

User Manual



3-Heights™ OCR Enterprise Add-On

Version 4.11.25.0



Contents

1	Introduction	6
1.1	Overview	6
1.2	Supported OCR Engines	6
1.2.1	ABBYY FineReader OCR Engine 10	6
1.2.2	ABBYY FineReader OCR Engine 11	6
1.2.3	OCR Service (using ABBYY)	6
1.3	Operating Systems	6
2	Installation and Configuration	7
2.1	ABBYY	7
2.1.1	Installation of ABBYY OCR FineReader Engine	7
2.1.2	License Activation of ABBYY OCR FineReader Engine	9
2.1.3	Upgrading to a Newer Version	10
2.1.4	Remove a Test License	10
2.2	OCR Service	10
2.2.1	Installation of the OCR Service	10
2.3	Installing Products Using the Add-On	11
2.3.1	Linux	11
3	Licensing	12
3.1	ABBYY FineReader Engine	12
3.1.1	License Types	12
	Renewable Volume Runtime Licenses	12
	CPU Core Runtime Licenses	12
	Project-Based Licenses	12
4	Interface Reference	13
4.1	Architecture	13
4.1.1	Use OCR Engine Directly	13
4.1.2	Use OCR via Service	14
4.2	OCR Related Switches and Functions	14
4.2.1	Switches and Functions	15
	GetOCRPluginCount, GetOCRPluginName	15
	SetOCREngine	16
	SetOCRLanguage	16
	SetOCRParams	16
4.3	Examples	16
4.3.1	Set OCR Engine	16
	Shell	16
	Service	17
	API	17
4.3.2	Set OCR Language	17
4.3.3	Set OCR Engine Specific Parameters	18
4.4	Impacts of Using OCR	18
4.4.1	Performance	18
4.4.2	File Size	18
4.4.3	Appearance	19

5	Configuration and Performance	20
5.1	Configuration for ABBYY	20
5.2	Performance Overview	20
5.2.1	Throughput	20
5.2.2	Recognition State	20
5.2.3	File Size	21
6	Version History	22
6.1	Changes in Version 4.11	22
6.2	Changes in Version 4.10	22
6.3	Changes in Version 4.9	22
6.4	Changes in Version 4.8	22
7	Licensing, Copyright, and Contact	23
A	ABBYY FineReader 10	24
A.1	Languages	24
A.2	String parameter for ABBYY 10	25
A.2.1	RecognizeBlankPages	25
A.2.2	DisableMaskEmbedding	26
A.2.3	RemoveGarbageOnly	26
A.3	Profiles for ABBYY 10	26
A.3.1	Predefined Profiles	26
A.3.2	Custom Profiles	27
A.4	ABBYY 10 Parameters	27
A.4.1	[PrepareImageMode] INI-File Section	27
	AutoOverwriteResolution	27
	CorrectSkewBy	28
	CorrectSkewMode	28
	CreatePreview	28
	DiscardColorImage	29
	ImageCompression	29
	InvertImage	29
	MirrorImage	29
	OverwriteResolution	29
	PreviewHeight	30
	PreviewWidth	30
	Rotation	30
	XResolutionToOverwrite	30
	YResolutionToOverwrite	31
A.4.2	[PageProcessingParams] INI-File Section	31
	DetectBarcodes	31
	DetectOrientation	31
	PerformPageAnalysis	31
	RemoveGeometricalDistortions	31
A.4.3	[PageAnalysisParams] INI-File Section	32
	DetectPictures	32
	DetectSeparators	32
	DetectVectorGraphics	32
	DetectTables	32
	NoShadowsMode	32
	ProhibitDoublePageMode	32
	ProhibitModelAnalysis	33

	SingleColumnMode	33
A.4.4	[BarcodeParams] INI-File Section	33
	HasChecksum	33
	IsCode39WithoutAsterisk	33
	Orientation	33
	PDF417CodePage	34
	SupplementType	35
	Type	35
A.4.5	[RecognizerParams] INI-File Section	36
	BalancedMode	36
	CaseRecognitionMode	36
	CellsCount	37
	CJKTextDirection	37
	ErrorHiliteLevel	37
	ExactConfidenceCalculation	38
	FastMode	38
	FieldMarkingType	38
	LowResolutionMode	39
	OneLinePerBlock	39
	OneWordPerLine	39
	PossibleTextTypes	39
	ProhibitHyphenation	40
	ProhibitInterblockHyphenation	40
	ProhibitItalic	40
	ProhibitSubscript	40
	ProhibitSuperscript	41
	SaveCharacterRecognitionVariants	41
	SaveCharacterRegions	41
	SaveWordRecognitionVariants	41
	TextLanguage	41
	TextTypes	42
	TrainUserPatterns	42
	UseBuiltInPatterns	42
	UserPatternsFile	43
	WritingStyle	43
A.4.6	[ObjectsExtractionParams] INI-File Section	44
	DetectMatrixPrinter	44
	DetectPorousText	44
	FastObjectsExtraction	44
	FlexiFormsDA	44
	FullTextIndexDA	44
	ProhibitColorImage	45
	RemoveGarbage	45
	RemoveTexture	45
A.4.7	[OrientationDetectionParams] INI-File Section	45
	OrientationDetectionMode	45
	ProhibitClockwiseRotation	45
	ProhibitCounterclockwiseRotation	46
	ProhibitUpsidedownRotation	46
A.4.8	[SynthesisParamsForDocument] INI-File Section	46
	DetectDocumentStructure	46
	DetectFontFormatting	46

	InsertEmptyParagraphsForBigInterlines	47
	PagePoolSize	47
	SaveRecognitionInfo	47
B	ABBYY FineReader 11	48
B.1	Languages for ABBYY 11	48
B.1.1	Natural languages	48
B.1.2	Technical	50
B.2	String parameter for ABBYY 11	50
B.2.1	RecognizeBlankPages	50
B.2.2	DisableMaskEmbedding	51
B.2.3	PreprocessingOnly	51
B.2.4	RemoveGarbage	51
B.3	Profiles for ABBYY 11	52
B.3.1	Predefined Profiles	52
B.3.2	Custom Profiles	53
B.3.3	Profile Parameters	54
	[PrepareImageMode] INI-File Section	54
	Main settings	54
	Resolution overwriting	56
	Creating image planes	57
	[ImageProcessingParams] INI-File Section	58
	InvertImage	58
	MirrorImage	59
	RotationType	59
	[DocumentProcessingParams] INI-File Section	59
	PerformSynthesis	59
	[PageProcessingParams] INI-File Section	59
	PerformPreprocessing	59
	ProhibitColorObjectsAtProcessing	60
	PerformAnalysis	60
	PerformRecognition	60
	ATTENTION: Removed or renamed properties	60
	[PagePreprocessingParams] INI-File Section	61
	CorrectInvertedImage	61
	CorrectOrientation	61
	CorrectShadowsAndHighlights	61
	CorrectSkew	62
	CorrectSkewMode	62
	GeometryCorrectionMode	63
	ResolutionCorrectionMode	63
	[PageAnalysisParams] INI-File Section	64
	DetectText	64
	EnableTextExtractionMode	64
	DetectTables	64
	AggressiveTableDetection	64
	DetectBarcodes	64
	DetectSeparators	64
	DetectPictures	65
	DetectVectorGraphics	65
	Additional settings	65
	ATTENTION: Removed or renamed properties	66

[TableAnalysisParams] INI-File Section	67
DetectCellsInversion	67
DetectCellsOrientation	67
SingleLinePerCell	67
SplitOnlyBySeparators	67
[BarcodeParams] INI-File Section	67
Main Settings	67
Settings for certain barcode types	68
ATTENTION: Removed or renamed properties	70
[ObjectsExtractionParams] INI-File Section	70
Common settings	70
Objects removing	71
Objects detection	71
ATTENTION: Removed or renamed properties	72
[OrientationDetectionParams] INI-File Section	73
Orientation Detection Parameters	73
[RecognizerParams] INI-File Section	74
Main settings	74
Recognition speed	75
Fine tuning	76
Handprint recognition	78
User patterns	79
ATTENTION: Removed or renamed properties	79
[SynthesisParamsForPage] INI-File Section	80
Main settings	80
Color settings	80
ATTENTION: Removed or renamed properties	81
[SynthesisParamsForDocument] INI-File Section	82
Main settings	82
Additional settings	82
[FontFormattingDetectionParams] INI-File Section	83
Text decoration	83
Font attributes	84
Scaling and spacing	84
ATTENTION: Removed or renamed properties	85

1 Introduction

1.1 Overview

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

The optional 3-Heights™ OCR Enterprise Add-On 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

1.2 Supported OCR Engines

Each OCR engine is interfaced via a corresponding plug-in. Plug-ins are provided by PDF Tools. Currently the following plug-ins are available:

1.2.1 ABBYY FineReader OCR Engine 10

This plug-in is based on the ABBYY FineReader OCR Engine 10 and comes in different throughput models.

1.2.2 ABBYY FineReader OCR Engine 11

This plug-in is based on the ABBYY FineReader OCR Engine 11 and comes in different throughput models.

1.2.3 OCR Service (using ABBYY)

This plug-in implements an OCR service that uses the Abbyy FineReader OCR Engine. The OCR service should be used in any situation where files are processed in parallel (for example with the Document Converter).

It is available in version 1.9.6.1 or later.

Supported OS: Windows XP/2003 or later

1.3 Operating Systems

The 3-Heights™ OCR Enterprise Add-On is available for the following operating systems:

- Windows 7, 8, 8.1, 10 – 32 and 64 bit
- Windows Server 2008, 2008 R2, 2012, 2012 R2, 2016 – 32 and 64 bit

The OCR plugin for the 3-Heights™ OCR Service is available for all operating systems supported by the product using the service, e.g. the 3-Heights™ PDF to PDF/A Converter Service.

2 Installation and Configuration

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 Enterprise Add-On so that it uses the OCR engine directly or via OCR service.

2.1 ABBYY

2.1.1 Installation of ABBYY OCR FineReader Engine

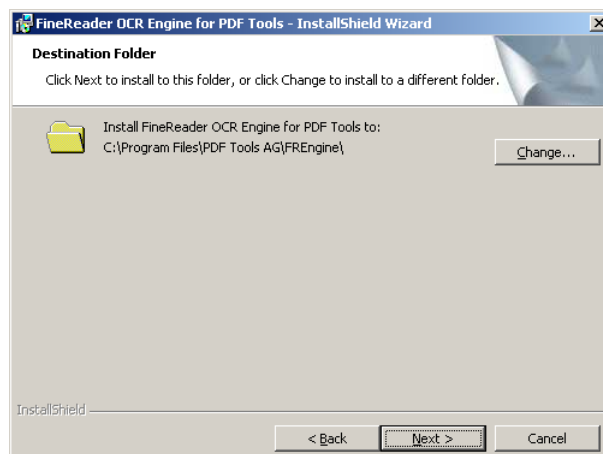
The Windows version of the 3-Heights™ OCR Enterprise Add-On is distributed in MSI-format (Microsoft Software Installation). The MSI file is named `FineReader OCR Engine for PDF Tools.msi`.

The screenshots are from ABBYY 8, the installation for ABBYY 10 and ABBYY 11 is equivalent.

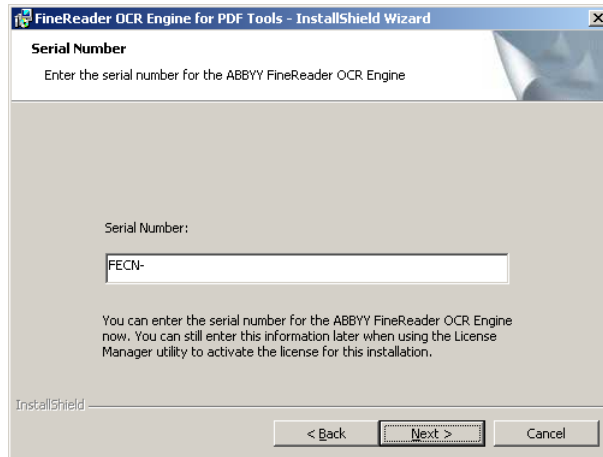
1. Download the ZIP archive `OCRE<version><platform>.zip` (e.g. `OCRE190Win32.zip`) from your download account at <http://www.pdf-tools.com>.
2. Extract the file `FineReader OCR Engine for PDF Tools.msi` from the ZIP archive.
3. Double-click the MSI file to start the installation wizard.
4. In the first dialog box of the installation wizard press the button "Next".



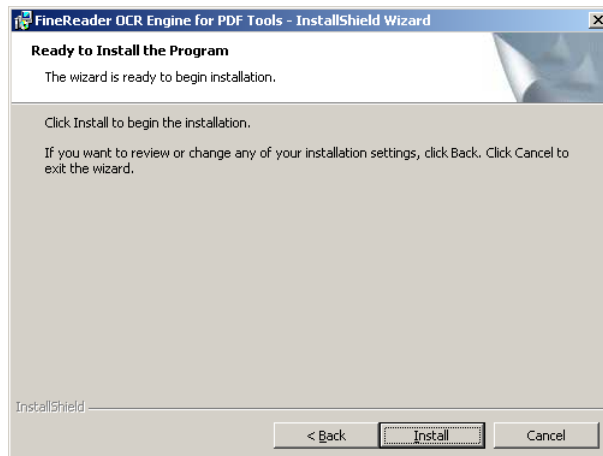
5. Adjust the destination folder if required using the button "Change..." press "Next" when done.



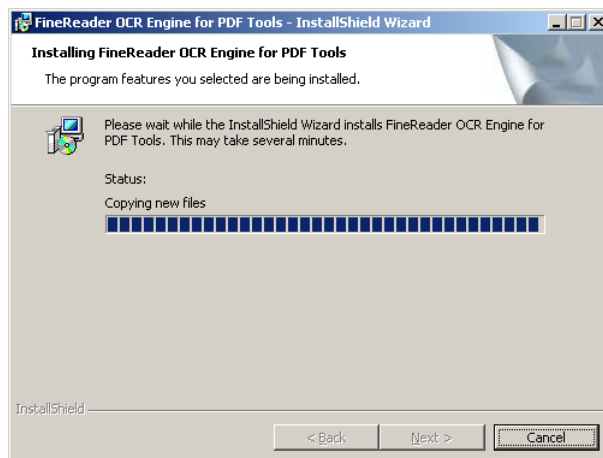
6. Enter the Serial Number. The serial number is provided by PDF Tools AG. The serial number is required for any license type (evaluation license, page limited license, unlimited license). If you do not have a serial number at this point, you can continue without entering the serial number. You can provide the serial number at a later time. Press "Next" when done.



7. When ready to install, press the "Install" button in the next dialog.



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

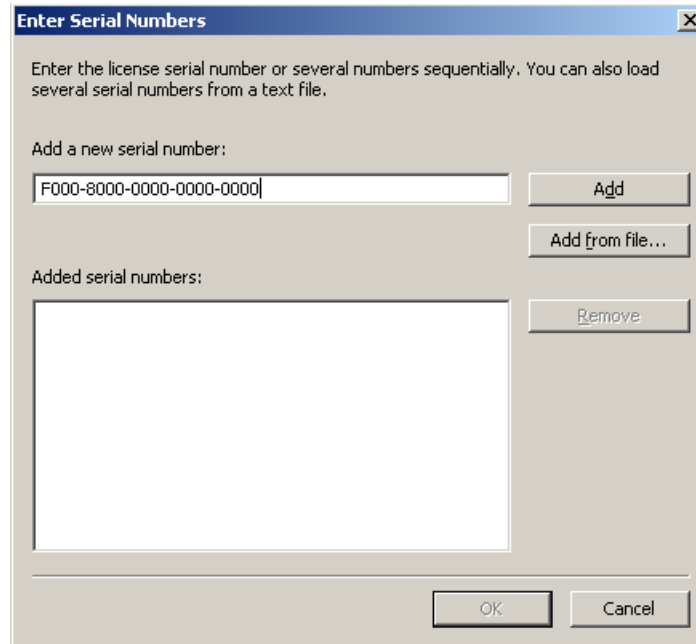


2.1.2 License Activation of ABBYY OCR FineReader Engine

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.

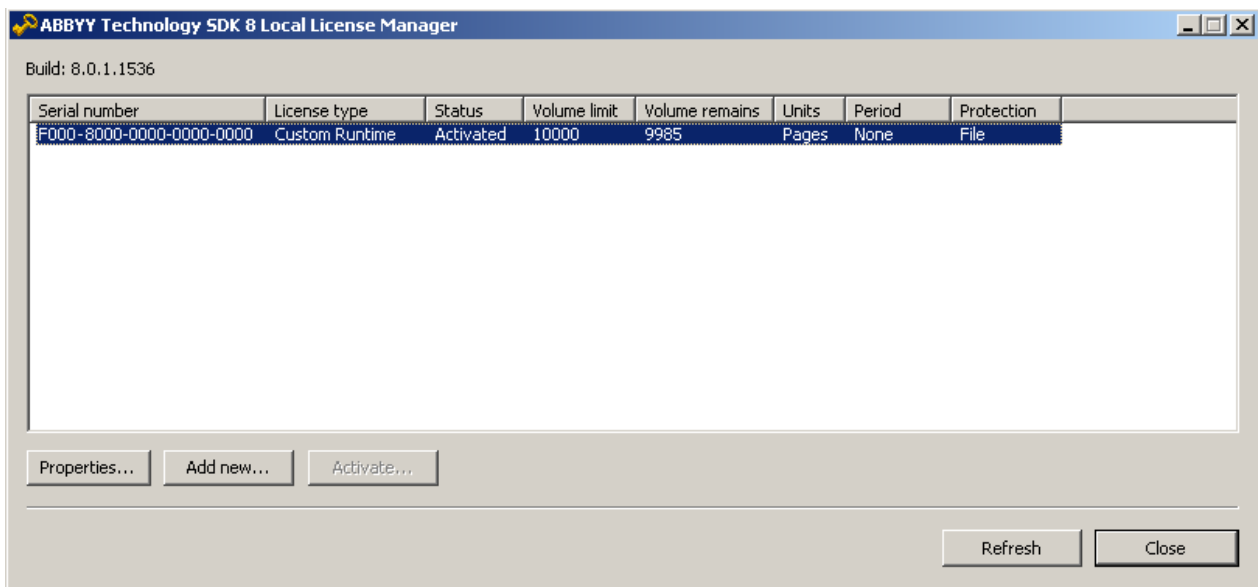


If you haven't provided the serial number at the time of installation, a dialog window asking for the serial number is shown. The serial number has a format like this:

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.



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.

2.1.3 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.

2.1.4 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 directories for version 10 and 11 respectively:

```
\AllUsersProfile\Application data\ABBYY\SDK\10\Licenses  
\AllUsersProfile\Application data\ABBYY\SDK\11\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.

2.2 OCR Service

If you are using the 3-Heights™ OCR Enterprise Add-On 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.

2.2.1 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- \langle version \rangle -Windows-(\langle platform \rangle).msi from your download account
2. Double-click the MSI file to start the installation wizard.
3. Follow the installation wizard.

2.3 Installing Products Using the Add-On

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

- 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.

2.3.1 Linux

Linux product kits do not contain any OCR plugins. However, selected plugins are available separately for selected UNIX platforms including major Linux versions. As of now, 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

3.1 ABBYY FineReader Engine

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.

3.1.1 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
<ul style="list-style-type: none">Renewable Volume Licenses<ul style="list-style-type: none">Volume per time periodCPU Core Licenses<ul style="list-style-type: none">No volume restrictionProject-Based Licenses<ul style="list-style-type: none">Volume per project, unlimited scalability	<ul style="list-style-type: none">1 Machine LicensesNetwork Licences. Allow distributing the licensed:<ul style="list-style-type: none">Number of coresPage volumes	<ul style="list-style-type: none">Standard FeaturesStandard & Add Ons Features, e.g.<ul style="list-style-type: none">OCR LanguagesICR (handprint recognition)

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.

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.

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 Interface Reference

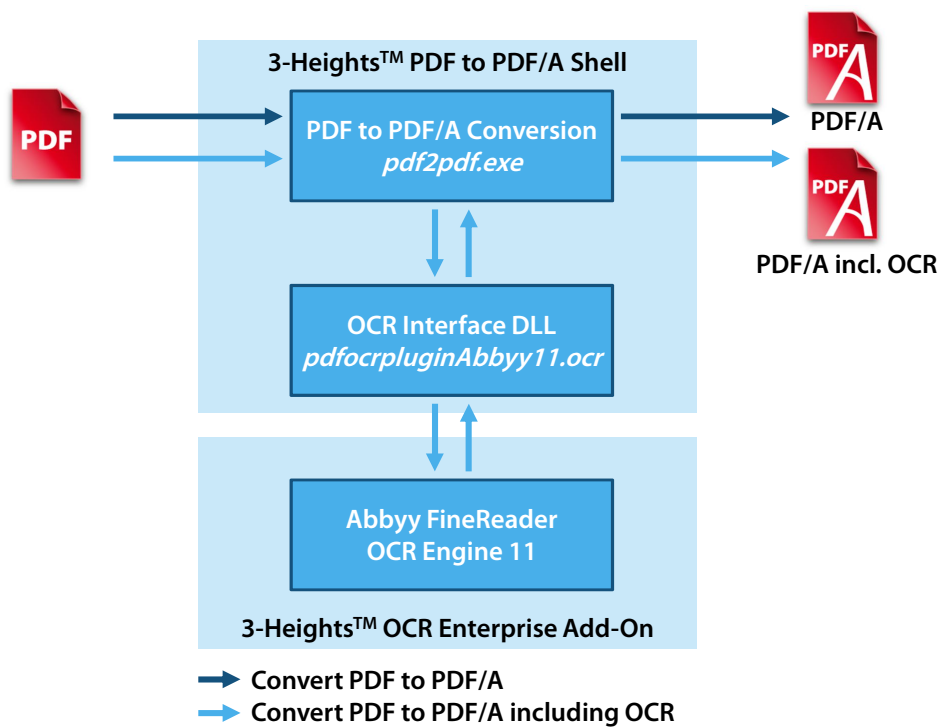
4.1 Architecture

4.1.1 Use OCR Engine Directly

How the 3-Heights™ OCR Enterprise Add-On 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 Enterprise Add-On installs the OCR engine which can be invoked from the 3-Heights™ PDF to PDF/A Shell.

The OCR interface DLL (e.g. `pdfocrpluginAbbyy11.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 Enterprise Add-On.

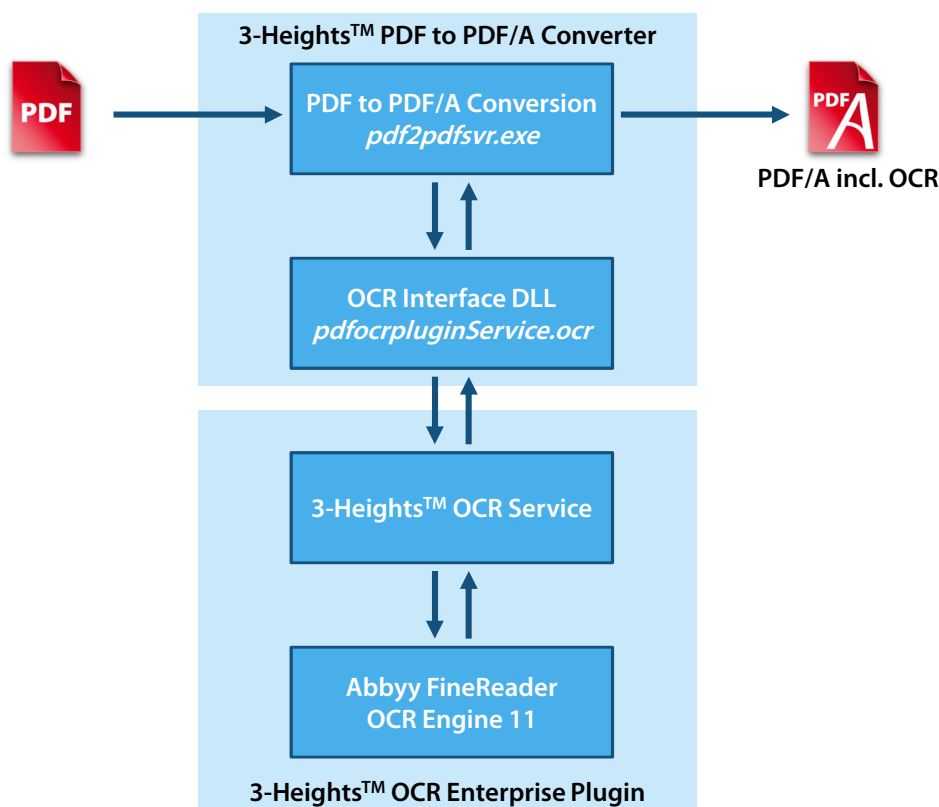


Once the 3-Heights™ OCR Enterprise Add-On is installed, it can be used by other applications of the 3-Heights™ line. How these other applications make use of the 3-Heights™ OCR Enterprise Add-On is individual for every product.

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 Enterprise Add-On 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	<code>GetOCRPluginCount()</code> <code>GetOCRPluginName(int)</code>	<p>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. pdfocrpluginAbbyy11.ocr.</p> <p>The shell version lists all available OCR Interface DLLs with the switch -le.</p> <p>In the API, the number of OCR engines is retrieved using <code>GetOCRPluginCount</code>. The string of the n-th engine is retrieved using <code>GetOCRPluginName(n)</code>.</p> <p>The name of the Abbyy FineReader OCR Engine 10 is "abbyy10".</p> <p>The name of the Abbyy FineReader OCR Engine 11 is "abbyy11".</p> <p>The name of the 3-Heights™ OCR Service is "service".</p> <p>This switch is not available for Windows services products, because they only support one engine type: "service".</p>
-ocr <name>	<code>SetOCREngine(String)</code>	Set OCR engine. The name of the engine is the string previously retrieved using the listing function.
-ocl <lang>	<code>SetOCRLanguage(String)</code>	This setting depends on the OCR engine.
-ocp <params>	<code>SetOCRParams(String)</code>	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. pdfocrpluginAbbyy11.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 Abbyy FineReader OCR Engine 11 is "abbyy11".

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 Enterprise Add-On 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 Enterprise Add-On.

```
pdf2pdf -c1 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:
- abby10
- abby11
- 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 Enterprise Add-On to be installed.

```
pdf2pdf -c1 pdfa-1b -ocr "abby11" 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 Enterprise Add-On including the OCR service to be installed. The OCR engine is defined in the configuration file of the main application (`pdf2pdfsvr.ini`).

```
-c1 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("abby11")
```

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 (Abbyy) - Set the languages to English and German.

```
pdf2pdf -ocr "abbyy11" -ocl "English, German" input.pdf output.pdf
```

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

```
SetOCREngine("abbyy11")  
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 "abbyy10" -ocp "Profile=C:\ocr.ini" input.pdf output.pdf
```

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

```
SetOCREngine("abbyy11")  
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).

5 Configuration and Performance

5.1 Configuration for ABBYY

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

5.2 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

5.2.1 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 Enterprise Add-On 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.

5.2.2 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.

5.2.3 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%.

6 Version History

6.1 Changes in Version 4.11

No functional changes.

6.2 Changes in Version 4.10

- Spaces in the ocr parameters at the beginning or at the end of key and value are allowed.
- OcrService made more robust to network errors/delays. Better error handling implemented.

6.3 Changes in Version 4.9

No functional changes.

6.4 Changes in Version 4.8

- **Updated** ABBYY FineReader Engine 11 to R8 (Version 11.1.19.48)

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 Enterprise Add-On 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
Kasernenstrasse 1
8184 Bachenbülach
Switzerland
<http://www.pdf-tools.com>
pdfsales@pdf-tools.com

A ABBYY FineReader 10

The FineReader Engine 10 is controlled using profiles.

A.1 Languages

Please note:

- Not all languages are supported in the base version. See also chapter [Licensing](#).
- 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 | ■ Czech | ■ Icelandic | ■ Malagasy |
| ■ Adyghe | ■ Danish | ■ Ido | ■ Malay |
| ■ Afrikaans | ■ Dargwa | ■ Indonesian | ■ Malinke |
| ■ Agul | ■ Dungan | ■ Ingush | ■ Maltese |
| ■ Albanian | ■ Dutch | ■ Interlingua | ■ Mansi |
| ■ Altaic | ■ DutchBelgian | ■ Irish | ■ Maori |
| ■ ArmenianEastern | ■ English | ■ Italian | ■ Mari |
| ■ ArmenianGrabar | ■ EskimoCyrillic | ■ Japanese | ■ Maya |
| ■ ArmenianWestern | ■ EskimoLatin | ■ Kabardian | ■ Miao |
| ■ Awar | ■ Esperanto | ■ Kalmyk | ■ Minankabaw |
| ■ Aymara | ■ Estonian | ■ KarachayBalkar | ■ Mohawk |
| ■ AzeriCyrillic | ■ Even | ■ Karakalpak | ■ Moldavian |
| ■ AzeriLatin | ■ Evenki | ■ Kasub | ■ Mongol |
| ■ Bashkir | ■ Faeroese | ■ Kawa | ■ Mordvin |
| ■ Basque | ■ Fijian | ■ Kazakh | ■ Nahuatl |
| ■ Belarusian | ■ Finnish | ■ Khakas | ■ Nenets |
| ■ Bemba | ■ French | ■ Khanty | ■ Nivkh |
| ■ Blackfoot | ■ Frisian | ■ Kikuyu | ■ Nogay |
| ■ Breton | ■ Friulian | ■ Kirgiz | ■ Norwegian |
| ■ Bugotu | ■ GaelicScottish | ■ Kongo | ■ NorwegianBokmal |
| ■ Bulgarian | ■ Gagauz | ■ Korean | ■ NorwegianNynorsk |
| ■ Buryat | ■ Galician | ■ Koryak | ■ Nyanja |
| ■ Catalan | ■ Ganda | ■ Kpelle | ■ Occidental |
| ■ Chamorro | ■ German | ■ Kumyk | ■ Ojibway |
| ■ Chechen | ■ GermanNewSpelling | ■ Kurdish | ■ OldEnglish |
| ■ ChineseSimplified | ■ GermanLuxembourg | ■ Lak | ■ OldFrench |
| ■ ChineseTraditional | ■ Greek | ■ Lappish | ■ OldGerman |
| ■ Chukcha | ■ Guarani | ■ Latin | ■ OldItalian |
| ■ Chuvash | ■ Hani | ■ Latvian | ■ OldSpanish |
| ■ Corsican | ■ Hausa | ■ Lezgin | ■ Ossetic |
| ■ CrimeanTatar | ■ Hawaiian | ■ Lithuanian | ■ Papiamentu |
| ■ Croatian | ■ Hebrew | ■ Luba | ■ PidginEnglish |
| ■ Crow | ■ Hungarian | ■ Macedonian | ■ 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

A.2 String parameter for ABBYY 10

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 A.3.1](#))
- Profile ([Section A.3.2](#))
- RecognizeBlankPages ([Section A.2.1](#))
- DisableMaskEmbedding ([Section A.2.2](#))
- RemoveGarbageOnly ([Section A.2.3](#))

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

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

Note: Spaces between two Key-Value pairs are not allowed.

A.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.

A.2.2 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 shall be embedded in the output file.

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

A.2.3 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.

A.3 Profiles for ABBYY 10

ABBYY 10 supports two types of profiles. A profile represents a configuration setting. There is a set of predefined profiles available. Alternatively custom profiles can be created.

A.3.1 Predefined Profiles

Supported are the following pre-defined profiles:

- 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"
```

A.3.2 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`

They are described in the chapter ABBYY 10 Parameters. Here is an example of a custom profile file:

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"
```

A.4 ABBYY 10 Parameters

A.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 `ImageDocument::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.

A.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.

A.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.

A.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_MS-DOS	Slavic_MS-DOS
Tatar	Tatar_MS-DOS	Turkish
Turkish_IBM	Turkish_ISO	Turkish_Macintosh
Ukrainian_Macintosh	US_MS-DOS	

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

A.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 40](#), 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	

A.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.

A.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.

A.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 RecognizerParams object is set to True, the value of this property is ignored.

B ABBYY FineReader 11

Between version 10 and 11, there are no fundamental changes.

Note: Some properties from ABBYY 10 have been removed or renamed. Those are listed in the documentation of the respective sections.

B.1 Languages for ABBYY 11

The following languages are supported:

B.1.1 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
- **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
- **GermanNewSpelling** – German (new spelling)
- **GermanLuxembourg** – German (Luxembourg)
- **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
- **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** – NorwegianNynorsk and 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
- **PidginEnglish** – Tok Pisin
- **Polish** – Polish
- **PortugueseBrazilian** – Portuguese (Brazil)
- **PortugueseStandard** – Portuguese (Portugal)
- **Provençal** – Provençal
- **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
- **UzbekCyrillic** – Uzbek (Cyrillic)
- **UzbekLatin** – Uzbek (Latin)
- **Vietnamese** – Vietnamese
- **Visayan** – Cebuano
- **Welsh** – Welsh
- **Wolof** – Wolof
- **Xhosa** – Xhosa
- **Yakut** – Yakut
- **Yiddish** – Yiddish
- **Zapotec** – Zapotec
- **Zulu** – Zulu

B.1.2 Technical

- **Basic** – Basic programming language
- **C++** – C/C++ programming language
- **Chemistry** – Simple chemical formulas
- **CMC7** – For MICR CMC-7 text type
- **Cobol** – Cobol programming language
- **Digits** – Numbers
- **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

B.2 String parameter for ABBYY 11

The string parameter for ABBYY 11 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 B.3.1](#))
- Profile ([Section B.3.2](#))
- RecognizeBlankPages ([Section B.2.1](#))
- DisableMaskEmbedding ([Section B.2.2](#))
- PreprocessingOnly ([Section B.2.3](#))
- RemoveGarbage ([Section B.2.4](#))

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

```
-ocr "abbyy11" -ocp "RemoveGarbage=100;PreprocessingOnly=true"
```

Note: Spaces between two Key-Value pairs are not allowed.

B.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 11 engine, recognize blank pages and store the recognition information (i.e. information about which pages are recognized as blank) in output.xml.

```
-ocr "abbyy11" -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.

B.2.2 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 11 engine, make ocr recognition, but don't embed a mask in the output file.

```
-ocr "abbyy11" -ocp "DisableMaskEmbedding=true" in.tif out.tif
```

B.2.3 PreprocessingOnly

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 11 without the necessity of performing optical character recognition. The value of the Key-Value pair takes either true or false.

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

```
-ocr "abbyy11" -ocp "Profile=C:\path\Profile.ini;PreprocessingOnly=true"
```

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

B.2.4 RemoveGarbage

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.

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

```
-ocr "abbyy11" -ocp "PredefinedProfile=BookArchiving_Speed;RemoveGarbage=50"
```

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

B.3 Profiles for ABBYY 11

ABBYY 11 supports two types of profiles. A profile represents a configuration setting. There is a set of predefined profiles available. Alternatively custom profiles can be created.

B.3.1 Predefined Profiles

Supported are the following pre-defined profiles:

DocumentConversion_Accuracy for converting documents into editable formats, optimized for accuracy

- Best quality. Enables font style detection.

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.**

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.

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**

BookArchiving_Accuracy for creating an electronic library, optimized for accuracy

- Best quality. Enables font style detection.

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.**

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.

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.**

FieldLevelRecognition for recognizing short text fragments.

- Currently this profile has default settings.

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.**

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.**

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.

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.

Version9Compatibility for compatibility

- sets the processing parameters to the default values of ABBYY FineReader Engine 9.0.

Default for default values

- Sets all the processing parameters to the default values.

Example: (Shell) Set the predefined profile DocumentArchiving_Accuracy.

```
-ocr "abbyy11" -ocp "PredefinedProfile=DocumentArchiving_Accuracy"
```

B.3.2 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](#)
- [ImageProcessingParams](#)
- [DocumentProcessingParams](#)
- [PageProcessingParams](#)
- [PagePreprocessingParams](#)
- [PageAnalysisParams](#)
- [TableAnalysisParams](#)
- [BarcodeParams](#)
- [ObjectsExtractionParams](#)
- [OrientationDetectionParams](#)
- [RecognizerParams](#)
- [SynthesisParamsForPage](#)
- [SynthesisParamsForDocument](#)
- [FontFormattingDetectionParams](#)

They are described in the chapter ABBYY 11 Parameters. Here is an example of a custom profile file:

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 "abbyy11" -ocp "Profile=C:\path\Profile1.ini"
```

B.3.3 Profile Parameters

Note: Some properties from ABBYY 10 have been removed or renamed. Those are listed in the documentation of the respective sections.

[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 [CorrectSkewMode 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 **False**.

Supported values are [CorrectSkewMode Enumeration](#) on [page 28](#).

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.

By default, this property is `PPM_Auto`, which means that FineReader Engine should define automatically whether the image is a photo.

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.

[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 54](#).

This property cannot be set to **RT_UnknownRotation**.

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

[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.

[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.

[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**
- the [TrainUserPatterns](#) property of the [RecognizerParams](#) cannot be set to **True**.

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 high-lighting 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 62](#).

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`.

The default value is `CSM_CorrectSkewByHorizontalText | CSM_CorrectSkewByVerticalText`

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_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.

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.

This property is `RCM_Auto` by default, which means that if resolution of the image is suspicious, FineReader Engine should automatically detect image resolution and change it.

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.

[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.

[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.

[BarcodeParams] INI-File Section

Main Settings

Type

Key: `Type` Type: Integer 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.

Orientation

Key: **Orientation** Type: Integer 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.

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

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.

Supported values are in [PDF417CodePage Enumeration](#)

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: Integer 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.

Supported values are in [SupplementType Enumeration](#) on [page 35](#).

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

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.

[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.

[OrientationDetectionParams] INI-File Section

Orientation Detection Parameters

OrientationDetectionMode

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

Specifies the mode of page orientation detection.

Supported values are in [OrientationDetectionMode Enumeration](#) on [page 45](#).

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**.

[RecognizerParams] INI-File Section

Main settings

TextLanguage

Key: **TextLanguage** Type: TextLanguage

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

By default this parameter is initialized with English language.

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 62](#).

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 set to **TT_Handprinted**, or **TT_Index**, the **TrainUserPatterns** property cannot be set to **True**.
- 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.

Supported values are in [CaseRecognitionMode Enumeration](#) on [page 36](#).

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

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.

Supported values are in [WritingStyle Enumeration](#) on [page 43](#).

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.

Supported values are in [FieldMarkingType Enumeration](#) on [page 39](#).

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

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.

[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 11 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 62](#).

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 62](#).

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 62](#).

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 62](#).

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.

[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 11 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.

[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`.