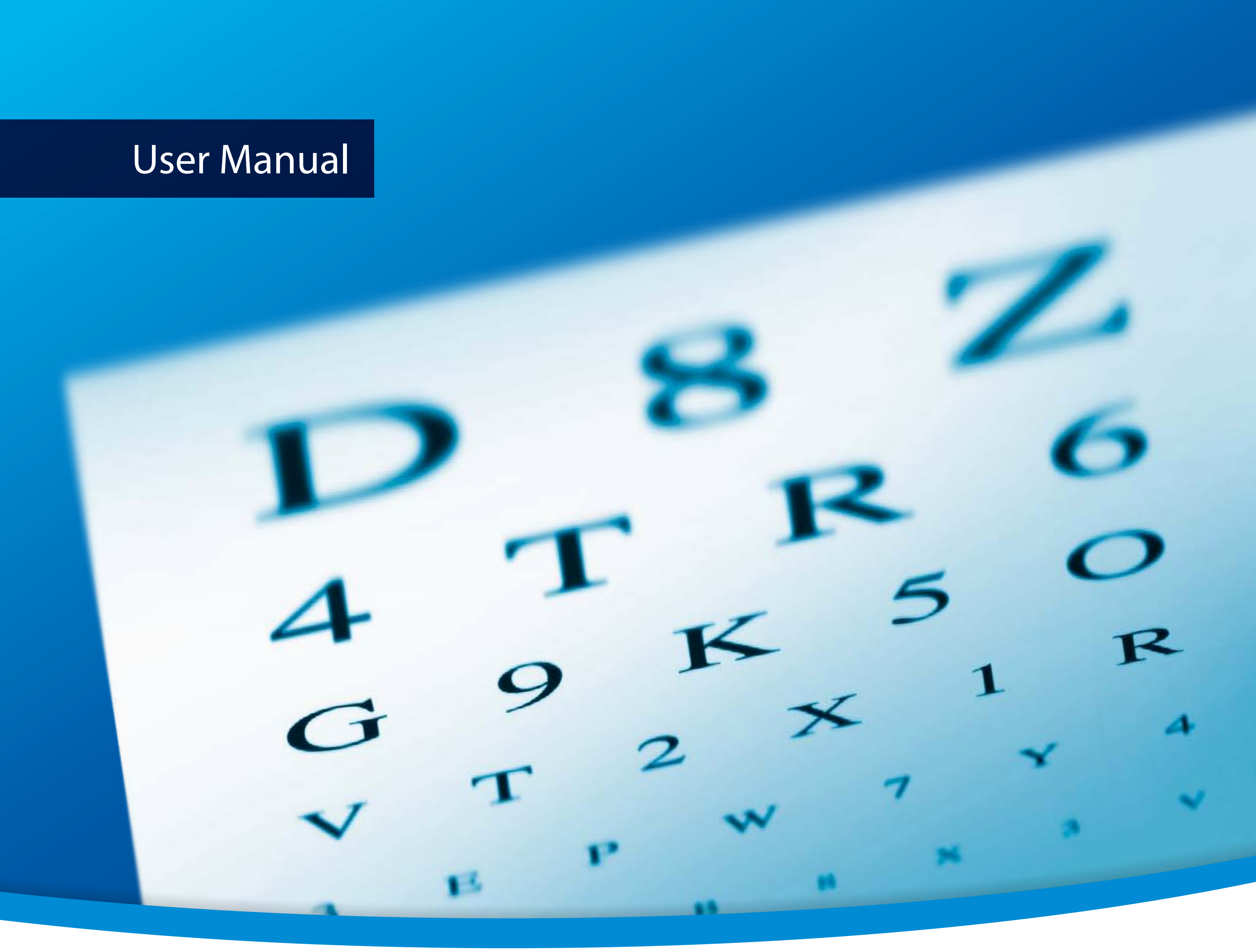
User Manual



3-Heights™ OCR Add-On for ABBYY FineReader Engine v12

Version 6.15.0

PDF-TOOLS.COM
Premium PDF Technology

Contents

1 Introduction

1.1 Overview

The 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 adds optical character recognition (OCR) to several 3-Heights™ products. Documents generated with a product that makes use of the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 contain searchable texts.

The optional 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 recognizes text content and embeds this text as Unicode text in a PDF or PDF/A file. Thanks to this process, the PDF files are searchable and the texts can be used in other applications, databases etc.

The add-on can be used in combination with one of the following products:

- 3-Heights™ Image to PDF Converter
- 3-Heights™ PDF to PDF/A Converter
- 3-Heights™ Document Converter
- 3-Heights™ PDF OCR

1.2 Operating Systems

The 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 is available for the following operating systems:

- Windows Client 7+ | x86 and x64
- Windows Server 2008, 2008 R2, 2012, 2012 R2, 2016, 2019 | x86 and x64

'+' indicates the minimum supported version.

By using the 3-Heights™ OCR Service, the engine can be made available to all platforms supported by 3-Heights™ products.

2 Installation

This chapter describes:

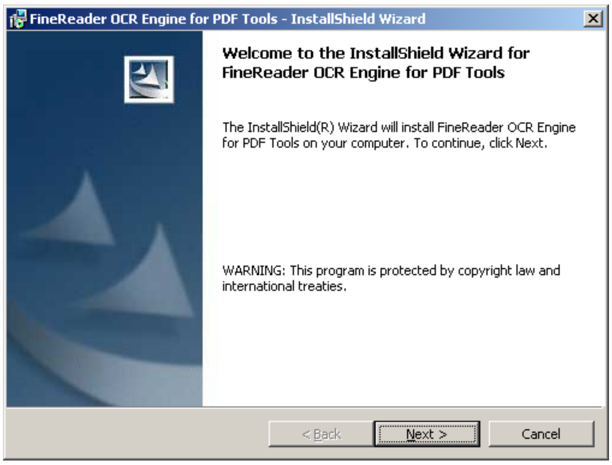
- How to install the ABBYY FineReader OCR Engine and how to activate its license.
- How to install the 3-Heights™ PDF OCR Service.
- How to setup a tool that uses the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 so that is uses the OCR engine directly or via OCR service.

2.1 Installation of ABBYY FineReader Engine

The Windows version of the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 is distributed in MSI-format (Microsoft Software Installation). The MSI file is named `OCR-FRE12-Installer.zip`.

The screenshots are from ABBYY 8, the installation for ABBYY 12 is equivalent.

1. Download the MSI installer `OCR-FRE12-Installer.zip` from your download account at [`http://www.pdf-tools.com`](http://www.pdf-tools.com).
2. Double-click the MSI file to start the installation wizard.
3. In the first dialog box of the installation wizard press the button "Next".



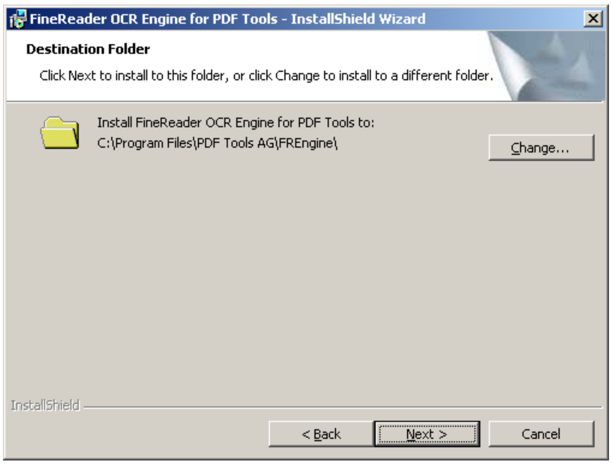
FineReader OCR Engine for PDF Tools - InstallShield Wizard

Welcome to the InstallShield Wizard for FineReader OCR Engine for PDF Tools

The InstallShield(R) Wizard will install FineReader OCR Engine for PDF Tools on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

4. Adjust the destination folder if required using the button "Change..." press "Next" when done.



FineReader OCR Engine for PDF Tools - InstallShield Wizard

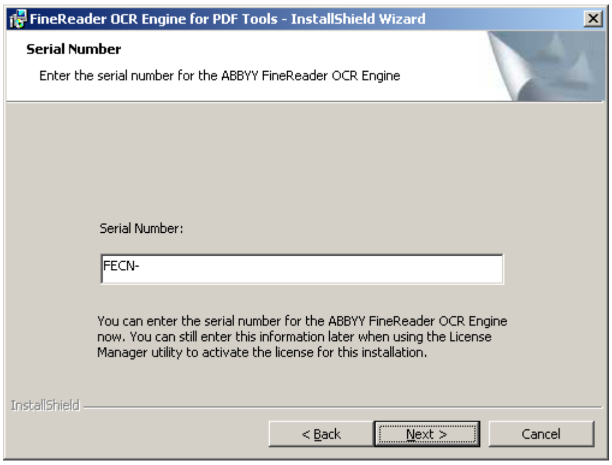
Destination Folder

Click Next to install to this folder, or click Change to install to a different folder.

Install FineReader OCR Engine for PDF Tools to:
C:\Program Files\PDF Tools AG\FREngine\

5. Enter the Serial Number. The serial number is provided by PDF Tools AG. The serial number is required for any license type (evaluation license, page limited license, unlimited license). If you do not have a serial number at

this point, you can continue without entering the serial number. You can provide the serial number at a later time. Press “Next” when done.



FineReader OCR Engine for PDF Tools - InstallShield Wizard

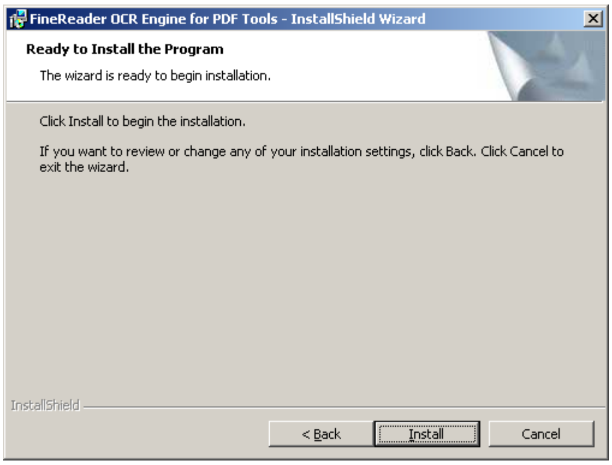
Serial Number

Enter the serial number for the ABBYY FineReader OCR Engine

Serial Number: FECN-

You can enter the serial number for the ABBYY FineReader OCR Engine now. You can still enter this information later when using the License Manager utility to activate the license for this installation.

6. When ready to install, press the “Install” button in the next dialog.



FineReader OCR Engine for PDF Tools - InstallShield Wizard

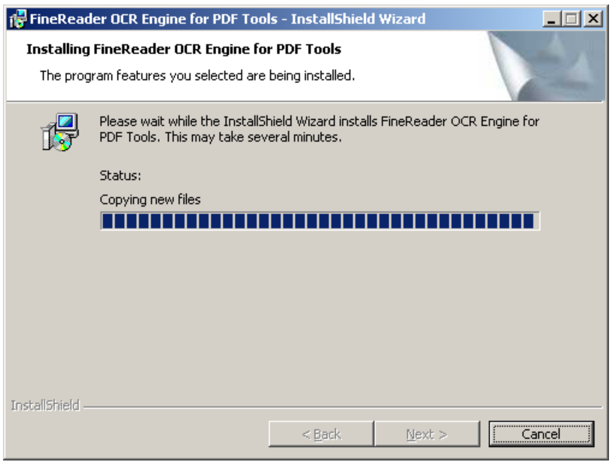
Ready to Install the Program

The wizard is ready to begin installation.

Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

7. The Fine Reader OCR Engine for PDF Tools is now being installed.



FineReader OCR Engine for PDF Tools - InstallShield Wizard

Installing FineReader OCR Engine for PDF Tools

The program features you selected are being installed.

Please wait while the InstallShield Wizard installs FineReader OCR Engine for PDF Tools. This may take several minutes.

Status:

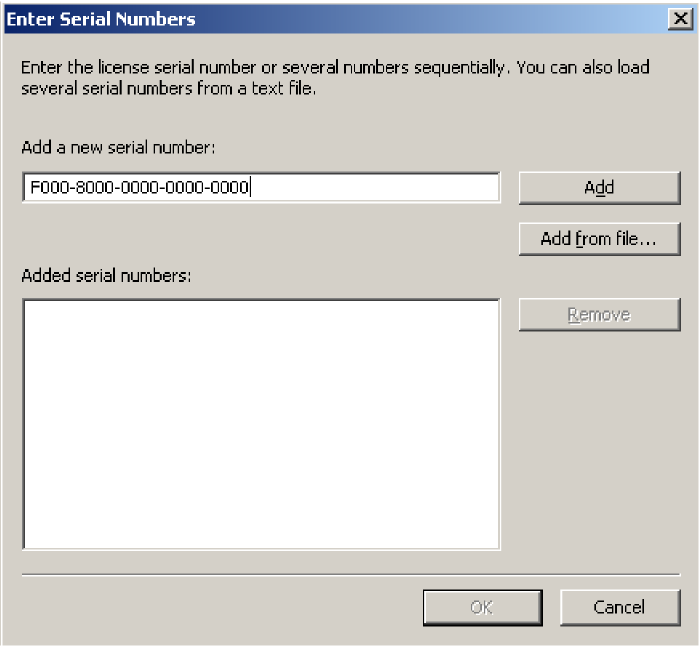
Copying new files

2.2 License Activation of ABBYY OCR FineReader Engine

After the installation, the license needs to be activated. This is done by doing the following steps.

1. Launch the License Manager.

Go to Start → Programs → PDF Tools AG → FineReader OCR Engine for PDF Tools AG → Launch License Manager.



Enter the license serial number or several numbers sequentially. You can also load several serial numbers from a text file.

Add a new serial number:

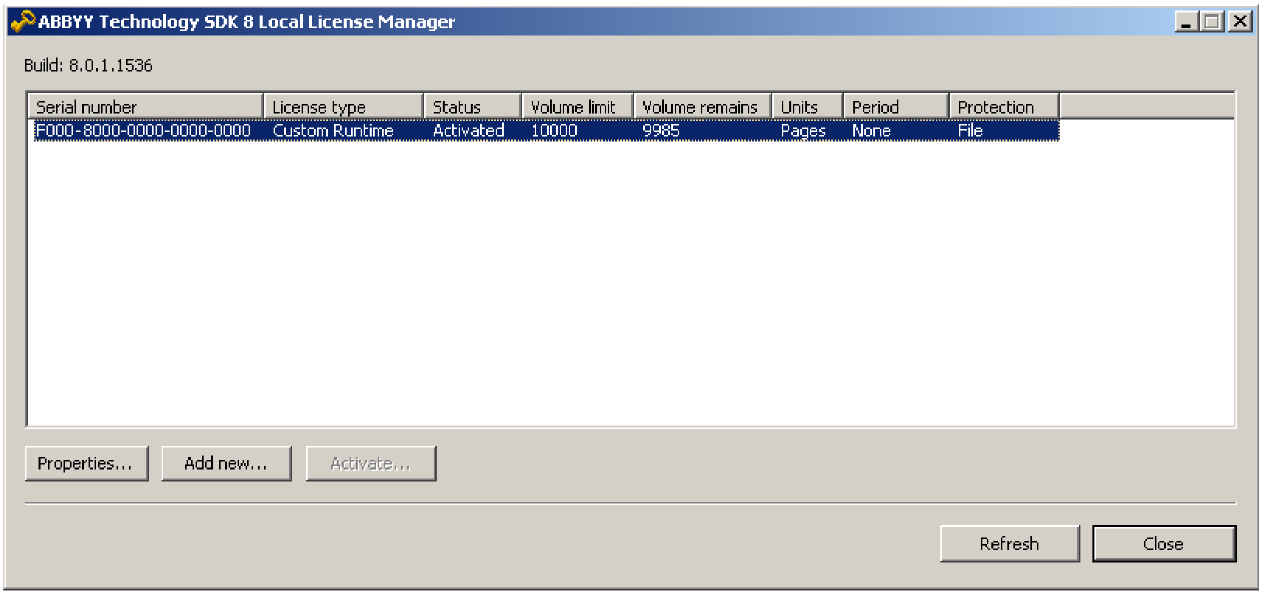
F000-8000-0000-0000-0000

Added serial numbers:

If you haven't provided the serial number at the time of installation, a dialog window asking for the serial number is shown. The serial number has a format like this:
`F000-8000-0000-0000-0000`

2. If a valid serial number is provided, the next dialog is shown. Press the button "Add..." in order to activate the serial number.

Note: Every serial number (including evaluation versions) can only be activated once.



Build: 8.0.1.1536

Serial number	License type	Status	Volume limit	Volume remains	Units	Period	Protection
F000-8000-0000-0000-0000	Custom Runtime	Activated	10000	9985	Pages	None	File

Should you accidentally have provided a wrong, but not necessarily an invalid serial number or want to change the serial number for another reason, do the following steps:

- Go to the following directory:
  `C:\Program Files\PDF Tools AG\FREngine\Bin`

Within this directory delete the file `FREngine.ini`.

- If in the same directory there is a file with a file name equal to your old license key, something similar to `F000-8000-0000-0000-0000.lprf`, delete that file.
- Launch the License Manager.
- Enter the new serial number in the dialog window.

2.3 Upgrading to a Newer Version

If a newer version of the Abbyy FineReader Engine is available and you would like to upgrade to that, do the following steps:

- Even though these files should be left alone, it's safer to make a backup of your configuration file and your license key file. These two files are located in the following directory:
  `C:\Program Files\PDF Tools AG\FREngine\Bin`
  The files named something similar to `F000-8000-0000-0000-0000.lprf` and `FREngine.ini`.
- Uninstall the FR Engine using the appropriate command from the Start menu.
- Install the new version of the FR Engine.

2.4 Remove a Test License

If you want to replace a previously installed test license with the production license, you need to deactivate or delete the test license. If deactivation via the license manager tool fails, follow the following steps.

Abbyy FineReader license information is stored in the following directory:

```
%AllUsersProfile%\ABBYY\SDK\12\Licenses
```

The license file is named according to the serial key; remove this file. After this, you need to restart the licensing service to expedite detection of the configuration change.

2.5 OCR Service

If you are using the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 in combination with a service, such as:

- 3-Heights™ Image to PDF Converter Service
- 3-Heights™ PDF to PDF/A Converter Service
- 3-Heights™ Document Converter

The OCR engine must be invoked via the 3-Heights™ OCR Service and not directly by the other service.

2.5.1 Installation of the OCR Service

The 3-Heights™ OCR Service can be downloaded from your customer account at http://www.pdf-tools.com.

1. Download the MSI `Ocr-Service-‹version›-Windows-(‹platform›).msi` from your download account
2. Double-click the MSI file to start the installation wizard.
3. Follow the installation wizard.

2.6 Installing Products Using the Add-On

The 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 can be used in combination with one of the following products. Other applications cannot access of the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12:

- 3-Heights™ PDF OCR (Shell, API)
- 3-Heights™ Image to PDF Converter (Shell, Service, API)
- 3-Heights™ PDF to PDF/A Converter (Shell, Service, API)
- 3-Heights™ Document Converter

How these products are installed is described in the corresponding manuals of these products.

2.6.1 Linux

On Linux and macOS, PDF Tools provides a plugin for the 3-Heights™ OCR Service. The OCR Service plugin is named `libpdfocrpluginService.ocr`.

Unlike on Windows, you do not need to install OCR plugins into the same directory as the tool using it. Plugins are rather located via the `PDFOCRPLUGINPATH` environment variable. When this environment variable is not set, it defaults to `${HOME}/.pdf-tools/ocr-plugins:/etc/opt/pdf-tools/ocr-plugin`

Also make sure to properly configure the `PDFFONTDIR` environment variable to point to a folder containing a collection of font files that include serif, non-serif and monospaced fonts.

> **Note (Applies only to versions prior to 4.6.13.0):** Whenever there is a change in the configuration of `PDFFONTDIR` (including fonts added or removed in one of the directories listed), make sure to delete the `font-database*dat` files in the tmp folder (folder defined by `PDFTMPDIR` environment variable, `TMP` environment variable, or `/tmp`).

When using the OCR Service plugin, also make sure that the `ocrserver.ini` configuration file is properly configured and also stored in one of the directories listed in `PDFOCRPLUGINPATH`. (Only HTTP-based service entries are supported).

3 Licensing

This is an overview of the licenses available. Details can be retrieved from a sales representative of PDF Tools AG at `pdfsales@pdf-tools.com`.

3.1 License Types

The type of license is defined in the serial key. There are 3 runtime license types available:

Runtime License Options	Network Options	Features
■ Renewable Volume Licenses ▪ Volume per time period ■ CPU Core Licenses ▪ No volume restriction ■ Project-Based Licenses ▪ Volume per project, unlimited scalability	■ 1 Machine Licenses ■ Network Licences. Allow distributing the licensed: ▪ Number of cores ▪ Page volumes ■ Cloud Licences. Allow distributing the licensed page volume.	■ Standard Features ■ Standard & Add Ons Features, e.g. ▪ OCR Languages ▪ ICR (handprint recoginition)

3.1.1 Renewable Volume Runtime Licenses

Volume pricing is based on the quantity of pages that can be processed in a certain period. The renewable period can be a month or a year. Volume licenses have no CPU core limitation. This network licensing allows the dynamic sharing of the licensed volume between multiple network computers, network license manager required.

3.1.2 CPU Core Runtime Licenses

CPU core pricing is based on the quantity of CPU cores that can be used for processing. These license have no page limitation. CPU core licenses are available for 1 machine or the number of licensed cores can be distributed over the network.

3.1.3 Project-Based Licenses

The project pricing scheme allows processing of a defined volume. Efficiency is guaranteed, because of unlimited number of multi-core machines can be use.

4 User's Guide

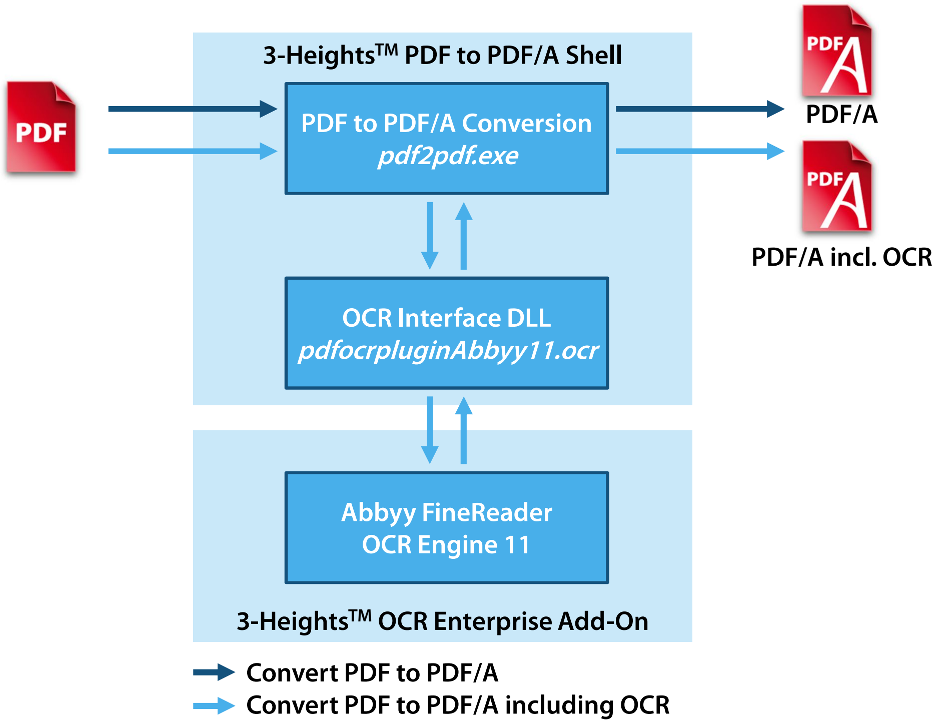
4.1 Architecture

4.1.1 Use OCR Engine Directly

How the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 is used in combination with another product of the 3-Heights™ line is shown in this chapter using the 3-Heights™ PDF to PDF/A Shell as an example. The product can be used in the same way with the 3-Heights™ Image to PDF Converter Shell/API and the 3-Heights™ PDF to PDF/A Converter API.

The 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 installs the OCR engine which can be invoked from the 3-Heights™ PDF to PDF/A Shell.

The OCR interface DLL (`pdfocrpluginabbyy12.ocr`) that communicates with the OCR engine is distributed with the 3-Heights™ PDF to PDF/A Shell. The OCR interface DLL does not contain any OCR technology itself. It is an interface between the PDF to PDF/A Shell and the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12.

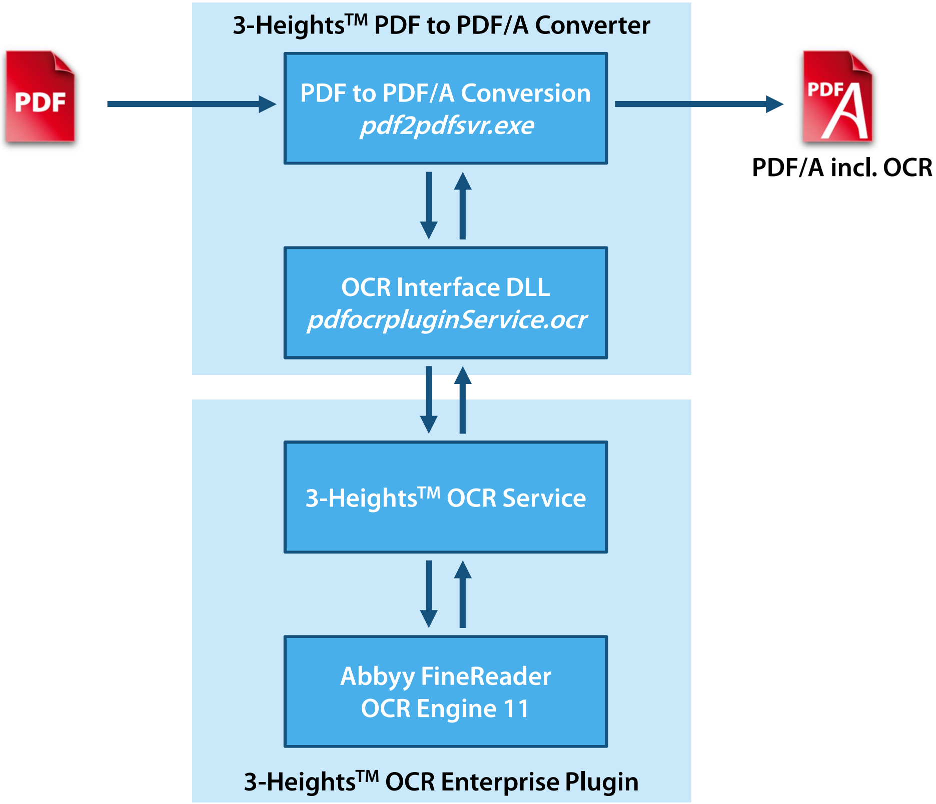


Once the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 is installed, it can be used by other applications of the 3-Heights™ line. How these other applications make use of the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 is individual for every product.

4.1.2 Use OCR via Service

Instead of the application invoking the OCR engine directly, the OCR engine can be accessed via a service. The interface DLL that communicates with the OCR service is `pdfocrpluginService.ocr`. The service invokes the OCR engine and returns the result back to the application. Using the service has the following advantages:

- If the application processes files in parallel, the requests to the OCR engine need to be serialized to match its technical capabilities and licensing limitations.
- The application and the OCR engine can reside on different computers to improve load-balancing.



All 3-Heights™ services must use the OCR service and not the OCR engine directly. Shell and API products can use either the OCR service or the OCR engine directly.

4.2 OCR Related Switches and Functions

The 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 can be used with the 3-Heights™ PDF to PDF/A Converter and the 3-Heights™ Image to PDF Converter. Both tools are available as shell, Windows service and as API version. The switches of shell and service version and the functions of the API version that are related to OCR are described in Table: Switches and Functions.

Switches and Functions

Shell/Service	API	Description

`-le`	`GetOCRPluginCount()`
	`GetOCRPluginName(int)`
		List available OCR engines. OCR engines are listed by means of the interface DLLs. The OCR Interface DLLs are distributed with converter and are named `*.ocr`, e.g. `pdfocrpluginabbyy12.ocr`.
		The shell version lists all available OCR Interface DLLs with the switch `-le`.
		In the API, the number of OCR engines is retrieved using `GetOCRPluginCount`. The string of the `n`-th engine is retrieved using `GetOCRPluginName(n)`.
		The name of the Abbyy FineReader OCR Engine 12 is `"abbyy12"`.
		The name of the 3-Heights™ OCR Service is `"service"`.
		This switch it not available for Windows services products, because they only support one engine type: `"service"`.

`-ocr ‹name›`	`SetOCREngine(String)`	Set OCR engine. The name of the engine is the string previously retrieved using the listing function.
`-ocl ‹lang›`	`SetOCRLanguage(String)`	This setting depends on the OCR engine.
`-ocp ‹params›`	`SetOCRParams(String)`	Set OCR engine dependent parameters. A parameter is always a key-value pair. Multiple pairs can be set as one comma-separated string.

4.2.1 Switches and Functions

GetOCRPluginCount, GetOCRPluginName

Method: `Integer GetOCRPluginCount()`

Method: `String GetOCRPluginName(Integer iOCREngine)`

List available OCR engines. OCR engines are listed by means of the interface DLLs. The OCR Interface DLLs are distributed with converter and are named `*.ocr`, e.g. `pdfocrpluginabbyy12.ocr`.

The shell version lists all available OCR Interface DLLs with the switch `-le`.

In the API, the number of OCR engines is retrieved using `GetOCRPluginCount`. The string of the nth engine is retrieved using `GetOCRPluginName(n)`.

The name of the Abbyy FineReader OCR Engine 12 is `"abbyy12"`.

The name of the 3-Heights™ OCR Service is `"service"`.

This switch is not available for Windows services products, because they only support one engine type: `"service"`.

SetOCREngine

Method: `Boolean SetOCREngine(String OCR engine)`

The shell version is `-ocr ‹name›`

Set OCR engine. The name of the engine is the string previously retrieved using the listing function.

SetOCRLanguage

Method: `Boolean SetOCRLanguage(String OCR language)`

The shell version is `-ocl ‹lang›`

This setting depends on the OCR engine.

SetOCRParams

Method: `Boolean SetOCRParams(String Parameter)`

The shell version is `-ocp ‹params›`

Set OCR engine dependent parameters. A parameter is always a key-value pair. Multiple pairs can be set as one comma-separated string.

4.3 Examples

This chapter describes how the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 is accessed from another product.

The shell samples in this chapter use the 3-Heights™ PDF to PDF/A Converter Shell (pdf2pdf). The commands for the 3-Heights™ Image to PDF Shell (img2pdf) are equivalent with respect to OCR. That is also true for the service version of both products, with the difference that the settings are not written to the console, but are applied in the configuration file instead.

4.3.1 Set OCR Engine

3-Heights™ products that support OCR allow for listing and selecting available OCR engines. At this time, the OCR engine "Abbyy FineReader 10", "Abbyy FineReader 11" and the OCR service are supported.

Shell

The following shell sample shows how to convert a PDF to a PDF/A document without applying OCR.

Example: Converting a PDF document to PDF/A without OCR does not require the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12.

```
pdf2pdf -cl pdfa-1b input.pdf output.pdf
```

OCR engines are listed using the switch `-le`. In order to list an OCR engine, its OCR interface DLL must be in the same directory as where the executable reside. OCR interface-DLLs have the file extension "`.ocr`".

Example: The listing function does not detect whether an OCR engine is actually installed.

```
pdf2pdf -le
List of available OCR engines:
- abbyy10
- abbyy11
- abbyy12
- service
End of list.
```

Once the name of the engine is known, it can be provided as parameter to the `-ocr` switch for converting.

Example: Converting a PDF document to a PDF/A document and add OCR information requires the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 to be installed.

```
pdf2pdf -cl pdfa-1b -ocr "abbyy12" input.pdf output.pdf
```

Service

In a service product (e.g. 3-Heights™ PDF to PDF/A Service), the OCR service must be defined as `"service"`.

Example: Converting a PDF document to PDF/A document and add OCR information using the 3-Heights™ PDF to PDF/A Converter Service requires the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 including the OCR service to be installed. The OCR engine is defined in the configuration file of the main application (`pdf2pdfsvr.ini`).

```
-cl pdfa-1b -ocr "service"
```

API

In the API version, there are three relevant calls, which are used in the following sequence:

`GetOCRPluginCount` to retrieve the number of available OCR engines.

`GetOCRPluginName(n)` to retrieve the name of the `n`-th OCR engine, where `n` is a value from `1` to `GetOCRPluginCount`.

`SetOCREngine` to set the OCR engine.

If the name of the OCR engine is known, the first two calls can be omitted.

Example: Set OCR engine if its name is known.

```
SetOCREngine("abbyy12")
```

4.3.2 Set OCR Language

Multiple languages can be set as one comma-separated string. The supported languages are listed in the appendix of this manual. Setting a language helps the OCR Engine to minimize errors by means of using dictionaries of the defined languages. Setting the language also improves the speed. Abbyy recommends to always set one or several languages, but not more than six. The default language is English.

Example (Shell): Set the languages to English and German.

```
pdf2pdf -ocr "abbyy12" -ocl "English, German" input.pdf output.pdf
```

Example (API): Set the languages to English and German.

```
SetOCREngine("abbyy12")
SetOCRLanguage("English, German")
```

This setting is optional.

4.3.3 Set OCR Engine Specific Parameters

Set OCR engine specific parameters. The parameter is a series of key-value pairs. The syntax of a key-value pair is: “key1=value1”. Multiple pairs can be set as one comma-separated string: “key1=value1, key2=value2”. The supported parameters are listed in the appendix of this manual.

Example (Shell): Set a custom profile.

```
pdf2pdf -ocr "abbyy12" -ocp "Profile=C:\ocr.ini" input.pdf output.pdf
```

Example (API): Set a predefined profile for ABBYY 11.

```
SetOCREngine("abbyy12")
SetOCRParams("PredefinedProfile=DocumentArchiving_Accuracy")
```

This setting is optional. This setting re-initializes the OCR languages to default. It must therefore be set before the OCR language.

4.4 Impacts of Using OCR

Applying OCR to a document compared to not applying OCR has mainly the following consequences.

4.4.1 Performance

OCR takes roughly 2-3 seconds per A4 page on a decent CPU. That is much more than the conversion of the content from one format to another (e.g. from PDF to PDF/A, or from an image to PDF/A). Therefore, in most cases, the OCR-process is the bottleneck for the throughput.

Performance can be improved by using multiple CPUs or configuring the OCR-process to run less accurate and thereby faster (e.g. FastMode, or BalancedMode).

4.4.2 File Size

Depending on the amount of text found, the file size is increased by roughly 0-10 KB for an A4 page.

4.4.3 Appearance

In general, when applying OCR, the original image in the document is replaced by the processed image that was used by the OCR-engine for text recognition. This image is de-skewed and possibly de-noised or has been modified in other ways.

As a consequence the text fits perfectly. However, the image may have visual changes.

If a main attribute of the image (color space, resolution) was changed by the OCR-engine for the purpose of text recognition, the original image is kept. In this case, it is possible that some text is not matching exactly (translation/skewing by a few pixels).

4.5 Best Practice and Recommendations

The Abbyy FineReader OCR Engine is optimized for A4 or smaller sized pages. Applying OCR to one A4 page, which has a resolution of 300 DPI, takes about 2-3 seconds. Larger pages or pages with higher resolution require more memory and time to apply OCR. Abbyy makes the following recommendations:

- Images should have a resolution of around 300 dpi
- Images with small text (font size under 10 points) should have a resolution of 400 to 600 dpi
- Grayscale images are suited best of OCR
- Setting the correct language improves the recognition rate and speed; Recommended is setting 3 to 6 languages
- Using the FastMode improve the throughput at the cost of recognition rate
- Using the normal mode yields the best recognition rate
- The BalancedMode is between the FastMode and the normal mode

4.6 Performance Overview

In the process of converting images to PDF/A, or PDF to PDF/A including OCR information there are basically three figures that are of interest. Depending on the process, one of these figures may be more important than others: Throughput, recognition rate and file size

4.6.1 Throughput

How long does it take to convert and apply OCR to one page? This figure is particularly important when dealing with large volumes, such as converting entire TIFF archives to PDF/A. Normally the throughput is measured in pages per minute (ppm). How long it takes to process a page depends on various factors. The most important are:

CPU Power A faster CPU processes the same file faster than a slower CPU.

Number of CPUs Whilst the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 does not support dividing one job to multiple CPUs, it does support processing multiple files in parallel. Thus if two or more CPUs are available, the load can be spread.

Input file Generally a larger image requires more processing time than a smaller image. The size of the image is given by the number of pixels.

Settings of OCR Engine An OCR engine can have different modes, which allow the user to choose a tradeoff between throughput and recognition rate. Normally it also helps to set the correct language of the text to which OCR is to be applied.

4.6.2 Recognition State

How much text can be identified depends on different factors. The most important one is the document itself. A low resolution document with text that is hardly readable by eye will result in a much lower recognition rate than a clear scan with non-skewed black text on white background.

According to the Abbyy documentation its OCR engine has a rate of 1-2 errors per page.

4.6.3 File Size

How large is the resulting file compared to the original image file?

Applying OCR to a document which contains scanned pages increases the file size by around 10%. At the same time it is possible to change the compression algorithm of the images and thereby reduce the file size. Changing the compression from G4 (which is supported in TIFF, PDF and PDF/A) to JBIG2 (which is supported by PDF and PDF/A) reduces the size of the images by around 50%.

5 Reference Manual

Between version 11 and 12, there are no fundamental changes.

5.1 Cloud Licensing

Starting with ABBYY FineReader Engine 12, a new type of licenses is available which uses the cloud protection mechanism.

Using a Cloud License requires:

1. Active Internet connection.
2. Allowed connections to *.abbyy.com on ports 80 and 443.

When you order a Cloud License, you should receive the following information from sales:

1. a license token file (a file named like `SWRTXXXXXXXXXXXXXXXXXXXX.ABBYY.Cloud.ActivationToken`), and
2. a password to the license token file (another string of text).

To use a Could License, the information must be set as parameters when creating the engine:

`"abbyy12@‹ActivationToken›;‹Password›"`

where `‹ActivationToken›` is the path to the license token file and `‹Password›` is the password.

Example: Passing Cloud Licensing information with engine creation parameters.

```
pdfocr -ocr
  "abbyy12@SWRTXXXXXXXXXXXXXXXXXXXX.ABBYY.Cloud.ActivationToken;G9gHyLT7lEJZ5nhpXoU3YQ=="
  -oim update input.pdf output.pdf
```

5.2 Supported Languages

The following languages are supported:

5.2.1 Natural Languages

- `Abkhaz` – Abkhaz
- `Adyghe` – Adyghe
- `Afrikaans` – Afrikaans
- `Agul` – Agul
- `Albanian` – Albanian
- `Altaic` – Altaic
- `Arabic` – Arabic (Saudi Arabia)
- `ArmenianEastern` – Armenian (Eastern)
- `ArmenianGrabar` – Armenian (Grabar)
- `ArmenianWestern` – Armenian (Western)
- `Awar` – Avar
- `Aymara` – Aymara
- `AzeriCyrillic` – Azerbaijani (Cyrillic)
- `AzeriLatin` – Azerbaijani (Latin)
- `Bashkir` – Bashkir
- `Basque` – Basque
- `Belarusian` – Belarussian
- `Bemba` – Bemba
- `Blackfoot` – Blackfoot
- `Breton` – Breton
- `Bugotu` – Bugotu
- `Bulgarian` – Bulgarian
- `Burmese` – Burmese
- `Buryat` – Buryat
- `Catalan` – Catalan
- `Chamorro` – Chamorro
- `Chechen` – Chechen
- `ChinesePRC` – Chinese Simplified

- `ChineseTaiwan` – Chinese Traditional
- `Chukcha` – Chukcha
- `Chuvash` – Chuvash
- `Corsican` – Corsican
- `CrimeanTatar` – Crimean Tatar
- `Croatian` – Croatian
- `Crow` – Crow
- `Czech` – Czech
- `Danish` – Danish
- `Dargwa` – Dargwa
- `Dungan` – Dungan
- `Dutch` – Dutch (Netherlands)
- `DutchBelgian` – Dutch (Belgium)
- `English` – English
- `EskimoCyrillic` – Eskimo (Cyrillic)
- `EskimoLatin` – Eskimo (Latin)
- `Esperanto` – Esperanto
- `Estonian` – Estonian
- `Even` – Even
- `Evenki` – Evenki
- `Faeroese` – Faeroese
- `Farsi` – Farsi
- `Fijian` – Fijian
- `Finnish` – Finnish
- `French` – French
- `Frisian` – Frisian
- `Friulian` – Friulian
- `GaelicScottish` – Scottish Gaelic
- `Gagauz` – Gagauz
- `Galician` – Galician
- `Ganda` – Ganda
- `German` – German
- `GermanLuxembourg` – German (Luxembourg)
- `GermanNewSpelling` – German (new spelling)
- `Greek` – Greek
- `Guarani` – Guarani
- `Hani` – Hani
- `Hausa` – Hausa
- `Hawaiian` – Hawaiian
- `Hebrew` – Hebrew
- `Hungarian` – Hungarian
- `Icelandic` – Icelandic
- `Ido` – Ido
- `Indonesian` – Indonesian
- `Ingush` – Ingush
- `Interlingua` – Interlingua
- `Irish` – Irish
- `Italian` – Italian
- `Japanese` – Japanese
- `JapaneseModern` – Japanese (Modern)
- `Kabardian` – Kabardian
- `Kalmyk` – Kalmyk
- `KarachayBalkar` – Karachay-Balkar
- `Karakalpak` – Karakalpak
- `Kasub` – Kasub
- `Kawa` – Kawa
- `Kazakh` – Kazakh
- `Khakas` – Khakas
- `Khanty` – Khanty
- `Kikuyu` – Kikuyu
- `Kirgiz` – Kirghiz
- `Kongo` – Kongo
- `Korean` – Korean
- `KoreanHangul` – Korean (Hangul)
- `Koryak` – Koryak
- `Kpelle` – Kpelle
- `Kumyk` – Kumyk
- `Kurdish` – Kurdish
- `Lak` – Lak
- `Lappish` – Sami (Lappish)
- `Latin` – Latin
- `Latvian` – Latvian
- `LatvianGothic` – Latvian language written in Gothic script
- `Lezgin` – Lezgin
- `Lithuanian` – Lithuanian
- `Luba` – Luba
- `Macedonian` – Macedonian
- `Malagasy` – Malagasy
- `Malay` – Malay
- `Malinke` – Malinke
- `Maltese` – Maltese
- `Mansi` – Mansi
- `Maori` – Maori
- `Mari` – Mari
- `Maya` – Maya
- `Miao` – Miao
- `Minankabaw` – Minangkabau
- `Mohawk` – Mohawk
- `Mongol` – Mongol
- `Mordvin` – Mordvin
- `Nahuatl` – Nahuatl
- `Nenets` – Nenets
- `Nivkh` – Nivkh
- `Nogay` – Nogay
- `Norwegian` – Norwegian Nynorskand Norwegian Bokmal
- `NorwegianBokmal` – Norwegian (Bokmal)
- `NorwegianNynorsk` – Norwegian (Nynorsk)
- `Nyanja` – Nyanja
- `Occidental` – Occidental
- `Ojibway` – Ojibway
- `OldEnglish` – Old English
- `OldFrench` – Old French

- `OldGerman` – Old German
- `OldItalian` – Old Italian
- `OldSlavonic` – Old Slavonic
- `OldSpanish` – Old Spanish
- `Ossetic` – Ossetian
- `Papiamento` – Papiamento
- `Pashto` – Pashto
- `PidginEnglish` – Tok Pisin
- `Polish` – Polish
- `PortugueseBrazilian` – Portuguese (Brazil)
- `PortugueseStandard` – Portuguese (Portugal)
- `Provencal` – Provencal
- `Quechua` – Quechua
- `RhaetoRomanic` – Rhaeto-Romanic
- `Romanian` – Romanian
- `RomanianMoldavia` – Romanian (Moldavia)
- `Romany` – Romany
- `Ruanda` – Ruanda
- `Rundi` – Rundi
- `RussianOldSpelling` – Russian (old spelling)
- `Russian` – Russian
- `RussianWithAccent` – Russian (with accents marking stress position)
- `Samoan` – Samoan
- `Selkup` – Selkup
- `SerbianCyrillic` – Serbian (Cyrillic)
- `SerbianLatin` – Serbian (Latin)
- `Shona` – Shona
- `Sioux` – Sioux (Dakota)
- `Slovak` – Slovak
- `Slovenian` – Slovenian
- `Somali` – Somali
- `Sorbian` – Sorbian
- `Sotho` – Sotho
- `Spanish` – Spanish
- `Sunda` – Sunda
- `Swahili` – Swahili
- `Swazi` – Swazi
- `Swedish` – Swedish
- `Tabassaran` – Tabassaran
- `Tagalog` – Tagalog
- `Tahitian` – Tahitian
- `Tajik` – Tajik
- `Tatar` – Tatar
- `Thai` – Thai
- `Tinpo` – Jingpo
- `Tongan` – Tongan
- `Tswana` – Tswana
- `Tun` – Tun
- `Turkish` – Turkish
- `Turkmen` – Turkmen
- `TurkmenLatin` – Turkmen (Latin)
- `Tuvin` – Tuvan
- `Udmurt` – Udmurt
- `UighurCyrillic` – Uighur (Cyrillic)
- `UighurLatin` – Uighur (Latin)
- `Ukrainian` – Ukrainian
- `Urdu` – Urdu
- `UzbekCyrillic` – Uzbek (Cyrillic)
- `UzbekLatin` – Uzbek (Latin)
- `Vietnamese` – Vietnamese
- `Visayan` – Cebuano
- `Welsh` – Welsh
- `Wolof` – Wolof
- `Xhosa` – Xhosa
- `Yakut` – Yakut
- `Yiddish` – Yiddish
- `Zapotec` – Zapotec
- `Zulu` – Zulu

5.2.2 Technical Languages

- `Basic` – Basic programming language
- `C++` – C/C++ programming language
- `Chemistry` – Simple chemical formulas
- `Digits` – Numbers
- `CMC7` – For MICR (CMC-7) text type
- `Cobol` – Cobol programming language
- `E13B` – For MICR (E-13B) text type
- `Fortran` – Fortran programming language
- `Java` – Java programming language
- `OcrA` – For OCR-A text type
- `OcrB` – For OCR-B text type
- `Pascal` – Pascal programming language

5.3 Engine Parameters

The parameter string for ABBYY 12 is composed by a sequence of key/value pairs separated by a semicolon (';').

Example (Shell): Remove garbage with size less than 100 and only preprocess (i.e. without text and picture recognition).

```
-ocr "abbyy12" -ocp "RemoveGarbage=100;PreprocessingOnly=true"
```

5.3.1 PredefinedProfile

Key: `PredefinedProfile` Type: `Name` Default: `Default`

The name of the predefined recognition profile. The available profiles are described in Section 5.4.

This parameter can be combined with `Profile` to customize the predefined profile.

5.3.2 Profile

Key: `Profile` Type: `Path`

The path to a custom recognition profile as described in Section 5.5.

This parameter can be combined with `PredefinedProfile` to choose a predefined base profile.

5.3.3 RecognizeBlankPages

Key: `RecognizeBlankPages` Type: `Boolean` Default: `false`

Recognize blank pages of a certain file. A blank page is considered to be a page with a uniform coloring containing only slight noise. Colored, grayscale and bi-tonal pages can be subject to blank page recognition. The value of the Key-Value pair takes either `True` or `False`.

> **Note:** If a page is recognized as blank no OCR is performed, hence ABBYY's count for remaining pages isn't lowered.

Example (Shell): Choose ABBYY 12 engine, recognize blank pages and store the recognition information (i.e. information about which pages are recognized as blank) in output.xml.

```
-ocr "abbyy12" -ocp "RecognizeBlankPages=true" -ocx "output.xml"
```

5.3.4 BlankPageMargin

Key: `BlankPageMargin` Type: `Double` Default: `0.02`

Set the ratio the margin takes with respect to the corresponding page length. The margin is excluded from the analysis whether a page is blank. The allowed values range from 0 to 0.5. This parameter is only active if at the same time the value of `RecognizeBlankPages` is `True`.

Example (Shell): Choose internal engine and analyze if a page is blank without taking into account a margin of 5% on every side.

```
tiffocr -ocr "abbyy12" -ocp "RecognizeBlankPages=true;BlankPageMargin=0.05"
```

5.3.5 DisableMaskEmbedding

Key: `DisableMaskEmbedding` Type: `Boolean` Default: `false`

If this option is set to `True`, no mask is embedded in the output TIFF. If this option is not set, a mask is embedded by default. The value of the Key-Value pair takes either `True` or `False`.

Example (Shell): Choose ABBYY 12 engine, make ocr recognition, but don't embed a mask in the output file.

```
-ocr "abbyy12" -ocp "DisableMaskEmbedding=true" in.tif out.tif
```

5.3.6 PreprocessingOnly

Key: `PreprocessingOnly` Type: `Boolean` Default: `false`

If this option is set to `True`, analysis and recognition of text and photographic picture regions is skipped. This key is conceived to make available the image transforming functionalities of ABBYY 12 without the necessity of performing optical character recognition. The value of the Key-Value pair takes either `True` or `False`.

Note: If you choose preprocessing only, ABBYY's count for remaining pages isn't lowered.

Example (Shell): Set the custom profile `Profile.ini` without performing OCR.

```
-ocr "abbyy12" -ocp "Profile=C:\path\Profile.ini;PreprocessingOnly=true"
```

5.3.7 RemoveGarbage

Key: `RemoveGarbage` Type: `Integer`

De-noise bitonal image before any OCR is done. The value of the Key-Value pair is an integer. The value specifies the maximum area of black dots that are to be considered garbage (in pixels). When setting -1 ABBYY calculates a suitable choice.

Note: Garbage is removed before any OCR is done. The changes are visible in the output file.

Example (Shell): Remove garbage with maximum size of 50 pixels and set the predefined profile BookArchiving_Speed.

```
-ocr "abbyy12" -ocp "PredefinedProfile=BookArchiving_Speed;RemoveGarbage=50"
```

5.4 Predefined Profiles

A predefined profile represents a predefined set of recognition parameters optimized for a specific use case.

The profile can be selected using the PredefinedProfile parameter.

Supported are the following profile names:

5.4.1 DocumentConversion_Accuracy

For converting documents into editable formats, optimized for accuracy.

- Best quality. Enables font style detection.

This profile corresponds to the following parameters:

```
[BarcodeParams]
EnableAdvancedExtractionMode = TRUE
```

5.4.2 DocumentConversion_Speed

For converting documents into editable formats, optimized for speed.

- Like DocumentConversion_Accuracy, but document analysis and recognition are speeded up.
- **Important! This profile requires the Fast Mode module available in the license.**

This profile corresponds to the following parameters:

```
[ObjectsExtractionParams]
ProhibitColorImage = TRUE
[PrepareImageMode]
UseFastBinarization = TRUE
[RecognizerParams]
FastMode = TRUE
```

5.4.3 DocumentArchiving_Accuracy

For creating an electronic archive, optimized for accuracy

- Enables detection of maximum text on an image, including text embedded into the image.
- Skew correction is not performed.
- Fonts and styles are not detected.

This profile corresponds to the following parameters:

```
[BarcodeParams]
EnableAdvancedExtractionMode = TRUE
[ObjectsExtractionParams]
DetectTextOnPictures = TRUE
EnableAggressiveTextExtraction = TRUE
[PageAnalysisParams]
EnableTextExtractionMode = TRUE
[SynthesisParamsForDocument]
DetectDocumentStructure = FALSE
DetectFontFormatting = FALSE
[SynthesisParamsForPage]
AllowGrayBackgroundColor = TSPV_Yes
AllowGrayTextColor = TSPV_Yes
DetectFontFormattingAtPageLevel = TRUE
DetectTextColor = TSPV_Yes
```

5.4.4 DocumentArchiving_Speed

For creating an electronic archive, optimized for speed

- Like DocumentArchiving_Accuracy, but document analysis and recognition are speeded up.
- **Important! This profile requires the Fast Mode module**

This profile corresponds to the following parameters:

```
[ObjectsExtractionParams]
DetectMatrixPrinter = FALSE
DetectPorousText = FALSE
DetectTextOnPictures = TRUE
EnableAggressiveTextExtraction = TRUE
FastObjectsExtraction = TRUE
ProhibitColorImage = TRUE
RemoveGarbage = TRUE
RemoveTexture = FALSE
[PageAnalysisParams]
EnableTextExtractionMode = TRUE
ProhibitModelAnalysis = TRUE
[PrepareImageMode]
CorrectSkew = FALSE
UseFastBinarization = TRUE
[RecognizerParams]
FastMode = TRUE
[SynthesisParamsForDocument]
DetectDocumentStructure = FALSE
DetectFontFormatting = FALSE
[SynthesisParamsForPage]
AllowGrayBackgroundColor = TSPV_Yes
AllowGrayTextColor = TSPV_Yes
DetectFontFormattingAtPageLevel = TRUE
DetectTextColor = TSPV_Yes
```

5.4.5 BookArchiving_Accuracy

For creating an electronic library, optimized for accuracy

- Best quality. Enables font style detection.

This profile corresponds to the following parameters:

```
[BarcodeParams]
EnableAdvancedExtractionMode = TRUE
```

5.4.6 BookArchiving_Speed

For creating an electronic library, optimized for speed

- Like BookArchiving_Accuracy, but document analysis and recognition are speeded up.
- **Important! This profile requires the Fast Mode module available in the license.**

This profile corresponds to the following parameters:

```
[ObjectsExtractionParams]
ProhibitColorImage = TRUE
[PrepareImageMode]
UseFastBinarization = TRUE
[RecognizerParams]
FastMode = TRUE
```

5.4.7 TextExtraction_Accuracy

For extracting text from documents, optimized for accuracy

- Enables detection of all text on an image, including small text areas of low quality (pictures and tables are not detected).
- Fonts and styles are not detected.

This profile corresponds to the following parameters:

```
[BarcodeParams]
EnableAdvancedExtractionMode = TRUE
[ObjectsExtractionParams]
DetectTextOnPictures = TRUE
EnableAggressiveTextExtraction = TRUE
[PageAnalysisParams]
DetectPictures = FALSE
EnableTextExtractionMode = TRUE
[SynthesisParamsForDocument]
DetectDocumentStructure = FALSE
DetectFontFormatting = FALSE
[SynthesisParamsForPage]
DetectFontFormattingAtPageLevel = TRUE
```

5.4.8 TextExtraction_Speed

For extracting text from documents, optimized for speed

- Like TextExtraction_Accuracy, but document analysis and recognition are speeded up.
- **Important! This profile requires the Fast Mode module available in the license.**

This profile corresponds to the following parameters:

```
[ObjectsExtractionParams]
```

```
DetectMatrixPrinter = FALSE
DetectPorousText = FALSE
DetectTextOnPictures = TRUE
EnableAggressiveTextExtraction = TRUE
FastObjectsExtraction = TRUE
ProhibitColorImage = TRUE
RemoveGarbage = TRUE
RemoveTexture = FALSE
[PageAnalysisParams]
DetectPictures = FALSE
EnableTextExtractionMode = TRUE
ProhibitModelAnalysis = TRUE
[PrepareImageMode]
CorrectSkew = FALSE
DiscardColorImage = TRUE
UseFastBinarization = TRUE
[RecognizerParams]
FastMode = TRUE
[SynthesisParamsForDocument]
DetectDocumentStructure = FALSE
DetectFontFormatting = FALSE
[SynthesisParamsForPage]
DetectFontFormattingAtPageLevel = TRUE
DetectTextColor = TSPV_Yes
```

5.4.9 FieldLevelRecognition

For recognizing short text fragments.

- Currently this profile has default settings.

This profile corresponds to the following parameters:

```
[DocumentProcessingParams]
PerformSynthesis = FALSE
[PageProcessingParams]
PerformAnalysis = FALSE
[SynthesisParamsForPage]
DetectFontFormattingAtPageLevel = FALSE
```

5.4.10 BarcodeRecognition_Accuracy

For barcode extraction, optimized for accuracy

- Extracts only barcodes (texts, pictures, or tables are not detected).
- **Important! This profile requires the Barcode Autolocation module available in the license.**

This profile corresponds to the following parameters:

```
[BarcodeParams]
MinRatioToTextHeight = 0.9
[ObjectsExtractionParams]
DetectMatrixPrinter = FALSE
DetectPorousText = FALSE
[PageAnalysisParams]
DetectBarcodes = TRUE
```

```
DetectPictures = FALSE
DetectTables = FALSE
DetectText = FALSE
DetectSeparators = FALSE
DetectVectorGraphics = FALSE
[PrepareImageMode]
CorrectSkew = FALSE
```

5.4.11 BarcodeRecognition_Speed

For barcode extraction, optimized for speed

- Like BarcodeRecognition_Accuracy, but document analysis and recognition are speeded up.
- **Important! This profile requires the Barcode Autolocation module available in the license.**

This profile corresponds to the following parameters:

```
[BarcodeParams]
MinRatioToTextHeight = 0.9
[ObjectsExtractionParams]
DetectMatrixPrinter = FALSE
DetectPorousText = FALSE
FastObjectsExtraction = TRUE
[PageAnalysisParams]
DetectBarcodes = TRUE
DetectPictures = FALSE
DetectTables = FALSE
DetectText = FALSE
DetectSeparators = FALSE
DetectVectorGraphics = FALSE
[PageProcessingParams]
PerformPreprocessing = FALSE
[PrepareImageMode]
CorrectSkew = FALSE
DiscardColorImage = TRUE
```

5.4.12 BusinessCardsProcessing

For recognizing business cards

- Detects only business cards.
- Enables detection of all text on an image, including small text areas of low quality (pictures and tables are not detected).
- Resolution correction is performed.

This profile corresponds to the following parameters:

```
[FontFormattingDetectionParams]
DetectFontFamily = FALSE
[ObjectsExtractionParams]
EnableAggressiveTextExtraction = TRUE
ProhibitColorImage = FALSE
[PageAnalysisParams]
DetectPictures = FALSE
DetectTables = FALSE
EnableTextExtractionMode = TRUE
```

```
ProhibitModelAnalysis = FALSE
ProhibitDoublePageMode = TRUE
ProhibitCJKColumns = FALSE
[PagePreprocessingParams]
ResolutionCorrectionMode = RCM_Correct
[SynthesisParamsForDocument]
DetectDocumentStructure = FALSE
DetectFontFormatting = FALSE
[SynthesisParamsForPage]
DetectFontFormattingAtPageLevel = TRUE
SynthesizeBusinessCards = TRUE
```

5.4.13 EngineeringDrawingsProcessing

For recognizing technical drawings

- It takes into account large size and complexity of engineering diagrams, as well as possibility of different text orientation within the image.
- Enables detection of all text on an image, including text blocks of vertical orientation.

This profile corresponds to the following parameters:

```
[PageAnalysisParams]
DetectPictures = FALSE
DetectVectorGraphics = FALSE
DetectVerticalEuropeanText = TRUE
EnableTextExtractionMode = TRUE
[SynthesisParamsForDocument]
DetectDocumentStructure = FALSE
DetectFontFormatting = FALSE
[SynthesisParamsForPage]
DetectFontFormattingAtPageLevel = TRUE
```

5.4.14 Default

For default values.

- Sets all the processing parameters to the default values.

Example (Shell): Set the predefined profile DocumentArchiving_Accuracy.

```
-ocr "abbyy12" -ocp "PredefinedProfile=DocumentArchiving_Accuracy"
```

5.5 Custom Profiles

A custom profile is defined by a configuration file that is set up like an INI file. It consists of sections and entries in each section.

Example:

```
[PrepareImageMode]
DiscardColorImage = false
```

```
[RecognizerParams]
BalancedMode = false
TextLanguage = English,German
```

The custom profile is provided using the parameter Profile. Ensure the profile file resides at a location that can be accessed by the OCR engine.

Example (Shell): Set the custom profile `Profile1.ini`.

```
-ocr "abbyy12" -ocp "Profile=C:\path\Profile1.ini"
```

5.5.1 [PrepareImageMode] INI-File Section

Main settings

Rotation

Key: `Rotation` Type: `RotationTypeEnum` Default: `RT_NoRotation`

This property specifies the rotation angle to apply to the image during preparation.

RotationTypeEnum

`RT_Clockwise` Rotate an image 90° clockwise.

`RT_Counterclockwise` Rotate an image 90° counterclockwise.

`RT_NoRotation` Do not rotate an image.

`RT_Upsidedown` Rotate an image upside down.

`RT_UnknownRotation` The rotation angle is undefined.

CorrectSkew

Key: `CorrectSkew` Type: `Boolean` Default: `True`

This property set to `True` tells ABBYY FineReader Engine to correct skew during image preparation. The type of skew correction is defined by the `CorrectSkewMode` property.

If this property is set to `False`, the value of the `CorrectSkewMode` property is ignored.

CorrectSkewMode

Key: `CorrectSkewMode` Type: `CorrectSkewModeEnum`
Default: `CSM_CorrectSkewByHorizontalText | CSM_CorrectSkewByVerticalText`

Specifies the mode of skew correction. The value of this property is an OR superposition of the CorrectSkewModeEnum constants which denote the types of skew correction. `0` means do not correct skew.

The value of this property is ignored, if the `CorrectSkew` property is set to `False`.

CorrectSkewModeEnum

`CSM_CorrectSkewByBlackSquaresHorizontally` The image skew angle is corrected based on so-called 'black squares' (the skew angle is calculated based on the horizontal pairs of squares). Black squares are often placed on forms. We recommend that you use this constant only when working with images of forms, otherwise you may obtain incorrect results.

`CSM_CorrectSkewByBlackSquaresVertically` The image skew angle is corrected based on so-called 'black squares' (the skew angle is calculated based on the vertical pairs of squares). Black squares are often placed on forms. We recommend that you use this constant only when working with images of forms, otherwise you may obtain incorrect results.

`CSM_CorrectSkewByHorizontalLines` The image skew angle is corrected based on horizontal lines. We recommend that you use this constant only when working with images which contain horizontal lines (e.g. invoices, price lists, or other documents which contain tables with visible borders), otherwise you may get incorrect results.

`CSM_CorrectSkewByHorizontalText` The image skew angle is corrected based on horizontal text lines.

`CSM_CorrectSkewByVerticalLines` The image skew angle is corrected based on vertical lines. We recommend that you use this constant only when working with images which contain vertical lines (e.g. invoices, price lists, or other documents which contain tables with visible borders), otherwise you may obtain incorrect results.

`CSM_CorrectSkewByVerticalText` The image skew angle is corrected based on vertical text lines. The constant may be useful when working with documents in Chinese, Japanese, or Korean language, or if page orientation is incorrect.

`BackgroundFillingColor`

Key: `BackgroundFillingColor` Type: `Integer` Default: `-1`

Specifies the color used for filling the areas which are added to the image after skew correction.

The default value of this property is `-1`, which means that the color is determined by ABBYY FineReader Engine automatically.

> **Note:** The int value is calculated from the RGB triplet using the formula: (red value) + (256 x green value) + (65536 x blue value), where red value is the first triplet component, green value is the second triplet component, blue value is the third triplet component. Hence the int value of the color white equals 16777215.

`InvertImage`

Key: `InvertImage` Type: `Boolean` Default: `False`

This property set to `True` tells ABBYY FineReader Engine to invert colors of the prepared image.

MirrorImage

Key: MirrorImage Type: Boolean Default: False

This property set to True tells ABBYY FineReader Engine to mirror the prepared image around its vertical axis.

EnhanceLocalContrast

Key: EnhanceLocalContrast Type: Boolean Default: False

Specifies whether the local contrast of the image should be increased. Such preprocessing may increase the quality of recognition.

> **Note:** Important Setting this property to True is meaningful for color and gray images only.

PhotoProcessingMode

Key: PhotoProcessingMode Type: PhotoProcessingModeEnum Default: PPM_Auto

Specifies whether the processing image should be treated as a photo. If a photo is processed, FineReader Engine uses special algorithms for such image on different stages of image processing.

PhotoProcessingModeEnum

PPM_Auto ABBYY FineReader Engine should automatically determine whether the image is a photo.

PPM_TreatAsPhoto The image should be treated as a photo.

PPM_TreatAsNonPhoto The image should not be treated as a photo.

Resolution overwriting

AutoOverwriteResolution

Key: AutoOverwriteResolution Type: Boolean Default: True

Specifies whether resolution of the prepared image should be automatically overwritten. The property is only available, if the value of the OverwriteResolution property is False. If the value of AutoOverwriteResolution is True, ABBYY FineReader Engine will automatically detect and overwrite image resolution.

Note: If you set this property to False and resolution of the prepared image is too low (less than 50 DPI), too high (more than 3200 DPI), or undefined, you should set the correct values of the XResolutionToOverwrite and YResolutionToOverwrite properties, otherwise an error is returned.

OverwriteResolution

Key: OverwriteResolution Type: Boolean Default: False

Allows you to overwrite resolution of the prepared image. The resolution is overwritten depending on the values of the XResolutionToOverwrite and YResolutionToOverwrite properties. In this case the new resolution will be used for image preprocessing (i.e. for binarization, deskewing, etc.). Image resolution can be automatically overwritten (see the description of the AutoOverwriteResolution property).

XResolutionToOverwrite

Key: XResolutionToOverwrite Type: Integer Default: 300

Specifies the horizontal resolution of the original image in DPI. This value is used to overwrite resolution of the prepared image when resolution of the original image is not specified or incorrect and only if the OverwriteResolution property is True. ABBYY FineReader Engine works with the prepared image which horizontal and vertical resolutions are equal, therefore the program stretches the image so that the horizontal and vertical resolutions of the prepared image are identical and equal to the maximum of XResolutionToOverwrite and YResolutionToOverwrite.

YResolutionToOverwrite

Key: YResolutionToOverwrite Type: Integer Default: 300

Specifies the vertical resolution of the original image in DPI. This value is used to overwrite resolution of the prepared image when resolution of the original image is not specified or incorrect and only if the OverwriteResolution property is True. ABBYY FineReader Engine works with the prepared image which horizontal and vertical resolutions are equal, therefore the program stretches the image so that the horizontal and vertical resolutions of the prepared image are identical and equal to the maximum of XResolutionToOverwrite and YResolutionToOverwrite.

Creating image planes

DiscardColorImage

Key: DiscardColorImage Type: Boolean Default: False

This property set to True tells ABBYY FineReader Engine to leave only black-and-white planes in the prepared image.In this case image binarization is performed during image preparation.

UseFastBinarization

Key: UseFastBinarization Type: Boolean Default: False

If this property is True, ABBYY FineReader Engine will use algorithms for fast image binarization. Binarization is performed either when the image is loaded (if the DiscardColorImage is True during preparation), or later when a black-and-white image is necessary. This property set to True will speed up binarization however its quality may deteriorate.

ImageCompression

Key: ImageCompression Type: ImageCompressionEnum Default: IC_Auto

This property specifies how an image should be compressed during conversion to the internal format.

ImageCompressionEnum

IC_Auto Automatic mode. ABBYY FineReader Engine automatically determines whether to compress temporary images or not.

IC_Compress Compress images with ZIP compression.

IC_NoCompression Keep images uncompressed.

CreatePreview

Key: CreatePreview Type: Boolean Default: False

This property set to True tells ABBYY FineReader Engine to create preview page for the prepared image.

PreviewHeight

Key: PreviewHeight Type: Integer Default: 90

Specifies the height in pixels of the preview page. This property is valid only if the CreatePreview property is True, otherwise it is ignored.

PreviewWidth

Key: PreviewWidth Type: Integer Default: 64

Specifies the width in pixels of the preview page. This property is valid only if the CreatePreview property is True, otherwise it is ignored.

5.5.2 [ImageProcessingParams] INI-File Section

InvertImage

Key: InvertImage Type: Boolean Default: False

Specifies if the image colors in the block are inverted. If this property is True, ABBYY FineReader Engine will invert the image of a block before recognition.

MirrorImage

Key: MirrorImage Type: Boolean Default: False

Specifies if the image in the block is mirrored around the vertical axis. If this property is True, ABBYY FineReader Engine will mirror the image of a block before recognition.

RotationType

Key: RotationType Type: RotationTypeEnum Default: RT_NoRotation

Specifies the orientation of a text in a block relative to the normal reading position.

Supported values are in RotationTypeEnum on page 33.

This property cannot be set to RT_UnknownRotation.

This property is RT_NoRotation by default, which means that the orientation is normal.

5.5.3 [DocumentProcessingParams] INI-File Section

PerformSynthesis

Key: PerformSynthesis Type: Boolean Default: True

Specifies if document synthesis is to be performed. If this property is False, the SynthesisParamsForDocument section is ignored.

5.5.4 [PageProcessingParams] INI-File Section

PerformPreprocessing

Key: PerformPreprocessing Type: Boolean Default: True

Specifies if page preprocessing is to be performed. Page preprocessing is performed before page analysis and includes correction of orientation, inversion, geometrical distortions. If this property is False, the PagePreprocessingParams section is ignored.

ProhibitColorObjectsAtProcessing

Key: ProhibitColorObjectsAtProcessing Type: Boolean Default: False

Specifies if color objects must be filtered out on the image before layout analysis and recognition.

PerformAnalysis

Key: PerformAnalysis Type: Boolean Default: True

Specifies if page analysis is to be performed. If this property is False, the PageAnalysisParams section is ignored.

> **Note:** If your license does not support the Analysis module, the default value of this property is False.

PerformRecognition

Key: PerformRecognition Type: Boolean Default: True

Specifies if recognition is to be performed. If this property is False, the RecognizerParams section is ignored.

> **Note:** If your license does not support the Recognition module, the default value of this property is False.

ATTENTION: Removed or renamed properties

DetectInvertedImage, DetectOrientation, RemoveGeometricalDistortions

Key: DetectInvertedImage Type: (Re-)moved
Key: DetectOrientation Type: (Re-)moved
Key: RemoveGeometricalDistortions Type: (Re-)moved

Use the CorrectInvertedImage, CorrectOrientation and GeometryCorrectionMode properties of the PagePreprocessingParams section and the properties of the OrientationDetectionParams section. There is a new stage concerned with image preprocessing, and settings used at that stage were grouped into a separate object to make the structure of parameters objects clearer and more logical.

PerformPageAnalysis

Key: PerformPageAnalysis Type: Renamed

Use the PerformAnalysis property to enable layout analysis and PerformPreprocessing property to enable correction of page orientation, inversion, and geometrical distortions. The image preprocessing stage has been separated from page analysis and has its own set of parameters.

DetectBarcodes

Key: DetectBarcodes Type: (Re-)moved

Use the corresponding properties of the PageAnalysisParams section. Many processing parameters were grouped into sections to make the structure of parameter sections clearer and more logical.

5.5.5 [PagePreprocessingParams] INI-File Section

CorrectInvertedImage

Key: CorrectInvertedImage Type: Boolean Default: False

This property set to True tells ABBYY FineReader Engine to detect whether the image is inverted (white text against black background). The text color is detected during page preprocessing, and if it differs from normal, ABBYY FineReader Engine automatically inverts the image.

CorrectOrientation

Key: CorrectOrientation Type: Boolean Default: False

If this property is True, the page orientation is detected during page preprocessing, and if it differs from normal, ABBYY FineReader Engine automatically rotates the image.

> **Note:** If this property is set to True:
> - the TextTypes property of the RecognizerParams cannot be set to TT_Handprinted

CorrectShadowsAndHighlights

Key: CorrectShadowsAndHighlights Type: ThreeStatePropertyValueEnum Default: TSPV_Auto

If this property is set to TSPV_Yes, the image preprocessing will include correction of excessive shadows and highlighting to improve recognition quality. This property is designed for use with photographs only.

ThreeStatePropertyValueEnum

TSPV_Auto ABBYY FineReader Engine will automatically determine if this processing mode should be used, depending on the situation (image characteristics etc.).

TSPV_No The processing mode in question will not be used.

TSPV_Yes The processing mode in question will be used.

CorrectSkew

Key: CorrectSkew Type: ThreeStatePropertyValueEnum Default: TSPV_Auto

If this property is set to TSPV_Yes, image skew will be corrected during page preprocessing. The type of skew correction is defined by the CorrectSkewMode property.

If this property is set to TSPV_No, the value of the CorrectSkewMode property is ignored.

Supported values are ThreeStatePropertyValueEnum on page 41.

CorrectSkewMode

Key: CorrectSkewMode Type: CorrectSkewModeEnum
Default: CSM_CorrectSkewByHorizontalText | CSM_CorrectSkewByVerticalText

Specifies the mode of skew correction. The value of this property is an OR superposition of the CorrectSkewModeEnum enumeration constants which denote the types of skew correction. 0 means do not correct skew.

The value of this property is ignored, if the CorrectSkew property is set to TSPV_No.

See CorrectSkewModeEnum.

GeometryCorrectionMode

Key: GeometryCorrectionMode Type: GeometryCorrectionModeEnum Default: GCM_Auto

Specifies whether geometrical distortions (perspective on photos, curved lines from scanned books, etc.) should be removed during page preprocessing.

GeometryCorrectionModeEnum

GCM_Auto Image geometry correction will be performed, if it is necessary. ABBYY FineReader Engine will determine automatically, whether the processed document is a photo and if it is, will perform geometry correction.

GCM_Correct Always correct image geometry. Photographs usually have perspective distortions, therefore if you process photos, you can select this option.

GCM_DontCorrect Do not correct image geometry. If you process scanned images of good quality, correction of image geometry is not necessary.

ResolutionCorrectionMode

Key: ResolutionCorrectionMode Type: ResolutionCorrectionModeEnum Default: RCM_Auto

Specifies whether resolution of the image should be corrected during page preprocessing.

ResolutionCorrectionModeEnum

RCM_Auto If resolution of the image is suspicious, FineReader Engine automatically detects image resolution and changes it.

RCM_Correct Detect image resolution and correct it.

RCM_DontCorrect Do not correct image resolution.

5.5.6 [PageAnalysisParams] INI-File Section

DetectText

Key: DetectText Type: Boolean Default: True

If this property is True, the text areas are detected during layout analysis.

EnableTextExtractionMode

Key: EnableTextExtractionMode Type: Boolean Default: False

This property set to True tells FineReader Engine that text blocks can be located anywhere on the page. Isolated text blocks are detected during layout analysis. Tables are not detected. Model analysis is not performed, as if ProhibitModelAnalysis property was set to True.

DetectTables

Key: DetectTables Type: Boolean Default: True

If this property is True, the tables are detected during layout analysis.

AggressiveTableDetection

Key: AggressiveTableDetection Type: Boolean Default: False

This 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.

DetectBarcodes

Key: DetectBarcodes Type: Boolean Default: False

Specifies if barcodes are detected, and accordingly barcode blocks created, during layout analysis. If this property is False, barcodes may be detected as blocks of some other type (e.g. pictures).

DetectSeparators

Key: DetectSeparators Type: Boolean Default: True

If this property is True, the separators are detected during layout analysis.

DetectPictures

Key: DetectPictures Type: Boolean Default: True

If this property is True, the pictures are detected during layout analysis.

DetectVectorGraphics

Key: DetectVectorGraphics Type: Boolean Default: True

If this property is `True`, the vector pictures are detected during layout analysis. Vector picture blocks may appear in the layout only if this property has been set to `True` during layout analysis.

Additional settings

`DetectMultipleBusinessCards`

Key: `DetectMultipleBusinessCards` Type: `Boolean` Default: `False`

Specifies whether a processing page can contain several business cards.

`NoShadowsMode`

Key: `NoShadowsMode` Type: `Boolean` Default: `False`

This property set to `True` tells ABBYY FineReader Engine to presume that an image has no shadows from scanning.

`DetectVerticalEuropeanText`

Key: `DetectVerticalEuropeanText` Type: `Boolean` Default: `False`

This property set to `True` makes FineReader Engine look for vertically oriented text. It applies to all languages other than CJK. For CJK languages, vertical text detection is managed by the `ProhibitCJKColumns` property.

`ProhibitCJKColumns`

Key: `ProhibitCJKColumns` Type: `Boolean` Default: `False`

The text in CJK languages can be written vertically as well as horizontally. Setting this property to `True` tells ABBYY FineReader Engine to ignore the possibility of vertical text and recognize the image with the supposition that all text is arranged horizontally.

This property is valid only for CJK languages.

`ProhibitDoublePageMode`

Key: `ProhibitDoublePageMode` Type: `Boolean` Default: `False`

This property set to `True` tells ABBYY FineReader Engine to presume that an image is not a book double page.

ProhibitModelAnalysis

Key: ProhibitModelAnalysis Type: Boolean Default: False

If this property is False, typical variants of page layout will be gone through during page analysis and the best variant will be selected, which can improve recognition quality. If the best variant of page layout cannot be selected, standard page layout analysis will be performed.

> **Note:** If EnableTextExtractionMode property is set to True, the value of this property is ignored and model analysis is not performed.

ATTENTION: Removed or renamed properties

FlexiFormsDA

Key: FlexiFormsDA Type: Behavior changed

This property is obsolete and will be removed in future versions.

The same functionality is now provided by EnableAggressiveTextExtraction property and EnableTextExtractionMode property.

If you do not set this property specifically, its value will be ignored. If you do, both properties mentioned above become automatically set to the value you assigned to this property.

SingleColumnMode

Key: SingleColumnMode Type: Removed

This property is no longer supported in FineReader Engine recognition technologies.

5.5.7 [TableAnalysisParams] INI-File Section

DetectCellsInversion

Key: DetectCellsInversion Type: Boolean Default: True

If this property is True, the cells inversion is detected during table block analysis.

DetectCellsOrientation

Key: DetectCellsOrientation Type: Boolean Default: True

If this property is True, the cells orientation is detected during table block analysis.

SingleLinePerCell

Key: SingleLinePerCell Type: Boolean Default: False

Set this property to True if you only recognize tables with one line of text per each cell. The table layout will be analyzed more readily.

SplitOnlyBySeparators

Key: SplitOnlyBySeparators Type: Boolean Default: False

Set this property to True if you only recognize tables with no hidden separators. The table layout will be analyzed more readily.

5.5.8 [BarcodeParams] INI-File Section

Main Settings

Type

Key: Type Type: BarcodeTypeEnum Default: BT_Autodetect

The value of this property is an OR superposition of the BarcodeTypeEnum enumeration constants which denote the types of barcodes. For example, if it is set to BT_EAN13 | BT_EAN8, ABBYY FineReader Engine will try to recognize barcode blocks in either EAN 13 or EAN 8 standard, ignoring all other variants.

By default, this property is set to BT_Autodetect, i.e. ABBYY FineReader Engine will detect the barcode type automatically.

> **Note:** The default value allows detecting barcodes of all supported types. However, we recommend excluding IATA 2 of 5, Interleaved 2 of 5, and Matrix 2 of 5 barcode types unless you are certain that they will occur on your images. Barcodes of these types do not have a checksum and can therefore be mistakenly found on image areas that do not actually contain any barcodes.

BarcodeTypeEnum

BT_Autodetect Forces ABBYY FineReader Engine to automatically detect the barcode type during recognition.

BT_Aztec Barcode in Aztec standard.

BT_Codabar Barcode in Codabar standard.

BT_Code128 Barcode in Code 128 standard.

BT_Code32 Barcode in Code 32 standard.

BT_Code39 Barcode in Code 39 standard.

BT_Code93 Barcode in Code 93 standard.

BT_DataMatrix Barcode in Data Matrix standard.

BT_EAN13 Barcode in EAN 13 standard.

BT_EAN8 Barcode in EAN 8 standard.

BT_FullAscii Barcode in Full ASCII Code 39 standard.

BT_IATA25 Barcode in IATA 2 of 5 standard. (**no checksum, see note**)

BT_Industrial25 Barcode in Industrial 2 of 5 standard.

BT_IntelligentMail Barcode in Intelligent Mail standard.

BT_Interleaved25 Barcode in Interleaved 2 of 5 standard. (**no checksum, see note**)

BT_Matrix25 Barcode in Matrix 2 of 5 standard. (**no checksum, see note**)

BT_MaxiCode Barcode in MaxiCode standard.

BT_Patch Barcode in Patch standard.

BT_PDF417 Barcode in PDF417 standard.

BT_PostNet Barcode in PostNet standard.

BT_QRCode Barcode in QR Code standard.

BT_UCC128 Barcode in GS1-128 standard. The former name UCC-128.

BT_Unknown Denotes unknown type of barcode.

BT_UPCA Barcode in UPC-A standard.

BT_UPCE Barcode in UPC-E standard.

Orientation

Key: Orientation Type: BarcodeOrientationEnum Default: BO_Autodetect

The value of this property is an OR superposition of the BarcodeOrientationEnum enumeration constants which denote the types of barcode orientation. For example, if it is set to BO_Left_To_Right | BO_Down_To_Top, ABBYY FineReader Engine will presume that barcode blocks may be oriented either from left to right or from down to top, ignoring all other variants.

BarcodeOrientationEnum

BO_Autodetect Detect the barcode orientation automatically.

BO_Down_To_Top Barcode is oriented from down to top.

BO_Left_To_Right Barcode is oriented from left to right.

BO_Right_To_Left Barcode is oriented from right to left.

BO_Top_To_Down Barcode is oriented from top to down.

BO_Unknown Denotes unknown type of barcode orientation.

MinRatioToTextHeight

Key: MinRatioToTextHeight	Type: double	Default: -1

This property defines the minimal acceptable height of the barcode in relation to the average letters height. Use this setting if you need to allow detecting low barcodes.

You can only assign a positive value to this property, or the default value -1. The upper limit to this property value is 2147483647 ($= 2^{31} - 1$).

The default value of this property is -1, which means that ABBYY FineReader Engine will adjust this setting automatically.

Settings for certain barcode types

CodePage

Key: CodePage	Type: CodePageEnum	Default: CP_Null

This property is used to recognize PDF417, Aztec, Data Matrix, QR Code, MaxiCode barcodes which do not conform to the barcode specifications. Do not use this property for barcodes created in conformity with the barcode specifications. Some barcode printers use code pages other than the one required by the specifications. In this case, use this property to specify the code page which was used by the barcode printer to create the barcode. In most cases this will be the code page of the operating system under which the barcode printer was running.

> **Note:** The following code pages required by the specifications:
> - for PDF417 – DOS United States (437)
>   CP_US_MSDOS
> - for Aztec, DataMatrix, QR Code, MaxiCode – ISO Latin 1 (8859-1)
>   CP_Latin_ISO

The property is used to convert recognized data to a Unicode string.

By default, this property is set to CP_Null, which means that the code page required by the specification should be used.

CodePageEnum

- CP_Armenian – Windows Armenian
- CP_Armenian_Macintosh – Macintosh Armenian
- CP_Armenian_MSDOS – DOS Armenian
- CP_Baltic – Windows Baltic (1257)
- CP_Baltic_ISO – ISO Baltic (8859-4)
- CP_Baltic_MSDOS – DOS Baltic (775)
- CP_Bashkir – Windows Bashkir
- CP_Chinese_Simpl_GB – Chinese Simplified (GB2312)
- CP_Chinese_Simpl_Mac – Chinese Simplified (Mac)
- CP_Chinese_Trad_Big – Chinese Traditional (Big5)
- CP_Chinese_Trad_Mac – Chinese Traditional (Mac)

- CP_Croatian_Macintosh – Macintosh Croatian
- CP_Cyrillic – Windows Cyrillic (1251)
- CP_Cyrillic_ISO – ISO Cyrillic (8859-5)
- CP_Cyrillic_Macintosh – Macintosh Cyrillic
- CP_Cyrillic_MSDOS – DOS Cyrillic (855)
- CP_Digits – Digits
- CP_EasternEuropean – Windows Central Europe (1250)
- CP_EasternEuropean_ISO – ISO Central Europe (8859-2)
- CP_Greek – Windows Greek (1253)
- CP_Greek_737 – DOS Greek (737)
- CP_Greek_869 – DOS Modern Greek (869)
- CP_Greek_ISO – ISO Greek (8859-7)
- CP_Greek_Macintosh – Macintosh Greek 1
- CP_Hebrew – Windows Hebrew (1255)
- CP_Hebrew_ISO – ISO Hebrew (8859-8)
- CP_Hebrew_Macintosh – Macintosh Hebrew
- CP_Hebrew_MSDOS – DOS Hebrew (862)
- CP_Icelandic_Macintosh – Macintosh Icelandic
- CP_Japan_Mac – Japanese (Mac)
- CP_Japan_SJIS – Japanese (Shift-JIS)
- CP_KOI8 – Russian KOI8
- CP_Korean – Korean
- CP_Korean_Johab – Korean (Johab)
- CP_Korean_Mac – Korean (Mac)
- CP_Latin – Windows Western Europe (1252)
- CP_Latin_ISO – ISO Latin 1 (8859-1)
- CP_Latin2_Macintosh – Macintosh Latin 2
- CP_Latin5_ISO – ISO Turkish (8859-9)
- CP_LatinI_MSDOS – DOS Multilingual Latin 1 (850)
- CP_Mathematical – Mathematical symbols
- CP_Null – Invalid code page.
- CP_Roman_Macintosh – Macintosh Roman
- CP_Russian_MSDOS – DOS Russian (866)
- CP_Slavic_MSDOS – DOS Latin 2 (852)
- CP_Tatar – Windows Tatar
- CP_Tatar_MSDOS – DOS Tatar
- CP_Thai – Windows Thai (874)
- CP_Thai_Macintosh – Macintosh Thai
- CP_Turkish – Windows Turkish (1254)
- CP_Turkish_IBM – DOS Turkish (857)
- CP_Turkish_ISO – ISO Latin 3 (8859-3)
- CP_Turkish_Macintosh – Macintosh Turkish
- CP_Ukrainian_Macintosh – Macintosh Ukrainian
- CP_US_MSDOS – DOS United States (437)
- CP_Vietnamese – Vietnamese

ContainsBinaryData

Key: ContainsBinaryData Type: Boolean Default: False

This property makes sense only for PDF417, Aztec, Data Matrix, QR Code barcodes which encode some binary data. The property affects the way binary data is represented in the recognized text.

If this property is set to True, the binary data encoded in a barcode is saved as a sequence of hexadecimal values for corresponding bytes. If this property is False, the binary data is translated to the Unicode string using the code page specified in the CodePage property.

EnableAdvancedExtractionMode

Key: EnableAdvancedExtractionMode Type: Boolean Default: False

This property is only valid for twodimensional barcodes.

When you set this property to True, the advanced mode of barcode extraction is turned on More possibilities of barcodes location are considered; in particular, it enables FineReader Engine to find barcodes which are positioned at 45° to the page orientation. However, this setting slows down the processing.

HasChecksum

Key: HasChecksum Type: Boolean Default: False

Specifies whether the barcode being recognized must be interpreted as the barcode of the same type but with a check sum. This property is only available for barcodes of types Code 39, Interleaved 2 of 5, Codabar, and Matrix 2 of 5.

> **Note:** While Codabar has no check digit, ABBYY FineReader Engine uses an algorithm for computing check digits according to Modulo 16. The check digit is computed as follows. Each Codabar character has a value assigned to it. The sum of all character values is taken, including the Start and the Stop characters. The data character whose value, when added to this sum, equals a multiple of 16 is the check digit.

IsCode39WithoutAsterisk

Key: IsCode39WithoutAsterisk Type: Boolean Default: False

Specifies that the Code 39 barcode being recognized has no start and stop symbol, the asterisk "*".

SupplementType

Key: SupplementType Type: BarcodeSupplementTypeEnum Default: BS_Autodetect

The value of this property is an OR superposition of the BarcodeSupplementTypeEnum enumeration constants. This property is only available for barcodes of the EAN 8, 13, UPC-A, and UPC-E types. For example, if it is set to `BS_Void | BS_2Digits`, ABBYY FineReader Engine will try to recognize barcode blocks either without supplementary barcode or with 2-digit supplementary barcode.

BarcodeSupplementTypeEnum

BS_2Digits 2-digit supplementary barcode.

BS_5Digits 5-digit supplementary barcode.

BS_Autodetect Forces ABBYY FineReader Engine to automatically detect the supplementary barcode type during recognition.

BS_Unknown Denotes unknown type of supplementary barcode.

BS_Void No supplementary barcode.

ATTENTION: Removed or renamed properties

PDF417CodePage

Key: PDF417CodePage Type: Renamed

The new name of the property is CodePage. This property can be used for barcodes of several types besides PDF417.

5.5.9 [ObjectsExtractionParams] INI-File Section

Common settings

FastObjectsExtraction

Key: FastObjectsExtraction Type: Boolean Default: False

If this property is True, objects extraction will speed up, but its quality may deteriorate.

ProhibitColorImage

Key: ProhibitColorImage Type: Boolean Default: False

This property set to True tells ABBYY FineReader Engine to use only black-and-white plane during objects extraction. In this case detection quality of colored tables and pictures can get worse.

Objects removing

RemoveGarbage

Key: RemoveGarbage Type: Boolean Default: False

Specifies if garbage (excess dots that are smaller than a certain size) is to be removed from the image during objects extraction.

RemoveTexture

Key: RemoveTexture Type: Boolean Default: True

This property set to True tells ABBYY FineReader Engine to remove the background noise from a temporary image used for recognition. The source image remains unaffected.

Objects detection

DetectMatrixPrinter

Key: DetectMatrixPrinter Type: Boolean Default: True

If this property is True, the text printed on matrix printer is detected during objects extraction.

DetectPorousText

Key: DetectPorousText Type: Boolean Default: True

If this property is True, the regions with porous text are detected during objects extraction.

DetectTextOnPictures

Key: DetectTextOnPictures Type: Boolean Default: False

This property set to True tells ABBYY FineReader Engine to detect all text on an image, including text embedded into images. Reading order is not changed to provide ability for full-text search later.

EnableAggressiveTextExtraction

Key: EnableAggressiveTextExtraction Type: Boolean Default: False

If this property is True, FineReader Engine will try to extract as much text on the image as possible. We recommend using this mode when the image contains some low-quality text. This processing mode may lead to mistaken interpretation of pictures as text or vertical rearranging of the horizontal text.

ProhibitDottedSeparators

Key: ProhibitDottedSeparators Type: Boolean Default: False

If this property is set to True, ABBYY FineReader Engine presumes that the document does not contain dotted separators. This property may be useful, if you are sure that the document does not contain dotted separators, or if some data is incorrectly recognized as a dotted separator. This is a temporary property, it may be removed in a future version.

ATTENTION: Removed or renamed properties

FlexiFormsDA

Key: FlexiFormsDA Type: Behavior changed

This property is obsolete and will be removed in future versions.

The same functionality is now provided by `EnableAggressiveTextExtraction` property and `EnableTextExtractionMode` property. If you do not set this property specifically, its value will be ignored. If you do, both properties mentioned above become automatically set to the value you assigned to this property.

`FullTextIndexDA`

Key: `FullTextIndexDA` Type: `Behavior changed`

This property is obsolete and will be removed in future versions.

The same functionality is now provided by the `DetectTextOnPictures` property.

`SourceContentReuseMode`

Key: `SourceContentReuseMode` Type: `Default value changed`

The default value of this property is now `CRM_Auto`. This setting uses both the text and the image layer of the PDF file and ensures the highest quality of objects extraction.

5.5.10 `[OrientationDetectionParams]` INI-File Section

Orientation Detection Parameters

`OrientationDetectionMode`

Key: `OrientationDetectionMode` Type: `OrientationDetectionModeEnum` Default: `ODM_Normal`

Specifies the mode of page orientation detection.

OrientationDetectionModeEnum

`ODM_Fast` Fast mode.

This mode provides the fastest speed of orientation detection at the cost of a moderately decreased quality.

`ODM_Normal` Normal mode.

The normal mode is an intermediate mode between thorough and fast modes.

`ODM_Thorough` Thorough mode.

This mode provides the best quality of orientation detection.

`ProhibitClockwiseRotation`

Key: `ProhibitClockwiseRotation` Type: `Boolean` Default: `False`

Disables clockwise page rotation when selecting the page orientation.

> **Note:** This property must not have the True value if the ProhibitCounterclockwiseRotation and ProhibitUpsidedownRotation properties are set to True.

ProhibitCounterclockwiseRotation

Key: ProhibitCounterclockwiseRotation Type: Boolean Default: False

Disables counterclockwise page rotation when selecting the page orientation.

> **Note:** This property must not have the True value if the ProhibitClockwiseRotation and ProhibitUpsidedownRotation properties are set to True.

ProhibitUpsidedownRotation

Key: ProhibitUpsidedownRotation Type: Boolean Default: False

Disables upside-down page rotation when selecting the page orientation.

> **Note:** This property must not have the True value if the ProhibitClockwiseRotation and ProhibitCounterclockwiseRotation properties are set to True.

5.5.11 [RecognizerParams] INI-File Section

Main settings

TextLanguage

Key: TextLanguage Type: TextLanguage Default: English

The languages used for text recognition, separated by a comma.

The supported languages are listed in chapter 5.2.

LanguageDetectionMode

Key: `LanguageDetectionMode` Type: `ThreeStatePropertyValueEnum` Default: `TSPV_Auto`

Manages automatic language detection.

When language autodetection is on, the recognition language is detected for each word in the text. It is selected from the list of languages specified in the `TextLanguage` property. Autodetection is intended to be used during recognition of documents the language of which is not known to you.

If you know for certain that all the languages you specified are present in the document, autodetection is useless. Turn it off by setting this property to `TSPV_No`.

Supported values are ThreeStatePropertyValueEnum on page 41.

TextTypes

Key: `TextTypes` Type: `TextTypeEnum` Default: `TT_Normal`

The value of this property is an OR superposition of the TextTypeEnum enumeration constants which denote possible text types used for recognition. For example, if it is set to `TT_Normal | TT_Index`, ABBYY FineReader Engine will presume that the text contains only common typographic text and digits written in ZIP-code style, ignoring all other variants. See also Using Text Type Autodetection.

> **Note:**
> - If this property is equal to any combination of `TT_Matrix`, `TT_Typewriter`, `TT_OCR_A`, and `TT_OCR_B`, italic fonts and superscript/subscript will not be recognized, regardless of the values of the ProhibitItalic, ProhibitSubscript and ProhibitSuperscript properties.
> - If this property is `TT_Handprinted`, the CorrectOrientation property of the PagePreprocessingParams section cannot be set to `True`.
> - If the TrainUserPatterns property is set to `True`, the TextTypes property must contain only one text type.

TextTypeEnum

`TT_Gothic` This value tells ABBYY FineReader Engine to presume that the text on the recognized image is printed with the Gothic type.

`TT_Handprinted` This value corresponds to handprinted text.

Note that automatic analysis is not available for handprinted text. The coordinates of blocks containing handprinted text should be set manually.

`TT_Index` This constant corresponds to a special set of characters including only digits written in ZIP-code style.

`TT_Matrix` This value tells ABBYY FineReader Engine to presume that the text on the recognized image is printed on a dot-matrix printer.

`TT_MICR_CMC7` This value corresponds to a special set of characters, which includes only digits and A, B, C, D, E characters, written in MICR barcode font (CMC-7).

`TT_MICR_E13B` This value corresponds to a special set of characters including only digits and A, B, C, D characters printed in magnetic ink. MICR (Magnetic Ink Character Recognition) characters are found in a variety of places, including personal checks.

TT_Normal This value corresponds to a common typographic type of text.

TT_OCR_A This value corresponds to a monospaced font, designed for Optical Character Recognition. Largely used by banks, credit card companies and similar businesses.

TT_OCR_B This value corresponds to a font designed for Optical Character Recognition.

TT_Receipt This value corresponds to a text of a receipt. 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.

TT_Typewriter This value tells ABBYY FineReader Engine to presume that the text on the recognized image is typed on a typewriter.

Recognition speed

BalancedMode

Key: BalancedMode Type: Boolean Default: False

If this property is True, the recognition will run in balanced mode. The balanced mode is an intermediate mode between full and fast modes. The fast mode can be activated with the help of the FastMode property.

This property is available for machine-printed texts only, for hand-printed texts the recognition will be run in full mode.

FastMode

Key: FastMode Type: Boolean Default: False

This property set to True provides 2-2.5 times faster recognition speed at the cost of a moderately increased error rate (1.5-2 times more errors). This property is available both for machine- and hand-printed texts. In the case of a hand-printed text (text type TT_Handprinted), a special recognition mode is used. On good print quality texts, ABBYY FineReader Engine makes an average of 1-2 errors per page, and such moderate increase in error rate can be easily tolerated in many cases, such as full text indexing with "fuzzy" searches, preliminary recognition, etc.

> **Note:** We do not recommend using this mode to recognize small image fragments (for example, fragments which consist of only one line or word) because the time advantage will be insignificant.

Fine tuning

LowResolutionMode

Key: LowResolutionMode Type: Boolean Default: False

Specifies whether a text on an image with low resolution is recognized. This property is useful when recognizing faxes, small prints, images with low resolution or bad print quality.

`OneLinePerBlock`

Key: `OneLinePerBlock` Type: `Boolean` Default: `False`

This property set to `True` tells ABBYY FineReader Engine to presume that the text in block to which the current RecognizerParams object belongs contains no more than one string.

`OneWordPerLine`

Key: `OneWordPerLine` Type: `Boolean` Default: `False`

This property set to `True` tells ABBYY FineReader Engine to presume that no text line may contain more than one word, so the lines of text will be recognized as a single word.

`ProhibitItalic`

Key: `ProhibitItalic` Type: `Boolean` Default: `False`

This property set to `True` tells ABBYY FineReader Engine not to recognize letters printed with italic -styled font. It is useful when a text with presumably no italic letters is recognized, in which case it may speed up the recognition. If there exist any italic letters on the image, and this property is `True`, these letters will be recognized incorrectly.

`ProhibitSubscript`

Key: `ProhibitSubscript` Type: `Boolean` Default: `False`

This property set to `True` tells ABBYY FineReader Engine not to recognize subscript letters. It is useful when a text with presumably no subscripts is recognized, in which case it may speed up the recognition. If there exist any subscript letters on the image, and this property is `True`, these letters will be recognized incorrectly.

`ProhibitSuperscript`

Key: `ProhibitSuperscript` Type: `Boolean` Default: `False`

This property set to `True` tells ABBYY FineReader Engine not to recognize superscript letters. It is useful when a text with presumably no superscripts is recognized, in which case it may speed up the recognition. If there exist any superscript letters on the image, and this property is `True`, these letters will be recognized incorrectly.

ProhibitHyphenation

Key: ProhibitHyphenation Type: Boolean Default: False

This property set to True prohibits recognition of hyphenation from line to line. It is useful when a text with presumably no hyphenations is recognized, in which case it may speed up the recognition.If there exist any hyphenations in the recognized block, and this property is True, the hyphenated words will be recognized incorrectly.

ProhibitInterblockHyphenation

Key: ProhibitInterblockHyphenation Type: Boolean Default: False

This property set to True tells ABBYY FineReader Engine to presume that text from one block cannot be carried over to the next block.

CaseRecognitionMode

Key: CaseRecognitionMode Type: CaseRecognitionModeEnum Default: CRM_AutoCase

This property specifies the mode of letter case recognition.

CaseRecognitionModeEnum

CRM_AutoCase This value tells ABBYY FineReader Engine to automatically detect the case of letters and to keep it in the output text.

CRM_CapitalCase The recognized text will be set in capitals.

CRM_SmallCase The recognized text will be set in lowercase letters.

Handprint recognition

WritingStyle

Key: WritingStyle Type: WritingStyleEnum Default: WS_Auto

Provides additional information about handprinted letters writing style.

By default the value of this property is WS_Auto, which means that the writing style is automatically detected by FineReader Engine.

WritingStyleEnum

- WS_American – The American writing style.
- WS_Arabic – The Arabic writing style. This style does not contain any specific characters. There is no need to use this constant.
- WS_Auto – The writing style is detected by FineReader Engine automatically.

- WS_Azerbaijan – The Azerbaijan writing style.
- WS_Baltic – The Baltic writing style.
- WS_British – The British writing style.
- WS_Bulgarian – The Bulgarian writing style.
- WS_Canadian – The Canadian writing style.
- WS_Chinese – The Chinese writing style.
- WS_Common – The Esperanto writing style.
- WS_Croatian – The Croatian writing style.
- WS_Czech – The Czech writing style.
- WS_Default – This constant is deprecated and will be removed in future versions. Please use WS_Auto instead to ensure the best recognition result. If you need to select the writing style corresponding to the current operating system language, use WS_DetectByLocale, which has the same value and behavior.
- WS_DetectByLocale – The writing style is selected depending on the current language of the operating system.
- WS_French – The French writing style.
- WS_German – The German writing style.
- WS_Greek – The Greek writing style.
- WS_Hungarian – The Hungarian writing style.
- WS_Italian – The Italian writing style.
- WS_Japanese – The Japanese writing style.
- WS_Kazakh – The Kazakh writing style.
- WS_Kirgiz – The Kirgiz writing style.
- WS_Latvian – The Latvian writing style.
- WS_Polish – The Polish writing style.
- WS_Romanian – The Romanian writing style.
- WS_Russian – The Russian writing style.
- WS_Slovak – The Slovak writing style.
- WS_Spanish – The Spanish writing style.
- WS_Thai – The Thai writing style.
- WS_Turkish – The Turkish writing style. This style does not contain any specific characters. There is no need to use this constant.
- WS_Ukrainian – The Ukrainian writing style.

FieldMarkingType

Key: FieldMarkingType Type: FieldMarkingTypeEnum Default: FMT_SimpleText

This property specifies the type of marking around letters (for example, underline, frame, box, etc.).

This property is valid only for the handprint recognition.

> **Note:** For correct handprint recognition use CellsCount property that allows you to set the number of character cells for a recognized block.

FieldMarkingTypeEnum

FMT_CharBoxSeries This value specifies that the field where the text is located is a set of separate boxes.

FMT_CombInFrame This value specifies that the field where the text is located is a comb and that this comb is also the bottom line of a frame.

FMT_GrayBoxes This value specifies that the text is located in white fields on a gray background.

FMT_PartitionedFrame This value specifies that the field where the text is located is a frame and this frame is split by vertical lines.

FMT_SimpleComb This value specifies that the field where the text is located is a comb.

FMT_SimpleText This value denotes the plain text.

FMT_TextInFrame This value specifies that the text is enclosed in a frame.

FMT_UnderlinedText This value specifies that the text is underlined.

CellsCount

Key: CellsCount Type: Integer Default: 1

Specifies the number of character cells for a recognized block.

This property is valid only for the handprint recognition.

It has a sense only for the field marking types (the FieldMarkingType property) that imply splitting the text in cells.

Default value for this property is 1, but you should set the appropriate value to recognize the text correctly.

User patterns

UseBuiltInPatterns

Key: UseBuiltInPatterns Type: Boolean Default: True

This property set to True means that ABBYY FineReader Engine will use its own built-in patterns for recognition. Patterns are files establishing relationship between character image and character itself. You may want to set this property to False when you do not want to use standard ABBYY FineReader Engine patterns for character recognition, but user patterns only. This may be useful for recognition of text typed with decorative or nonstandard fonts. In this case it is better not to use ABBYY FineReader Engine built-in patterns, but use your own userdefined patterns trained for these fonts.

A path to user-defined pattern file is stored in the UserPatternsFile property. If the UserPatternsFile property is empty the UseBuiltInPatterns property is ignored. See also Recognizing with Training.

UserPatternsFile

Key: UserPatternsFile Type: String Default: ""

Contains the full path to a file of the user pattern used for recognition. If the value of this property is not empty, information from the user pattern file will be used during recognition.

If the UseBuiltInPatterns property is False, which means that standard ABBYY FineReader Engine patterns are not used during recognition, this property should contain a path to userdefined pattern file, as only information stored in it will be used. See also Recognizing with Training.

ATTENTION: Removed or renamed properties

TextType

Key: PossibleTextTypes Type: Removed

Key: TextType Type: Removed

The properties are obsolete and were deprecated in the previous version. Use the TextTypes property instead.

CJKTextDirection

Key: CJKTextDirection Type: Removed

If you set this property to TRT_Unknown, text direction will be detected automatically. By default, this property is TRT_Unknown as well.

5.5.12 [SynthesisParamsForPage] INI-File Section

Main settings

ParagraphExtractionMode

Key: ParagraphExtractionMode Type: ParagraphExtractionModeEnum Default: PEM_NormalExtraction

Specifies the mode of paragraph extraction.

ParagraphExtractionModeEnum

PEM_NormalExtraction Normal paragraph extraction.

PEM_RoughExtraction Extracts the minimal number of paragraphs (either one paragraph per block or only paragraphs which start with a dropped capital).

PEM_SingleLineParagraphsWithSpaceFormatting This constant is deprecated and will be removed in future versions. Each line is extracted to a separate paragraph formatted with spaces.

PEM_SingleLineParagraphsWithWordSeparationOnly Each line is extracted to a separate paragraph without space formatting, blank spaces are to separate words only.

DetectFontFormattingAtPageLevel

Key: DetectFontFormattingAtPageLevel Type: Boolean Default: False

If this property is set to True, font parameters are detected at the stage of page synthesis. This property set to True enables detection of subscripts, superscripts, italic-face type, small capital letters at the stage of page synthesis and allows you to set additional parameters using FontFormattingDetectionParams. If this property is False, the FontFormattingDetectionParams is ignored.

Important: With the default settings, ABBYY FineReader Engine 12 detects font parameters at the stage of document synthesis. If you set the value of this property to True, you must turn off detection of font parameters during document synthesis. To do this, set the DetectFontFormatting property to False. Detection of font parameters during page synthesis enables the program to speed up the subsequent document synthesis and decrease memory usage. However, the quality of font detection may deteriorate.

Color settings

DetectBackgroundColor

Key: DetectBackgroundColor Type: ThreeStatePropertyValueEnum Default: TSPV_Auto

If this property is set to TSPV_Yes, the background color is detected during page synthesis.

Supported values are ThreeStatePropertyValueEnum on page 41.

AllowGrayBackgroundColor

Key: AllowGrayBackgroundColor Type: ThreeStatePropertyValueEnum Default: TSPV_Auto

If this property is set to TSPV_Yes, the gray color is detected for background. Otherwise, background will be detected as black or white.

The value of this property is taken into account only if the DetectBackgroundColor property is set to TSPV_Yes or TSPV_Auto.

Supported values are ThreeStatePropertyValueEnum on page 41.

DetectTextColor

Key: DetectTextColor Type: ThreeStatePropertyValueEnum Default: TSPV_Auto

If this property is set to TSPV_Yes, the text color is detected during page synthesis.

Supported values are ThreeStatePropertyValueEnum on page 41.

CorrectDynamicRange

Key: CorrectDynamicRange Type: ThreeStatePropertyValueEnum Default: TSPV_Auto

If this property is set to TSPV_Yes, image colors will be corrected so that the background is white and the text is black, or vice versa, which improves image quality. Recognition, however, will slow down.

We recommend using this property only if the DetectBackgroundColor and DetectTextColor properties are set to TSPV_Yes or TSPV_Auto.

Supported values are ThreeStatePropertyValueEnum on page 41.

ATTENTION: Removed or renamed properties

[Deprecated] **Key:** AllowGrayBackgroundColor Type: Type changed

[Deprecated] **Key:** AllowGrayTextColo Type: Type changed

[Deprecated] **Key:** CorrectDynamicRange Type: Type changed

[Deprecated] **Key:** DetectBackgroundColor Type: Type changed

[Deprecated] **Key:** DetectTextColor Type: Type changed

These properties now take values of the new ThreeStatePropertyValueEnum,which provides the opportunity not only to turn the setting on and off, but also make the selection automatically (TSPV_Auto constant). All these properties have TSPV_Auto value by default.

5.5.13 [SynthesisParamsForDocument] INI-File Section

Main settings

DetectDocumentStructure

Key: DetectDocumentStructure Type: Boolean Default: True

Specifies whether document structure detection should be performed while document synthesis.

DetectFontFormatting

Key: DetectFontFormatting Type: Boolean Default: True

Specifies whether font formatting detection should be performed while document synthesis.

If this property is set to False, the FontFormattingDetectionParams section is ignored.

Important: With the default settings, ABBYY FineReader Engine 12 detects font parameters at the stage of document synthesis. If you set the value of this property to False, you must turn on detection of font parameters during page synthesis. To do this, set the DetectFontFormattingAtPageLevel property in the Synthesis-ParamsForPage section to True. Detection of font parameters during page synthesis enables the program to speed up the subsequent document synthesis and decrease memory usage. However, the quality of font detection may deteriorate.

Note: If your license does not support the Synthesis module, the default value of this property is False.

Additional settings

LowMemoryMode

Key: LowMemoryMode Type: Boolean Default: False

Specifies whether low memory mode should be used during document synthesis. If you set this property to True, ABBYY FineReader Engine will try to use no more than 600 Mb of memory during document synthesis by loading less pages to memory simultaneously. However, document synthesis may slow down and the quality may slightly deteriorate.

PagePoolSize

Key: PagePoolSize Type: Integer Default: 64

Specifies how many pages may be loaded by document synthesis simultaneously. This property allows you to decrease memory usage. We recommend to use the value in range from 32 to 64. The more the value, the more speed of processing. However, for processing big documents it is not recommended to use the highest values of this property, as this may lead to an out of memory error. The value less than 5 is ignored.

5.5.14 [FontFormattingDetectionParams] INI-File Section

Text decoration

DetectBold

Key: DetectBold Type: Boolean Default: True

If this property is set to True, boldface is detected during synthesis.

DetectItalic

Key: DetectItalic Type: Boolean Default: True

If this property is set to True, italic typeface is detected during synthesis.

DetectSubscriptsSuperscripts

Key: DetectSubscriptsSuperscripts Type: Boolean Default: True

If this property is set to `True`, subscript and superscript are detected during synthesis.

`DetectUnderlineStrikeout`

Key: `DetectUnderlineStrikeout` Type: `Boolean` Default: `True`

If this property is set to `True`, underline and strikeout are detected during synthesis.

`DetectSmallCaps`

Key: `DetectSmallCaps` Type: `Boolean` Default: `True`

If this property is set to `True`, small capital letters are detected during synthesis.

Font attributes

`DetectFontFamily`

Key: `DetectFontFamily` Type: `Boolean` Default: `True`

If this property is set to `True`, font name is detected during synthesis.

`DetectFontSize`

Key: `DetectFontSize` Type: `Boolean` Default: `True`

If this property is set to `True`, font size is detected during synthesis.

`DetectSerifs`

Key: `DetectSerifs` Type: `Boolean` Default: `True`

If this property is set to `True`, serif is detected during synthesis, i.e. if serif has been detected, serif typeface is selected to represent the recognized text. If this property is set to `False`, serif is ignored. This means that the most suitable font (from both serif and sans serif typefaces) is selected to represent the recognized text, no matter whether the text is serif or sans serif.

Scaling and spacing

DetectScaling

Key: DetectScaling Type: Boolean Default: True

If this property is set to True, scaling is detected during synthesis.

DetectSpacing

Key: DetectSpacing Type: Boolean Default: True

If this property is set to True, spacing is detected during synthesis.

MonospaceDetectionMode

Key: MonospaceDetectionMode Type: MonospaceDetectionModeEnum Default: MDM_Auto

Specifies the mode of monospace font detection.

MonospaceDetectionModeEnum

MDM_Auto	Detect whether the font is monospace automatically.
MDM_Ignore	Do not detect whether the font is monospace.
MDM_Monospace	Set the font to monospace.
MDM_NotMonospace	Set the font to non-monospace.

ATTENTION: Removed or renamed properties

DetectMonospace

Key: DetectMonospace Type: Removed

Use the MonospaceDetectionMode property instead. If you do not want monospaced typeface to be detected, set the MonospaceDetectionMode property to MDM_Ignore.

6 Version History

6.1 Changes in Version 6

No functional changes.

6.2 Changes in Version 5

- **New** additional supported operating system: Windows Server 2019.

6.3 Changes in Version 4.12

- **New** HTTP proxy setting in the GUI license manager.

6.4 Changes in Version 4.11

- **New** support for reading PDF 2.0 documents.

6.5 Changes in Version 4.10

- **Improved** robustness against corrupt input PDF documents.

6.6 Changes in Version 4.9

- **Improved** support for and robustness against corrupt input PDF documents.
- **Improved** repair of embedded font programs that are corrupt.
- **New** support for OpenType font collections in installed font collection.
- **Improved** metadata generation for standard PDF properties.

6.7 Changes in Version 4.8

- **Improved** creation of annotation appearances to use less memory and processing time.
- **Added** repair functionality for TrueType font programs whose glyphs are not ordered correctly.

7 Licensing, Copyright, and Contact

PDF Tools AG is a world leader in PDF (Portable Document Format) software, delivering reliable PDF products to international customers in all market segments.

PDF Tools AG provides server-based software products designed specifically for developers, integrators, consultants, customizing specialists and IT-departments. Thousands of companies worldwide use our products directly and hundreds of thousands of users benefit from the technology indirectly via a global network of OEM partners. The tools can be easily embedded into application programs and are available for a multitude of operating system platforms.

Licensing and Copyright The 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12 is copyrighted. This user's manual is also copyright protected; It may be copied and given away provided that it remains unchanged including the copyright notice.

Contact

PDF Tools AG
Brown-Boveri-Strasse 5
8050 Zürich
Switzerland
http://www.pdf-tools.com
pdfsales@pdf-tools.com