R1 GM –

PART NUMBER, BUILD NUMBER

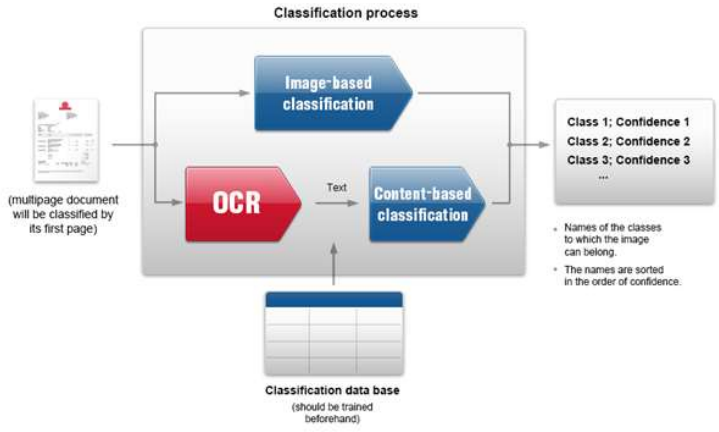
Part#	1041/17
Build#	11.1.4.118

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)	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

	○ 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.	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.  The following fields can be extracted: • Personal name	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

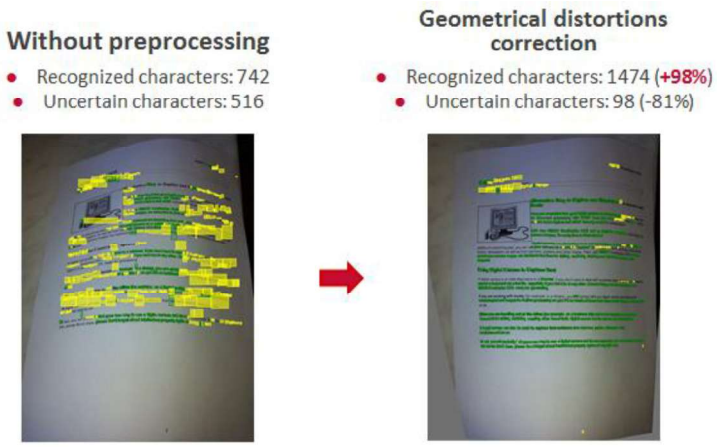
	• Company name • Position in the company • Company address • Phone number • Fax • Mobile phone number • E-mail • Web site **Export to vCard format** • Recognized data can be saved in vCard format, which is often used to pass business card by e-mail or networks. **Business Card reading in 27 languages** • 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.	

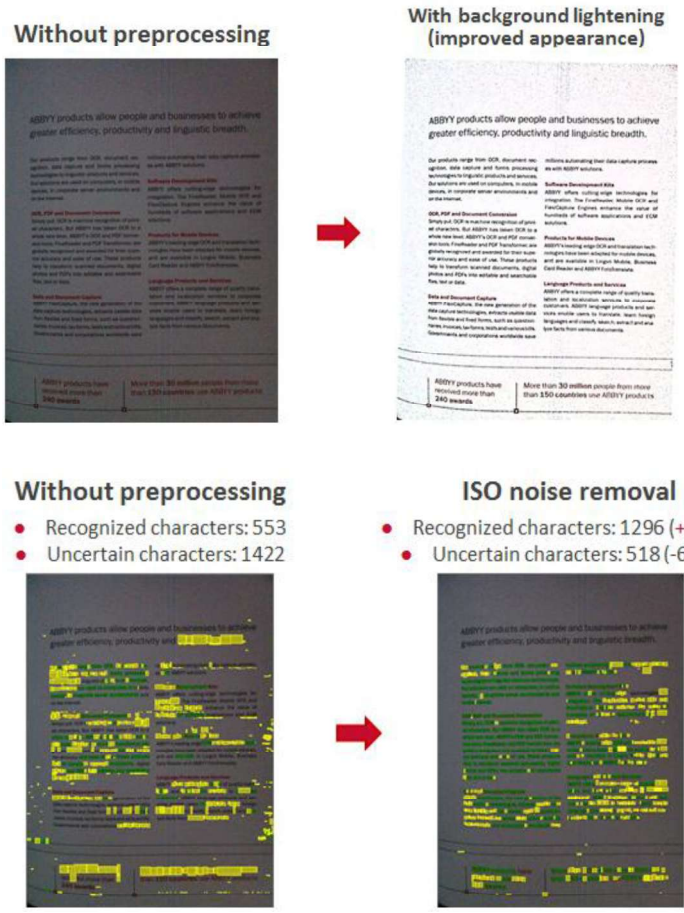
OCR/ICR IMPROVEMENTS

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 	• 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 	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 	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. 	

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: 40304885 WEST SILVER 3,80 2 9900000000021049000555 6 Vensafe ********* Alterskontrolle **********	
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	Better recognition results

	quality of low contrast images.	

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

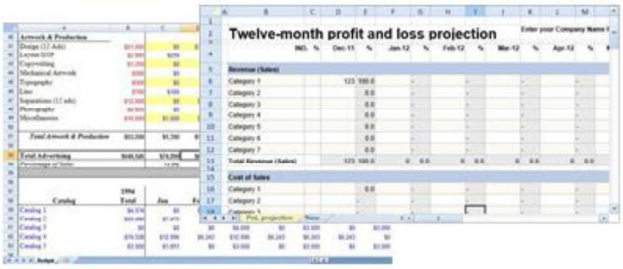
Feature	Description	Benefits
Improved PDF Export		
Faster PDF Export	Export to PDF now is up to 12% faster than in previous version of FineReader Engine.	Converting images into searchable PDF is one of the most needed scenarios on the market. Better appearance and minimum size of the document.
Higher 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*	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	

development	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
 	
PDF association
	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.*	

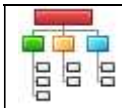
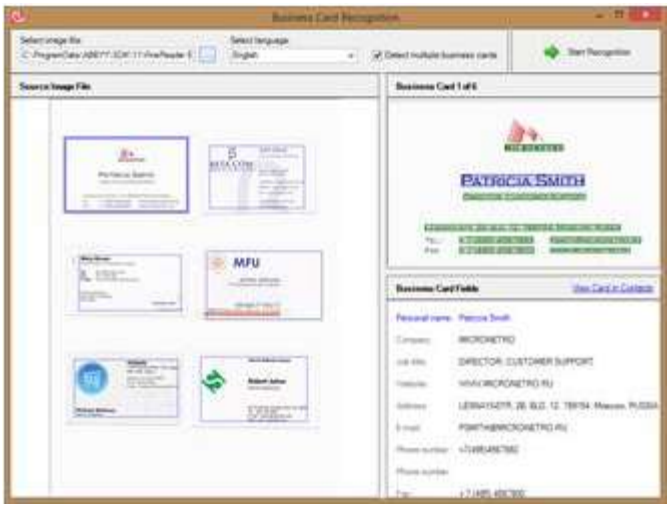
Export to PDF/A-3* PDF association ** still under development*	PDF/A-3 is an extension of the A-2 Standard and it allows including not only PDF/A files, but also all kind of other binary formats, like XML or Office formats. Long-term archiving and readability of the PDF/A part is still guaranteed, and 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 PDF files have internal data arranged in a page order. A page of	

linear or non-linear PDF files	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	This sample demonstrates how ABBYY FineReader Engine can be	• Ready-to-use UI for your

	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.	application • Quick learning curve • No need in specialized training courses
Business Card Recognition 	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	This sample was updated with the following new options to play with: • Correction of geometric distortions	• Select the required image pre-processing technologies for your particular document type

	• Image crop • Remove noise • Image contrast	
New How to…		
Convert a batch of documents 	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 	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 	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 operation and repeat processing of a page with Batch Processor • FineReader Engine collections can be iterated using the foreach statement 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.

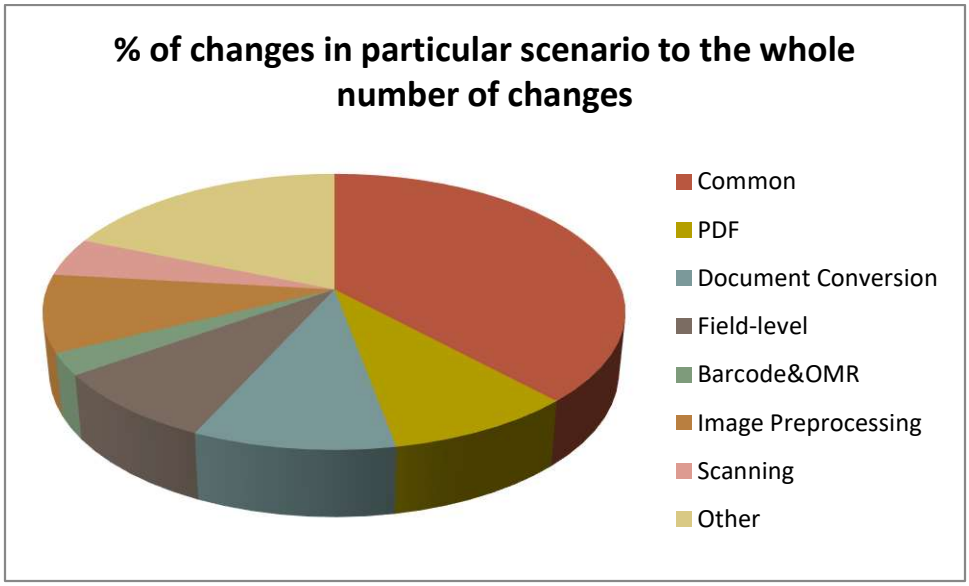
Short summary of API changes

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:

Possible limitations for standalone license are:

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

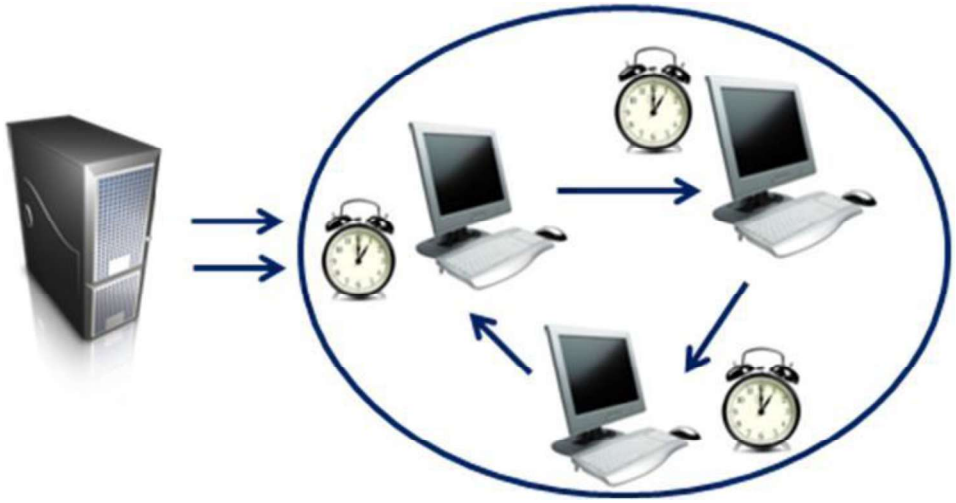
For example:

- 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 runtime licenses 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).

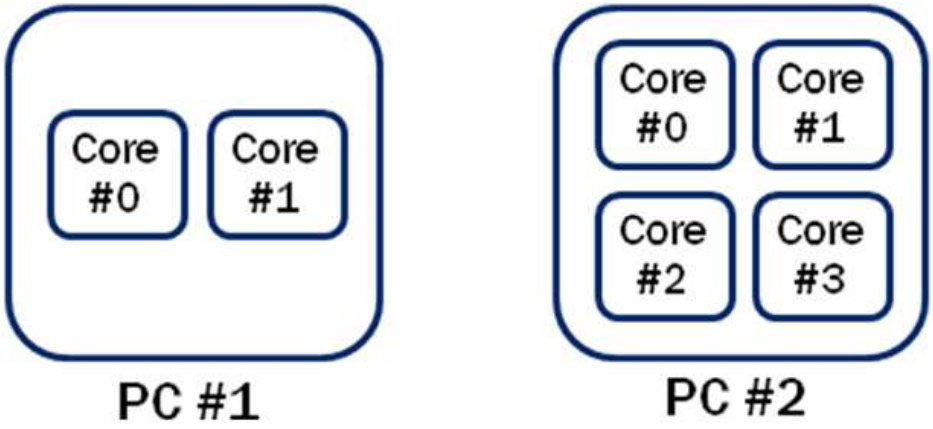
By CPU Cores

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:

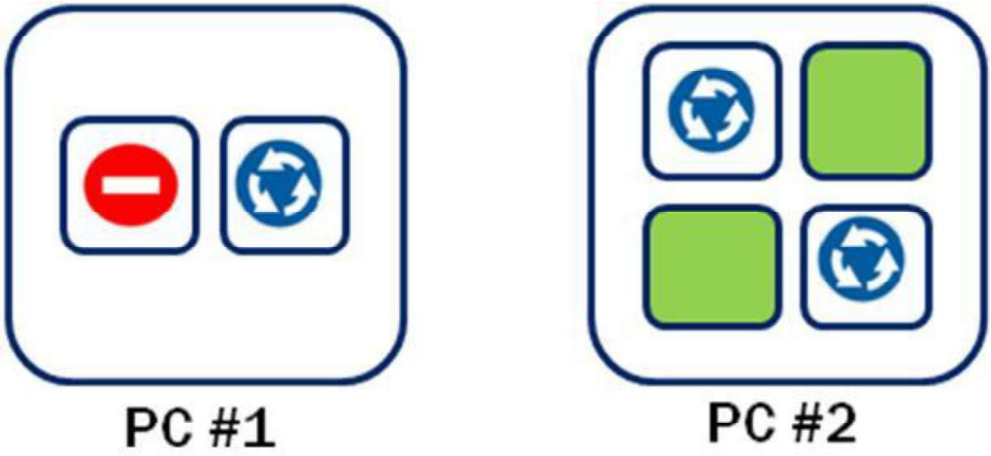
- 1-core voucher – if we use CPU cores limitation in license, customer pays for and we provide the precise number of vouchers, one for each core.
- Unlimited voucher – if number of cores is unlimited and we use other types of limitations in license.

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

SUPPORTED ACTIVATION TYPES

A serial number is required to activate the product.

1. Software protection keys activation types:
    a. Via the Internet;
    b. By e-mail;
    c. By e-mail from another computer.
2. Hardware protection keys activation types:
    a. Pre-activation at an ABBYY local distribution office (using Dongle Update utility);
    b. 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
			185, included in Runtime Professional

		• **2** special languages (included by default if corresponding text type is chosen)
		  ○ CMC7
		  ○ E13B
		• **6** programming languages
		• Chemistry
		• Digits	

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:
    a. DOCX
    b. XLS/XLSX
    c. PPTX
3. PDF file formats
    a. PDF
    b. PDF/A (1b, 1a, and NEW 2a and 2u)
    c. 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:

-.ЁІАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯабвгдежзийклмнопрстуфхцчшщъыьэюяёіѢѣѲѳVv

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:
   8.1. the file is language-specific. Include it if you need this language (consult columns 6 and 7).
   8.2. 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:

9. Path — file path in the root installation folder.
10. FileName — file name.

11. 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).

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:

CLI_export_patch.zip

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.