **Load...** methods simply load into memory an **ImageDocument** which has been created before (possibly using one of the abovementioned methods); * Methods with the name containing **ImageFile** work with a file from disk or memory; * Methods with the name containing **ImageDoc** or **ImageDocument** work with an **ImageDocument**. |
| Scanning | **7 (4,52%)** | The changes for new functionality support (asynchronouse scanning, extended access to scanning settings, etc.). |
| Other | **18,71%** | * The changes for 64-bit support:   + **long** data type has been replaced with **int**,   + **\_\_int64** data type has been introduced,   + new **Handle** interface that simplifies memory management. * There are also some changes that correct spelling errors in the names of properties and methods. |

The following chart illustrates these numbers:

## Components Delivery

### FTP Delivery

FTP delivery is mostly used for trial versions but can be also used for usual sales. It includes:

1. FTP address of the Distribution Pack.
2. Serial Number.
3. Paper User Guide (optional).

### DVD Box

It includes:

1. Common DVD Box with DVD Box Cover
2. CD/DVD with the Distribution Pack copy and the CD/DVD Label.
3. Serial Number.
4. Paper User Guide (optional).

## Compatibility Issues with Version 10

### New default installation folders

Comparing to the previous version the release has different installation folders.

|  |  |
| --- | --- |
| Folder | Description |
| “%ProgramFiles%\ABBYY SDK\11\FineReader Engine” | **Default** value for a folder storing binaries, the Help file and the guides, and USB dongle driver redistribution. Also Read Me file takes place there. User can change the destination upon installation process. |
| “%ProgramData%\ABBYY\SDK\11\Licenses” | This is permanent place for protection subsystem files. |
| “%ProgramData%\ABBYY\SDK\11\FineReader Engine” | This is permanent place for auxiliary Engine files like include files and samples. |

### Licensed 3rd-party software

This version uses several licensed 3rd-party libraries. They enable the product with useful functionality and require us to add certain acknowledgements and items in the product documentation and/or LA.

The list of newly licensed technologies is below.

#### JPEG 2000 Kakadu library

11th version uses JPEG 2000 Kakadu library for saving image files in JPEG 2000 format, or to export to PDF format with embedded JPEG 2000 pictures. That obliges us to specify certain copyrights in the product documentation:

* Working with JPEG2000 image format:  
  Portions of this software are copyright ©2011 University of New South Wales All rights reserved.Product Components

## Distribution Components

### Documentation

|  |  |  |  |
| --- | --- | --- | --- |
| Material | Language | File name | Description |
| Readme | English | CD\ABBYY SDK\11\FineReader Engine\Readme.htm | A short summary of the distribution package and the setup procedure. |
| Product Installation Guide | English | CD\ABBYY SDK\11\FineReader Engine\Help\FREngine11AdminGuide.pdf | The guide describes how to install the software library. |
| License Server Installation Guide | English | CD\License Server\ABBYY SDK 11 License Server\LicenseServer11AdminGuide.pdf | The guide describes how to install the License Server. |
| Help File | English | CD\ABBYY SDK\11\FineReader Engine\Help\FREngine11.chm | A full and detailed description of the product functionality. It also includes chapters on License Manager. |
| License Server Help File | English | CD\License Server\ABBYY SDK 11 License Server\LicenseManager11.chm | A full and detailed description of the License Service and License Manager. |
| User’s Guide | English | CD\ ABBYY SDK\11\FineReader Engine\Help\FREngine11UserGuide.pdf | Printing version of the Help File. |
| Distribution List | English | CD\ABBYY SDK\11\FineReader Engine\Help\FREngine11\_Distribution.csv | A list of files to distribute with description of responsibility and necessity to distribute. |

### Sample images

There are prepared sample images for demonstration of basic scenarios and advanced technologies.

The distribution DVD contains all images in the following folder:

* DVD \CommonAppData\ABBYY\SDK\11\FineReader Engine\Samples\SampleImages

### Code samples

The distribution contains samples described below in the following folder:

* DVD \CommonAppData\ABBYY\SDK\11\FineReader Engine\Samples

To view samples description and run them or open source files a customer should use “Code Samples Library”, the HTML-based browsing utility, or go directly to the location of samples. A developer can find shortcuts for both the utility and the samples folder under the Start menu after the installation process is completed.

#### Samples for developers

|  |  |  |
| --- | --- | --- |
| Name | Available in | Description |
| Hello | * Delphi 5 * VB .Net * VB 6 * C++ (COM) * C++ * C# * Java * VBScript (New) * JavaScript (New) * Perl (New) | Performs document conversion with just a few lines of code. This sample will help you to start development using ABBYY SDK. |
| BatchProcessing (New) | * VB .Net * C++ (COM) * C++ * C# | This sample shows how to use resources of computer effectively when processing a large amount of one-page or several-page documents (using Batch Processor). |
| Engines Pool (New) | * C# | This sample on the one hand provides a complete reusable solution for a pool of Engines in a multithreaded application, and on the other hand demonstrates the gain in speed when using multiprocessing. |
| EventsHandling | * VB .Net * VB 6 * C++ (COM) * C++ * C# | Illustrates the use of the callback interfaces using the FRDocument callback interface (IFRDocumentEvents) as an example. The sample shows the progress of recognition and export during image processing. You can use the callback interfaces to control image processing. |
| Visual Components | * VB .Net * VB 6 * C++ (COM) * C# | This sample illustrates all the steps you need to perform to create a simple application with graphical user interface similar to that of ABBYY FineReader. The sample allows you to open an image, recognize it, verify recognition results and save recognized text in RTF format. |
| CustomLanguage | * VB .Net * VB 6 * C++ (COM) * C++ * C# | Creates a new recognition language and changes its dictionary to a manually-created sample dictionary. After recognition, calculates the number of words in the text and how many of them were found in the user dictionary. You can create a custom recognition language, which will help your program to read specific types of data correctly. |
| RecognizedTextProcessing | * VB .Net * VB 6 * C++ (COM) * C++ * C# | Calculates recognition statistics (e.g. the number of suspicious characters and rejects, the number of words that are not in the dictionary). You can use information about uncertainly recognized characters and words for checking the results of recognition. |
| CommandLineInterface | * C++ | Provides the command line interface of ABBYY FineReader Engine. The sample produces a CommandLineInterface utility, which supports most of the ABBYY FineReader Engine API functions through numerous keys. |
| **Scanning (New)** | * C# | This sample shows how to use scanning functionality available in FineReader Engine. |

#### **Samples for technology advantages demonstration**

All these sample codes are available in C# only.

|  |  |
| --- | --- |
| Name | Description |
| **Business Card Recognition (New)** | The sample shows how FREngine can extract data from business cards and illustrates how several business cards scanned on one page can be split. |
| **Classification (New)** | The sample shows how FREngine can classify documents. It allows one to train its own classification database and use it for document classification. The sample also includes a set of pre-trained classification data bases for the following document types:   * For English, French, German, Italian, Russian, Spanish: **Receipt, Invoice, Contract, Business card**. * For Chinese (PRC), Japanese, Korean: **Receipt, Invoice, Business card**. * For Portuguese (Brazilian): **Receipt, Business card**.   By default, Classification code sample uses Quality mode of classification, which uses full text OCR information for classification. |
| **Camera OCR (Updated)** | This sample shows how you can improve recognition quality of photographed documents with the help of the tools of ABBYY FineReader Engine. You can correct 3D perspective distortions, motion blur, ISO noise, page orientation, and image skew of the photos.  The sample is available in FREngine 10 too. The new version includes new and enhanced photo preprocessing tools:   * white background of a document, * auto-cropping (in previous version we have only estimated rectangle of a page, which can be used for cropping), * correcting geometrical distortions (not only trapezium distortions as in previous version, but any type of geometrical distortions), * removing ISO noise (improved as compared to FREngine 10)   With this sample, you can modify the source photo so that it has “scanning quality”. |
| **Image Preprocessing (Updated)** | This sample includes a set of image preprocessing tools and allows you to watch how this or that tool influences recognition quality. You can use general preprocessing tools (like page orientation and skew correction), filter colors, use special preprocessing tools for photos, and enhance appearance of the images.  The sample is available in FREngine 10 too. The new version includes new Camera OCR features and new auto-splitting features. |
| **Engine Predefined Processing Profiles** | This sample shows how you can easily configure FineReader Engine with predefined processing profiles. All you need is to select the profile suitable for your task before processing and ABBYY FineReader Engine will choose the best settings automatically. |
| **PDFExportProfiles** | Shows the advantages of using PDF export profiles during export to PDF format. These export profiles contain optimal settings for popular export variants and allow you to tune export to PDF with only several parameters. |
| **MultiProcessingRecognition** | Shows the gain in speed when using multiprocessing recognition. Built-in multi-processing makes your application scalable and efficient without any efforts from your side. |
| **BatchProcessingRecognition** | **Shows the gain in speed when using multiprocessing recognition with Batch Processor.** This means that:   * Image files are taken from a custom image source, i.e. you can implement image processing queue in a custom way. * Image files are taken one-by-one from the image source and immediately passed for processing to available recognition processes. When a recognition process completes recognition of an image, it receives the next image from the source. * Recognized pages are returned to the user in the order they have been taken from the image source. |
| **User Pattern Training Utility** | It allows training of patterns to unusual fonts using Engine and Visual Components in a way similar to that available in ABBYY FineReader desktop product. The utility saves patterns in a format compatible with the Engine.  A need in the utility appeared due to changes in ABBYY FineReader 11 project format, which was incompatible with SDK and incorporated patterns along with other document (batch) data. Thus the utility is an alternative for people who prefer (or like) to train patterns in FR GUI and then pass them to SDK. |

## Licensing Model and Parameters

### Definitions

**Product, FRE** – FineReader Engine 11.

**FRE (or Product) instance** – is the 1 object - FRE copy, running in 1 process on 1 CPU core. There are several FRE instances could be, run by user (or application) on 1 or several CPU cores. If FRE instance occupies some CPU cores, it can create new FRE instances for each core internally, without interaction with user or application.

**License** – a set of parameters that regulates Product using.

**Activation** – action that makes License available for a customer.

**TPC** - Total Page Count – the most useful absolute page counter for general characters

**FRXIXPC** – FineReader XIX Page Count - absolute page counter for FineReader XIX (Gothic) characters

**PPM** – Pages per Month – the most useful renewable page counter for general characters

### License types

In this section we describe the general types of licenses, based on different characteristics:

* Activation types (hardware, software and open)
* Target users (developer and runtime)
* Installation types (standalone and network)
* Usage cases (trial and commercial)

#### Hardware, Software and Open Licenses

There are several licenses separated by different activation types:

**Software License**

The Software License is activated by binding to particular computer parameters and is available on the particular computer only. The Software License is mostly used for trial versions or in the case there is no possibility to use Hardware Key (special security rules).

**Hardware License**

The Hardware License is activated by binding to a Hardware Protection Key (USB dongle) and is available on any computer the Hardware Protection Key is ported to.

Now iKey dongles only are supported. Wibu dongles support will be added in the mid of 2013 (FRE11 R2/R3).

**Open License**

Open License is a standalone license in the form of a file key that does not require activation. It is already generated at the moment of generation. This license is distributed as a file.

Open Licenses are intended for those developers who do not wish to encumber their users with the activation process. ABBYY does not advertise the availability of this type if license, but it is available to customers that we can trust.

#### Developer and Runtime Licenses

**Developer License**

Under the License Agreement for Software Development, this license allows the client to use ABBYY SDK for development purposes, without the right to include ABBYY SDK functions in the distributed software.

This license includes full functionality and the only restriction is the number of pages that can be processed within a month or year.

**Runtime License**

This license gives developers the right to use and distribute ABBYY SDK functions in the client’s software. This license is regulated by the License Agreement for Usage. The License Agreement stipulates the royalties for the application with ABBYY SDK libraries, and other distribution conditions.

When ordering a Runtime license, one must make sure it corresponds to the Developer license, under which the application was developed. During the initialization of each application copy Runtime license checks its correspondence to Developer license ID and blocks application if they are not satisfied.

The developers must buy a Runtime license from ABBYY and provide end users with it, together with the application.

**Runtime Emulated mode**

Developer license could be also used in the ***Runtime emulated mode***. That allows running an application on a single computer.

Runtime emulated mode is very helpful for Developer to check which FRE functional modules application really uses, what have to be added to Runtime license.

#### Standalone and Network Licenses

There are two possible license installation types – ***standalone*** and ***network***.

**Standalone license**

Standalone license allows installation on the only one station – Server or PC:



Server



PC

Possible limitations for standalone license are:

* Absolute volume limitation – Total pages
* Performance limitation – Pages per month, CPU cores
* Usage Time

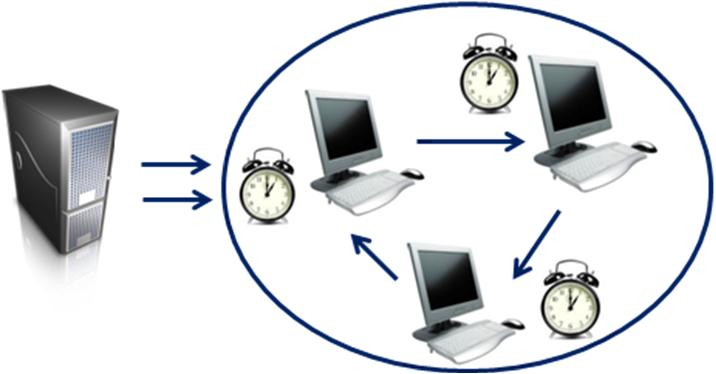
For example:

* Trial license – PPM=10K pages, time = 60 days
* Standalone desktop license (like FineReader) – PPM = 10K or 25K (time, cores - unlimited)
* Server license (e.g. BPO for 1 project) – total pages = 1M or cores=4, time= 60 days (PPM - unlimited)

**Network license**

Network licensed Product available for stations in the network and its usage controls by ***concurrent licensing*** scheme.

For example if we have 3 PCs in network and concurrent network license for 2 stations, we can use Product for all stations but only for 2 at the same time.



***Lease time*** represents the minimal time of using license for 1 station (in seconds). With this parameter we can simulate both variants of FRE network licensing – per seat and concurrent.

For example we can use these values of Lease time (T):

* T = 1 sec. => true concurrent license
* T>= 28800 sec.(8 hours) => per seat license, every license is tied to one PC for 8 or more hours
* 1<T<28800 => different middle variant

Possible limitations for network license are:

* Absolute volume limitation – Total page count (TPC)
* Performance limitation – Pages per month (PPM), CPU cores
* Usage time
* Lease time

**Proper activation for different license types**

|  |  |  |  |
| --- | --- | --- | --- |
| ***License Types*** | ***USB Key*** | ***Software*** | ***Open*** |
| **Standalone** | Preferable | Only if USB-key cannot be used | Only if other licenses cannot be used |
| **Network** | Preferable | Only if USB-key cannot be used | Not used |

#### Trial and Commercial Licenses

Developer and Runtime licenses could be one of two types – Trial or Commercial.

**Trial License**

We offer a full-functioned trial version of the Product for testing its usability.

Trial version limits:

* number of processed pages (10.000)
* usage time (60 days)

By default in trial license the most of functions are accessible and the others (e.g. FineReader XIX - old fonts’ recognition) could be added by special request from customer.

To receive the right of using the trial version, it is necessary to sign the Trial License Agreement (but in practice usually Non-Disclosure Agreement is enough).

The Trial License uses the software protection key (software license).

**Commercial License**

It is a paid License and its price depends on available features. Product is offered in some ***predefined license packages***:

Developer Licenses:

* Professional
* FineReader XIX
* Barcode/OMR\*

Runtime Licenses:

* ASCII
* Professional
* FineReader XIX
* Barcode/OMR\*
* Visual/Harmony/Business/VIP/Barcode

\*OMR is available in Windows version only.

And also could be generated ***custom*** developer and runtimelicenses with some special set of available features.

**Proper activation for different license types**

|  |  |  |  |
| --- | --- | --- | --- |
| ***License Types*** | ***USB Key*** | ***Software*** | ***Open*** |
| **Developer licenses** | Preferable | Only if USB-key cannot be used | Not used |
| **Runtime licenses** | Preferable | Only if USB-key cannot be used | Only if other licenses cannot be used |
| **Trial licenses** | Not used | Preferable | Not used |

### Non-functional License Limitations

#### Volume Limitation

Volume Limitation Licensing is intended to control the ***maximum number of characters or pages*** that can be processed by a license.

There are some possible variants of limitation:

* Absolute page count
  + Total Page Count (TPC)
  + FRXIX Page Count (FRXIXPC)
* Absolute characters count
  + Total Chars Count (TCC)
  + FRXIX Chars Count (FRXIXCC)
* Unlimited Pages

#### Time Limitation

Limitation of working period of License is mostly used in Trial Licenses and can be specified by the ***relative period of time*** – the number of days starting from the first launch of program and ***absolute period of time*** i.e. available till 31.05.2010.

The Commercial Licenses (both Hardware and Software) are usually ***time unlimited***.

#### Performance Limitations

**By pages (characters) for a period**

There are several possible variants:

* Renewable Page Count (RPC)
  + Pages per Month (PPM)
  + Pages per Year (PPY)
* Renewable Chars Count (RCC)
  + Charts per Second (CPS)

The most useful counter – Pages per Month (PPM).

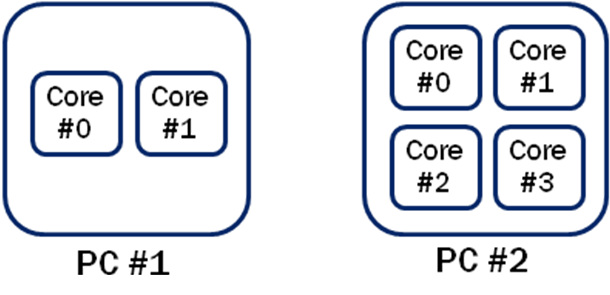
**By CPU Cores**

In case when the monthly (annual) volume is unpredictable or very changeable we can use another type of productivity control – assigning the ***maximum number of CPU cores (X)*** that can be used by processing stations during execution of resource-intensive operations (import, recognition or export).

The number X is specified as a sum of values for all processing stations of one license.

For example if we have 2 PCs:

* the first – of 2 CPU cores
* the second – of 4 CPU cores.



In this case we need to pay for 6 cores in order to use the whole system resources.

CPU cores limitations are implemented by CPU core vouchers distribution. ***Voucher*** is the right to use core that Licensing Server gives to FRE instance. It could be of 2 types:

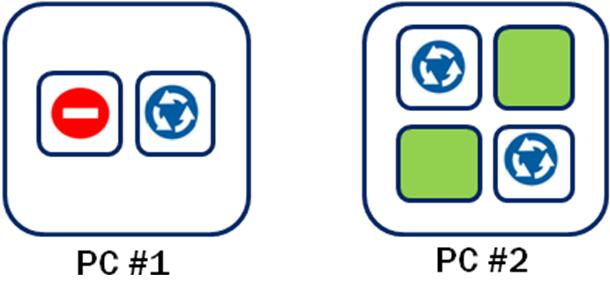
* 1-core voucher – if we use CPU cores limitation in license, customer pays for and we provide the precise number of vouchers, one for each core.
* Unlimited voucher – if number of cores is unlimited and we use other types of limitations in license.

Unlimited voucher allows any FRE instance using any cores of any available CPUs.

1-core voucher could be taken by FRE instance either ***exclusively*** or ***jointly***. *Exclusively* means that the only particular FRE instance can use this CPU core. In this case License server won’t give access to CPU core to other instances of FRE.

Voucher also could be taken jointlywith other instances of FRE. In this case License Server defines the ***mask*** that specifies which cores could be used.

Let us consider the same example (2 PCs, 6 cores, 6 vouchers for all cores) to illustrate the notions of exclusively and jointly.

****

* Core1 (PC#1, core#0) – occupied by FRE#1 instance exclusively
* Core2 (PC#1, core#1) – occupied by FRE#2 instance jointly
* Core3 (PC#2, core#0) – occupied by FRE#2 instance jointly
* Core4 (PC#2, core#1) – free
* Core5 (PC#2, core#2) – free
* Core6 (PC#2, core#3) – occupied by FRE#3 instance jointly

So 4 cores are used: 1 exclusively and 3 jointly.

For FRE#2 mask is (0, 1, 1, 0, 0, 0), i.e. it can use only Core2 and Core3

If new instance FRE#4 runs and asks Licensing Server for two CPU cores jointly, it has to specify what cores it wants, e.g. Core3 and Core4. So it will share Core3 with FRE#2, and receives a new voucher for Core4

If FRE#4 asks for 2 cores exclusively, so it receives voucher for available cores Core4 and Core5 that are free now.

Despite the difference between exclusive and joint vouchers, they conduct almost the same power, limited by number of CPU cores. If you have 6 vouchers for 6 cores you may either run 6 FRE instances exclusively (1 instance for each core) or many-many FRE instances jointly, but anyway the total performance will be almost the same.

### Functional License Limitations

Functional limitations defines if some functional modules available for customer or not.

Now we use the following basic packs:

* Developer
  + Professional
  + Barcode/OMR\*
  + Custom
* Runtime
  + Professional
  + Barcode/OMR\*
  + Custom
  + ASCII
* Trial
  + Developer Professional
  + Runtime Professional

For each type of license some of modules are included into basic functionality and always available, some – are not available at all, and the others could be appended as add-on modules.

General usage scenarios:

| ***License pack*** | ***Developer*** | ***Runtime*** | ***Trial*** | ***Description*** |
| --- | --- | --- | --- | --- |
| **Professional** | + | + | + | Basic license for common usage scenarios |
| **ASCII** | - | + | - | Special license for text extraction and document indexing. |
| **Barcode/OMR** | + | + | - | Special license for forms processing, using barcodes recognition and OMR. |
| **Custom** | + | + | - | Custom license based on Professional set where all modules could be added/removed. |

## Protection Key Types, Activation, Deactivation, Registration

### Supported Protection Types

1. Software protection keys:
   1. Out of a process (requires separate licensing service installed on a PC);
   2. In a process (requires no licensing service installed on a PC);
   3. Open (like b. and requires no activation, i.e. binding to a certain PC).
2. Hardware protection keys:
   1. iKey;
   2. Wibu (planned for a maintenance release).

### Supported Activation Types

A serial number is required to activate the product.

1. Software protection keys activation types:
   1. Via the Internet;
   2. By e-mail;
   3. By e-mail from another computer.
2. Hardware protection keys activation types:
   1. Pre-activation at an ABBYY local distribution office (using Dongle Update utility);
   2. On-site activation at a customer location (using License Manager).

The number of activations is limited (“Activation count” parameter). The limit is set up during license creation and can be modified manually afterwards. The activations counter is stored in the license database and can be checked by the manager.

### License Deactivation

This release supports deactivation of protection keys.

In order to deactivate license on one computer and activate it on another a customer should use License Manager to deactivate a software protection key.

Upon deactivation, the activation counter is decremented for the deactivated license.

The number of deactivations is also limited (“Deactivation count” parameter).

## License Limitations

### Page Counter

Any “Out of a process” (see “Supported Protection Types”) may have a limit for recognition volume to process (pages or characters, total or renewable count).

Engine treats a page as processed upon a call of the following methods:

* IFRDocument::
  + Analyze
  + AnalyzePages
  + Process
  + ProcessPages
  + Recognize
  + RecognizePages
* IFRPage::
  + Analyze
  + AnalyzeRegion
  + AnalyzeTable
  + PreprocessAnalyzeRecognize
  + Recognize
  + RecognizeBlocks
  + ExtractBarcodes
* IDocumentAnalyzer::
  + AnalyzePage
  + AnalyzePagesEx
  + AnalyzeRegion
  + AnalyzeTable
  + PreprocessAnalyzeRecognizePage
  + PreprocessAnalyzeRecognizePagesEx
  + RecognizePage
  + RecognizePagesEx
  + RecognizeBlocks
  + RecognizeImageAsPlainText
  + RecognizeImageDocumentAsPlainText
  + ExtractBarcodes
* IEngine::
  + RecognizeImageFile
  + ProcessPage
  + ProcessPagesEx

A page counter is increased by one only once for the same ImageDocument object regardless to how many times one uses analysis, recognition, or exporting methods with that object.

In case of character counter, the following methods have no effect:

* IFRDocument::
  + Analyze
  + AnalyzePages
* IFRPage::
  + Analyze
  + AnalyzeRegion
  + AnalyzeTable
* IDocumentAnalyzer::
  + AnalyzePage
  + AnalyzeRegion
  + AnalyzeTable

## Installation

There are two types of the product installations:

* Developer. Used for software development. To set up the product one should run “Setup.exe” from distribution CD (or its copy).
* Runtime. Used for developed software distribution. To set up runtime copy of the product one should copy required files and register certain libraries or run “Setup.exe” in command line mode with proper options.

Please read the Administrator’s Guide document distributed with the DVD for details.

## Supported Languages, Types and Formats

### Supported Recognition Languages

#### OCR Languages

|  |  |  |  |
| --- | --- | --- | --- |
|  | With dictionaries | Without dictionaries | Overall count |
| Common (default) languages | **40** | **145**, including   * **4** artificial languages   + Esperanto   + Ido   + Interlingua   + Occidental * **2** special languages (included by default if corresponding text type is chosen)   + CMC7   + E13B * **6** programming languages * Chemistry * Digits | **185**, included in Runtime Professional |
| Additional languages | **12**, including:   * Arabic * Japanese * Hebrew * Korean * Korean Hangul * Thai * Vietnamese * FR XIX   + Old English   + Old French   + Old German   + Old Italian   + Old Spanish | **5** additional languages   * Chinese Simplified (PRC) * Chinese Traditional (Taiwan) * Yiddish (under Hebrew Add-On) * FR XIX   + Old Slavonic   + Latvian Gothic | **17**, included in Add-Ons |
| Total | **52** | **150** | **202** |

#### ICR languages

|  |  |  |
| --- | --- | --- |
| With dictionaries | Without dictionaries | Overall count |
| **38**, new are:   * Dutch (Netherlands) * Norwegian (Bokmal) * Norwegian (Nynorsk) * Portuguese (Brazil) * Portuguese (Portugal) * Swedish * Old English * Azerbaijani (Latin) * Latin | **88**, new are   * Serbian (Cyrillic) * Tajik * Turkmen (Latin) | **126**,   * **121** are included in Data Capture add-on in Runtime Professional license. * **5** are additionally included in Data Capture add-on in Runtime FineReader XIX license. |

#### Classification languages

FineReader Engine supports all languages for classification.

#### BCR languages

**27** languages including:

* **4** hieroglyphic languages (CJK) - Chinese Traditional (Taiwan), Chinese Simplified (PRC), Japanese, Korean

### Supported barcode types

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1D Barcodes | | | 2D Barcodes | Overall count |
| **15**:   * Codabar * Code 128 * Code 39 * Code 93 * EAN 8 * EAN 13 * IATA 2 of 5 * Industrial 2 of 5 * Interleaved 2 of 5 * Matrix 2 of 5 * Patch * PostNet * UCC-128 * UPC-A * UPC-E | Including 4 with checksum:   * Code 39 * Interleaved 2 of 5 * Codabar * Matrix 2 of 5 | Including 4 with supplemental:   * EAN 8 * EAN 13 * UPC-A * UPC-E | **5**:   * PDF417 * Aztec * DataMatrix * QR Code * MaxiCode (NEW) | **20** |

### Supported text types and field marking types

#### OCR text types

1. Normal
2. Fax
3. Typewriter
4. Matrix
5. OCR\_A
6. OCR\_B
7. MICR\_E13B
8. MICR\_CMC7
9. Fraktur/Gothic (available only under FineReader XIX add-on)
10. Receipt (NEW)

#### Field marking types

1. Simple Text
2. Underlined Text
3. Text in Frame
4. Grey Boxes
5. Char Box Series
6. Simple Comb
7. Comb in Frame
8. Partitioned Frame

### Supported import and export formats

#### Supported import formats

|  |  |  |
| --- | --- | --- |
| Format | Open | Save |
| BMP: uncompressed black and white 4- and 8-bit — uncompressed Palette 16-bit — uncompressed, uncompressed Mask 24-bit — uncompressed 32-bit — uncompressed, uncompressed Mask | + | + |
| BMP: 4- and 8-bit — RLE compressed Palette | + |  |
| DCX: black and white 2-, 4- and 8-bit palette 24-bit color | + | + |
| DjVu: black and white, gray, color | + |  |
| GIF: black and white — LZW-compressed 2-, 3-, 4-, 5-, 6-, 7-, 8-bit palette — LZW-compressed | **+** |  |
| JBIG2: black and white | + | + |
| JPEG:  gray, color | + | + |
| JPEG 2000:  gray — Part 1 color — Part 1 | + | + |
| PCX: black and white 2-, 4- and 8-bit palette 24-bit color | + | + |
| PDF (version 1.7 or earlier) | + | + |
| PNG: black and white, gray, color | + | + |
| TIFF: black and white — uncompressed, CCITT3, CCITT4, Packbits, ZIP, LZW gray — uncompressed, Packbits, JPEG, ZIP, LZW 24-bit color — uncompressed, JPEG, ZIP, LZW 1-, 4-, 8-bit palette — uncompressed, Packbits, ZIP, LZW (including multi-page TIFF) | + | + |
| TIFF: black and white — CCITT3FAX | + |  |
| WDP: black and white, gray, color (WIC or Microsoft .NET Framework 3.0 required) | + |  |
| WIC-compatible (WIC or Microsoft .NET Framework 3.0 required) | + |  |

#### Supported export formats

1. RTF
2. Microsoft Office file formats:
   1. DOCX
   2. XLS/XLSX
   3. PPTX
3. PDF file formats
   1. PDF
   2. PDF/A (1b, 1a, and NEW 2a and 2u)
   3. MRC (Mixed Raster Content) for both PDF and PDF/A
4. HTML
5. TXT/CSV
6. ABBYY XML
7. EPUB, ALTO, FB2
8. ODT
9. vCard — for export of business cards only (NEW)

## GetEngineObject function changes

The functions and methods, which load the Engine object, have their syntax changed:

The **GetEngineObject** function and **IEngineLoader::GetEngineObject** method do no longer have the parameters for **Open License**. To use Open License, one should use the **GetEngineObjectEx** function or **IEngineLoader::GetEngineObjectEx** method.

The **GetEngineObjectEx** function and **IEngineLoader::GetEngineObjectEx** method have one more additional parameter, which specifies whether CPU cores should be used in shared mode.

## Full native 64-bit support

FineReader Engine 11 provides both 32-bit and 64-bit versions of libraries, including libraries for classification and BCR. Both 32-bit and 64-bit versions can be installed during Developer installation on 64-bit operating systems. Details on the distribution package can be found in the Help file.

There is a known issue with **64-bit Visual Studio designer**: you cannot add Visual components to a Windows Form application in Visual Studio if you have only the 64-bit version of Visual components installed (a similar issue is described here: <http://support.microsoft.com/kb/980533>). This means that in order to use Visual Components in designer, 32-bit Visual Components should be installed too. Therefore, if a user selects 64-bit Visual Components for installation, both 64-bit and 32-bit components are installed. It is not necessary for compiled applications, which use Visual Components.

## Classification

Classification is a new feature in FineReader Engine 11. You can find the detailed description of the feature in marketing materials of the product. Below are several implementation details.

### Classification modes

Classification in FineReader Engine 11 can be performed in two modes:

* **Fast**. This mode is useful for documents that contain not much text, and the difference between classes is visible in the appearance of the documents. This mode uses image pattern (black pixels location template) and recognized titles of a document for classification.
* **Quality**.This mode is useful for documents that contain a lot of text, and the difference between classes can be determined only when text content is taken into account. This mode uses full text OCR for classification.

The mode is specified in I**ClassificationParams::ClassificationMode** property.

### Classification confidence

The resutls of classification provide information both on the detected category of the document and the confidence, probability that a document belongs to this category (**IClassificationClass::Confidence**). There is also a flag (**IClassificationClasses::IsSuspicious**) indicating whether classification of a document was uncertain, e.g. if classification detected two classes with equal confidence for one document. One may use these values to determine the way of further processing for the classified documents, for example, re-classify some of the documents manually, if necessary.

## FlexiFormsDA and FullTextIndexDA options in API and licensing

Starting from FREngine 11, **DA for Invoices** and **DA for Full-text Indexing** add-ons are free of charge and removed from licensing scheme. This is because of the following reasons:

* “TextExtraction” and “DocumentArchiving” predefined profiles require these add-ons. That is because the add-on modules were initially designed for those usage scenarios and therefore were included into corresponding profiles. Without add-ons Engine performs not optimally in those scenarios.
* Indexing and text extraction years ago were ‘additional’ usage scenarios for FineReader Engine while ‘document conversion’ was main one. It seems that all three scenarios are basic now for Engine.

Changes in licensing required additional changes in API:

* We declare the **FlexiFormsDA** and **FillTextIndexDA** properties as obsolete. These properties will be removed in the next version of the product.
* The old FlexiFormsDA property is replaced with **IPageAnalysisParams::EnableTextExtractionMode** and **IObjectsExtractionParams::EnableAggressiveTextExtraction** properties.
* The old FullTextIndexDA property is replaced with **IObjectsExtractionParams::DetectTextOnPictures** property.

## Installation on a virtual machine and activation

A license that does not have virtual machine support now cannot be activated on a virtual machine. The reason is that there are many customers installing and activating, then develop or wait for a while, and run their application later and only at this moment find that the license does not work on a virtual machine. It’s very late into the game, and the customer requires this working urgently.

Therefore, we’ve done the following changes in the installation and activation procedures:

* When installing on a virtual machine, installer displays a warning that a license with virtual machine support is required for programm operation.
* A license without virtual machine support cannot be activated on a virtual machine. Activation server returns an error message on activation request.

## CJK doesn’t Use User Patterns

CJK recognizer does not use cache so it is impossible to teach it with user patterns. The same situation is in ABBYY FineReader Engine 10.

## Software and Hardware Requirements

### ABBYY FineReader Engine 11 Requirements

* PC with x86-compatible processor (1 GHz or higher)
* Operating system:
  + Windows Server 2012 (64-bit)
  + Windows 8 (32-bit and 64-bit)
  + Windows Server 2008 R2 (64-bit)
  + Windows 7 (32-bit and 64-bit)
  + Windows Server 2008 SP1-SP2 (32-bit and 64-bit)
  + Windows Vista SP1-SP2 (32-bit and 64-bit)
  + Windows Server 2003 SP1-SP2, R2 (32-bit and 64-bit)
  + Windows XP SP1-SP3 (32-bit and 64-bit)

ABBYY FineReader Engine has been tested on the following cloud computing platforms:

* + Windows Azure
  + Amazon EC2

ABBYY FineReader Engine has been tested in the following virtual environments:

* + Microsoft Virtual PC
  + Microsoft Hyper-V (only with software protection key)
  + Oracle VM VirtualBox 3, 4
  + Parallels Desktop 4
  + Parallels Virtuozzo Containers 4
  + VMware Server 2
  + VMware Workstation 7, 8
  + VMware Player 3
  + VMware ESXi 5
* Memory:
  + for processing one-page documents — minimum 400 MB RAM, recommended 1 GB RAM
  + for processing multi-page documents — minimum 1 GB RAM, recommended 1,5 GB RAM
* Hard disk space: 800 MB for library installation and 100 MB for program operation plus additional 15Mb for every processing page of a multi-page document
* 100% TWAIN-compatible scanner, digital camera, or fax modem — for scanning or image import only
* Video card and monitor (min. resolution 1024\*768 — for pattern training, dictionary editing, scanning with a GUI displayed, Visual Components)
* Keyboard, mouse or other input device
* The following registry branches should be accessible from the workstation:
  + "HKEY\_CURRENT\_USER\Software\ABBYY\SDK\11\FineReader Engine" — full control
  + "HKEY\_CURRENT\_USER\Software\ABBYY\SDK\11" — full control for installation only
  + "HKEY\_LOCAL\_MACHINE\Software\ABBYY\SDK\11" — full control for installation only
* The following folders should be accessible from the workstation:
  + Folder with ABBYY FineReader Engine binary files — access for reading
  + System temporary folder — full control access
  + For Windows XP, Windows Server 2003:
    - folder Documents and Settings\All Users\Application Data\ABBYY\SDK\11\FineReader Engine — full control access;
    - folder Documents and Settings\All Users\Application Data\ABBYY\SDK\11\Licenses — full control access
  + For Windows Vista, Windows Server 2008, Windows 7, Windows 8, Windows Server 2012:
    - folder ProgramData\ABBYY\SDK\11\FineReader Engine — full control access;
    - folder ProgramData\ABBYY\SDK\11\Licenses — full control access
* The following components should be installed:
  + Microsoft Internet Explorer 5.0 or higher
  + If your application uses pattern training, dictionary editing, scanning with a GUI displayed, Microsoft Windows Common Controls must have version 5.80 or later and RichEdit Control must have version 3.0 or later
* For ABBYY FineReader Engine Visual Components only:
  + Microsoft Windows Common Controls must have version 6.0 or later.
  + If you use Microsoft Visual Studio 2010 and .NET Framework 4.0 for development of your application, you may need to install COM Interop assemblies for Visual Components manually. Refer to the Developers's Help for details.

### ABBYY SDK 11 License Server Requirements

* PC with x86-compatible processor (1 GHz or higher)
* Operating system:
  + Windows Server 2012 (64-bit)
  + Windows 8 (32-bit and 64-bit)
  + Windows Server 2008 R2 (64-bit)
  + Windows 7 (32-bit and 64-bit)
  + Windows Server 2008 SP1-SP2 (32-bit and 64-bit)
  + Windows Vista SP1-SP2 (32-bit and 64-bit)
  + Windows Server 2003 SP1-SP2, R2 (32-bit and 64-bit)
  + Windows XP SP1-SP3 (32-bit and 64-bit)

ABBYY SDK 11 License Server has been tested in the following virtual environments:

* + Microsoft Virtual PC
  + Microsoft Hyper-V (only with software protection key)
  + Oracle VM VirtualBox 3, 4
  + Parallels Desktop 4
  + Parallels Virtuozzo Containers 4
  + VMware Server 2
  + VMware Workstation 7, 8
  + VMware Player 3
  + VMware ESXi 5
* 25 MB of free hard-disk space
* For Windows XP, Windows Server 2003:
  + folder Documents and Settings\All Users\Application Data\ABBYY\SDK\11\Licenses — full control access
* For Windows Vista, Windows Server 2008, Windows 7, Windows 8, Windows Server 2012:
  + folder ProgramData\ABBYY\SDK\11\Licenses — full control access

## New Features and Improvements

### Pattern training for Vietnamese language

This release includes ability to train user patterns for Vietnamese language.

Pattern training helps in cases when an image quality is not sufficient or the Engine works with a document font not good enough. In this case one can teach the Engine’s raster classifier with characters’ images as they are presented in document images. After that recognition accuracy should become higher.

### Support for Russian with Accents language

Russian with Accents is an ordinary Russian language where tone stress is written under a character, sort of an alternative way of writing a character. This alternative writing is supported in FRE 11 as a separate language: RussianWithAccent.

### Support for Russian Old Spelling language

FRE 11 supports OCR in Russian language used before the Russian Revolution in 1917. This language includes few outdated characters and slightly different morphology.

Language alphabet:

|  |
| --- |
| -.ЁІАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯабвгдежзийклмнопрстуфхцчшщъыьэюяёіѢѣѲѳѴѵ |

### Support for FullAscii and Code32 1D barcodes

The following new barcode types are supported in this release:

|  |  |
| --- | --- |
| *Barcode Type* | *Description* |
| Code 32 | Code 32 is a variable length self-checking bar code. It is a variant version of Code 39, with the characters that may be encoded limited to digits and capital letters excluding the vowels ('A', 'E', 'I', 'O', 'U'). It is used in Italian medication packaging and is also known as Italian Pharmacode. |
| Full ASCII Code 39 | This is an extended version of Code 39. It is used to encode all 128 ASCII characters by using pairs of Code 39 characters to represent the lowercase ASCII characters not in the Code 39 character set. |

Both types are supported in automatic barcode location and type detection.

### Backward compatibility with prior versions, new Help article

To assist existing customer in migrating to FRE 11 from older versions the product documentation has been enhanced. The Help includes new compatibility articles in addition to the standard article about compatibility with the previous version:

* ABBYY FineReader Engine 11 and 9.0/9.5 Compatibility
* ABBYY FineReader Engine Visual Components 11 and 9.0/9.5 Compatibility
* ABBYY FineReader Engine 11 and 8.0/8.1/8.5 Compatibility

### Re-formatted Distribution.csv file for easier runtime file list composing

Well known Distribution.csv file got new structure bringing ease into a process of runtime distribution file list composing. It is also suitable for making automated (script) procedures.

The file contains the following data (columns):

1. Stage — the stage of working with FineReader Engine which your application uses.
2. Part — the way in which you are going to use this stage. For example, the Opening stage includes Scanning and Pdf parts. If this field is empty, the file is needed for the working stage in general. Do not filter the blank values out.
3. Details — further specific information about the operations in which the file is used.
4. x64/x86 — the operating system architecture. Again, the files marked "x64,x86" are necessary for both.
5. RequiredByModule — the values in this column are equal to Stage.Part.Details, and there is no need to filter this column if the first three have been specified correctly. But it can be used to check which modules have been included.
6. RequiredByInterfaceLanguage — the interface language for which the file is necessary. The files marked "Any" are necessary independent of language settings.
7. RequiredByRecognitionLanguage — the recognition language for working with which the file is necessary. The files marked "Any" are necessary independent of recognition language.
8. Optional — specifies if the file is necessary for the module functionality. If the value is No, this file must be included in your distribution kit. The value can be set to Yes in the following cases:
   1. the file is language-specific. Include it if you need this language (consult columns 6 and 7).
   2. the functionality for which this file is responsible is not always necessary. For example, it can be used for opening images in a specific format. Consult ABBYY FineReader Engine Distribution Kit for further information about this file which will help you decide if you need it.

Finally, you receive the list of files. In the last three columns you will find the information about these files' location and size:

1. Path — file path in the root installation folder.
2. FileName — file name.
3. Size — file size in bytes.

### New export formats are supported in CLI sample: ALTO, EBook

CLI sample was updated to support additional export formats: ALTO and EBook.

### JBIG2 lossless compression is supported

This release includes new compression format for black and white images – JBIG 2 lossless.

It is useful if somebody works with low quality b/w images where lossy format may lead to similar characters substitution, e.g. “6” to “8” and vice versa.

This format is available for choosing in PDFPictureCompressionParams::BwPictureFormats.

### Java interop throws more informative errors

From now on in a case an error occurs in Engine Java interop will return com.abbyy.FREngine.EngineException object inheritor of java.lang.Exception. This object exposes getHResult() method returning a meaningful error number.

Nothing is changed for existing customers; they can use legacy code as is. But now anybody can get additional error number and provide it to ABBYY technical support for investigation.

Sample code:

|  |
| --- |
| try {  …  } catch( Exception ex ) {  displayMessage( "Message = " + ex.getMessage() );               displayMessage( "HResult = " + ( ( EngineException )ex ).getHResult() );  } |

### Improvements in MRC visual quality

It is possible to use black and white image prepared during binarization process as a source for MRC mask. In this case all contrast graphical elements will be saved into PDF MRC file with the highest quality.

This technique allows saving handwritten data, stamp details, signatures, etc. in a foreground layer with high quality (low compression) and keep a document look as close to original as possible. Previously enumerated objects were placed into background layer with high compression.

The following PDF export parameter enables the feature: PDFMRCParams::UseBwImageAsTextMask.

### Method for checking if a page is empty

FRPage::IsEmpty() method checks if the page is empty. It uses the same parameters as analysis methods to find out if the page contains any relevant objects, for example text, tables, or pictures.

If there is a high probability that some of your pages contain only barcodes, and they are relevant for your procedure, set the boolean NeedCheckBarcodes parameter to TRUE to help the method detect barcode-only pages.

This method is useful in a batch scanning scenario when there is a need to separate scanned images flow into documents. In this scenario a batch contains separating paper sheets which are blank or with certain barcodes printed on them.

### New OnBlockAdded callback for ImageViewer and ZoomViewer interfaces

OnBlockAdded method is implemented on the client side. It is called by ABBYY FineReader Engine after a new block has been added in Image Viewer or Zoom Viewer.

It delivers to the client the index of the newly added block.

### Aggressive table detection mode

PageAnalysisParams::AggressiveTableDetection property manages the table detection mode. If you set it to TRUE, FineReader Engine tries to find as many tables as possible on the page. This setting is recommended only for the documents which contain a lot of tables.

This property is FALSE by default.

### Predefined processing profile for writing highly compressed image-only PDFs

The Engine got new processing profile “*HighCompressedImageOnlyPdf*” for creating highly compressed image-only PDF files.

It is suitable for creating high-compressed PDF files which contain entire documents saved as pictures. The following settings are used:

* Document recognition and synthesis of the logical structure of a document are not performed.
* Skew correction is not performed.
* PDF export is optimized for the minimum size of the resulting file.
* The entire document is saved as a picture (PEM\_ImageOnly mode).

### XPS export format

Starting from this release FRE supports exporting to XPS format.

This format is developed by Microsoft in contrast to PDF format ([see Wiki](http://en.wikipedia.org/wiki/Open_XML_Paper_Specification)).

This is optional export format and to enable it one should enable this format in a license.

### Retaining document layout in textual export

TextExportParams::RetainLayout property turns on the export mode in which the original layout is simulated by inserting spaces. When displayed with a monospace font, the text and table columns will be level. The distance between blocks will be approximately the same.

This mode is not intended for right-to-left or vertical texts.

When this property is set to TRUE, the InsertEmptyLineBetweenParagraphs property is ignored, and the presence of empty lines is determined by the size of empty space between paragraphs in the original. The ExportParagraphsAsOneLine property is also ignored, and the line breaks are always kept.

This property is FALSE by default.

## Fixed Bugs

A list of bugs reported by customers that have been fixed:

|  |  |  |
| --- | --- | --- |
| *Office* | *HD Case* | *Description* |
| EU | 371886 | IPE: “.\Src\ReadingOrderFinder.cpp, 52”, - during synthesis of an image from a customer. |
| 352045 | Left column with paragraph numbers is lost during DA. |
| 372316 | IPE: “.\Src\TextWriter.cpp, 462”, - during recognition of an image from a customer. |
| 371889 | IPE: “.\Src\DocumentModelGenerator.cpp, 125”, - during analysis of an image from a customer. |
| 371885 | Error: “Stack overflow. A new guard page for the stack cannot be created.”, - during analysis of an image from a customer. |
| US | 369068 | Opening of XFA forms in PDF files is malfunction. |
| 366788 | Colors are not detected on a photo. |
| 341888 | Option of writing linearized PDF is missing. |
| 366788 | IPE: “d:\build\13.0.trunk\0\synthesis\src\pagesectonseriesfinder.cpp, 66”, - during synthesis of an image from a customer. |
| 359925 | Incorrect table structure analysis if a table has white separators. |

## Known Issues and Workarounds

### CLI sample does not allow exporting if a license lacks of any exporting module

If a license lacks of any exporting module then it is not possible to use CLI sample for saving recognition result.

This will be fixed in the maintenance release.

As an immediate patch one can use the following source code files instead of those provided in the distribution:



### VB6 samples are not compatible with 64-bit version of the Engine

VB6 programming language has 32-bit nature and it is not possible to write native 64-bit applications using it. Thus it is not possible to reference to 64-bit version of the Engine from VB6 code.

If one does not install 32-bit version of the Engine VB6 samples are still installed and are not functional.

Installer of the next FRE 11 release will be not installing VB6 samples if only 64-bit Engine version is installed.

### IFontSet::EnablePdfStandardFonts is non-functional

It is not possible to use the standard PDF fonts during synthesis. That may lead to font embedding even though the text is typed using one of the standard PDF fonts.

This will be fixed in the maintenance release.

### FRE installer fails on localized Chinese Windows 7 x64

It is not possible to install FRE (developer installation) on Windows 7 x64 with Chinese locale.

This is planned to be fixed in the maintenance release.

### OnPageProcessed comes once per a document

During opening, synthesis, exporting stages the callback OnPageProcessed arrives only once per a document rather than after each page.

This will be fixed in the maintenance release.

### Some API is not implemented

The following API is not implemented in FRE 10 and FRE 11:

* IFootnoteSeries::IsNumberingWithSuperscript. Always returns “false”.
* IFootnoteSeries::PositionOnPage. Always returns “FPPT\_SingleColumnSection”.
* IFootnoteSeries::PositionInDocument. Always returns “FPDT\_PageEnd”.
* IFootnoteSeries::HasSeparator. Always returns “true”.
* ITextPicture::ColumnNumber. Always returns “0”.
* ICharParams::IsWordStart. Always returns “false”. It is true only for character parameters got through IWordRecognitionVariants interface.
* IIncut::TextWrapping. Always returns “TW\_Undefined”.
* IRunningTitlesSeriesText::HasSeparator. Always returns “false”.

The implementation is not planned.

### Initialization of APDFL library fails if a path to the contains hieroglyphs

It is not possible to open PDF files if FRE (and APDFL library) are placed into a folder with a path containing Chinese hieroglyphs.

This will be fixed in the maintenance release.

### Recognition of a PDF file with text in Vietnamese may fail

Recognition fails with internal program error on certain PDF files with text layer in Vietnamese when “content reuse mode” is on.

This will be fixed in the maintenance release.

### Long font names in MS Word

If a user uses fonts from non-system folder then exported MS Word document has long font names. These names are visible in MS Word GUI in the font choosing combo.

This will be fixed in the maintenance release.

### Out of memory exception during export very large image into PDF using LZW compression

LZW codec allocates too much memory during exporting large images (construction plans, maps, etc.) into PDF file and processing ends up with “out of memory exception”.

This issue is valid for 32-bit Engine version only. 64-bit can address more memory and the issue was not reproduced.

This will be fixed in the maintenance release.

### PDF/A validation report

The following issues are known for PDF/A files produced by this release of FRE 11:

1. Adobe Acrobat 11.0.3 reports “Text cannot be mapped to Unicode” for 2% of images in CJK languages recognized and exported into PDF/A-1a or PDF/A-2a formats. On the other hand <http://www.pdf-tools.com/pdf/validate-pdfa-online.aspx> on-line validator finds no issue in the same documents.
2. callas pdfaPilot, 3.1 (156) and 4 report “Image is not valid” for few images exported into PDF/A-2a (-2u) format. At the same time Adobe Acrobat 10.1.4, 11 report no issues with these files.

# R2 GM –

## Part number, build number

|  |  |
| --- | --- |
| Part# | 1041/20 |
| Build# | 11.1.5.104 |

## New Features and Improvements

### New samples for Java

New samples are available for Java:

• BatchProcessing

This sample shows how to use Batch Processor for processing a large amount of one-page documents.

• EventsHandling

The sample illustrates the use of the callback interfaces using the FRDocument callback interface (IFRDocumentEvents) as an example. The sample shows the progress of recognition and export during image processing. You can use the callback interfaces to control image processing.

• EnginesPool

This sample on the one hand provides a complete reusable solution for a pool of Engines in a multithreaded application, and on the other hand demonstrates the gain in speed when using multiprocessing.

The samples have the same scenarios as the samples with the corresponding title provided in C#.

### Improved Java wrapper

In previous release Java wrapper had some limitations. Now the complete version of JAVA wrapper is implemented, which means that entire SDK functionality is available through it. Now FRE can be easily integrated into Java environment.

New features are supported:

1. Inheritance of interfaces

If interfaces are inherited in IDL, the corresponding wrapper interfaces are also inherited (For example, ITextBlock and IBlock)

1. Methods that load data from memory( e.g. IFRDocument.AddImageFileFromMemory)
2. Interfaces which are implemented on the client side (callbacks and events) :
   1. The [IExternalDictionary](mk:@MSITStore:C:\Program%20Files\ABBYY%20SDK\11\FineReader%20Engine\Help\FREngine11.chm::/APIReference/IExternalDictionary.htm) interface
   2. The [IAsyncProcessingCallback](mk:@MSITStore:C:\Program%20Files\ABBYY%20SDK\11\FineReader%20Engine\Help\FREngine11.chm::/APIReference/IAsyncProcessingCallback.htm), [IImageSource](mk:@MSITStore:C:\Program%20Files\ABBYY%20SDK\11\FineReader%20Engine\Help\FREngine11.chm::/APIReference/IImageSource.htm), [IFileAdapter](mk:@MSITStore:C:\Program%20Files\ABBYY%20SDK\11\FineReader%20Engine\Help\FREngine11.chm::/APIReference/IFileAdapter.htm) interfaces
   3. The [IDocumentAnalyzerEvents](mk:@MSITStore:C:\Program%20Files\ABBYY%20SDK\11\FineReader%20Engine\Help\FREngine11.chm::/APIReference/IDocumentAnalyzerEvents.htm) interface
   4. The [IExporterEvents](mk:@MSITStore:C:\Program%20Files\ABBYY%20SDK\11\FineReader%20Engine\Help\FREngine11.chm::/APIReference/IExporterEvents.htm) interface
   5. The [IFRDocumentEvents](mk:@MSITStore:C:\Program%20Files\ABBYY%20SDK\11\FineReader%20Engine\Help\FREngine11.chm::/APIReference/IFRDocumentEvents.htm) interface
   6. The [IFRPageEvents](mk:@MSITStore:C:\Program%20Files\ABBYY%20SDK\11\FineReader%20Engine\Help\FREngine11.chm::/APIReference/IFRPageEvents.htm) interface
   7. The [IFRPagesEvents](mk:@MSITStore:C:\Program%20Files\ABBYY%20SDK\11\FineReader%20Engine\Help\FREngine11.chm::/APIReference/IFRPagesEvents.htm) interface
   8. The [IImageDocumentEvents](mk:@MSITStore:C:\Program%20Files\ABBYY%20SDK\11\FineReader%20Engine\Help\FREngine11.chm::/APIReference/IImageDocumentEvents.htm) interface
   9. The [IImagePasswordCallback](mk:@MSITStore:C:\Program%20Files\ABBYY%20SDK\11\FineReader%20Engine\Help\FREngine11.chm::/APIReference/IImagePasswordCallback.htm) interface
   10. The [IScanCallback](mk:@MSITStore:C:\Program%20Files\ABBYY%20SDK\11\FineReader%20Engine\Help\FREngine11.chm::/APIReference/IScanCallback.htm) interface
3. Methods with out-parameters (For example, IPlainText::GetCharacterData method)

### IEngine::InjectTextLayer method was added

This is a new method which creates a copy of the input PDF file and adds the text layer which corresponds to the recognized text of the document.

### Support for IntelligentMail barcodes

Starting with this release, ABBYY FineReader Engine can recognize ABBYY FineReader Engine recognizes IntelligentMail barcodes. Intelligent Mail is a height-modulated 65-bar barcode which is used on mail in the United States. It is also known as USPS 4-CB.

BT\_ IntelligentMail was added to BarcodeTypeEnum enumeration constants.

### Predefined languages OcrA and OcrB were added

The OCR A and OCR B modules now license not only corresponding text types but also the predefined languages which are provided for these text types. Apart from regular characters the alphabet for these languages also contains special symbols from OCR\_A and OCR\_B standards.

### New special predefined languages for phone numbers and EIN

English\_US\_Address\_PhoneNumber and English\_US\_WellKnownCode\_EIN special languages were added.

English\_US\_Address\_PhoneNumber contains phone numbers (English, United States).

English\_US\_WellKnownCode\_EIN contains Employer Identification Numbers (English, United States).

### Detection of vertical text for all languages

IPageAnalysisParams::DetectVerticalEuropeanText property is a new property which allows you to detect vertical text in languages other than CJK (helpdesk request 369221).

### New IClassificationTrainer::AddPage method

From now on FRE 11 supports IClassificationTrainer::AddPage method.

This is a new method which allows you to add an already recognized page to the classification database.

### The possibility to add custom features for classification

Now it is possible to add custom features that can enhance the accuracy of classification.

The following methods were implemented:

1. IClassificationTrainer::AddFeaturesForPage

This is a new method which supports training the classification database with the help of user-defined additional features.

2. IFRPage::ClassifyEx

This is a new method which supports classification with the help of user-defined additional features.

### New predefined profile EngineeringDrawingsProcessing

This profile is specially developed for engineering drawings and maps recognition.

EngineeringDrawingsProcessing would be useful for text extraction disregarding its orientation and conversion of engineering drawings from PDF to searchable PDF.

### Export with original layout to XLSX

New property IXLExportParams::LayoutRetentionMode that allows to export files to XLSX with original layout was added.

It is possible to choose the mode of retaining tables’ formatting during export to XLSX. XLSXLayoutRetentionModeEnum gives the ability to choose the mode that is better corresponding to the assigned task.

### New properties for IBarcodeParams object

This release includes a new property which allows you to detect more barcodes but slows down the processing. IBarcodeParams::EnableAdvancedExtractionMode property enables the feature.

New IBarcodeParams::MinRatioToTextHeight property defines the minimal acceptable height of the barcode in relation to the average letters height. This setting can help to detect low barcodes.

### Support for embedded files in PDF

IPDFExportFeatures::WriteSourceAttachments property specifies if attachments from the original PDF file should be written to the output file. This property is ignored if the original file was not PDF.

New values PCM\_Pdfa\_3a and PCM\_Pdfa\_3u were added to enumeration constants PDFAComplianceModeEnum for PDF/A-3a and PDF/A-3u formats correspondingly .

### Enhanced JBIG2 lossless compression

The technology JBIG2 lossless compression format for black and white images was significantly improved in this release.

### New property IEngine::Version

IEngine::Version returns the build number of the ABBYY FineReader Engine version you are using.

### Version for FREngine.jni.dll was added

From this release FREngine.jni.dll version was added. It coincides with the version of FRE you are using.

### Screenshot detection

From now on our technologies are able to identify screenshots in documents. The document analyzer detects screenshots as image blocks if EnableTextExtractionMode property set to false (it is default value). Otherwise, text blocks can be detected on the screenshots. (helpdesk request 343728 )

### Launch of AInfo utility from the command line

It is possible now to call AInfo utility that is used to gather information for technical support from the command line in silent mode. (helpdesk request 388454)

### Improved key value tables detection

Significant improvements were made for key value tables’ detection in this release.

### Serbian Cyrillic ICR support

From now on Serbian Cyrillic is supported. Earlier It was not included in the distributive due to internal limitations.

### Improved memory allocation

The developers have successfully managed to significantly reduce memory fragmentation. Thus, the error OUT\_OF\_MEMORY was fixed in many cases.

## Fixed Bugs

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

The following terminology is used for frequency in the list:

General bugs are frequently reproduced.

Rare bugs are seldom reproduced on certain conditions.

Specific bugs are reproduced in custom images in very specific scenarios.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Frequency | Description | Subsystem | HD # | Office |
| General | The Arabic symbols ٢ and ٣ were often mixed up. The Classifiers for these characters were updated improved. | Recognizer | 397942 | RU |
| Temporary files were recorded in the folder %ALLUSERSPROFILE%\Application Data\ABBYY\SDK\11\Licenses and were not deleted. Now all temporary files are deleted after they become unnecessary. | Protection | 391063 | 3A |
| As the INSTALLDIR option is marked as required it shouldn’t have default values. From now on INSTALLDIR option has no default values and users have to specify the path to the installation directory. | Install | 401453 | EU |
| Internal Program Error after two successive runtime installations. | Install | 394275 | EU |
| Problem with ASCII RTL. The output file with recognition results was empty. | API | 393613 | 3A |
| There was a change in the section of Help file: after the installation the files of fonts for synthesis and export to PDF are placed to Data folder not in Bin\Bin64. | Help | 393406 | US |
| Table blocks were detected in the mode IPageAnalysisParams::DetectTables=FALSE. Issue was fixed; from now on the parameter will work as documented. | DA | 389740 | US |
| In Java wrapper in ITableBlock interface the method FindBaseCellFromPoint wasn’t supported. In Help file this limitation wasn’t mentioned. From now on this method will be kept in the API as documented. | API | 385844 | EU |
| The extra spaces were inserted in numbers with decimal point. | Recognizer | 302474 | US |
| DataMatrix barcodes weren’t detected | BarcodesDetector | 383444 | US |
| There was problem with this Windows Installer package on Chinese Windows 7 | Install | 386824 | US |
|  | IRTFExportParams::KeepPages property didn’t work correctly. There was a disparity between the API and the Help file. Now this property is working as described in the Help file: the value of this property is used only if the PageSynthesisMode property is set to PSM\_RTFFormatParagraphs or PSM\_RTFPlainText, otherwise it is ignored. | API |  |  |
| Rare | There was problem with this Windows Installer package on Chinese Windows 7 | Install | 389818 | 3A |
| An error occurs during the opening of pdf files using AddImageFileFromMemory Method with MultiProcessingMode = MPM\_Auto | API | 394159 | UA |
| Aztec code wasn’t recognized. | BarcodesDetector | 385839 | EU |
| Access violation in java project during classification. | API | 361888 | RU |
| Specific | The table was detected incorrectly. | DA | 362040 | RU |
| The layout of output docx file is not the same with original pdf file. | Export | 344028 | US |
| Some tables weren’t detected on the specific image. | DA | 359925 | US |
| IPE in the IFRDocument::Analyze method. | DA | 341072 | EU |
| The QR barcodes weren’t detected on the specific images. | BarcodesDetector | 386532 | EU |
| After the export from a specific tif file to TXT and one of the columns wasn’t displayed correctly. | Export | 387248 | EU |
| An IPE occurs in the process of analyze of Korean document. | DA | 381549 | EU |
| The result of export to TXT with RetainLayout = true was incorrect. | Export | 350589 | US |
| The table wasn’t detected on the specific picture. | Image | 384343 | RU |
| The table wasn’t detected on the specific picture. | DA | 347439 | US |
| The precision on Regular Expressions section about the fact that backslashes in the path must be doubled. | Help | 372998 | UA |
| IPE in the parallel processing mode. | DA | 384351 | EU |
| The text in the specific pdf file wasn’t detected completely. | DA | 385218 | EU |
| The color was detected incorrectly on the specific image. | DA | 344028 | US |
| The barcode PDF417 wasn’t recognized. | BarcodesDetector | 386257 | US |
| The table blocks weren’t detected correctly on the specific image. | DA | 389219 | 3A |
| The numbers were detected as a picture with the property EnableTextExtractionMode = true on the specific image. | DA | 391601 | US |
| The specific PDF files were opened very slowly with IFRDocument::AddImageFile method | Image/PDF | 395329 | EU |
| The left column of the table wasn’t detected. | DA | 393181 | EU |
| The text wasn’t recognized on the specific image. | DA | 390245 | US |
|  | The Croatian characters Č, Ć, Ž were recognized as C, C, Z. | Synthesis | 397849 | US |

## Known Issues and Workarounds

### Some API is not implemented

The following API is not implemented in FRE 10 and FRE 11:

* IFootnoteSeries::IsNumberingWithSuperscript. Always returns “false”.
* IFootnoteSeries::PositionOnPage. Always returns “FPPT\_SingleColumnSection”.
* IFootnoteSeries::PositionInDocument. Always returns “FPDT\_PageEnd”.
* IFootnoteSeries::HasSeparator. Always returns “true”.
* ITextPicture::ColumnNumber. Always returns “0”.
* ICharParams::IsWordStart. Always returns “false”. It is true only for character parameters got through IWordRecognitionVariants interface.
* IIncut::TextWrapping. Always returns “TW\_Undefined”.
* IRunningTitlesSeriesText::HasSeparator. Always returns “false”.

The implementation is not planned.

### OnPageProcessed comes once per a document

During opening, synthesis, exporting stages the callback OnPageProcessed arrives only once per a document rather than after each page.

### Skew Correction works on the image without text

The skew is corrected on the image without text. The settings for skew correction are default. (helpdesk request 402457)

The fix of this issue is postponed.

### In parallel mode IWords collection is empty after processing

After recognition with IEngine::ProcessPage in parallel mode IWords collection is empty.

### Skew is not corrected properly on the specific images

Problem with text extraction accuracy and block detection due to inexact skew correction. (helpdesk request 372736)

### Problem with number recognition

The recognition results for numbers on the specific client’s images are worse than the results in FRE 8. (helpdesk request 369894)

### Incorrect positioning of images in the results of export to HTML

Images from the PDF file are shifted in the HTML export results. (helpdesk request 344900)

### The results of export to XML in several cases don’t pass XML validation

The results of export to XML with property WriteParagraphStyles = true for right to left text don’t pass XML validation. The following error occurs: “The 'rtl' attribute is not declared”

### The text doesn’t fit the size of cells during export to XLSX.

The text doesn’t fit the cells when it is exported to XLSX in ExactCopy mode.

### Wrong detection of footers and headers

In some cases extra footers and headers are detected on the processed documents.

### InjectTextLayer method works incorrectly on some Japanese images

New IFREngine::InjectTextLayer methods that processes the input PDF file and creates and injects he text layer created from the recognized text in a searchable PDF file has a shortcoming. On some pictures with Japanese text the injected text layer contains wrong characters.

### PDF/A validation report

The following issues are known for PDF/A files produced by this release of FRE 11:

1. Adobe Acrobat 11.0.5 (Preflight 11.0.4) detects an error for PDF/A with attachments (PCM\_Pdfa\_3a):

“Embedded file does not have AF entity”

# R2 GM Patch –

## Part number, build number

|  |  |
| --- | --- |
| Part# | 1041/21 |
| Build# | 11.1.5.110 |

## Fixed Bugs

This section contains a list of bugs that have been fixed in this release.

The following terminology is used for severity in the list:

**Critical**: the software will not run

**High**: unexpected fatal errors (includes crashes and data corruption)

**Medium**: a feature is malfunctioning

**Low**: a cosmetic issue

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Severity | Description | Subsystem | HD # | Office |
| Critical | During the first setup of FREngine if the folder FREngineDataFolder (the parameter of GetEngineObjectEx function) didn’t exist, FREngine crashed on the GetEngineObjectEx function with "Unhandled exception at 0x7769dcbb (ntdll.dll) in Reclamation.exe: 0xC0000374: A heap has been corrupted." The issue didn’t reproduce during the second setup. | API | - | - |
| High | The project hung on the IFRDocument::Recognize method with the following settings:  RecognizerParams::TextLanguage = English, CMC7 and RecognizerParams::TextTypes = (TT\_MICR\_CMC7 | TT\_Normal) | Recognizer | 405817 | US |
| Medium | Some text was lost during analysis stage on specific documents. | Analysis | 407460 | EU |
| Medium | Extra spaces appeared in the recognition results of addresses in envelopes in Italian. | Recognizer | 364232 | EU |
| Medium | During export to PDF with PDFAComplianceMode = PCM\_Pdfa\_1a and FontEmbeddingMode = FEM\_Embed an error occurred "Could not embed the font 'Helvetica'. Embedding is not allowed for this font." | PDFToolkit | [407718](https://helpdesk.abbyy.com/Request/Edit/397942) | EU |

# R3 GM –

## Part number, build number

|  |  |
| --- | --- |
| Part# | 1041/29 |
| Build# | 11.1.6.94 |

## New Features and Improvements

### Possibility to extract and to add attachments from PDF

From this release it is possible to extract attachments from input PDF file and to add attachments to the output PDF file.

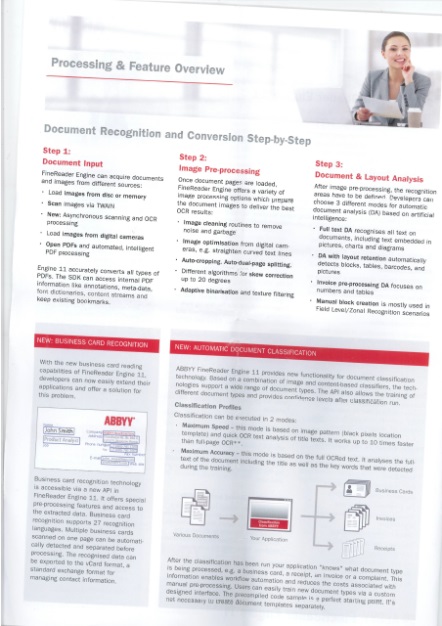
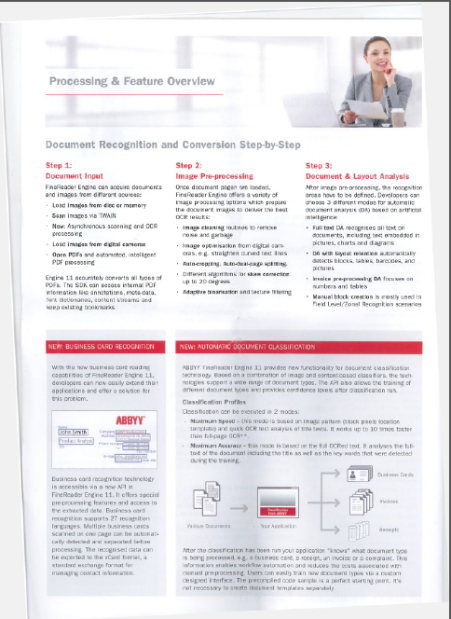
Each attachment is represented by PDFAttachment object that exposes methods which allow you to access the attached file by saving it on disk or into the global memory.

PDFAttachment object also provides access to the original file name, description added by the author and the type of binding of the attachment. The binding can take values from PDFAttachmentBindingEnum: PAB\_Annotation and PAB\_Document. PAB\_Annotation means that attached file is associated with a specific annotation on a specific page. PAB\_Document value is set if the file is attached to the whole document. Note that for an attachment which is added via FineReader Engine API binding value is always PAB\_Document.

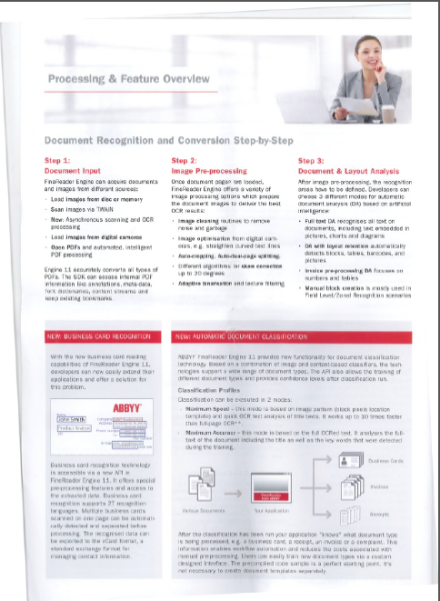
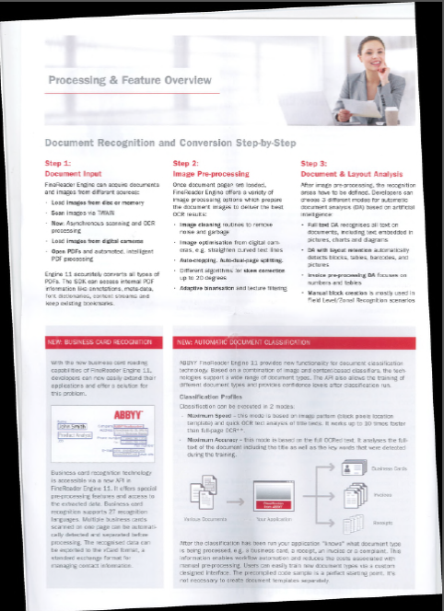
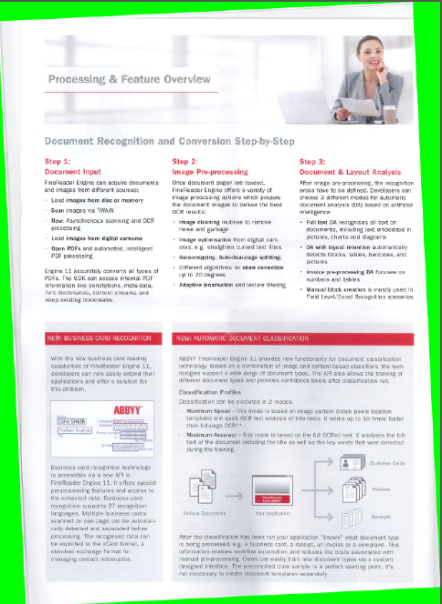
All the attachments are contained in FRDocument::PDFAttachments. They are extracted from the input PDF document during opening, or you can add your own files to be attached to the output PDF file during export. To attach all the files of this collection to the output PDF file, it is necessary set the IPDFExportFeatures::WriteSourceAttachments property to TRUE.

### New property IPrepareImageMode::BackgroundFillingColor to fill areas added after skew correction

After skew correction supplementary areas are added to the edges of document. The color of these areas is set automatically by FineReader Engine:

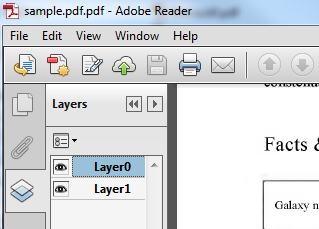
Now it is possible to specify the color used for filling the areas manually with IPrepareImageMode::BackgroundFillingColor property:

The default value of this property is -1, which means that the color is determined by ABBYY FineReader Engine automatically (HelpDesk request 401344).

### Possibility to see PDF layers in PDF Viewers

A new property IPDFMRCParams::AssignPdfLayersToMrcPlanes was added in this release. This property manages the availability of PDF layers in the output PDF file. If you set it to TRUE, PDF layers are assigned to MRC image planes, so that in some PDF viewers such as Adobe Acrobat you can choose which layers to view:



This property is only taken into account for image-only and image-on-text PDF files.

The default value of this property is FALSE.

### Parallel export to PDF and to PPTX

Now export of multi-page documents to PDF and to PPTX will be performed in parallel mode. This feature increases processing speed of multi-page documents in parallel processing scenario using Batch Processor or FRDocument. In the table below you can see the number of pages processed per minute and exported to PDF in this release compared to the previous release:

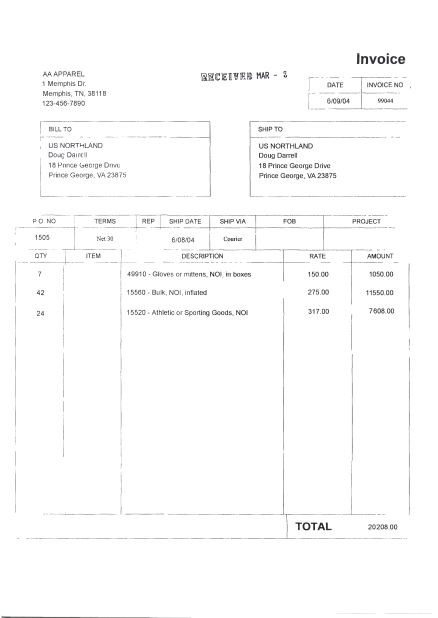
|  |  |  |
| --- | --- | --- |
|  | Release 3 with parallel export | Release 2 without parallel export |
| Processing with **FRDocument** | 111 | 79 |
| Processing with **FRDocument** (with [PageFlushingPolicy](mk:@MSITStore:D:\Documents\FREngine11.chm::/APIReference/FRDocument.htm#PageFlushingPolicy) = PFP\_KeepInMemory) | 143 | 96 |
| Processing using Batch Processor | 117 | 82 |

The processor of the testing machine is Intel® Core™ i5-3450 (3.10GHz, 4 physical cores) with 8 GB of RAM, the number of simultaneously run processes is 4. Performance was tested on 300 English-language images, with the settings of the DocumentArchiving\_Speed predefined profile.

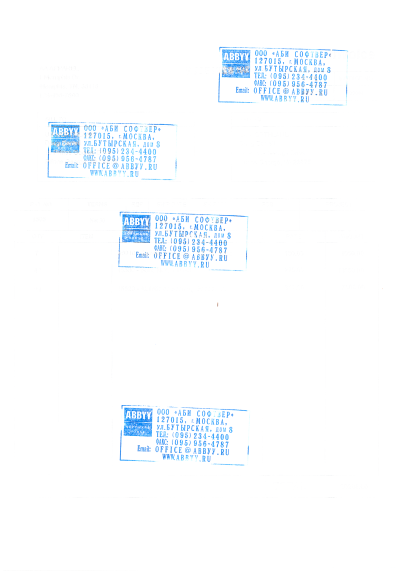
### New method IImageDocument::RemoveColorObjectsEx

This feature can be useful for documents with black text and white background that contain colored elements that don’t need to be recognized. For example, for monochromic documents with colored stamps only black-and-white layer can be recognized and the colored elements can be added to output document.

IImageDocument::RemoveColorObjectsExmethod allows you to remove color objects of specified hues from original document

C:\Users\TKOLES~1\AppData\Local\Temp\Rar$DR10.2125\Input\StampScan0001.tif 

The hues of removed objects are set by HSL representation. The method allows replacing removed objects with the specified color. It also allows saving a separate image containing only the extracted objects:



### New advanced language detection mode

New property RecognizerParams::LanguageDetectionMode was added on this release. This property will be useful for recognition of documents the language of which is not known to you. If language detection mode is on, the recognition languages are selected from the list of languages specified in the TextLanguage property. Language detection was significantly improved from the previous release and now it turns on automatically if it is necessary.

This property has three modes: TSPV\_Auto, TSPV\_No and TSPV\_Yes. The default value of this property is TSPV\_Auto. In this mode ABBYY FineReader Engine will automatically determine if this processing mode should be used, depending on the situation.

Auto detection will be useful only if TextLanguage set contains a combination of CJK and European languages. New feature will increase the speed and improve the quality of recognition for documents with European languages in the scenario of processing of input batch that contains documents only with CJK languages and documents only with European languages.

In the scenario when you know that all languages, specified in TextLanguage set, are present in the document, that you process, we recommend setting RecognizerParams::LanguageDetectionMode to TSPV\_No.

Old property RecognizerParams::DetectLanguage is marked as deprecated.

### New predefined filter FNF\_PDF in FontNamesFiltersEnum

FontNamesEnum enumeration constants describe predefined filters of font family names. These filters specify the set of fonts to be used during document synthesis. A new value FNF\_PDF was added to FontNamesFiltersEnum. If this filter is applied, document synthesis uses font families the names of which are specified in the resources of the input PDF file. However, the fonts themselves are not extracted from the PDF file; they need to be installed on the workstation to be used.

### New property IFRDocument::PDFFontNames

New read-only property IFRDocument::PDFFontName returns the collection of fonts extracted from PDF file.

### Skew correction in preprocess stage

Now it is possible correct skew at preprocessing stage instead of during image opening.

CorrectSkew property was added to PagePreprocessingParams object. The type of skew correction is defined by the CorrectSkewMode property.

### Advanced IsEmptyEx method for checking if the page is empty

IsEmptyEx is a new method which allows you to specify additional parameters during empty page detection.

WithEmptyPageDetectionParams object you can define the number of letters and text objects that a page can contain and still be considered empty. It is possible to set maximum black percentage and to specify if the page must be searched for barcodes. You can also set the page rectangle, so that any garbage on the margins does not affect the result.

A method IEngine::CreateEmptyPageDetectionParamswas included in API to create theEmptyPageDetectionParams object.

### WibuKey support

WibuKey protection hardware keys are supported in this release. It is necessary to use new pricelists to generate licenses with license Storage CodeMeter Key. HQ will add new pricelists for each particular regional office to Registration Server upon request. You can address directly to HQ product analysts.

### Possibility to export TIFF files with one strip

A strip is a subsection of the image composed of one or more rows. A TIFF image may be composed of one or more strips. Now it is possible to deliberately produce TIFF file composed of one strip by setting ITiffExtendedParams::WriteSingleStrip to TRUE. The parameters for TIFF files producing can be used IImage::WriteToFile method.

That feature was implemented as some archive writers don’t support valid TIFF files with several strips (HelpDesk request 421941).

### New property Recognition Set

A property ITextLanguage::RecognitionSet returns the full letter set used for recognition with this TextLanguage, combining all letter sets of its base languages and additional letter sets.

(HelpDesk requests 375306, 420377, 300708).

### Deactivation in silent mode

Now it is possible to deactivate licenses in silent mode via License Manager by entering the parameter /SilentDeactivation through the command line (HelpDesk request 401395).

### Improved CRM\_ContentOnly mode

New technology is used in CRM\_Content Mode for document analysis of PDF files. In CRM\_ContentOnly mode only content of the source PDF file is used (the image is not rasterized for recognition as it was before, all the information about text layer, images, separators, etc. is taken directly from the PDF). This mode is designed for PDF that contain not only raster elements. This changing helps to preserve the original layout of the source document and improve the quality of the output picture. This feature is particularly useful for conversion from PDF to formats of MS Office.

Comparing to the previous release CRM\_Content Mode works faster and some shortcomings and bugs of this mode were eliminated.

To use the feature you can set IObjectsExtractionParams::SourceContentReuseMode to CRM\_ContentOnly and use these parameters during analysis stage.

### Correct display of PDF files with PMingLIU/MingLIU fonts in PDF viewers

Now all exported PDF files with PMingLIU and MingLIU fonts are displayed correctly in all PDF viewers. Previously there were problems with display due to errors of some viewers. As a workaround these fonts are now embedded to the output PDF file.

Please note that if for a Chinese-language document the PMingLiU/MingLiU font is used, it will be embedded into output PDF file regardless of the value of the property IPDFExportFeatures::FontEmbeddingMode (HelpDesk request 414470).

### Information about adding comments in profiles in Help file

The information and an example on how to add comments, were in the section Working with Profiles Comments can be added by starting a line with a semicolon (HelpDesk request 406671).

## Performance results

This section contains performance results of FRE 11 R3 comparing to the previous releases and the latest version of FRE 10.5. The processor of the testing machine is Intel® Core™ 2 Duo CPU E6750 (2.66GHz, 2 physical cores) with 3,9 GB of RAM.

### English

FRE 10.5 R3

FRE 11 R1/R2/R3

FRE 10.5 R3

FRE 11 R1/R2/R3

### Japanese

### Korean

### Chinese

## Fixed Bugs

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

Four-point scale will help you to evaluate the severity of each issue, enabling you to make informed decision 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 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 following table contains bugs fixed in this release sorted 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 | Subsystem | HD # | Office |
| Critical | The project hangs on the IFRDocument::Recognize method with the following settings:  RecognizerParams::TextLanguage = English, CMC7 and RecognizerParams::TextTypes = (TT\_MICR\_CMC7 | TT\_Normal). | Recognizer | 405817 | US |
| Critical | The project hangs on the IFRDocument::Recognize method after adding of two blocks with hand-printed text (FMT\_PartitionedFrame and FMT\_SimpleComb). | Handprint Recognizer | 405607 | EU |
| Critical | Unhandled access violation during call of IMultiProcessingParams::RecognitionProcessesCount  **Root cause**: interface IMultiProcessingParams lacked "dual" attribute. | API | 405818 | EU |
| 335175 | US |
| Critical | IPE: .\src\ProcessorsCallsHandler.cpp, 74. during analysis stage with MultiProcessingMode set to MPM\_Parallel on specific image. | DA | 412308 | EU |
| Critical | IPE: .\src\PrepareImageEmulator.cpp, 59 on the IEngine::PrepareDib method on specific image. | API | 419854 | US |
| Critical | IPE: Division by zero during analysis stage on specific image in multiprocessing mode. | DA | 426015 | EU |
| Critical | IPE: .\Src\TableVerticalSplitter.cpp, 141 during analysis stage on specific image in multiprocessing mode. | DA | 426015 | EU |
| Critical | An error "...\Data\Arabic.oce was not found." on analysis of CJK documents  **Workaround:** leave Arabic.oce in the Data folder in the distribution | DA | 418831 | EU |
| Major | During export to PDF with PDFAComplianceMode = PCM\_Pdfa\_1a and FontEmbeddingMode = FEM\_Embed an error occurrs "Could not embed the font 'Helvetica'. Embedding is not allowed for this font." | PDFToolkit | [407718](https://helpdesk.abbyy.com/Request/Edit/397942) | EU |
| Major | ImageOnly PDF files exported with MRC are fuzzy. | Image | 416816  423131 | US |
| Major | Documents exported to PDF with PDFAComplianceMode = PCM\_Pdfa\_1a and PDFAComplianceMode = PCM\_Pdfa\_1b don’t pass online validation on <http://www.pdf-tools.com/pdf/validate-pdfa-online.aspx> . | Export | 418803 | US |
| Major | With IRecognizerParams::SaveCharacterRecognitionVariants = true only one character recognition variant is saved | Recognizer | 413018 | US |
| Major | An error “An incorrect structure was found in the PDF file” in Adobe Acrobat XI PRO during page insertion (Insert Pages... > From File...). The problem is reproduced on exported tagged PDF with links. | PDFToolkit | 419801 | US |
| Major | Chinese text written with MingLIU and PMingLIU fonts isn’t displayed in Adobe Reader.  **Workaround**: set FontEmbeddingMode = FEM\_Embed  **Root cause:** Adobe Readerdoesn’t display Chinese text written with MingLIU and PMingLIU fonts if they are nor embedded. | PDFToolkit | 414470 | EU |
| Minor | In results of export with ObjectsExtractionParams::SourceContentReuseMode=CRM\_ContentOnly the size and the boldness of print are different from those of text content in PDF file. | Synthesis | 398687 | RU |
| Minor | During call of FREngine::LoadModule with FREM\_EuropeanPatterns parameter page counter decreases on one page. | API | 416835 | US |
| Minor | The content of IParagraph::Words sometimes differs from the content of IParagraph::Text. | Synthesis | 410630 | EU |
| Minor | The tabulation isn’t saved in table cells. | Synthesis | 302091 | US |
| Minor | Extra spaces appear on the sides of letter “i” in the recognition results of addresses in envelopes in Italian. | Recognizer | 364232 | EU |
| Minor | Some text is lost during analysis stage on specific documents. | DA | 407460 | EU |
| Minor | A table on specific document is detected as three text blocks.  **Workaround:** define table block manually | DA | 412314 |  |
| Trivial | User profiles are case sensitive. The property specified in user profile isn’t change if it is defined with case errors in a profile. In this release the property doesn’t depend on the case. | API | 406671 | US |

## Known Issues and Workarounds

### Some API is not implemented

The following API is not implemented in FRE 10 and FRE 11:

* IFootnoteSeries::IsNumberingWithSuperscript. Always returns “false”.
* IFootnoteSeries::PositionOnPage. Always returns “FPPT\_SingleColumnSection”.
* IFootnoteSeries::PositionInDocument. Always returns “FPDT\_PageEnd”.
* IFootnoteSeries::HasSeparator. Always returns “true”.
* ITextPicture::ColumnNumber. Always returns “0”.
* ICharParams::IsWordStart. Always returns “false”. It is true only for character parameters got through IWordRecognitionVariants interface.
* IIncut::TextWrapping. Always returns “TW\_Undefined”.
* IRunningTitlesSeriesText::HasSeparator. Always returns “false”.

The implementation is not planned.

### IWords collection is empty and IWordRecognitionVariants and ICharacterRecognitionVariant return NULL after processing

The issue is reproduced after the recognition with methods of IEngine and IDocumentAnalyzer objects, developed only for one-page documents processing (e.g. IEngine::ProcessPage, IDocumentAnalyzer::PreprocessAnalyzeRecognizePage or IDocumentAnalyzer::RecognizePage). After the recognition IWords collection is empty and IWords collection properties aren’t filled up, IWordRecognitionVariants and CharacterRecognitionVariant collections return NULL.

**Workaround**: use IFRDocument object or IFRPage object for processing.

This issue will be fixed in the next maintenance release.

### The text doesn’t fit the size of cells during export to XLSX.

The text doesn’t fit the cells when it is exported to XLSX in ExactCopy mode.

### Wrong detection of footers and headers

In some cases extra footers and headers are detected on the processed documents.

### InjectTextLayer method works incorrectly on some Japanese images

New IFREngine::InjectTextLayer method that processes the input PDF file and creates and injects the text layer created from the recognized text in a searchable PDF file has a shortcoming. On some pictures with Japanese text the injected text layer contains wrong characters.

### IPDFMRCParams::MonochromeText doesn’t work correctly

Different algorithms of compressions during export to PDF are used with IPDFMRCParams::MonochromeText set to TRUE (HelpDesk request 416986).

### Errors during opening of PDF file

Unknown errors during opening of PDF file (HelpDesk requests 414136, 395847).

### CSS files during export to HTML

During export to HTML apart from .html files, .css files appears in the output results.

### PDF/A validation report

The following issues are detected by Adobe Acrobat 11.0.6 (Preflight 11.0.6) for PDF/A files produced by this release of FRE 11:

1. Adobe Acrobat rarely detects an error for exported PDF/A files "Text cannot be mapped to Unicode". Exported files don’t pass validation.
2. Adobe Acrobat rarely detects an error for exported PDF/A -1a files "Syntax problem: Array with more than 8191 elements The issue is reproduced on files with a lot of pages. Exported files don’t pass validation.
3. Adobe Acrobat detects an error "Text is mapped to Unicode Private Use Area but no ActualText entry is present" for exported PDF/A-2a и PDF/A-3a in the scenario of conversion from PDF to PDF/A with SourceContentReuseMode set to CRM\_ContentOnly.

# R4 GM –

## Part number, build number

|  |  |
| --- | --- |
| Part# | 1041/41 |
| Build# | 11.1.9.75 |

## New Features and Improvements

### Protection redundancy

While working in a network, the failure of the license server stops the processing on all workstations in the network. To prevent this situation, a failproof redundant license server configuration has been introduced in this release. Now you can set up a backup license server which will take over the license management if the main license server breaks down.

For detailed instructions on setting up a redundant configuration, see “Working with the LicensingSettings.xml File” section in Developer’s Help.

### Export to memory

From now on FRE 11 can save recognized documents into a file stream.

In the previous release all files had to be saved on disk before they could be used in other applications. This may be not suitable, for example, due to security considerations.

IFRDocument::ExportToMemory method allows you to save the document into memory in an external format. Available file formats are represented by the FileExportFormatEnum enumeration constants. There is a limitation: the document cannot be exported into memory in XLS format.

IFileWriter interface is implemented for a file writing stream. The pointer to IFileWriter interface is an input parameter of IFRDocument::ExportToMemory method. This interface and all its methods are implemented on the client side.

### Load from memory method in outproc

IFRDocument::AddImageFileFromStream method opens an image file from the input stream implemented by the user, and adds the pages corresponding to the opened file to the document. This method differs from the AddImageFileFromMemory method in that it can be used when the Engine object is created using the OutprocLoader object.

IReadStream interface is implemented for a read stream. The pointer to IReadStream interface is an input parameter of IFRDocument::AddImageFileFromStream method. This interface and all its methods are implemented on the client side. A read stream may be implemented as reading from file.

### Possibility to enable and disable interpolation in PDF viewers

Interpolation in PDF viewers can insignificantly affect the visual quality of PDF file. A new property IPDFPictureCompressionParams::EnableInterpolationMode allows to disable interpolation in PDF Viewers. The property has tree modes: TSPV\_Yes, SPV\_No and TSPV\_Auto. Note that if the IPDFExportParams::PDFAComplianceMode is set to PDF/A, interpolation will be always disabled as it is required by specification of PDF/A format.

The default value of this property is TSPV\_Auto, which means that the interpolation will be turned off for PDF/A-compliant formats and on otherwise.

### New property IEngine::AvailablePredefinedLanguages

New property IEngine::AvailablePredefinedLanguages returns the collection of predefined languages that are available under the current license.

### New method OnChangeBlockType in Visual Components Sample

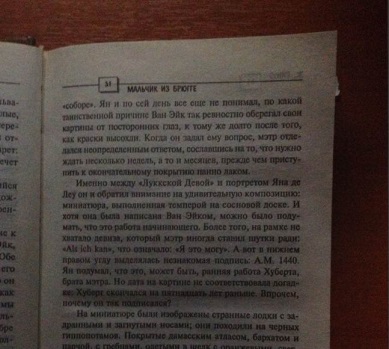
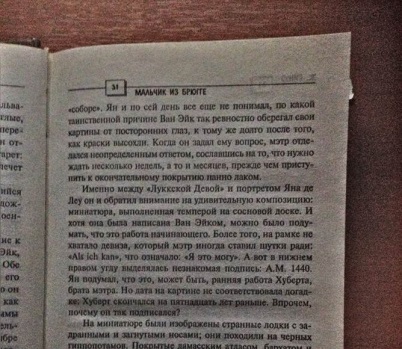
The method OnChangeBlockType must be implemented on the client side. It is called by ABBYY FineReader Engine after the block type has been changed in Image Viewer or in Zoom Viewer synchronized with Image Viewer. It allows you to cancel the type change.

The block type can be changed via the block properties toolbar or popup menu, or indirectly, by adding table separators to a block which previously was of non-table type.

### New property for shadows and highlights correction in photographs

IPagePreprocessingParams:: CorrectShadowsAndHighlightsproperty was added in this release. This property allows correcting excessive shadows and highlighting during image preprocessing. This property is designed for use with photographs only.

Three states are available: TSPV\_Yes, TSPV\_No and TSPV\_Auto. By default the property is set to TSPV\_Auto.

Before After

### Support of Farsi

From this release Farsi language is supported. It is added to the list of Predefined Languages. Win32 standard language identifier is equal to 1065.

The language is available, if Arabic module is included in license. Please note that as of now, dictionary support for this language is not available.

### New property IHTMLExportParams::UseCss

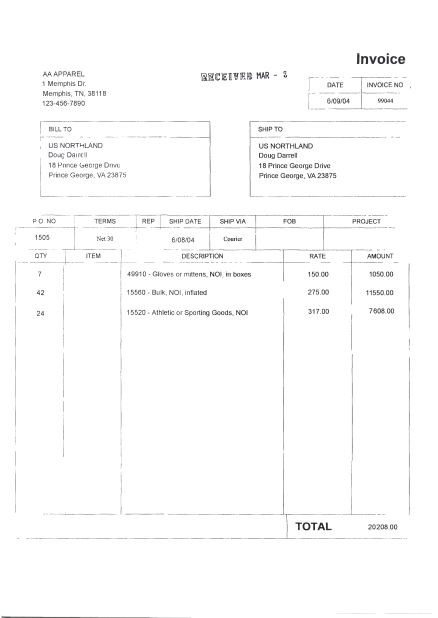
This property determines if a separate style sheet file (.css) is created. In previous releases style sheet file was always created. Now it is possible to use built-in style sheet file by setting IHTMLExportParams::UseCss to FALSE.

The default value of this property is TRUE.

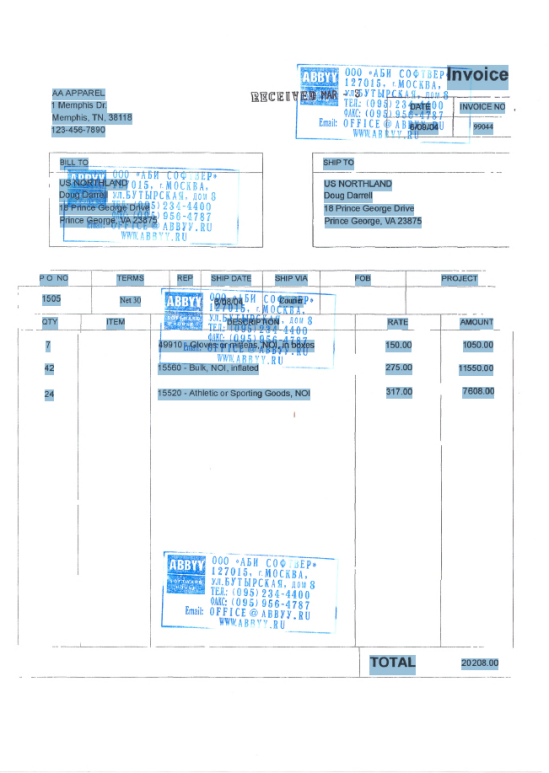
### Possibility to improve recognition quality by removing color objects before recognition during preprocessing stage

This feature can be useful for documents with black text and white background that contain colored elements (for example, stamps) that don’t need to be recognized.

New property IPageProcessingParams::ProhibitColorObjectsAtProcessing allows filtering out color objects on the image before layout analysis and recognition. After processing is complete the color objects can be put back on the image again:

C:\Users\TKOLES~1\AppData\Local\Temp\Rar$DR10.2125\Input\StampScan0001.tif 

**1.Input document contains color objects 2.Color objects are ignored during processing**



**3.Color objects are restored in export result**

If this property is set to FALSE, the ColorObjectsProhibitingParams property is ignored. The default value of this property is FALSE.

A new ColorObjectsProhibitingParams object has been added. It is used for tuning parameters of filtering out the color objects on the image before starting processing. This kind of preprocessing can be useful in cases when the document to be recognized has color stamps, signatures etc., which can reduce recognition quality. The parameters are only taken into account if the ProhibitColorObjectsAtProcessing property is set to TRUE.

It is possible to specify in properties of this object the following parameters:

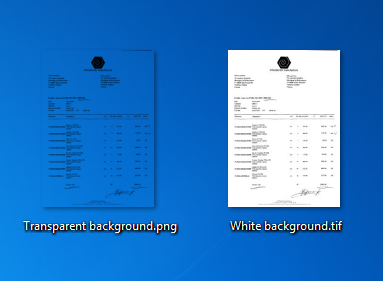
* the color which must replace the removed color objects
* the collection of the hues of the objects which must be filtered, in HSL representation.
* if, after processing is complete, the color objects must be put back on the image again.

A new property IPageProcessingParams::get\_ColorObjectsProhibitingParams returns a ColorObjectsProhibitingParams object.

Another way to improve recognition quality by removing color objects is to use IImageDocument::RemoveColorObjectsEx method, added in previous release, but it doesn’t allow restoring color objects on the image again before export.

### Possibility to replace black or white color of exported PNG images with transparent

From now on it is possible to replace black or white color of exported PNG images with transparent:



A new object PngExtendedParams was added. This object provides functionality for tuning the parameters of saving a black-and-white image to PNG format (IFF\_Png format) using the IImage::WriteToFile method.

To replace the color with transparent it necessary to specify the color in IPngExtendedParams::TransparentColor property. Only black and white colors are currently supported.

The default value of this property is -1, which means that no color will be replaced with transparent color.

### New method IEngine::CreateMultipageImageWriterEx

The method IEngine::CreateMultipageImageWriterEx allows to set extended parameters of image saving (JpegExtendedParams, PngExtendedParams or TiffExtendedParams) when creating the MultipageImageWriter object.

### New attribute ‘rotation’ in xml export scheme

New attribute ‘rotation’ in ‘page’ tag defines the type of rotation applied to original page image before processing. It can have one of the following values: Normal, RotatedClockwise, RotatedUpsideDown, RotatedCounterclockwise.

### New property IRTFExportParams::KeepPageBreaks

IRTFExportParams::KeepPageBreaks property specifies if the page breaks must be retained in the output RTF document. (HelpDesk request #433300)

### New profile for speed up of barcode recognition

New profile BarcodeRecognition\_Speed was added in this release. It enables fast barcodes extraction.

New BarcodeRecognition\_Accuracy profile is equivalent to the BarcodeRecognition profile available in previous releases.

In some cases BarcodeRecognition\_Speed speeds up the processing almost by 2 times. However, it can decrease the recognition quality. Below you can see the changes in results of processing with the BarcodeRecognition\_Speed profile compared to the BarcodeRecognition\_Accuracy profile for some types of barcodes.

|  |  |  |
| --- | --- | --- |
| Barcode type | Speed up in **BarcodeRecognition\_Speed**  profile | Lost in quality in **BarcodeRecognition\_Speed**  profile |
| 1D Barcodes, Code 128 | 14% | 10% |
| 1D Barcodes, Code 93 | 73% | 0% |
| 1D Barcodes, EAN 8 | 29% | 12% |
| 1D Barcodes, UPC-E | 116% | 0% |
| 2D barcodes, Aztec | 59% | 50% |
| 2D barcodes, MaxiCode | 94% | 37% |

Tests were run on the machine with following configuration: CPU - Intel(R) Core(TM) i5-3450 CPU @ 3.10GHz, Memory - 2.0 Gb.

### New section in Developer’s Help ”Predefined Profiles Specification”

The new section contains a full list of all settings used by ABBYY FineReader Engine predefined profiles.

### The changing of the GUI forms height of some samples

The height of the main form of Image Preprocessing sample didn’t entirely fit on the screen with small resolution. A scroll has been added to the main form of Image Preprocessing sample, and now it fits to the monitors with small display size.

The heights of GUI forms of BCR, Camera OCR and samples were adjusted to fit the screens with vertical resolution equal to 800 pixels.

### Support of corrupted tiff files opening

Starting with this release, FineReader Engine can open corrupted TIFF files. It can be done only if one last line is corrupted, otherwise an error occurs and the corrupted file is not opened.

If the TIFF image which has been opened is corrupted, а warning message containing the number of the damaged page in the image file will be shown. (HelpDesk request #344473)

## Performance results

This section contains performance results of FRE 11 R4 comparing to the previous releases and the latest version of FRE 10.5. The processor of the testing machine is Intel® Core™ 2 Duo CPU E6750 (2.66GHz, 2 physical cores) with 3,9 GB of RAM.

### English

### Japanese

### Korean

### Chinese

## Fixed Bugs

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

Four-point scale will help you to evaluate the severity of each issue, enabling you to make informed decision 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 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 following table contains bugs fixed in this release sorted 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 | Subsystem | HD # | Office |
| Critical | Error “Page 0 of the image file cannot be opened due to the following error: Not enough memory! Page numbering starts with 0” during adding of PDF files to FRDocument | API | 412723 | RU |
| Critical | Access violation during opening of a corrupted file. | Image | 445991 | 3A |
| Critical | During processing of 47 page document IPE:.\src\ProcessorsCallsHandler.cpp, 74. occurs during IFRDocument::Preprocess and IFRDocument::Analyze stages. | API | 443163 | US |
| Critical | E\_OUTOFMEMORY during synthesis stage. A multipage document is processed. | API | 431135 | EU |
| Critical | Access violation on method IRecognizerParams::SetTextLanguage. Java language is used in project. | API | 440006 | EU |
| Critical | IPE: .\src\ProcessorsCallsHandler.cpp, 74. during processing of specific tiff files with default settings. | Document Analysis | 429750 | EU |
| Critical | IPE:.\Src\KeyValuePageGenerator.cpp, 917. During processing of PDF file with default settings. | Document Analysis | 436388 | EU |
| Critical | IPE: .\src\languages.cpp, 257 occurs during export.  All image blocks of a document are replaced by the text block. Document is exported to PDF in TextOnImage mode. | Export | 433170 | US |
| Critical | JPEG image is recognized with default settings and exported to PDF.  IPE:.\Build\13.0.14\0\PdfTools\PdfToolkit\Inc\PdfToolkit.Types.inl, 283. occurs during export. | Export | 435031 | EU |
| Critical | IPE:.\Src\OutParagraph.cpp, 58. during synthesis stage in a specific usage scenario. | Synthesis | 418444 | EU |
| Critical | IPE:.\Src\KeyValuePageGenerator.cpp, 1269. during analysis of specific image. | Document Analysis | 430857 | EU |
| Critical | IPE:.\Src\Fonts\FontDescriptorImpl.cpp, 107 during export to PDF while using BatchProcessor. | Export | 430298 | US |
| Major | All XLS and XLSX produced by FRE 11 are not indexed in Windows Search. | Export | 429237 | US |
| Major | A significant slowdown during sequential processing of large multipage documents (200 pages and more). | Image | 430252 | EU |
| Major | The issue is reproduced after the recognition with methods of IEngine and IDocumentAnalyzer objects, developed only for one-page documents processing (e.g. IEngine::ProcessPage, IDocumentAnalyzer::PreprocessAnalyzeRecognizePage or IDocumentAnalyzer::RecognizePage). After the recognition IWords collection is empty and IWords collection properties are not filled up, IWordRecognitionVariants and CharacterRecognitionVariant collections return NULL. | API | 419443  421025  436055 | US |
| Minor | After processing of a specific multipage document with BatchProcessor and with settings IRecognizerParams::TextLanguage = "English,Japanese" IPDFExportFeatures::FontEmbeddingMode = FEM\_Embed  text layer in exported PDF file contains only “?” symbols.  **Workaround**: the problem occurs only for multipage documents, the workaround is to export document page by page | Export | 445547 | US |
| Minor | The number of pages in input file is 40 and the output RTF document contains 38 pages.  **Root cause**: Page breaks weren’t retained in the output document. New property IRTFExportParams::KeepPageBreaks is added. | Synthesis | 433300 | US |
| Minor | Lost separator in export results to DOCX. | Export | 406089 | US |
| Minor | Error in validation of exported PDF/A-1 document in Adobe Acrobat Preflight “Indirect object "endobj" keyword not followed by an EOL marker”. | PDFToolkit | 437768 | US |
| Minor | Bold font style wasn't detected on a specific image. | Synthesis | 422183 | EU |
| Minor | Only capital letters are recognized on specific image. | Recognizer | 437944 | US |
| Minor | First paragraphs are not detected on the specific documents with IObjectsExtractionParams:​:SourceContentReuseMode = CRM\_ContentOnly | Recognizer | 398687 | RU |
| Minor | Text layer of output PDF file is not selected correctly Adobe Reader. Recognition languages are ChinesePRC and English. | TextRendering | 381998 | EU |
| Minor | CJK text string with vertical orientation is not detected on some images. | Document Analysis | 437555 | US |
| Minor | Empty page is detected as image block. Document analysis takes a lot of time. | Document Analysis | 426809 | EU |
| Minor | The first line in Japanese is not recognized in a specific document.  **Workaround:** To set IPageAnalysisParams::EnableTextExtractionMode to true. | Recognizer | 444511 | US |
| Minor | Orientation of QR-code is not detected while recognition. | Barcodes | 421513 | US |
| Minor | PDF417 barcode is not found on the image. | Barcodes | 417110 | RU |
| Minor | Extra spaces during recognition of MICR E13B text on checks. | Recognizer | 425106 | US |
| Minor | During processing with BatchProcessor document fonts are detected as Default Metric Font which is not true. | API | 427765 | US |
| Minor | The coordinates of ", " symbol are not correct. | Recognizer | 426877 | US |
| Trivial | In Help BitmapBitsFormatEnum values are MIF\_BlackAndWhite, MIF\_Gray and MIF\_Color. The correct values are BBF\_BlackAndWhite, BBF\_Gray and BBF\_Color. | Help | 438707 | US |
| Trivial | IFRPage::CorrectResolutio​n() is on Help file that is not available in FRE 11. | Help | 438707 | US |
| Trivial | Misprints in ABBYY FineReader Engine 11 and 8.0/8.1/8.5 Compatibility and in ABBYY FineReader Engine 11 and 9.0/9.5 Compatibility in Help | Help | 439554 | RU |

## Known Issues and Workarounds

### Some API is not implemented

The following API is not implemented in FRE 10 and FRE 11:

• IFootnoteSeries::HasSeparator. Always returns “true”.

• ITextPicture::ColumnNumber. Always returns “0”.

• ICharParams::IsWordStart. Always returns “false”. It is true only for character parameters got through IWordRecognitionVariants interface.

• IIncut::TextWrapping. Always returns “TW\_Undefined”.

• IRunningTitlesSeriesText::HasSeparator. Always returns “false”.

The implementation is not planned.

### The text doesn’t fit the size of cells during export to XLSX.

The text doesn’t fit the cells when it is exported to XLSX in ExactCopy mode.

### Wrong detection of footers and headers

In some cases extra footers and headers are detected on the processed documents.

### InjectTextLayer method works incorrectly on some Japanese images

New IFREngine::InjectTextLayer methods that processes the input PDF file and creates and injects he text layer created from the recognized text in a searchable PDF file has a shortcoming. On some pictures with Japanese text the injected text layer contains wrong characters.

### IPDFMRCParams::MonochromeText doesn’t work correctly

Different algorithms of compressions during export to PDF are used with IPDFMRCParams::MonochromeText set to TRUE. (HelpDesk request 416986)

### Errors during opening of PDF file

Unknown errors during opening of PDF file. (HelpDesk requests 414136, 395847)

### Small displacement of the text rectangle in PDF files

The selection rectangle around a text line in exported PDF files is slightly shifted down (relative to text symbols).

### PDF/A validation report

The following issues are detected by Adobe Acrobat 11.0.9 (Preflight 11.0.9) for PDF/A files produced by this release of FRE 11:

1. Adobe Acrobat rarely detects an error for exported PDF/A files "Text cannot be mapped to Unicode". Exported files don’t pass validation.

2. Adobe Acrobat rarely detects an error for exported PDF/A -1a files "Syntax problem: Array with more than 8191 elements The issue is reproduced on files with a lot of pages. Exported files don’t pass validation.

3. Adobe Acrobat detects an error "Text is mapped to Unicode Private Use Area but no ActualText entry is present" for exported PDF/A-2a и PDF/A-3a in the scenario of conversion from PDF to PDF/A with SourceContentReuseMode set to CRM\_ContentOnly.

# R5 GM –

## Part number, build number

|  |  |
| --- | --- |
| Part# | 1041/50 |
| Build# | 11.1.10.100 |

## New Features and Improvements

### Exporting to ALTO up to version 3.0

It is possible to export OCR data into ALTO format according to the following ALTO standard versions:

• 2.0

• 2.1

• 3.0

The default version is 2.0, same as it was in the previous release.

### Memory saving in case of no need in coordinates on original image

Instruction IPrepareImageMode::KeepOriginalCoordinatesInfo = false tells the Engine not keeping image transformation matrix required for backward object coordinates calculation used for getting coordinates on original (source) image.

If there is no need in keeping the transformation data, it helps to preserve storage and memory space. For example, during processing of b/w images (~100Kb) the transformation data could be up to 1Mb per image.

### Speeding up multi-page image opening

Starting from this release the Engine opens multi-page documents using available Engine processors as well as it uses them for analysis, recognition and exporting stages. This functionality is available for the following image opening methods:

* AddImageFile
* AddImageFileFromMemory
* AddImageFileFromStream
* AddImageFileWithPassword
* AddImageFileWithPasswordCallback

New functionality brings up to 2 times speed up (see tables in this section below) for image opening step in 2 and 4 core configurations.

Table Processing time reduction with parallel image opening comparing to sequential opening, 2 cores

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **2 cores** | **Pages in a document** | | | | | |
| **2** | **3** | **5** | **10** | **30** | **100** |
| **PDF b/w** | -80% | -72% | -75% | -76% | -76% | -74% |
| **PDF color** | -78% | -65% | -68% | -73% | -73% | -76% |
| **TIFF b/w** | -91% | -88% | -90% | -90% | -88% | -88% |
| **TIFF color** | -86% | -87% | -88% | -88% | -88% | -84% |

Table Processing time reduction with parallel image opening comparing to sequential opening, 4 cores

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **4 cores** | **Pages in a document** | | | | | |
| **2** | **3** | **5** | **10** | **30** | **100** |
| **PDF b/w** | -80% | -80% | -80% | -75% | -85% | -83% |
| **PDF color** | -78% | -76% | -77% | -82% | -84% | -84% |
| **TIFF b/w** | -93% | -88% | -91% | -90% | -89% | -88% |
| **TIFF color** | -68% | -90% | -91% | -92% | -92% | -91% |

### Crop function supports greyscale and b/w images

Starting from this release it is possible to run ImageDocument::CropImage() function on greyscale and black and white images. Previously that was possible only for color images.

### New Java wrapper functions for loading Engine able to throw exceptions

FRE 11 inherited Java wrapper functions that load FREngine object from the wrapper released with FRE 10. In case of any issue with loading the Engine those functions do not throw exceptions, instead they write to a log file or a console. FRE 11 presented more advanced Java wrapper and new methods in it are capable of throwing exceptions what is common.

In this release, we completed Java wrapper by adding the Engine loading functions mirroring similar in native API and capable of throwing exceptions instead of logging them.

Here is the list of functions added:

* GetEngineObject
* GetEngineObjectEx
* DeinitializeEngine
* GetEngineInprocLoader
* GetEngineOutprocLoader

### Fine tuning of paper size detection

The Engine is able to detect paper boundaries on a scan automatically (PageAnalysisParams::NoShadowsMode = false, default value).

This helps to rid of garbage detection as text near borders of a scan. In some cases, this detector can fail. For example, if a scanned document contains a big dark picture it might be taken as a scanning background and removed from a document area as a scanning shadow.

To prevent such mistakes a host application may advise the Engine to limit the scanning shadow detector hypotheses by providing information on what part a source document occupies on a scan.

API provides PageAnalysisParams::PaperSizeDetectionMode for this purpose.

### Improved MRC quality

We improved all predefined MRC modes

#### MinSize predefined mode.

In the case of a 32-38% increase in file size, you get 6-15% faster speed and significantly better document image quality:

|  |  |
| --- | --- |
| Old settings | New settings |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

#### MaxQuality predefined mode

In the case of a negligible image size increase (~0%), the new settings provide 2-11% faster speed and better document image quality:

|  |  |
| --- | --- |
| Old settings | New settings |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

#### Balanced predefined mode

In the case of a 14-15% increase in file size, you get a 6-15% speed increase and significantly better document image quality:

|  |  |
| --- | --- |
| Old settings | New settings |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

#### MaxSpeed predefined mode

In the case of a very small increase in file size (2-8%), the new settings provide better image quality (see modes above) and noticeably faster processing speeds (30-45%).

#### Convenient method for getting recognized word region

Many customers work with recognized words instead of recognized characters. In most of the cases, such customers need to highlight a word of matter in graphic user interface. The highlighting requires a word-bounding rectangle (region).

Since words may include characters of different height and could span several lines the task becomes non-trivial. To make it easy and facilitate our customers we added new property to IWord interface that returns a word-bounding region – IWord::Region.

#### WIA 2.0 support

FRE 11 R5 is capable to work with scanners supporting WIA interface of version 2.0. This is the most recent version of WIA interface standard gradually substituting the previous one.

## Performance results

This section contains performance results of FRE 11 R5 comparing to the previous releases and the latest version of FRE 10.5. The processor of the testing machine is Intel® Core™ i5-4690 CPU (3.50GHz, 4 physical cores) with 8GB of RAM.

### English

### Japanese

### Korean

### Chinese PRC

### Chinese Taiwan

### Arabic

## Fixed Bugs

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

Four-point scale will help you to evaluate the severity of each issue, enabling you to make informed decision 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 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 following table contains bugs fixed in this release sorted 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 | Technology Subsystem | HD # | Office |
| Critical | Synthesis step crashes on a customer image with the following settings: IFontFormattingDetectionParams::DetectScaling = false. | Synthesis | 471624 | EU |
| Critical | VC crashes in the following scenario:  0. Set up 'Exact Copy' MS Word export mode.  1. Add an imgae file.  2. Recognize.  3. Add same image file again.  4. Scroll down few pages in the Document View.  5. Recognize. **Root cause:** incorrect (Undo Group) object freeing in destructor. | API | 452619 | RU |
| Critical | Recognition crashes on a customer image having incorrect orientation. Text Editor conponent (VC) must be in 'Editable Copy' mode. | TextRendering | 442908 | US |
| Medium | After deleting a page via the context menu VC does not automatically updates page numbers in the Document View component. **Root cause:** no code for updating. | API | 458348 | EU |
| Medium | The Engine loses IWords data in parallel processing mode, the issue is absent in sequential processing mode. **Root cause:** incorrect initialization of internal strutures. | API | 463315 | EU |
| Medium | The Engine does not accept the following PDF export parameters same time: PVN\_Version14 and PDFKL\_128Bit. **Root cause**: invalid parameters check. | Export | 458504 | US |
| Medium | PDF export in 'image on text' mode does not ignore the setting IPDFExportFeatures::ReplaceUncertainWordsWithImage = true as stated in the product documentation. **Workaround:** set the setting to 'false'. | Export | 465901 | US |
| Medium | Automatic document analysis lasts very long time (few hours) on a customer image. | DA | 469564 | US |
| Minor | Manually created text block in VC has Inversion property default value equal to false. Should be 'Auto'. | API | 452626 | RU |
| Minor | The Engine throws an exception during de-initialization in a customer's scenario. **Root cause:** incorrect order of object freeing. | API | 458518 | RU |
| Minor | Synthsis subsystem returns text of lower height than it is on source image. Reproduced on a customer image. | Synthesis | 430128 | US |
| Trivial | The Engine does not write IFRPage::DetectOrientation (ITextOrientation) and IBatchProcessor::Start(IPageSplittingParams \*PageSplittingParams) parameters into log file. | API | 456734 | US |
| Trivial | Searchable text is written as visible text under foreground image. | Export | 446166 | US |
| Trivial | EnginePool sample in Java has a bug leading to a deadlock when ReleaseEngine() follows GetEngine() provided all Engine instances are busy. | Samples | 459716 | US |

## Known Issues and Workarounds

### Some API is not implemented

The following API is not implemented in FRE 10 and FRE 11:

* IFootnoteSeries::HasSeparator. Always returns “true”.
* ITextPicture::ColumnNumber. Always returns “0”.
* ICharParams::IsWordStart. Always returns “false”. It is true only for character parameters got through IWordRecognitionVariants interface.
* IIncut::TextWrapping. Always returns “TW\_Undefined”.
* IRunningTitlesSeriesText::HasSeparator. Always returns “false”.

The implementation is not planned.

### The text doesn’t fit the size of cells during export to XLSX.

The text doesn’t fit the cells when it is exported to XLSX in ExactCopy mode.

### Wrong detection of footers and headers

In some cases extra footers and headers are detected on the processed documents.

### InjectTextLayer method works incorrectly on some Japanese images

New IFREngine::InjectTextLayer methods that processes the input PDF file and creates and injects the text layer created from the recognized text in a searchable PDF file has a shortcoming. On some pictures with Japanese text the injected text layer contains wrong characters.

### IPDFMRCParams::MonochromeText doesn’t work correctly

Different algorithms of compressions during export to PDF are used with IPDFMRCParams::MonochromeText set to TRUE. (HelpDesk request 416986)

### Errors during opening of PDF file

Unknown errors in APDFL library during opening of PDF several files. (HelpDesk requests 414136, 395847)

### Small displacement of the text rectangle in PDF files

The selection rectangle around a text line in exported PDF files is slightly shifted down (relative to text symbols).

### ‘User Pattern Training Utility’ sample crashes for ‘FieldLevelRecognition’ profile

Recognition command will result into IPE: “…\Interfaces\Implementations\inc\NotifyImpl.h, 62”.

The fix will be available in the next release.

### PDF/A validation report

The following issues are detected by Adobe Acrobat 11.0.9 (Preflight 11.0.9) for PDF/A files produced by this release:

In SourceContentReuseMode = CRM\_ContentOnly the Engine sometimes (5% of the test base in Korean language) generates PDF/A-2a and PDF/A-3a files which Adobe Acrobat Preflight validates with the following message “Text is mapped to Unicode Private Use Area but no ActualText entry is present”.

# R6 GM –

## Part number, build number

|  |  |
| --- | --- |
| Part# | 1041/74 |
| Build# | 11.1.14.63 |

## New Features and Improvements

### Windows 10 support

This release officially supports Microsoft Windows 10.

### Large document conversion to searchable PDFs

Use new API for converting PDF documents over 500 pages in order to reduce processing time up to 4 times. Please check “Exporting Large Documents” section in the Help file.

The main advantages of the new scenario:

• Dramatic speed increase when exporting large documents

• Less RAM resources required

• Convenient error handling without losing all the exported data. In case of simultaneous export a single error may cause failure of the whole export stage. If you are processing a large number of pages, the restarting of export can take significant time. Using the new functionality you do not have to stop processing. The processing errors can be caught and handled inside subsets of pages and export will be finished faster even if some errors occur during the processing.

Recommendations

• Using the new export mode is reasonable for documents of 50 pages or more. For 500 pages and more using ExportFileWriter is highly recommended.

• For the best possible speed choose Batch Processor (see “Processing using Batch Processor” in the help file).

• Export a fixed number of pages at a time. You may need to do experiments to pick the best number of pages for your documents. During internal ABBYY tests the portion size of 30 pages was found best for generic documents.

The results of speed testing are presented on the diagram.

Testing machine: Intel® Core™ i5-3450 (3.10 GHz, 4 physical cores), 8 GB of RAM, the number of simultaneously run processes is 4. Documents were saved by 30 pages at a time.

### Japanese OCR improvements

FineReader Engine 11 R6 contains the following improvements in Japanese OCR, resulting in 5% reduction of recognition errors:

• JapaneseModern is new OCR language with excluded outdated Kanji characters and it improves the accuracy of OCR for modern documents.

• Several frequent modern Japanese characters were trained and added into the recognition set.

• There are new models are added, which help with similar characters and mixed language recognition:

o Date: 2014年10月19日 （日）.

o Address: 東京都港区赤坂２丁目１７−２２.

• Updated dictionary

### Ability to process single-page documents from memory in batch mode

IBatchProcessor interface allowed to process single-page documents in batch mode, i.e. using multi-processing on all available CPU cores. In previous releases of FineReader Engine it worked with image files only.

Starting from this release it is possible to process single-page documents stored in memory in batch mode using new API:

• IImageSourceEx. Comparing to IImageSource interface the new one allows implementing callbacks for image files and images stored in memory.

• IBatchProcessor::StartEx(). Comparing to Start() method it accepts new IImageSourceEx interface instead of outdated IImageSource one.

### Arabic Fathatan (ً) diacritics support

FineReader Engine 11 R6 supports [Arabic Fathatan diacritics](http://www.fileformat.info/info/unicode/char/064b/index.htm), which increases Arabic OCR accuracy.

### Thai OCR accuracy improvement

Improved support of the most popular Thai fonts (Tahoma, TH SarabunPSK, TH Niramit AS, TH Kodchasal, TH Baijam, TH Fah Kwang) resulted in 11%-13% better Thai recognition accuracy overall, according to internal tests.

### Extended information on character position in a word

New “IsWordLeftmost” and “IsWordFirst” allows capturing the accurate location of word’s characters.

Previously the IsWordStart property supposed to define if a character starts a word. For right-to-left writing, this parameter was ambiguous: logically a word starts from the right end whereas geometrically it starts form the left. To solve the ambiguity we added these two new properties to API and XML export:

• IsWordLeftmost. Specifies whether the character is the leftmost character in a word.

• IsWordFirst. Specifies whether the character is the first character in a word.

The old property IsWordStart is declared deprecated and will be removed from API and XSD in FRE12.

### Smaller PDF with pages of mixed colority

In previous FRE 11 releases, we added PDF exporting API for setting up an individual compression type for each page. Now we introduce additional API for setting up individual resolution:

• BwPictureResolutionParams

• ColorPictureResolutionParams

• GrayPictureResolutionParams

Note: New API does not work in MRC mode. MRC and individual resolution/compression setting are two different approaches to the same goal – visual quality and small size.

### ‘BarcodeRecognition\_Accuracy’ became more accurate

FineReader Engine 11 R6 includes to following improvements in barcode detection:

• QRCode: 5.6% less errors

• EAN 13 with supplemental: 20.5% less errors

• EAN 13: 15.38% less errors

• UPC-E with supplemental: 10.5% less errors

• EAN 8 with supplemental: 5.9% less errors

The accuracy improvement brings also a slowdown, so it was applied to accuracy profile only. The slowdown varies from 5% to 70% depending on image complexity, detection type, and test-bed content (length and specifics).

### Callback from parallel processing

While using a parallel processing, one of Engine processors may ‘hang up’ with an image taking more time to process than it is allowed by default. There are several reasons for that:

• Limited hardware resources

• Specific image (large, complex layout, etc.)

• Logical error in FineReader Engine

Previously that led to process breaking with an error, that is not always a desirable behavior. In order to unlock new scenarios FRE 11 R6 presents new API:

• IParallelProcessingCallback to implement desired behavior on a client side

• IEngine::SetParallelProcessingCallback() to activate the handling.

### Warnings are connected to particular pages

FineReader Engine 11 R6 introduces an updated IFRDocument::IFRDocumentEventsEx interface with extended OnWarning() callback returning PageIndex property.

### Updated Admin’s Guide

We updated the ‘Admin’s Guide’ describing various installation scenarios of the product. It covers installation and de-installation processes for the following main distribution scenarios:

• Standalone mode using setup.exe with silent command line SN activation

• Standalone mode w/o setup.exe with SN activation using License Manager GUI

• Network mode using setup.exe, Licensing Service set up with command line SN activation

• Network mode w/o setup.exe, Licensing Service set up SN activation using GUI

### ‘Classification’ sample became faster

We speeded up the ‘Classification’ sample by adding multi-processing to the most calculation intensive step – OCR.

### CodeMeter drivers update

The release includes version 5.21 (5.21.1478.500) of runtime and driver for CodeMeter protection.

### ‘Is Redundancy’ option is listed in LM

License Manager is showing now ‘Is Redundancy’ licensing option introduced in FineReader Engine 11 R5.

## Performance results

This section compares the performance of FRE 11 R6 to previous releases and the latest version of FRE 10.5. The test machine’s processor was an Intel® Core™ i5-4690 CPU (3.50GHz, 4 physical cores) with 8GB of RAM. Test batches for different languages may include significantly different documents, making it impossible to accurately compare documents featuring different languages.

Here are the results of our comparisons:

### English

### Japanese

### Korean

### Chinese PRC

### Chinese Taiwan

### Arabic

## Fixed Bugs

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

This four-point scale will help you 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 disparity between feature functionality and internal specifications, memory leaks or data corruption. |
| **Minor** | A bug that leads to feature malfunctions or affects minor functionality of the software. 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 which 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, sorted 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 | Technology Subsystem |
| Critical | Unhandled exception on a customer's image with PageAnalysisParams::DetectBarcodes=true property set. | Document Analysis |
| Major | One of simultaneously running processes executing conversion to PDF in multi-processing mode crashes soon after start. | Export |
| Major | In 'PDF with MRC' scenario overriden image resolution is not applied. **Root cause**: FRE initializes not all internal parameters for MRC passed to Export subsystem. | API |
| Minor | Convertion to RTF 'Editable Copy'. The output contains more pages than original document even with the PageAnalysisParams::PaperSizeDetectionMode = PSDM\_CloseToImageSize parameter set. **Root cause**: DA makes a mistake in text block detection, Synthesis is not able to correct the DA mistake. | Synthesis |
| Minor | Incorrect table cell text indent on an image. | Document Analysis |
| Minor | Text frame has incorrect parameters on a customer's image. | Synthesis |
| Minor | FRE Java interop does not install itself properly being renamed. | Install |
| Minor | 'ト' is recognized as '卜' on a customer's image. **Workaround**: add undesired character to the list of prohibited characters. | Recognizer |
| Minor | Output contains no tabs between text line parts (words) as they are on a customer's image. | Synthesis |
| Minor | In case of Hebrew or any other right-to-left writing languages the property IsWordStart = true for the leftmost character instead of the rightmost. **Root cause**: the property was designed for left-to-right writing and does not work properly for reversed writing. **Workaround**: check for a word language and interpret the property value correspondingly. | Synthesis |
| Minor | Incorrectly recognized words in English in a document with a text mixture of English and Japanese words. | Recognizer |
| Minor | Incorrectly recognized words in English in a document with a text mixture of English and Chinese words. | Recognizer |
| Minor | Incorrect business cards splitting on customer images. | Document Analysis |
| Minor | In 'source PDF content reuse' scenario 'Franklin Gothic Heavy' fonts is taken for output instead of 'Arial' on a file provided by a customer. | Synthesis |
| Trivial | Incomplete article 'Using Text Type Autodetection'. | Help |
| Trivial | The 'Business Card Recognition' sample uses incorrect image preprocessing sequence that may lead to skew correction mulfunctioning. **Workaround**: do frDocument->Preprocess with pageSplittingParams->SplitType = PST\_BusinessCardSplit instead of frDocument->SplitPages. | Samples |
| Trivial | Java sample code for Engine Pool creation uses string variables where enums are more appropriate. | Samples |
| Trivial | The product documentation attribtes NrmlPart.ssc and NrmlPart.slp files as required for CJK OCR only, whereas they are required also for Arabic OCR. | Help |

## Known Issues and Workarounds

### Some API is not implemented

The following API is not implemented in FRE 10 and FRE 11:

• IFootnoteSeries::HasSeparator. Always returns “true”.

• ITextPicture::ColumnNumber. Always returns “0”.

• ICharParams::IsWordStart. Always returns “false”. It is true only for character parameters got through IWordRecognitionVariants interface.

• IIncut::TextWrapping. Always returns “TW\_Undefined”.

• IRunningTitlesSeriesText::HasSeparator. Always returns “false”.

The implementation is not planned.

### Farsi language is based on Arabic language

Farsi OCR output indicates Language ID as Arabic (CharParams::LanguageId = LI\_ArabicSaudiArabia). This is because Farsi technology preview is based on Arabic OCR language.

The fix will be available in FRE 12.

### IPDFMRCParams::MonochromeText doesn’t work correctly

Different algorithms of compressions during export to PDF are used with IPDFMRCParams::MonochromeText set to TRUE.

### ‘User Pattern Training Utility’ sample crashes for ‘FieldLevelRecognition’ profile

Run the sample, select ‘FieldLevelRecognition’ processing profile, recognize an image. The Text Editor will show “The page has been recognized, but no page view has been created. Click Read to complete the process.” message. A subsequent recognition command will result into IPE: “…\Interfaces\Implementations\inc\NotifyImpl.h, 62”.

The fix will be available in the next release.

### Infinite loop in Java crashes

In 32-bit JDK 8 Update 45, JDK 8 Update 51 applications infinite loop for ‘Hello’ sample processing crashes after 5-9 cycles.

The fix will be available in the FRE 11 R6 patch.

### Password protected PDF files couldn’t be opened in parallel processing mode

PDF file may be password protected and to open a file for processing a corresponding password must be provided. API has two possibilities:

• AddImageFileWithPassword, accepts a password as a parameter.

• AddImageFileWithPasswordCallback, calls for a password as soon as it is required.

We found that in parallel processing mode AddImageFileWithPasswordCallback scenario does not work properly.

The fix will be available in the FRE 11 R6 patch.

### JapaneseModern OCR language is not available for BCR

‘JapaneseModern’ OCR language is not available for BCR.

The fix will be available in the FRE 11 R6 patch.

### Language auto detection uses CJK resources though none is selected for OCR

‘Cjk.\*’ resource files are used while RecognizerParams::LanguageDetectionMode = TSPV\_Yes even though recognition languages set does not include any of CJK languages.

The fix will be available in the FRE 11 R7.

# R6 GM Patch –

## Part number, build number

|  |  |
| --- | --- |
| Part# | 1041/76 |
| Build# | 11.1.14.64 |

## New Features and Improvements

There are no new features.

## Performance results

The same as in FRE 11 R6.

## Fixed Bugs

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

This four-point scale will help you 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 disparity between feature functionality and internal specifications, memory leaks or data corruption. |
| **Minor** | A bug that leads to feature malfunctions or affects minor functionality of the software. 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 which 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, sorted 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 | Technology Subsystem | HD # | Office |
| Critical | In 32-bit JDK 8 Update 45, JDK 8 Update 51 applications infinite loop for ‘Hello’ sample processing crashes after 5-9 cycles. | API | 510416 | US |
| Major | In parallel processing mode AddImageFileWithPasswordCallback (password protected PDF files opening) scenario does not work properly. | API | n/a | n/a |
| Major | ‘JapaneseModern’ OCR language is not available for BCR. | API | n/a | n/a |
| Minor | IEngine::AvailablePredefinedLanguages returns no JapaneseModern OCR language. | API | n/a | n/a |

## Known Issues and Workarounds

### Some API is not implemented

The following API is not implemented in FRE 10 and FRE 11:

• IFootnoteSeries::HasSeparator. Always returns “true”.

• ITextPicture::ColumnNumber. Always returns “0”.

• ICharParams::IsWordStart. Always returns “false”. It is true only for character parameters got through IWordRecognitionVariants interface.

• IIncut::TextWrapping. Always returns “TW\_Undefined”.

• IRunningTitlesSeriesText::HasSeparator. Always returns “false”.

The implementation is not planned.

### Farsi language is based on Arabic language

Farsi OCR output indicates Language ID as Arabic (CharParams::LanguageId = LI\_ArabicSaudiArabia). This is because Farsi technology preview is based on Arabic OCR language.

The fix will be available in FRE 12.

### IPDFMRCParams::MonochromeText doesn’t work correctly

Different algorithms of compressions during export to PDF are used with IPDFMRCParams::MonochromeText set to TRUE.

### ‘User Pattern Training Utility’ sample crashes for ‘FieldLevelRecognition’ profile

Run the sample, select ‘FieldLevelRecognition’ processing profile, recognize an image. The Text Editor will show “The page has been recognized, but no page view has been created. Click Read to complete the process.” message. A subsequent recognition command will result into IPE: “…\Interfaces\Implementations\inc\NotifyImpl.h, 62”.

The fix will be available in the next release.

### Language auto detection uses CJK resources though none is selected for OCR

‘Cjk.\*’ resource files are used while RecognizerParams::LanguageDetectionMode = TSPV\_Yes even though recognition languages set does not include any of CJK languages.

The fix will be available in the FRE 11 R7.

### PDF/A validation report

The following issues are detected by Adobe Acrobat 11.0.9 (Preflight 11.0.9) for PDF/A files produced by this release:

1. In SourceContentReuseMode = CRM\_ContentOnly the Engine sometimes (5% of the test base in Korean language) generates PDF/A-2a and PDF/A-3a files which Adobe Acrobat Preflight validates with the following message “Text is mapped to Unicode Private Use Area but no ActualText entry is present”.

**R7 GM –**

## Part number, build number

|  |  |
| --- | --- |
| Part# | 1041/89 |
| Build# | 11.1.16.94 |

## New Features and Improvements

### Conversion documents to searchable PDFs at full throttle requires no page numbers setup any more

It is possible to convert documents containing undefined number of pages to searchable PDF using ExportFileWriter. The PagesCount parameter must be set to -1 in this case (this parameter is deprecated and will be removed in future versions).

We have introduced this feature in FRE 11 in the R6 release. And from now our customers have no necessity to know an amount of pages in the document at the moment of creating a session of recognition.

It can be useful for effective work with scanners when one has to process a lot of pages and doesn’t know the amount of them in the document until the end of scanning, but needs the processing to start before the scanning will be finished.

### Simultaneous network and standalone usage of one FRE installation

Now it is possible to define network and standalone licenses in one LicensingSettings.xml file.

This possibility was important for one of our customers when he decided to put his application into a network folder and let local (to a host of the folder) and remote (other PC in the same network) end-users run the same copy of the application. It is reasonable that ‘local’ end-users should use a standalone FRE license, whereas ‘remote’ end-users should use a network FRE license. To do so we specified two licensing configurations (standalone and network) in one LicensingSettings.xml file.

### Garbage removal from color images

New method ImageDocument::RemoveGarbageEx works with both color and white-and-black images. From now it is possible to remove garbage from color images using this method.

|  |  |
| --- | --- |
| Input image | Output image |
| C:\Users\KPavlov\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1input.png | C:\Users\KPavlov\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1output.png |
| C:\Users\KPavlov\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2input.png | C:\Users\KPavlov\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2output.png |
| C:\Users\KPavlov\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3input.png | C:\Users\KPavlov\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3output.png |

Dealing with black-and-white images the method works the same way as ImageDocument::RemoveGarbage which will be removed in future versions of FRE.

### Possibility to inject text layer to selected pages of PDF documents

A new method IEngine::InjectTextLayerEx allows to process specific pages in the "image only" or "image on text" PDF documents. It creates a searchable PDF file which contains the same page images and the invisible text layer created from the recognized text of the document.

This new method works the same way as IEngine::InjectTextLayer and has new arguments:

• PageIndices. This parameter refers to the IntsCollection object which specifies the indices of the document pages, to which the text will be injected. This parameter is optional and may be 0, in which case the text will be injected to all the pages of the document.

• ProcessingEvents. Refers to the interface of the user-implemented object that is used for reporting events to the listeners. This parameter may be 0, in which case no callback will be attached.

The IEngine::InjectTextLayer method is now deprecated and will be removed in future versions.

### A new article about working with screenshots in documentation

An article ‘I am working with the screenshot image. Are there any special recommendations for screenshot processing?’ about suitable settings of FRE for screenshot processing was added. It is in the FAQ section.

## Performance results

This section contains performance results of the release comparing to the previous releases. The processor of the testing machine is Intel® Core™ i5-4690 CPU (3.50GHz, 4 physical cores) with 8GB of RAM.

### English

### Japanese

### Korean

### Chinese PRC

### Chinese Taiwan

### Arabic

## Fixed Bugs

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

Four-point scale will help you to evaluate the severity of each issue, enabling you to make informed decision 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 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 following table contains bugs fixed in this release sorted 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 | Technology Subsystem | HD # | Office |
| Critical | Internal program error occurs when document structure is being read and IWord::get\_Region is being used. | API | 526665 | RU |
| Critical | In multi-processing mode on Java the program crashes with ‘CoInitialize has not been called’ error after a long time of work when a lot of threads process the documents in infinite cycle.  **Root cause**: when big amount of Java-threads is created, Java reuses internal threads of OS to its Java-threads. In this case two Java-threads can process using one thread of OS. And when one of the Java-threads finishes its work and deinicialize COM, the second one is still trying to use this thread. So, an error occurs.  **Workaround**: not to use unlimited number of threads | API | 542063 | RU |
| Critical | An error ‘Access violation’ occurs during the processing our customer’s input file without skew correction during image preparation and without pictures detection during layout analysis. | Recognizer | 535364 | EU |
| Critical | Use method FRPage::CorrectGeometricalDistortions, then on trying to look for vertically oriented text (PageAnalysisParams::DetectVerticalEuropeanText = true) internal program error occurs on our customers’ PDF file. | Document Analysis | 529645 | EU |
| Critical | If region is laid on the customer’s image and AnalyzeRegion method is used, internal program error occurs. | Document Analysis | 511192 | EU |
| Critical | In case of Network license usage, an error occurs when connection with server is lost after Engine has been loaded even if the interface for managing the connection breakdowns is used.  **Root cause**: some problem situations with server were not processed in the right way and this is why the OnDisconnect() method was not called in this cases. | API | 528727 | US |
| Critical | Internal program error occurs on opening customer’s image file with incorrect color masks. | Image | 519469 | RU |
| Critical | Internal program error occurs on the analysis stage of image processing on specific customer’s image.  **Workaround**: process image without choice of variants of page layout (ProhibitModelAnalysis = true) | Document Analysis | 512057 | EU |
| Critical | Internal program error occurs during the document analysis of specific customer’s image with active orientation correction but without skew correction. | Document Analysis | 537863 | EU |
| Critical | Internal program error occurs when a document analysis is processing on specific customer’s document. | Barcodes | 537010 | EU |
| Critical | An internal program error occurs when one tries to get a ParagraphAlignment value for Chinese text. | API | 548896 | EU |
| Major | ‘Cjk.\*’ resource files are demanded by FRE on processing with automatic language detection even though recognition languages set does not include any of CJK languages. | Document Analysis | 509942 | EU |
| Major | Throwing an exception when IParagraph::NextGroup() method gets SF\_BackgroundColor or SF\_BaseLineRise flags in StyleFlagMask parameter.  **Root cause**: an internal enumerator for recognizing the mask for style parameters didn’t contain necessary elements. | API | 543033 | US |
| Major | Orientation is fixed only for the first 30 pages of the document when the page flushing policy is set to automatic mode (IFRDocument::PageFlushingPolicy = PFP\_Auto), FREngine is set to distribute analysis and recognition of multi-page documents to CPU cores in automatic mode (IMultiProcessingParams::MultiProcessingMode = MPM\_Auto) and the page orientation is detected during page preprocessing (IPagePreprocessingParams::CorrectOrientation = true).  **Workaround**: set ABBYY FineReader Engine not to distribute analysis and recognition of multi-page documents to CPU cores. (IMultiProcessingParams::MultiProcessingMode = MPM\_Sequential) | API | 535634 | US |
| Major | In case of processing PDF file using only text layer, when the pages of the document are recognized sequentially (MPM\_Sequential) with image skew correction (CorrectSkew = TSPV\_Yes), the output file doesn’t contain a text layer.  **Workaround**: use CorrectSkew = TSPV\_Auto and/or MPM\_Parallel mode  **Root cause**: in technologies when the flag CorrectSkew is active, the copy of document is created with loss of information about text layer. | API | 530146 | US |
| Major | In case of multi-page documents, when the images are being recognized with BatchProcessor in parallel processes, pages of different documents can be mixed in output of the method.  **Root cause**: while the BatchProcessor is used together with ExportFileWriter, the following situation was possible: the page has already been deleted from indices of unissued pages set, but has not been deleted from the document yet. The next pages are put to the indices set at and to the temp document with the same indices. The page order is mixed. | API | 529263 | US |
| Major | Memory leaks in the scenario: FRE initialization, image processing in cycle with default settings, FRE deinitialization. | Recognizer | 535895 | EU |
| Minor | In case of loading image with saving the coordinates on the image after modifications (PrepareImageMode:: KeepOriginalCoordinatesInfo = false), on saving the recognized text from the PlainText object into an XML file, Export saves coordinates on original image instead of transformed image.  **Root cause**: SaveToAsciiXMLFile() method couldn’t recognize what coordinates should be used during the export: coordinates on a non-deskewed image or coordinates on deskewed image; and always saved coordinates on a non-deskewed image. | API | 529807 | US |
| Minor | In case of export to XML with words variants saving, several characters for one wordRecVariant have same coordinates.  **Root cause**: export has no information about character variants’ rectangles that are not equal to the basic variant’s rectangle. So, output file contains same rectangles for every characters’ varints. | Export | 523884 | US |
| Minor | In case of export to ALTO format in mode with saving characters’ coordinates on image with skew correction (AltoExportParams::WriteNondeskewedCoordinates = FALSE), <Shape> elements are missed. | API | 520823 | US |
| Minor | When the document is processed using Farsi language, most of characters have CharConfidence = -1  **Root cause**: for Farsi language the normalized weight was not calculated correctly | Recognizer | 512578 | EU |
| Minor | Mobile phone number subscribed as ‘wireless’ recognized as BCFT\_Phone, not BCFT\_Mobile.  **Root cause**: ‘wireless’ was not related to mobile phone number. | Document Analysis | 538150 | US |
| Minor | Images in the customer’s document are detected even with settings when pictures are not detected during the layout analysis. | Document Analysis | 528220 | EU |
| Minor | QR-code is not detected on customer’s document processing.  **Root cause**: the edges of the dots are too noisy. | Barcodes Detector | 518784 | RU |
| Minor | A part of the text is not recognized on customer’s document.  **Root cause**: this area is recognized as an image. | Document Analysis | 513141 | 3A |
| Minor | The output DOCX file contains not all of horizontal lines. | Synthesis | 508370 | US |
| Minor | Numbers like ’60.01’ at the beginning of the raw are synthesized as elements of the list during the export to rtf format.  **Root cause**: Synthesis recognized numerators like ‘01’ or ‘7.05’ as list elements. | Synthesis | 540540 | 3A |
| Minor | Numbers like ’60.01’ at the beginning of the raw are synthesized as elements of the list during the export to rtf format. | Synthesis | 540540 | 3A |
| Minor | In case of export to PDF with embedded fonts and no predefined filters, the text layer is narrower than characters border. | Export | 546487 | EU |
| Trivial | The default value of IBackgroundLayer::IsPicture property is FALSE, but it is specified to be TRUE in documentation. | Help | 546581 | US |
| Trivial | The tab character is always replaced by one character during the processing.  **Root cause**: an internal flag was active with no opportunity to switch it off. | API | 537658 | 3A |
| Trivial | Incorrect sample in the ‘Document Classification’ article. Classify method must return ClassificationClasses object instead of StringsCollection. | Help | 520210 | EU |
| Trivial | The EnginePool sample returns ‘There are no free licenses’ when the Trial license with CPU limit is used. | Samples | 451719 | US |
| Trivial | Generated linearized PDF/A-1a file doesn’t pass the compliance test in Preflight. | Export | 546220 | US |
| Trivial | In case of Runtime installation of FREngine, the SN parameter is not obligated and should be deleted from several articles related to installation in documentation. | Samples | 545923 | EU |

## Known Issues and Workarounds

### Some API is not implemented

The following API is not implemented in FRE 10 and FRE 11:

* IFootnoteSeries::HasSeparator. Always returns “true”.
* ITextPicture::ColumnNumber. Always returns “0”.
* ICharParams::IsWordStart. Always returns “false”. It is true only for character parameters got through IWordRecognitionVariants interface.
* IIncut::TextWrapping. Always returns “TW\_Undefined”.
* IRunningTitlesSeriesText::HasSeparator. Always returns “false”.

The implementation is not planned.

### Farsi language is based on Arabic language

Farsi OCR output indicates Language ID as Arabic (CharParams::LanguageId = LI\_ArabicSaudiArabia). This is because Farsi technology preview is based on Arabic OCR language.

The fix will be available in FRE 12.

### IPDFMRCParams::MonochromeText doesn’t work correctly

Different algorithms of compressions during export to PDF are used with IPDFMRCParams::MonochromeText set to FALSE or DEFAULT whereas default value is FALSE. (HelpDesk request 416986)

### ‘User Pattern Training Utility’ sample crashes for ‘FieldLevelRecognition’ profile

Run the sample, select ‘FieldLevelRecognition’ processing profile, recognize an image. The Text Editor will show “The page has been recognized, but no page view has been created. Click Read to complete the process.” message. A subsequent recognition command will result into IPE: “…\Interfaces\Implementations\inc\NotifyImpl.h, 62”.

The fix is planned to be implemented in the next release with low priority because the scenario is not correct ant there is a workaround for it.

### The process hangs when the multiprocessing methods works on the machine with one CPU.

The machine has one core, IMultiProcessingParams::MultiProcessingMode = MPM\_Parallel and IMultiProcessingParams::RecognitionProcessesCount = 2 settings are set, operation system has radio System Properties->Advanced->Performance->Settings...->Advanced->Processor scheduling set to Background services value. The process hangs.

### PDF/A validation report

The following issues are detected by Adobe Acrobat 11.0.9 (Preflight 11.0.9) for PDF/A files produced by this release:

In SourceContentReuseMode = CRM\_ContentOnly the Engine sometimes (5% of the test base in Korean language) generates PDF/A-2a and PDF/A-3a files which Adobe Acrobat Preflight validates with the following message “Text is mapped to Unicode Private Use Area but no ActualText entry is present”.

**R7 GM Patch –**

## Part number, build number

|  |  |
| --- | --- |
| Part# | 1041/90 |
| Build# | 11.1.16.94 with Export.Pdf.dll from 11.1.16.96 |

## New Features and Improvements

None

## Performance results

This section contains performance results of the release comparing to the previous releases. The processor of the testing machine is Intel® Core™ i5-4690 CPU (3.50GHz, 4 physical cores) with 8GB of RAM.

### English

### Japanese

### Korean

### Chinese PRC

### Chinese Taiwan

### Arabic

## Fixed Bugs

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

Four-point scale will help you to evaluate the severity of each issue, enabling you to make informed decision 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 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 following table contains bugs fixed in this release sorted 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 | Technology Subsystem | HD # | Office |
| Minor | The time of exported file’s creation/modification displayed in Adobe Reader is not equal to this file’s creation/modification time displayed in file properties. | Export | 530461 | US |

## Known Issues and Workarounds

Same as R7.

**R8 GM –**

## Part number, build number

|  |  |
| --- | --- |
| Part# | 1041/104 |
| Build# | 11.1.19.48 |

### Change in Behavior

We have changed the pattern of working with redundancy time. From now in LicensingSettings.xml there is no opportunity to set redundancy live time (RedundancyLiveTime parameter) and redundancy synchronization time (RedundancySynchronizationTime parameter).

Redundancy live time was fully moved to serial number parameters. Redundancy synchronization time is an internal parameter that was available for editing by mistake.

## New Features and Improvements

### Orientation and skew correction using InjectTextLayer method

All settings in the PrepareImageMode and PageProcessingParams opject such as CorrectSkew and CorrecrOrientation work when passed to InjectTextLayer method.

Sometimes customers have PDFs that contain a mixture of digitally published files and scans. Pages that are digitally published do not need any correction, but scans may need it.

From FRE 11 R8, the new InjectTextLayerEx2 method is available. This method works the same as InjectTextLayerEx, but it additionally can correct orientation and skew of input images.

New TextLayerInjectionParams object allows to tune the parameters of processing the input "image only" or "image on text" PDF files and creation of a searchable PDF file using InjectTextLayerEx2 method.

### An opportunity to check PDF files that placed in memory for text layer

Sometimes it is necessary to work with InputStream for working with the files.

Previous versions of FRE 11 has only method IsPdfWithTextualContent for checking PDFs for text layer that required a string for FileName. As the customer works with InputStream it is necessary to pass a byte array as the first parameter for the method. In order to be able to check the files earlier they wrote the stream first in a temp file, but it is not acceptable for them since it would have performance impact.

Now method IsPdfWithTextualContentFromStream accepts FileName also as a byte array besides a string.

### An opportunity not to write BOM on export to TXT

FRE 11 always wrote Byte Order Mark (BOM) in case of export to TXT.

UTF-8 permits the BOM, but does not require or recommend its use. Byte order has no meaning in UTF-8 and is used only to signal at the start that the text stream is encoded in UTF-8.

Java's UTF-8 encoding has a known behavior and does not recognize this character as a BOM; the result of reading such a stream is a set of characters beginning with FEFF.

So, there are scenarios when clients may need not to write BOM in case of export to TXT. New export option WriteBomCharacter that solves this problem is available since FRE 11 R8.

### Human-readable names of paragraph style on export to XML

Default values of names of paragraph styles were modified and now they can be easier read by a human. Now the names are formed basing on a role of the paragraph and modifications, applied to the style. Earlier GUID was used for naming the styles and it is was not informative.

Users also have an opportunity to set paragraph style name manually using IParagraphStyle::Name method.

### An opportunity to rasterize FreeText annotations

In earlier versions of FRE it was not possible to retain annotations that have FreeText type in the output document. Now users can do it using new property RasterizeFreeText of IPrepareImageMode.

### Added ‘Not enough disk space’ error code

A new error message ‘Not enough disk space’ for cases when there is not enough space on the disk was added.

### New option for faster printing of PDF using MRC

Some our customers complained on too slow printing of PDFs with Mixed Raster Content.

FRE 11 R8 has new option UseMultipleMasks of IPDFMRCParams for tuning MRC parameters. This option activates a special mode in which different monochrome masks are used instead of one multicolor. This leads to faster document printing.

This mode allows to create only documents with monochrome characters. E.g. it is not possible to create documents with characters with a gradient filling using the new option.

The feature is implemented as technical preview and has not been tested for wide use. So, usage of new mode can cause some issues.

### New article about OMR/Barcode licenses in documentation

An article ‘I am working with the screenshot image. Are there any special recommendations for screenshot processing?’ about suitable settings of FRE for screenshot processing was added. It is in the FAQ section.

## Performance results

This section contains performance results of the release comparing to the previous releases. The processor of the testing machine is Intel® Core™ i5-4690 CPU (3.50GHz, 4 physical cores) with 8GB of RAM.

### English

### Japanese

### Korean

### Chinese PRC

### Chinese Taiwan

### Arabic

## Fixed Bugs

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

Four-point scale will help you to evaluate the severity of each issue, enabling you to make informed decision 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 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 following table contains bugs fixed in this release sorted 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 | Technology Subsystem | HD # | Office |
| Critical | Internal program error occurs when document structure is being read and IWord::get\_Region is being used. | Document Analysis | 564859 | US |
| Critical | Internal program error occurs in Visual Components when client tries to merge and then split the paragraphs. | Recognizer | 573385 | US |
| Major | Inability to move a table region using Move method. | Document Processing | 570026 | RU |
| Major | AV error occurred on Analyze step on client’s document  **Workaround**: set the parameters for not to reuse content and to detect text on pictures (SourceContentReuseMode = CRM\_DoNotReuse, DetectTextOnPictures = true) | Image | 569835 | EU |
| Minor | FreeText annotation in PDF was lost during the processing. | Image | 523432 | US |
| Minor | Properties of TabPosition object that was created using TabPositions::AddNew had always have the default values. | API | 560979 | EU |
| Minor | An error on PDF documents with annotations of CPDFNullObjectImpl type. | API | 567061 | US |
| Minor | In ‘Classification’ in Demo Tools an AV error occurred if one tried to add to class a PDF file as an image.  **Root cause**: APDFL must be used in the same thread where FRE was initialized. | Samples | 561740 | US |
| Minor | Incorrect recognizing of barcodes when barcodes without start and stop symbols are processed (IsCode39WithoutAsterisk = TRUE).  **Root cause**: incorrect order of hypothesizes | Barcodes | 555941 | EU |
| Minor | On client’s document a part of the text was detected as images.  **Workaround**: prohibit image searching using IObjectsExtractionParams::DetectTextOnPictures=true | Document Analysis | 551011 | 3A |
| Minor | Too long processing of document with extremely big amount of dashes | Document Analysis | 551484 | EU |
| Minor | Several rows of the table detected as one. | Document Analysis | 565558 | 3A |
| Minor | On client’s document on processing with settings for detection all text on an image, including text embedded into images (ObjectsExtractionParams::DetectTextOnPictures = true), text was detected an image. | Document Analysis | 569559 | US |
| Minor | In ImageOnText mode ReplaceUncertainWordsWithImage property was not ignored despite the documentation says it must be ignored. | PDF Export | 558695 | US |
| Minor | On recognizing digits, the ‘^’ character occurs instead of space character.  **Root cause**: the text layer of the source PDF file contains the character 65533 (FFFD) aka 'REPLACEMENT CHARACTER', which is used by our OCR engine to show the UNKNOWN (unrecognized) symbol. So, the character was used as the whitespace symbol in the PDF document (1 067,00), but our Engine shows it as '^' (1^067,00).  **Workaround**: postprocessing to replace the circumflex with the whitespace character. | Image | 569910 | EU |
| Minor | Method for straightening out distorted lines on an image (CorrectGeometricalDistortions) works incorrect on photo of receipt: the photo is stretched and non-readable. | Image | 567855 | RU |
| Minor | On exporting to PDF with parameters of page size set to be equal to the original image size, size of output page is a little bigger. | PDFToolkit | 535881 | US |
| Minor | FRE 11 returns incorrect coordinates of characters on image.  **Root cause**: internal error in algorithm for detection of rows’ coordinates in case of rows consisting of one fragment. | Recognizer | 557988 | US |
| Minor | Four dots in a row were recognized as underscore or hyphen. | Recognizer | 568536 | 3A |
| Minor | On client’s document ‘Ill’ was recognized as ‘111’. | Recognizer | 569751 | 3A |
| Minor | On recognizing text with underlined words, words in the last row are not underlined in the output file. | Synthesis | 569043 | 3A |
| Trivial | Wrong id for Maori language. | API | 569449 | EU |

## Known Issues and Workarounds

Same as R7.

**R8 Patch GM –**

## Part number, build number

|  |  |
| --- | --- |
| Part# | 1041/111 |
| Build# | 11.1.19.49 |

## New Features and Improvements

Same as R8.

## Performance results

Same as R8.

## Fixed Bugs

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

Four-point scale will help you to evaluate the severity of each issue, enabling you to make informed decision 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 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 following table contains bugs fixed in this release sorted 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 | Technology Subsystem | HD # | Office |
| Critical | "Integer division by zero" error caused by recognizing barcode area on documents with manually changed orientatoin. | Barcodes | 586835 | EU |
| Major | Problem with zero license counter in case of license written (activated) onto a Wibu key Update. | Protection | 590620 | 3A |
| Minor | Internal program error occurs in Visual Components when client tries to merge and then split the paragraphs. | PDF Export | 582150 | US |

## Known Issues and Workarounds

Same as R7.

**R8 Patch 2 GM –**

## Part number, build number

|  |  |
| --- | --- |
| Part# | 1041/117 |
| Build# | 11.1.19.59 |

## New Features and Improvements

### Changes in behavior

#### an ability to save license counters value on license

A new section 'ZeroLevel' was added into License Wizard for FCE 11. The only one available option in this section is ZeroLevel.

If this option is activated, a license counter value will be saved in ABBYY Registration Server during the license deactivation. On further activation counter will take value from ABBYY Registration Server.

## Performance results

This section contains performance results of the release comparing to the previous releases. The processor of the testing machine is Intel® Core™ i5-4690 CPU (3.50GHz, 4 physical cores) with 8GB of RAM.

Test batches for different languages may include significantly different documents, making it impossible to accurately compare documents featuring different languages.

### English

### Japanese

### Korean

### Chinese PRC

### Chinese Taiwan

### Arabic

## Fixed Bugs

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

Four-point scale will help you to evaluate the severity of each issue, enabling you to make informed decision 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 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 following table contains bugs fixed in this release sorted 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 | Technology Subsystem | HD # | Office |
| Critical | Access violation error in case of processing several business cards in a row with calling Close() method for every business card. | API | 582481 | US |
| Critical | An error occurs when specific customer’s document is processed with skew correction. | DA | 600099 | US |
| Critical | Internal program error occurs in case of recognition a document with noised background on Hebrew. | DA | 596597 | UA |
| Critical | An error occurs when specific customer’s document is recognized using aggressive text extraction mode. | DA | 581189 | EU |
| Minor | Incorrect detection of separator role on specific customer’s document. | Synthesis | 537006 | US |
| Minor | In case of export to RTF, using synthesis in Editable Copy mode, text blocks in an output files intersects each other. | Synthesis | 600479 | US |
| Minor | Too long closure of FRDocument when PFP\_KeepInMemory page flushing policy is used and large document is processed. | API | 580213 | US |
| Minor | Vertical CJK text is displayed as rotated on 90 degrees lines, not column of characters, in case of export to RTF. | Export | 585885 | EU |
| Minor | On transferring ImageDocument from FCE to FRE using OutprocLoader, license counter decreases again when ImageDocument is re-recognized in FRE. | API | 597379 | EU |
| Minor | A sequence of dots is not recognized of recognized as a sequence of dashes. | Recognizer | 568536 | 3A |
| Minor | Page orientation is not recognized on a specific customer’s document. | DA | 587222 | UA |
| Minor | On recognition of PDF document with default settings, digits is recognized wrong when a font that looks like Courier is used. | Recognizer | 595504 | EU |
| Minor | Error in Visual Components sample occurs in case of deleting a large block of text from any page of recognized customer’s document. | TextLayout | 584301 | US |
| Minor | Wrong page orientation detection on specofoc customer’s document | DA | 585707 | US |
| Minor | Specific customer’s document is not opened. Other viewers can open the document. | Image | 588641 | US |
| Minor | Wrong footnote detection on synthesys stage on specific customer’s document. | Synthesis | 551264 | EU |
| Trivial | Documentation has not represented that in case of using CSS on exporting to HTML, it is not possible to insert <HR> tags between exported pages. | Help | 599496 | US |
| Trivial | Incorrect information about default value for MultiProcessingMode in case of using OutprocLoader. | Help | 597077 | EU |
| Trivial | Wrong spelling of Tabassaran language name in the documentation. | Help | 549609 | EU |

## Known Issues and Workarounds

### Some API is not implemented

The following API is not implemented in FRE 10 and FRE 11:

* IFootnoteSeries::HasSeparator. Always returns “true”.
* ITextPicture::ColumnNumber. Always returns “0”.
* ICharParams::IsWordStart. Always returns “false”. It is true only for character parameters got through IWordRecognitionVariants interface.
* IIncut::TextWrapping. Always returns “TW\_Undefined”.
* IRunningTitlesSeriesText::HasSeparator. Always returns “false”.

The implementation is not planned.

### Farsi language is based on Arabic language

Farsi OCR output indicates Language ID as Arabic (CharParams::LanguageId = LI\_ArabicSaudiArabia). This is because Farsi technology preview is based on Arabic OCR language.

The fix will be available in FRE 12.

### IPDFMRCParams::MonochromeText doesn’t work correctly

Different algorithms of compressions during export to PDF are used with IPDFMRCParams::MonochromeText set to FALSE or DEFAULT whereas default value is FALSE. (HelpDesk request 416986)

### ‘User Pattern Training Utility’ sample crashes for ‘FieldLevelRecognition’ profile

Run the sample, select ‘FieldLevelRecognition’ processing profile, recognize an image. The Text Editor will show “The page has been recognized, but no page view has been created. Click Read to complete the process.” message. A subsequent recognition command will result into IPE: “…\Interfaces\Implementations\inc\NotifyImpl.h, 62”.

The fix is planned to be implemented in the next release with low priority because the scenario is not correct ant there is a workaround for it.

### The process hangs when the multiprocessing methods works on the machine with one CPU.

The machine has one core, IMultiProcessingParams::MultiProcessingMode = MPM\_Parallel and IMultiProcessingParams::RecognitionProcessesCount = 2 settings are set, operation system has radio System Properties->Advanced->Performance->Settings...->Advanced->Processor scheduling set to Background services value. The process hangs.

### Internal program error on exporting PDF using MRC in MaxSpeed scenario

Internal program error occurs on export to PDF in case of using Mixed Raster Content with several monochrome masks instead of one multicolor background plane and one mask (IPDFMRCParams::UseMultipleMasks = true) in MaxSpeed scenario (IPDFExportParams::Scenario = PES\_MaxSpeed).

### PDF/A validation report

The following issues are detected by Adobe Acrobat 11.0.9 (Preflight 11.0.9) for PDF/A files produced by this release:

In SourceContentReuseMode = CRM\_ContentOnly the Engine sometimes (5% of the test base in Korean language) generates PDF/A-2a and PDF/A-3a files which Adobe Acrobat Preflight validates with the following message “Text is mapped to Unicode Private Use Area but no ActualText entry is present”.