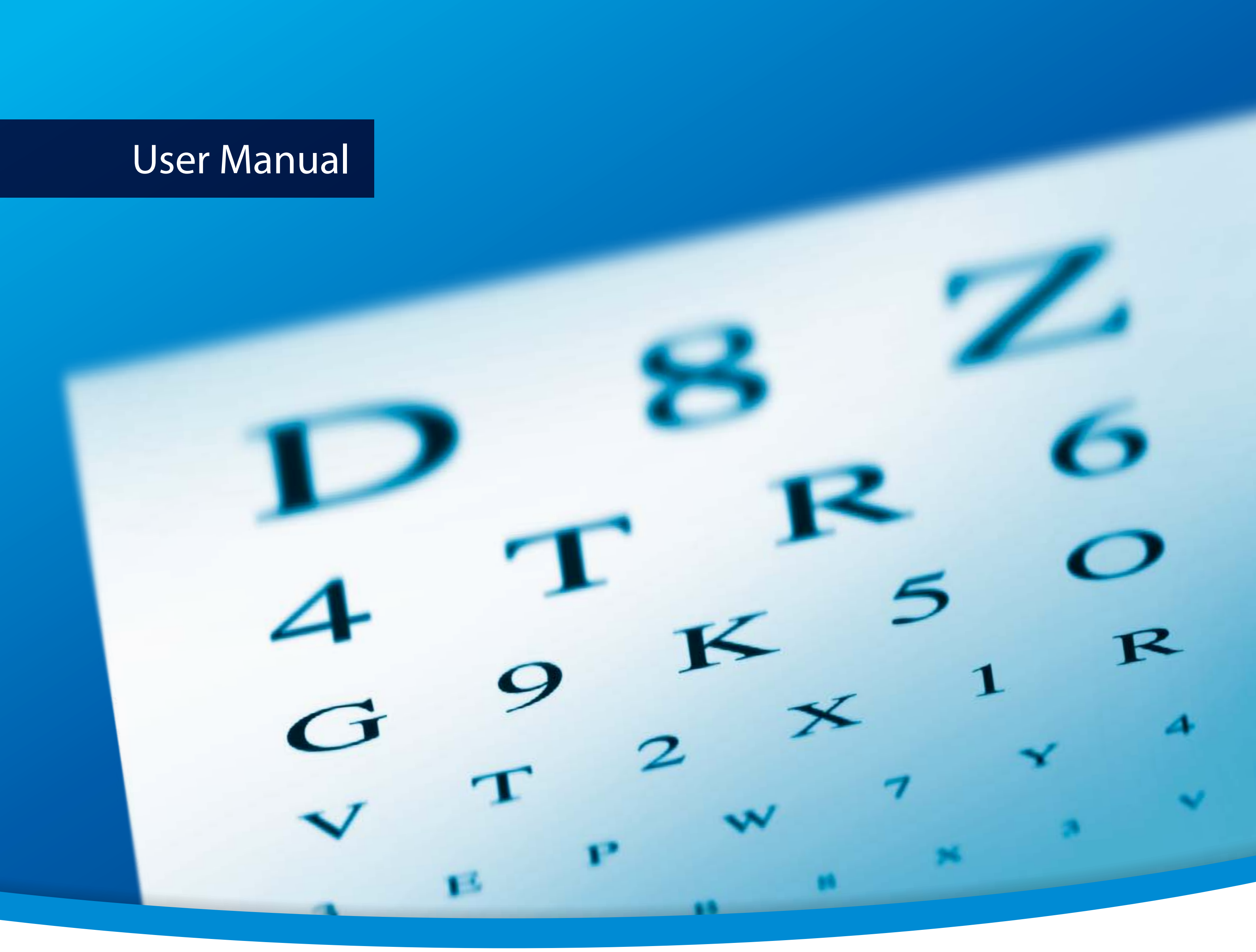
User Manual



3-Heights™ OCR Add-On for ABBYY FineReader Engine v10

Version 6.12.1

PDF-TOOLS.COM
Premium PDF Technology

Contents

1 Introduction

1.1 Overview

The 3-Heights™ OCR Add-On for ABBYY FineReader Engine v10 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 v10 contain searchable texts.

The optional 3-Heights™ OCR Add-On for ABBYY FineReader Engine v10 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 v10 is available for the following operating systems:

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

'+' 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:

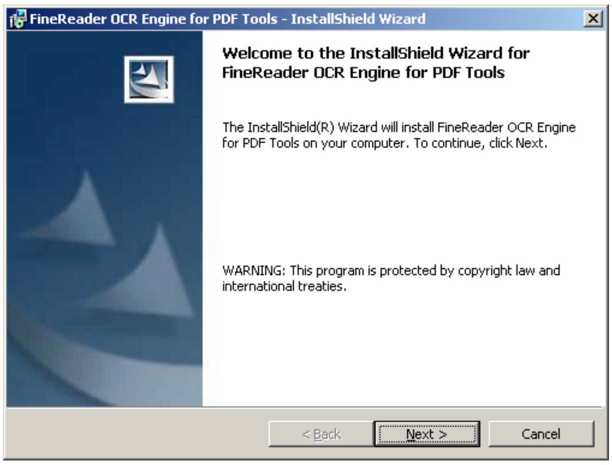
- How to install the ABBYY FineReader OCR Engine and how to activate its license.
- How to install the 3-Heights™ PDF OCR Service.
- How to setup a tool that uses the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v10 so that is uses the OCR engine directly or via OCR service.

2.1 Installation of ABBYY FineReader Engine

The Windows version of the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v10 is distributed in MSI-format (Microsoft Software Installation). The MSI file is named `OCR-FRE10-Installer.zip`.

The screenshots are from ABBYY 8, the installation for ABBYY 10 is equivalent.

1. Download the MSI installer `OCR-FRE10-Installer.zip` from your download account at 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”.



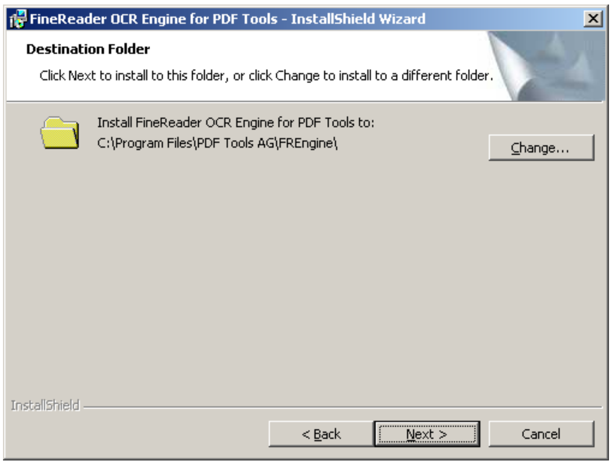
FineReader OCR Engine for PDF Tools - InstallShield Wizard

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

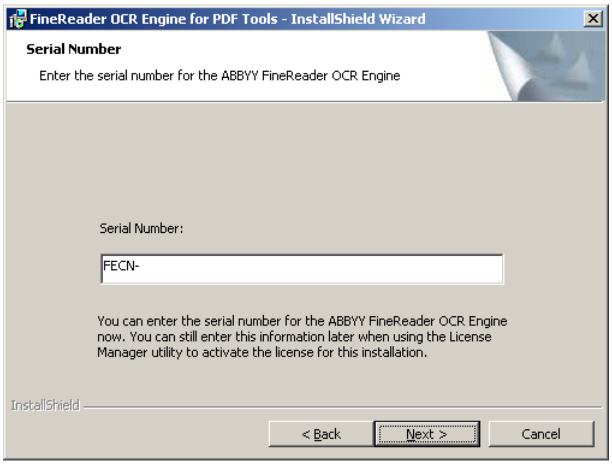
Destination Folder

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

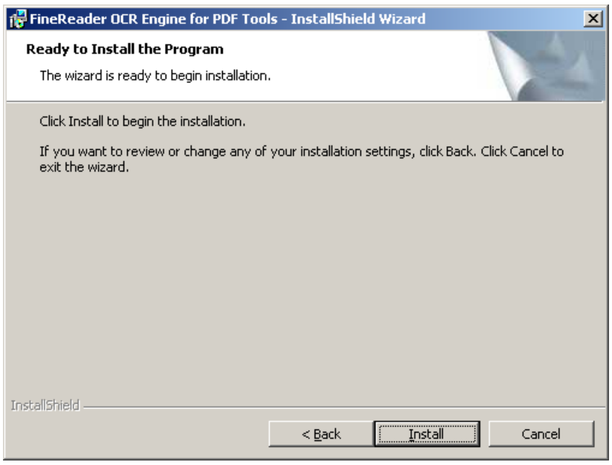
Serial Number

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

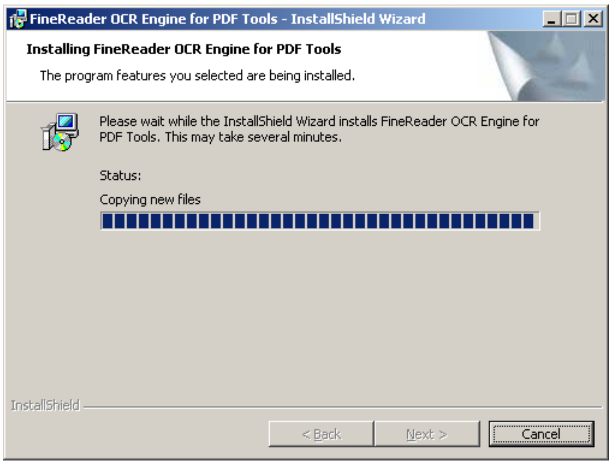
Ready to Install the Program

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:

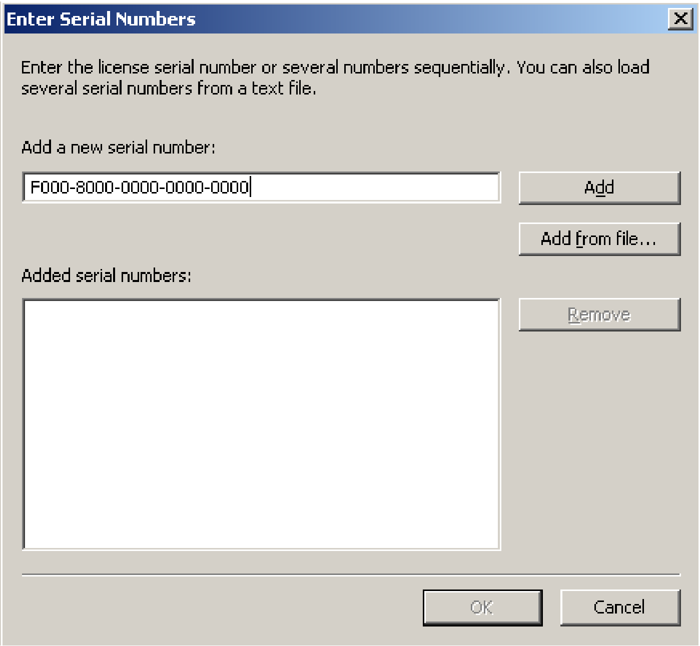
Copying new files

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 Serial Numbers

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

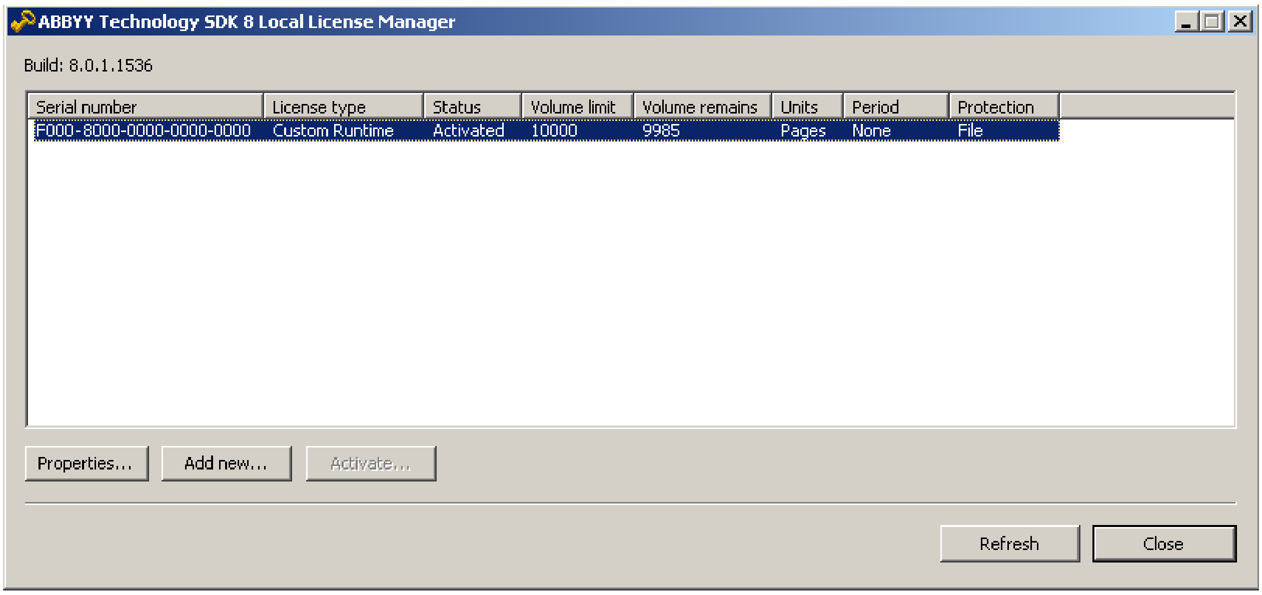
Add a new serial number: F000-8000-0000-0000-0000

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.



ABBYY Technology SDK 8 Local License Manager

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\10\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 v10 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 v10 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 v10:

- 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 User's Guide

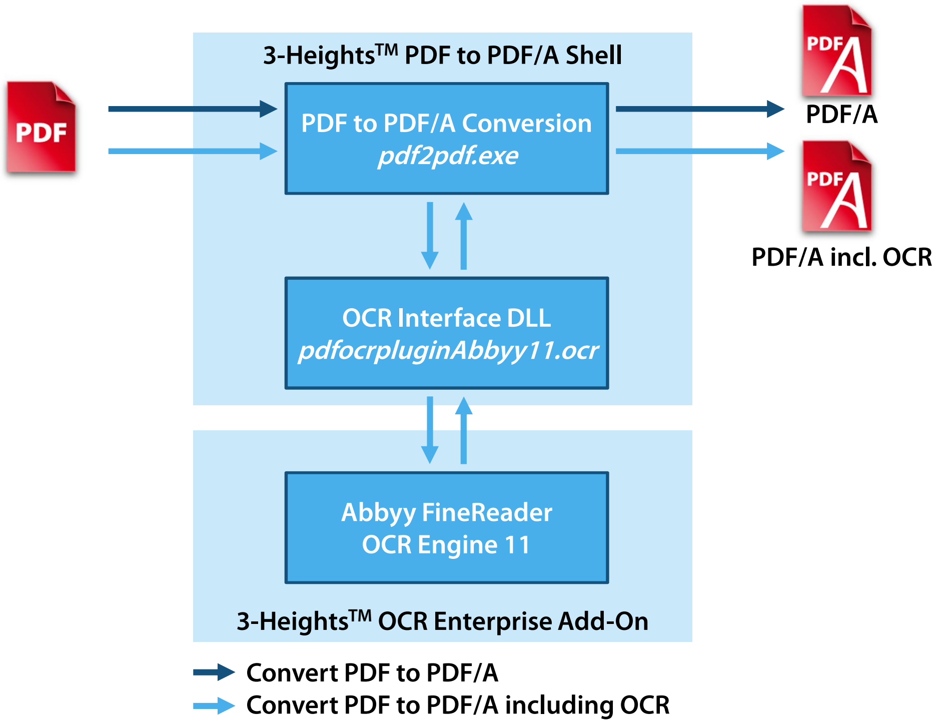
3.1 Architecture

3.1.1 Use OCR Engine Directly

How the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v10 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 v10 installs the OCR engine which can be invoked from the 3-Heights™ PDF to PDF/A Shell.

The OCR interface DLL (`pdfocrpluginabbyy10.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 v10.

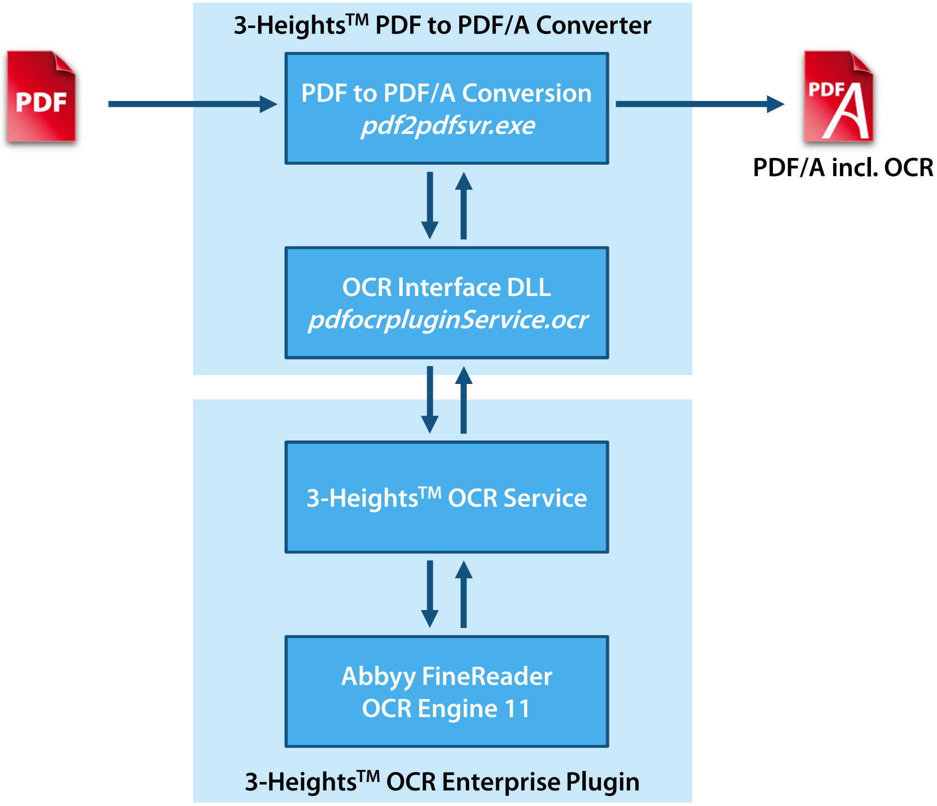


Once the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v10 is installed, it can be used by other applications of the 3-Heights™ line. How these other applications make use of the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v10 is individual for every product.

3.1.2 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.

3.2 OCR Related Switches and Functions

The 3-Heights™ OCR Add-On for ABBYY FineReader Engine v10 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. `pdfocrpluginabbyy10.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 10 is `"abbyy10"`.
		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.

3.2.1 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. `pdfocrpluginabbyy10.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 10 is `"abbyy10"`.

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.

3.3 Examples

This chapter describes how the 3-Heights™ OCR Add-On for ABBYY FineReader Engine v10 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.

3.3.1 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 v10.

```
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 v10 to be installed.

```
pdf2pdf -cl pdfa-1b -ocr "abbyy10" 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 v10 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("abbyy10")
```

3.3.2 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 "abbyy10" -ocl "English, German" input.pdf output.pdf
```

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

```
SetOCREngine("abbyy10")
SetOCRLanguage("English, German")
```

This setting is optional.

3.3.3 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 "abbyy10" -ocp "Profile=C:\ocr.ini" input.pdf output.pdf
```

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

```
SetOCREngine("abbyy10")
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.

3.4 Impacts of Using OCR

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

3.4.1 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).

3.4.2 File Size

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

3.4.3 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).

3.5 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

3.6 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

3.6.1 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 v10 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.

3.6.2 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.

3.6.3 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%.

4 Reference Manual

The FineReader Engine 10 is controlled using profiles.

4.1 Supported Languages

Please note:

- Not all languages are supported in the base version.
- The Engine does not support language combinations consisting of two hieroglyphic languages or combinations of a hieroglyphic language and a non-hieroglyphic language. However, the Engine will recognize bilingual texts written in English and a hieroglyphic language - when you select “ChineseTraditional”, “ChineseSimplified”, “Japanese” or “Korean”, the Engine will add the “English” recognition language automatically.

The following languages are supported

- Abkhaz
- Adyghe
- Afrikaans
- Agul
- Albanian
- Altaic
- ArmenianEastern
- ArmenianGrabar
- ArmenianWestern
- Awar
- Aymara
- AzeriCyrillic
- AzeriLatin
- Bashkir
- Basque
- Belarusian
- Bemba
- Blackfoot
- Breton
- Bugotu
- Bulgarian
- Buryat
- Catalan
- Chamorro
- Chechen
- ChineseSimplified
- ChineseTraditional
- Chukcha
- Chuvash
- Corsican
- CrimeanTatar
- Croatian
- Crow
- Czech
- Danish
- Dargwa
- Dungan
- Dutch
- DutchBelgian
- English
- EskimoCyrillic
- EskimoLatin
- Esperanto
- Estonian
- Even
- Evenki
- Faeroese
- Fijian
- Finnish
- French
- Frisian
- Friulian
- GaelicScottish
- Gagauz
- Galician
- Ganda
- German
- GermanNewSpelling
- GermanLuxembourg
- Greek
- Guarani
- Hani
- Hausa
- Hawaiian
- Hebrew
- Hungarian
- Icelandic
- Ido
- Indonesian
- Ingush
- Interlingua
- Irish
- Italian
- Japanese
- Kabardian
- Kalmyk
- KarachayBalkar
- Karakalpak
- Kasub
- Kawa
- Kazakh
- Khakas
- Khanty
- Kikuyu
- Kirgiz
- Kongo
- Korean
- Koryak
- Kpelle
- Kumyk
- Kurdish
- Lak
- Lappish
- Latin
- Latvian
- Lezgin
- Lithuanian
- Luba
- Macedonian
- Malagasy
- Malay
- Malinke
- Maltese
- Mansi
- Maori
- Mari
- Maya
- Miao
- Minankabaw
- Mohawk
- Moldavian
- Mongol
- Mordvin
- Nahuatl
- Nenets
- Nivkh
- Nogay
- Norwegian
- NorwegianBokmal
- NorwegianNynorsk
- Nyanja
- Occidental
- Ojibway
- OldEnglish
- OldFrench
- OldGerman
- OldItalian
- OldSpanish
- Ossetic
- Papiamento
- PidginEnglish
- Polish
- PortugueseBrazilian
- PortugueseStandard
- Provencal
- Quechua

- RhaetoRomanic
- Romanian
- RomanianMoldavia
- Romany
- Ruanda
- Rundi
- RussianOldSpelling
- Russian
- Samoan
- Selkup
- SerbianCyrillic
- SerbianLatin
- Shona
- Sioux
- Slovak
- Slovenian
- Somali
- Sorbian
- Sotho
- Spanish
- Sunda
- Swahili
- Swazi
- Swedish
- Tabassaran
- Tagalog
- Tahitian
- Tajik
- Tatar
- Thai
- Tinpo
- Tongan
- Tswana
- Tun
- Turkish
- Turkmen
- Tuvin
- Udmurt
- UighurCyrillic
- UighurLatin
- Ukrainian
- UzbekCyrillic
- UzbekLatin
- Visayan
- Welsh
- Wolof
- Xhosa
- Yakut
- Zapotec
- Zulu
- Basic
- C++
- Cobol
- Fortran
- Java
- Pascal
- Chemistry
- E13B
- CMC7
- Mixed
- Digits

4.2 Engine Parameters

The string parameter for ABBYY 10 is composed by a sequence of Key-Value pairs that are separated by semicolons (;). In order to form the string parameter the following keys are supported.

- PredefinedProfile (Section 4.3)
- Profile (Section 4.4)
- RecognizeBlankPages (Section 4.2.1)
- BlankPageMargin (Section 4.2.2)
- DisableMaskEmbedding (Section 4.2.3)
- RemoveGarbageOnly (Section 4.2.4)

Example: (Shell) Set the custom profile `Profile.ini` and use predefined profile `TextExtraction_Speed`.

```
-ocp "Profile=C:\path\Profile.ini;PredefinedProfile=TextExtraction_Speed"
```

4.2.1 RecognizeBlankPages

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`.

Example: (Shell) Choose ABBYY 10 engine, recognize blank pages and store the recognition information (i.e. information about which pages are recognized as blank) in output.xml.

```
-ocr "abbyy10" -ocp "RecognizeBlankPages=true" -ocx "output.xml"
```

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

4.2.2 BlankPageMargin

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. The default value is 0.02. This parameter is only active if at the same time the Key-Value pair RecognizeBlankPages=true is set.

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 "abbyy10" -ocp "RecognizeBlankPages=true;BlankPageMargin=0.05"
```

4.2.3 DisableMaskEmbedding

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 10 engine, make ocr recognition, but no mask shell be embedded in the output file.

```
-ocr "abbyy10" -ocp "DisableMaskEmbedding=true" in.tif out.tif
```

4.2.4 RemoveGarbageOnly

De-noise bitonal image, no further ocr processing is performed. 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.

Example: (Shell) Remove garbage with maximum size of 50 pixels.

```
-ocr "abbyy10" -ocp "RemoveGarbageOnly=50" in.tif out.tif
```

Note: In any case no OCR is performed, hence ABBYY's count for remaining pages isn't lowered. The changes are visible in the output file.

4.3 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:

- DocumentConversion_Accuracy – for converting documents into editable formats, optimized for accuracy
- DocumentConversion_Speed – for converting documents into editable formats, optimized for speed
- DocumentArchiving_Accuracy – for creating an electronic archive, optimized for accuracy
- DocumentArchiving_Speed – for creating an electronic archive, optimized for speed
- BookArchiving_Accuracy – for creating an electronic library, optimized for accuracy
- BookArchiving_Speed – for creating an electronic library, optimized for speed

- TextExtraction_Accuracy – for extracting text from documents, optimized for accuracy
- TextExtraction_Speed – for extracting text from documents, optimized for speed
- FieldLevelRecognition – for recognizing short text fragments
- BarcodeRecognition – for extracting barcodes
- Version9Compatibility – provided for compatibility, sets the processing parameters to the default values of ABBYY FineReader Engine 9.0.

Example: (Shell) Set the predefined profile DocumentArchiving_Accuracy.

```
-ocr "abbyy10" -ocp "PredefinedProfile=DocumentArchiving_Accuracy"
```

4.4 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. The following sections are supported:

- PrepareImageMode
- PageProcessingParams
- PageAnalysisParams
- BarcodeParams
- RecognizerParams
- ObjectsExtractionParams
- OrientationDetectionParams
- SynthesisParamsForDocument

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 "abbyy10" -ocp "Profile=C:\path\Profile1.ini"
```

4.4.1 [PrepareImageMode] INI-File Section

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 the AutoOverwriteResolution property is True, ABBYY FineReader Engine will automatically detect and overwrite image resolution.

CorrectSkewBy

Key: CorrectSkewByBlackSquaresHorizontally Type: Boolean

Key: CorrectSkewByBlackSquaresVertically Type: Boolean

Key: CorrectSkewByHorizontalLines Type: Boolean

Key: CorrectSkewByHorizontalText Type: Boolean

Key: CorrectSkewByVerticalLines Type: Boolean

Key: CorrectSkewByVerticalText Type: Boolean

These properties are obsolete. Use the CorrectSkewMode property instead.

CorrectSkewMode

Key: CorrectSkewMode Type: Long

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 default value is `CSM_CorrectSkewByHorizontalText | CSM_CorrectSkewByVerticalText`

CorrectSkewMode Enumeration

CSM_CorrectSkewByBlackSquaresHorizontally
CSM_CorrectSkewByBlackSquaresVertically
CSM_CorrectSkewByHorizontalLines
CSM_CorrectSkewByHorizontalText
CSM_CorrectSkewByVerticalLines
CSM_CorrectSkewByVerticalText

CreatePreview

Key: CreatePreview Type: Boolean Default: False

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

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.

ImageCompression

Key: ImageCompression Type: ImageCompressionEnum Default: IC_Auto

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

Support enums are:

ImageCompression Enumeration

IC_Auto
IC_Compress
IC_NoCompression

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.

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).

See also IImageDocument::ChangeResolution.

PreviewHeight

Key: PreviewHeight Type: Long 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: Long 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.

Rotation

Key: Rotation Type: RotationTypeEnum Default: no rotation

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

Support enums are:

RotationType Enumeration, reference=rotationtypeenum

RT_UnknownRotation
RT_NoRotation
RT_Clockwise
RT_Counterclockwise
RT_Upsidedown

XResolutionToOverwrite

Key: XResolutionToOverwrite Type: Long 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 ResolutionToOverwrite and YResolutionToOverwrite.

YResolutionToOverwrite

Key: YResolutionToOverwrite Type: Long 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.

4.4.2 [PageProcessingParams] INI-File Section

DetectBarcodes

Key: DetectBarcodes Type: Boolean Default: False

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

DetectOrientation

Key: DetectOrientation Type: Boolean Default: False

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

PerformPageAnalysis

Key: PerformPageAnalysis Type: Boolean Default: True

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

RemoveGeometricalDistortions

Key: RemoveGeometricalDistortions Type: Boolean Default: False

Specifies if geometrical distortions (perspective on photos, curved lines from scanned books, etc.) should be removed during layout analysis.

4.4.3 [PageAnalysisParams] INI-File Section

DetectPictures

Key: DetectPictures Type: Boolean Default: True

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

DetectSeparators

Key: DetectSeparators Type: Boolean Default: True

If this property is True, the separators 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.

DetectTables

Key: DetectTables Type: Boolean Default: True

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

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.

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.

SingleColumnMode

Key: `SingleColumnMode` Type: `Boolean` Default: `False`

If this property is set to True, the analysis procedure presumes that there is only one column of text on a page.

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

4.4.4 [BarcodeParams] INI-File Section

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 "*".

Orientation

Key: `Orientation` Type: Default: `BO_Autodetect`

The value of this property is an OR superposition of the Orientation Enumeration constants on 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.

By default, this property is set to `BO_Autodetect`, i.e. ABBYY FineReader Engine will detect the barcode orientation automatically.

Orientation Enumeration

Autodetect
Unknown
Left_To_Right
Down_To_Top
Right_To_Left
Top_To_Down

PDF417CodePage

Key: `PDF417CodePage` Type: Default: `CP_Null`

This property is used to recognize 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 US-MSDOS 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.

Supported values are shown in PDF417CodePage Enumeration, use prefix CP_.

PDF417CodePage Enumeration

Null	Armenian	Armenian_Macintosh
Armenian_MSDOS	Baltic	Baltic_ISO
Baltic_MSDOS	Bashkir _ISO	Croatian_Macintosh
Cyrillic	Cyrillic_ISO	Cyrillic_Macintosh
Cyrillic_MSDOS	EasternEuropean	EasternEuropean_ISO
Greek	Greek_ISO	Greek_Macintosh
Greek_737	Greek_869	Icelandic_Macintosh
KOI8	Latin	Latin_ISO
LatinI_MSDOS	Latin2_Macintosh	Latin5_ISO

PDF417CodePage Enumeration

Roman_Macintosh	Russian_MSDOS	Slavic_MSDOS
Tatar	Tatar_MSDOS	Turkish
Turkish_IBM	Turkish_ISO	Turkish_Macintosh
Ukrainian_Macintosh	US_MSDOS	

SupplementType

Key: SupplementType Type: 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.

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

Supported values are shown in SupplementType Enumeration, use prefix BS_.

SupplementType Enumeration

Autodetect
Unknown
Void
2Digits
5Digits

Type

Key: Type Type: 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.

Supported values are shown in Type Enumeration, use prefix BT_.

Type Enumeration

Autodetect	Codabar
Code39	Code93
Code128	EAN8
EAN13	IATA25
Industrial25	Interleaved25
Matrix25	PDF417
PostNet	UCC128
Unknown	UPCE

4.4.5 [RecognizerParams] INI-File Section

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.

CaseRecognitionMode

Key: CaseRecognitionMode Type: CaseRecognitionModeEnum Default: CRM_AutoCase

This property specifies the mode of letter case recognition.

By default the value of this property is CRM_AutoCase, which corresponds to automatic case recognition.

Supported values are:

Supported values are shown in CaseRecognitionMode Enumeration, use prefix CRM_.

CaseRecognitionMode Enumeration

AutoCase
SmallCase
CapitalCase

CellsCount

Key: CellsCount Type: Long 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.

CJKTextDirection

Key: CJKTextDirection Type: CJKTextDirectionEnum Default: CJKTD_Autodetect

Sets the direction of the text to be recognized. This property is valid only for the hieroglyphic languages.

Supported values are in CJKTextDirection Enumeration, use prefix CJKTD_.

CJKTextDirection Enumeration

AutoDetect
Horizontal
Vertical

ErrorHiliteLevel

Key: ErrorHiliteLevel Type: ErrorHiliteLevelEnum Default: EHL_Standard

Specifies the level at which the CharParams::IsSuspicious property is set to True for a recognized character. The name of the property reflects the fact that the uncertain characters are highlighted with color in ABBYY FineReader.

Supported values are:

ErrorHiliteLevel Enumeration

EHL_AllText
EHL_None
EHL_Scanty
EHL_Standard
EHL_Thorough

ExactConfidenceCalculation

Key: `ExactConfidenceCalculation` Type: `Boolean` Default: `False`

If this property is True, character and word confidence will be defined more accurately, but recognition speed may get slower. The value of character confidence is stored in the CharConfidence property of the CharacterRecognitionVariant and PlainText objects. The value of word confidence is stored in the WordConfidence property of the WordRecognitionVariant object.

This property is automatically set to True if the SaveCharacterRecognitionVariants or SaveWordRecognitionVariants property is True.

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.

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.

By default the value of this property is `FMT_SimpleText` which means the plain text.

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

Supported values are:

FieldMarkingType Enumeration

FMT_CharBoxSeries	FMT_CombInFrame
FMT_GrayBoxes	FMT_PartitionedFrame
FMT_SimpleComb	FMT_SimpleText
FMT_TextInFrame	FMT_UnderlinedText

LowResolutionMode

Key: LowResolutionMode Type: Boolean Default: False

Specifies whether a text on an image with low resolution is recognized.

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.

The property is obsolete. Use the TextTypes property instead.

PossibleTextTypes

Key: PossibleTextTypes Type: LongCollection Default: TT_Normal

This property contains a collection of TextType Enumeration values.

The property tells ABBYY FineReader Engine to presume that the text to recognize is of one of the types the collection contains. If the value of the TextType property is not `TT_ToBeDetected`, the value of this property will be ignored. The property returns a copy of the collection but not a reference to it. In order to modify the value of the property it is necessary to create a new collection, add required values to it, and then assign the collection to the property.

The collection should contain at least one element and cannot contain `TT_ToBeDetected`. When this property is changed, the TextType property is automatically set to `TT_ToBeDetected`. By default it contains `TT_Normal`.

TextType Enumeration

Gothic	Handprinted
Index	Matrix
MICR_CMC7	MICR_E13B
Normal	OCR_A
OCR_B	ToBeDetected
Typewriter	

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.

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.

SaveCharacterRecognitionVariants

Key: SaveCharacterRecognitionVariants Type: Boolean Default: False

Specifies whether the variants of characters recognition are saved. The ICharParams::CharacterRecognitionVariants property returns a collection of recognition variants for a character.

See also Using Voting API.

SaveCharacterRegions

Key: SaveCharacterRegions Type: Boolean Default: Flase

Specifies whether the exact characters regions (ICharParams::CharacterRegion) are saved.

SaveWordRecognitionVariants

Key: SaveWordRecognitionVariants Type: Boolean Default: False

Specifies whether the variants of recognition of a word are saved. The IParagraph::GetWordRecognitionVariants method and ICharParams::WordRecognitionVariants property return a collection of recognition variants for a word.

See also Using Voting API.

TextLanguage

Key: TextLanguage Type: TextLanguage Default: English

This property refers to the TextLanguage object used for image recognition.

By default this parameter is initialized with English language.

This property may be easily set via the SetPredefinedTextLanguage method.

TextTypes

Key: TextTypes Type: Long Default: TT_Normal

This property tells ABBYY FineReader Engine to presume that the text to recognize is of that type.

> **Note:** If this property is set to TT_ToBeDetected, TT_Handprinted, or TT_Index, the TrainUserPatterns property cannot be set to True.
> 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.

The property cannot be set to TT_ToBeDetected. See also Using Text Type Autodetection.

> **Note:** If this property is set to TT_Handprinted, or TT_Index, the TrainUserPatterns property cannot be set to True. 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.

Supported values are in TextType Enumeration on Page 34, use prefix TT_.

TrainUserPatterns

Key: TrainUserPatterns Type: Boolean Default: False

This property specifies whether user patterns should be trained during the recognition. If this property is True, some user pattern file should be specified in the UserPatternsFile property. The Pattern Training dialog box will display during recognition. For correct operation of pattern training process it is necessary to set the value of the parent window HWND handle (IEngine::ParentWindow property). See also Recognizing with Training.

If this property is set to True, the TextType and TextTypes properties cannot be set to TT_ToBeDetected, TT_Handprinted, or TT_Index.

> **Note:** Pattern training is not supported for hieroglyphic languages.

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 non- standard fonts. In this case it is better not to use ABBYY FineReader Engine built-in patterns, but use your own user-defined patterns trained for these fonts.

`UserPatternsFile`

Key: `UserPatternsFile` Type: `String` Default: `""`

UserPatternsFile property. If the UserPatternsFile property is empty the UseBuiltInPatterns property is ignored. See also Recognizing with Training. Contains the full path to a file of the user pattern used for recognition.

By default this property stores an empty string.

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 user-defined pattern file, as only information stored in it will be used.

See also Recognizing with Training.

`WritingStyle`

Key: `WritingStyle` Type: `WritingStyleEnum` Default: `WS_Default`

Provides additional information about handprinted letters writing style.

By default the value of this property is `WS_Default`, which means that the writing style is selected depending on the current language of the operating system.

Supported values are shown in WritingStyle Enumeration, use prefix `WS_`.

WritingStyle Enumeration

American	Arabic
Baltic	British
Bulgarian	Canadian
Common	Croatian
Czech	French
German	Greek
Hungarian	Italian
Japanese	Polish
Romanian	Russian

WritingStyle Enumeration

Slovak	Spanish
Thai	Turkish
Ukrainian	

4.4.6 `[ObjectsExtractionParams]` INI-File Section

`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.

`FastObjectsExtraction`

Key: `FastObjectsExtraction` Type: `Boolean` Default: `False`

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

`FlexiFormsDA`

Key: `FlexiFormsDA` Type: `Boolean` Default: `False`

This property set to True tells ABBYY FineReader Engine to locate all text on the page, including small text areas of low quality and text in diagrams and pictures. Tables are recognized as plain text.

`FullTextIndexDA`

Key: `FullTextIndexDA` Type: `Boolean` Default: `False`

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

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.

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.

4.4.7 [OrientationDetectionParams] INI-File Section

OrientationDetectionMode

Key: OrientationDetectionMode	Type: OrientationDetectionModeEnum	Default: ODM_Normal

Specifies the mode of page orientation detection.

OrientationDetectionMode Enumeration

ODM_Fast
ODM_Normal
ODM_Thorough

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.

4.4.8 `[SynthesisParamsForDocument]` INI-File Section

`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.

InsertEmptyParagraphsForBigInterlines

Key: InsertEmptyParagraphsForBigInterlines Type: Boolean Default: False

If this property is set to True, empty paragraphs are inserted to reproduce big line spacing of the original text.

PagePoolSize

Key: PagePoolSize Type: Long 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.

SaveRecognitionInfo

Key: SaveRecognitionInfo Type: Boolean Default: True

SaveCharacterRecognitionVariants property or the SaveWordRecognitionVariants property of the Recognizer-Params object is set to True, the value of this property is ignored.

5 Version History

5.1 Changes in Version 6

No functional changes.

5.2 Changes in Version 5

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

5.3 Changes in Version 4.12

- **New** HTTP proxy setting in the GUI license manager.
- **New** key `BlankPageMargin` for the formation of the OCR parameter. The corresponding value denotes the relative margin. The margin is excluded from the analysis if a page is blank.

5.4 Changes in Version 4.11

- **New** feature: Recognition of individual character position and suspicious characters.
- **Improved** license reloading in case the credits are exhausted.

5.5 Changes in Version 4.10

- **Improved** parameter parsing: Spaces in the ocr parameters at the beginning or at the end of key and value are now allowed.

5.6 Changes in Version 4.9

No functional changes.

5.7 Changes in Version 4.8

No functional changes.

6 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 v10 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