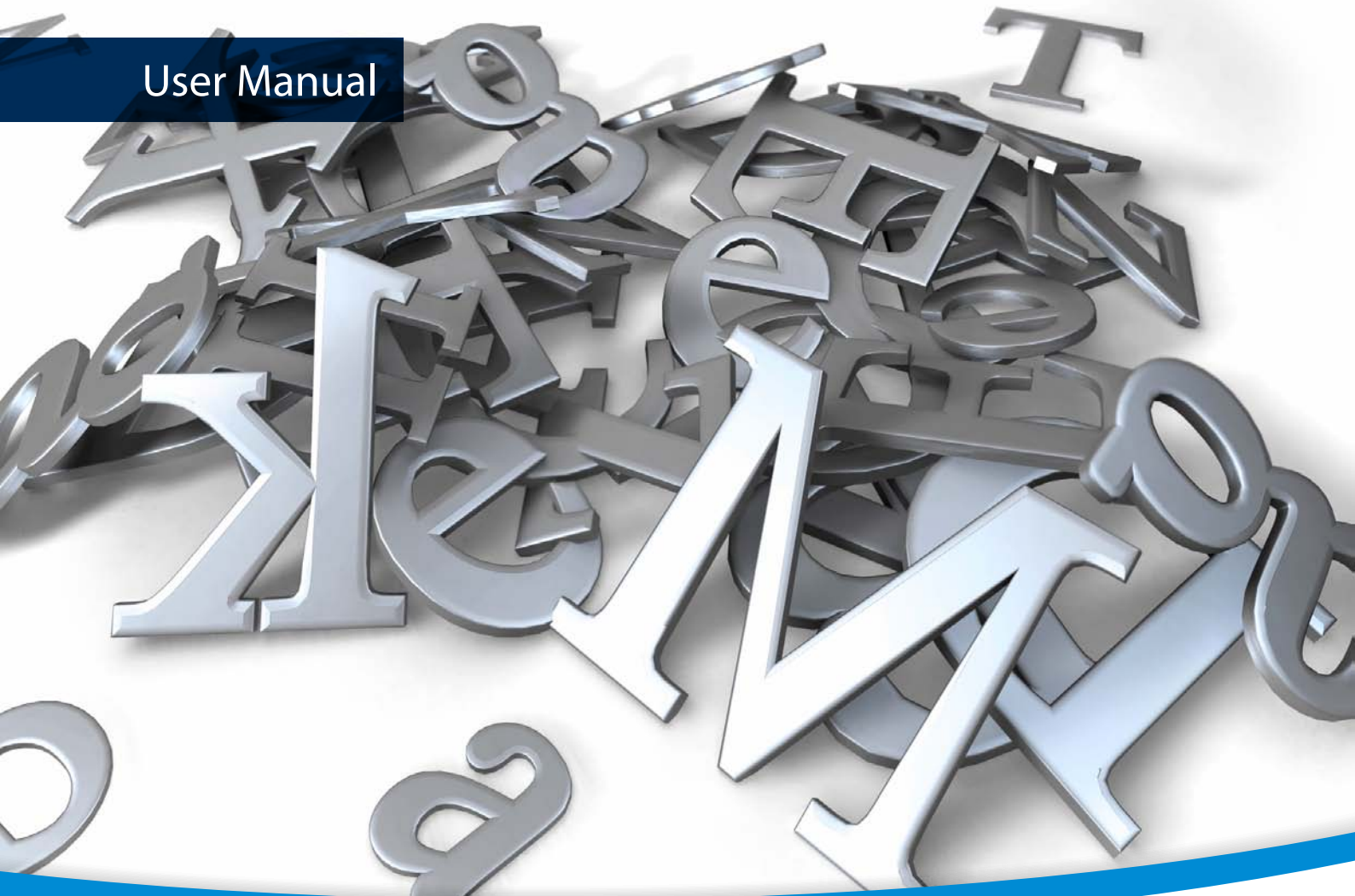


User Manual



3-Heights™ PDF OCR Shell

Version 4.12.26.2



Contents

1	Introduction	3
1.1	Description	3
1.2	Functions	3
1.2.1	Features	3
1.2.2	Formats	3
1.2.3	Compliance	4
1.3	Operating Systems	4
2	Installation	5
2.1	Windows	5
2.1.1	How to set the Environment Variable "Path"	5
2.2	Unix	6
2.2.1	All Unix Platforms	6
2.3	Uninstall	7
2.4	Fonts	7
2.4.1	Font Cache	7
2.5	Note about the Evaluation License	8
2.6	Special Directories	8
2.6.1	Directory for temporary files	8
2.6.2	Cache Directory	8
2.6.3	Font Directories	9
3	License Management	10
3.1	License Installation and Management	10
3.1.1	Graphical License Manager Tool	10
	List all installed license keys	10
	Add and delete license keys	10
	Display the properties of a license	10
3.1.2	Command Line License Manager Tool	10
	List all installed license keys	11
	Add and delete license keys	11
	Display the properties of a license	11
3.2	License Selection and Precedence	12
3.2.1	Selection	12
3.2.2	Precedence	12
3.3	Key Update	12
3.4	License activation	13
3.4.1	Activation	13
3.4.2	Reactivation	14
3.4.3	Deactivation	14
3.5	Offline Usage	14
3.5.1	First Step: Create a Request File	15
3.5.2	Second Step: Use Form on Website	15
3.5.3	Third Step: Apply the Response File	15
3.6	License Key Versions	16
3.7	License Key Storage	16
3.7.1	Windows	16
3.7.2	macOS	16
3.7.3	Unix/Linux	16

3.8	Troubleshooting	17
3.8.1	License key cannot be installed	17
3.8.2	License is not visible in license manager	17
3.8.3	License is not found at runtime	17
3.8.4	Eval watermark is displayed where it should not	17
3.8.5	Activation is not recognized	18
3.8.6	Activation is invalidated too often	18
3.8.7	Connection to the licensing service fails	19
3.8.8	Offline usage fails due to a request/response mismatch	19
4	Getting Started	20
4.1	Basics	20
4.1.1	Usage	20
4.2	Specify the Folder of the Output File	20
4.3	Process Description	20
4.4	Example	21
4.5	Use Cases	21
4.5.1	How to make text extractable	21
4.5.2	How to tag scans for accessibility (PDF/A level A)	22
4.5.3	How to detect barcodes	24
5	Interface Reference	25
5.1	OCR Engine Configuration	25
5.1.1	-1e List OCR Engines	25
5.1.2	-ocr Load OCR Engine	25
5.1.3	-oc1 Set OCR Language	26
5.1.4	-ocp Set OCR Parameters	26
5.2	Image OCR Options	26
5.2.1	-0im Image OCR Mode	27
5.2.2	-oca Rotate Scan	27
5.2.3	-occs Deskew Scan	27
5.3	Text OCR Options	27
5.3.1	-otm Text OCR Mode	27
5.4	Page OCR Options	28
5.4.1	-opm Page OCR Mode	28
5.4.2	-tm Tagging mode	28
5.5	Barcode Options	28
5.5.1	-obx Process Recognized Barcodes	28
5.6	General Options	29
5.6.1	-ocd OCR resolution	29
5.6.2	-pef Process Embedded Files	29
5.6.3	-pw Read an Encrypted PDF File	29
5.6.4	-v Verbose Mode	29
5.6.5	-1k Set License Key	30
5.7	Frequent Error Source	30
5.8	Return Codes	30
6	Version History	31
6.1	Changes in Version 4.12	31
7	Licensing, Copyright, and Contact	32

1 Introduction

1.1 Description

The 3-Heights™ PDF OCR Shell enhances PDF documents using information detected by an OCR engine.

All text in PDF documents can be made extractable, regardless of how text is included in the document. Specifically, text in images, text written with vector graphics or other graphical effects (e.g. transparency effects), or text using a font that does not provide extractable text (i.e. Unicode information) can be detected.

Tagging of OCR text for accessibility is supported. This is useful as preparation for PDF/A level A conversion or to process tagged documents.

Detected barcodes or QR codes can be extracted or embedded into the document's metadata.

The product is optimized for performance, which guarantees low latency and high document throughput. This is achieved by minimizing the number of required OCR operations. Furthermore, OCR operations are executed asynchronously, if supported by the OCR engine.

1.2 Functions

1.2.1 Features

- Make text extractable
 - Text contained in images
 - Text with fonts that have no Unicode information
 - Text written using vector graphics (e.g. in CAD drawings)
 - Any visible text, regardless of the type of graphics objects used
- Scan improvements
 - Deskew scanned images
 - Rotate pages according to the recognized rotation of scan
- Detect barcodes and QR codes
- Process embedded files
- Tagging of OCR text for accessibility
- High performance
 - Asynchronous processing
 - Page analysis and result caching to minimize OCR operations
- High quality
 - PDF/A compliant
 - High-fidelity conversion of existing page content
 - 3-Heights™ PDF Rendering Engine 2.0.
 - Automatic detection of optimal OCR resolution

1.2.2 Formats

- PDF 1.x (PDF 1.0, ..., PDF 1.7)
- PDF 2.0
- PDF/A-1, PDF/A-2, PDF/A-3

1.2.3 Compliance

Standards:

- ISO 32000-1 (PDF 1.7)
- ISO 32000-2 (PDF 2.0)
- ISO 19005-1 (PDF/A-1)
- ISO 19005-2 (PDF/A-2)
- ISO 19005-3 (PDF/A-3)

1.3 Operating Systems

The 3-Heights™ PDF OCR Shell 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
- HP-UX 11i and later PA-RISC2.0 – 32 bit
- HP-UX 11i and later ia64 (Itanium) – 64 bit
- IBM AIX 6.1 and later – 64 bit
- Linux 2.6 – 32 and 64 bit
- Oracle Solaris 2.8 and later, SPARC and Intel
- FreeBSD 4.7 and later (32 bit) or FreeBSD 9.3 and later (64 bit, on request)
- macOS 10.4 and later – 32 and 64 bit

2 Installation

2.1 Windows

The 3-Heights™ PDF OCR Shell comes as a ZIP archive or MSI installer.

The installation of the software requires the following steps.

1. You need administrator rights to install this software.
2. Log in to your download account at <http://www.pdf-tools.com>. Select the product “PDF OCR Shell”. If you have no active downloads available or cannot log in, please contact pdfsales@pdf-tools.com for assistance.

You will find different versions of the product available. We suggest to download the version, which is selected by default. If another is required, it can be selected using the combo box.

There is an MSI (*.msi) and a ZIP (*.zip) version available. The MSI (Microsoft Installer) provides an installation routine that installs and uninstalls the product for you. The ZIP version allows you to select and install everything individually.

There are 32 and 64-bit versions of the product available. While the 32-bit version runs on both, 32 and 64-bit platforms, the 64-bit version runs on 64-bit platforms only. The MSI installs the 64-bit version, whereas the ZIP file contains both the 32-bit and the 64-bit version of the product. Therefore, on 32-bit systems, the ZIP file must be used.

3. If you select an MSI version, start it and follow the steps in the installation routine.
4. If you are using the ZIP version, follow the steps below. Unzip the archive to a local folder, e.g. C:\Program Files\PDF Tools AG\.

This creates the following subdirectories:

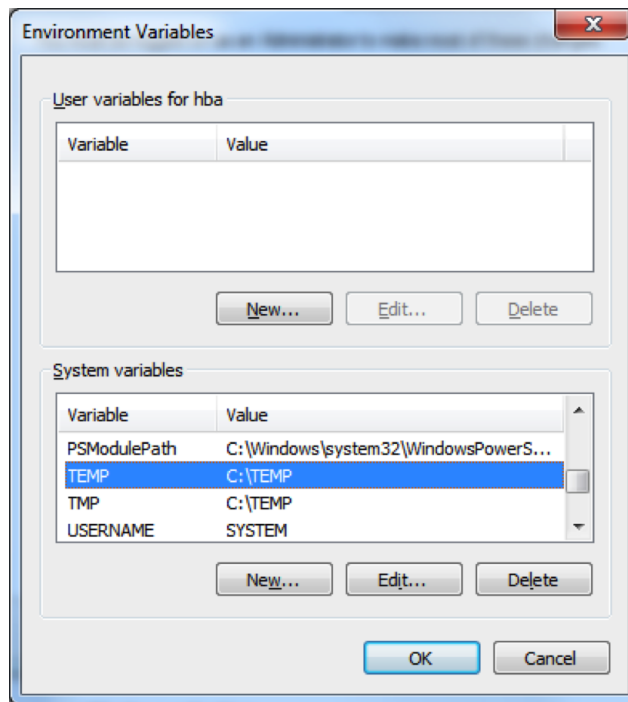
Subdirectory	Description
bin	Contains the runtime executable binaries.
doc	Contains documentation.

5. (Optional) To easily use the 3-Heights™ PDF OCR Shell from a shell, the directory needs to be included in the “Path” environment variable.
6. (Optional) Register your license key using the [License Management](#).
7. Make sure your platform meets the requirements regarding fonts described in chapter [Fonts](#).
8. Download and install the 3-Heights™ OCR Service, the OCR Service client plugin, and the OCR Engine as described in the respective manuals:
 - 3-Heights™ OCR Add-On for ABBYY FineReader Engine v10: [OcrAbbyy10.pdf](#)
 - 3-Heights™ OCR Add-On for ABBYY FineReader Engine v11: [OcrAbbyy11.pdf](#)
 - 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12: [OcrAbbyy12.pdf](#)
 - 3-Heights™ OCR Add-On for Barcode and QR Code Recognition: [OcrBarcodes.pdf](#)
 - 3-Heights™ OCR Service: [OcrService.pdf](#) from the separate product kit.

2.1.1 How to set the Environment Variable “Path”

To set the environment variable “Path” on Windows, go to Start → Control Panel (classic view) → System → Advanced → Environment Variables.

Select “Path” and “Edit”, then add the directory where `pdfocr.exe` is located to the “Path” variable. If the environment variable “Path” does not exist, create it.



2.2 Unix

This section describes installation steps required on all Unix platforms, which includes Linux, macOS, Oracle Solaris, IBM AIX, HP-UX, FreeBSD and others.

Here is an overview of the files that come with the 3-Heights™ PDF OCR Shell:

File Description

Name	Description
bin/<platform>/pdfocr	This is the main executable. The directory <platform> is either x86 containing the 32-bit version of the product, or x64 for 64-bit.
bin/<platform>/libPdfOcrAPI.so	This is a shared library required by pdfocr.
bin/<platform>/*.ocr	These are OCR plugin modules.
doc/*.*	Documentation

2.2.1 All Unix Platforms

1. Unpack the archive in an installation directory, e.g. /opt/pdf-tools.com/
2. Verify that the GNU shared libraries required by the product are available on your system:

- On Linux:

```
ldd pdfocr
```

- On AIX:

```
dump -H pdfocr
```

In case the above reports any missing libraries you have three options:

- a. Download an archive that is linked to another version of the GNU shared libraries and verify whether they are available on your system. Use any version whose requirements are met. Note that this option is not available for all platforms.
 - b. Use your system's package manager to install the missing libraries. On Linux it usually suffices to install the package `libstdc++6`.
 - c. Use GNU shared libraries provided by PDF Tools AG:
 1. Go to <http://www.pdf-tools.com> and navigate to "Support" → "Utilities".
 2. Download the GNU shared libraries for your platform.
 3. Install the libraries manually according your system's documentation. On Linux this typically involves copying them to your library directory, e.g. `/usr/lib` or `/usr/lib64`, and running `ldconfig`.
 4. Verify that the GNU shared libraries required by the product are available on your system now.
3. Create a link to the executable from one of the standard executable directories, e.g:

```
ln -s /opt/pdf-tools.com/bin/<platform>/pdfocr /usr/bin
```

4. Create a link to the shared library from one of the standard library directories, e.g:

```
ln -s /opt/pdf-tools.com/bin/<platform>/libPdfOcrAPI.so /usr/lib
```

5. Optionally register your license key using the [Command Line License Manager Tool](#).
6. Make sure your platform meets the requirements regarding fonts described in chapter [Fonts](#).
7. Download and install the 3-Heights™ OCR Service, the OCR Service client plugin, and the OCR Engine as described in the respective manuals:
 - 3-Heights™ OCR Add-On for ABBYY FineReader Engine v10: [OcrAbbyy10.pdf](#)
 - 3-Heights™ OCR Add-On for ABBYY FineReader Engine v11: [OcrAbbyy11.pdf](#)
 - 3-Heights™ OCR Add-On for ABBYY FineReader Engine v12: [OcrAbbyy12.pdf](#)
 - 3-Heights™ OCR Add-On for Barcode and QR Code Recognition: [OcrBarcodes.pdf](#)
 - 3-Heights™ OCR Service: [OcrService.pdf](#) from the separate product kit.

2.3 Uninstall

If you have used the MSI for the installation, go to Start → 3-Heights™ PDF OCR Shell... → Uninstall...

If you have used the ZIP file for the installation: In order to uninstall the product, undo all the steps done during installation.

2.4 Fonts

Fonts are required, if OCR is performed and OCR text is added to a PDF document. Hereby it is crucial, that the fonts available in the [Font Directories](#) contain all characters required for the OCR text. For example, when recognizing Japanese OCR text, it is recommended to add the fonts "MS Mincho" or "MS Gothic" to the [Font Directories](#).

2.4.1 Font Cache

A cache of all fonts in all [Font Directories](#) is created. If fonts are added or removed from the font directories, the cache is updated automatically.

In order to achieve optimal performance, make sure that the cache directory is writable for the 3-Heights™ PDF OCR Shell. Otherwise the font cache cannot be updated and the font directories have to be scanned on each program startup.

The font cache is created in the subdirectory <CacheDirectory>/Installed Fonts of the [Cache Directory](#).

2.5 Note about the Evaluation License

With the evaluation license the 3-Heights™ PDF OCR Shell automatically adds a watermark to the output files.

2.6 Special Directories

2.6.1 Directory for temporary files

This directory for temporary files is used for data specific to one instance of a program. The data is not shared between different invocations and deleted after termination of the program.

The directory is determined as follows. The product checks for the existence of environment variables in the following order and uses the first path found:

Windows

1. The path specified by the %TMP% environment variable.
2. The path specified by the %TEMP% environment variable.
3. The path specified by the %USERPROFILE% environment variable.
4. The Windows directory.

Unix

1. The path specified by the \$PDFTMPDIR environment variable.
2. The path specified by the \$TMP environment variable.
3. The /tmp directory.

2.6.2 Cache Directory

The cache directory is used for data that is persisted and shared between different invocations of a program. The actual caches are created in subdirectories. The content of this directory can safely be deleted to clean all caches.

This directory should be writable by the application, otherwise caches cannot be created or updated and performance will degrade significantly.

Windows

- If the user has a profile:
%LOCAL_APPDATA%\PDF Tools AG\Caches
- If the user has no profile:
<TempDirectory>\PDF Tools AG\Caches

Linux, macOS and other Unixes

- If the user has a home directory:
~/ .pdf-tools/Caches

- If the user has no home directory:
 <TempDirectory>/pdf-tools/Caches

where <TempDirectory> refers to the [Directory for temporary files](#).

2.6.3 Font Directories

The location of the font directories depends on the operating system. Font directories are traversed recursively in the order as specified below.

If two fonts with the same name are found, the latter one takes precedence, i.e. user fonts will always take precedence over system fonts.

Windows

1. %SystemRoot%\Fonts
2. directory Fonts, which must be a direct sub-directory of where pdfocr.exe resides.

macOS

1. /System/Library/Fonts
2. /Library/Fonts

Linux and other Unixes

1. /usr/share/fonts
2. /usr/local/share/fonts
3. ~/.fonts
4. \$PDFFONTDIR or /usr/lib/X11/fonts/Type1

3 License Management

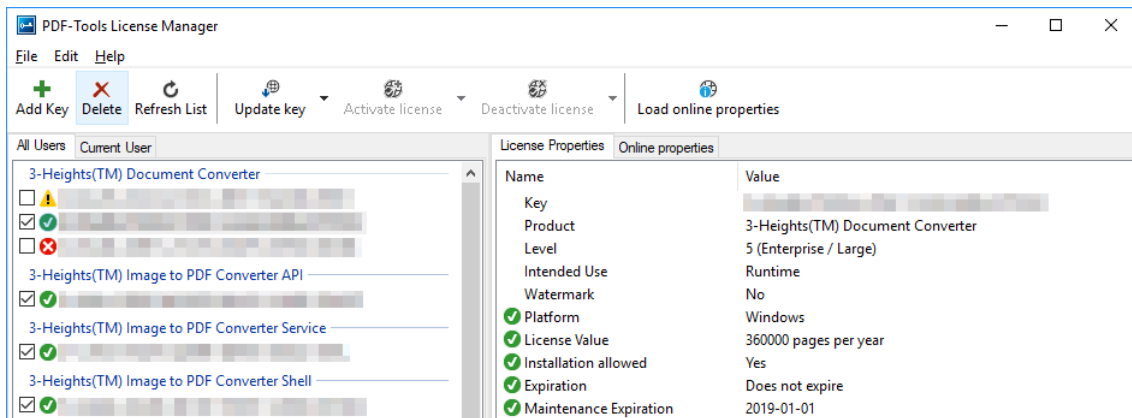
3.1 License Installation and Management

There are three possibilities to pass the license key to the application:

1. The license key is installed using the GUI tool (graphical user interface). This is the easiest way if the licenses are managed manually. It is only available on Windows.
2. The license key is installed using the shell tool. This is the preferred solution for all non-Windows systems and for automated license management.
3. The license key is passed to the application at run-time via the switch `-lk`. This is the preferred solution for OEM scenarios.

3.1.1 Graphical License Manager Tool

The GUI tool `LicenseManager.exe` is located in the `bin` directory of the product kit (Windows only).



List all installed license keys

The license manager always shows a list of all installed license keys in the left pane of the window. This includes licenses of other PDF Tools products. The user can choose between:

- Licenses available for all users. Administrator rights are needed for modifications.
- Licenses available for the current user only.

Add and delete license keys

License keys can be added or deleted with the “Add Key” and “Delete” buttons in the toolbar.

- The “Add key” button installs the license key into the currently selected list.
- The “Delete” button deletes the currently selected license keys.

Display the properties of a license

If a license is selected in the license list, its properties are displayed in the right pane of the window.

3.1.2 Command Line License Manager Tool

The command line license manager tool `licmgr` is available in the `bin\x86` and `bin\x64` directory.

Note: The command line tool licmgr is not included in Windows platform kits, as the GUI tool is the recommended tool for managing Licenses. A Windows licmgr shelltool is available on request.

A complete description of all commands and options can be obtained by running the program without parameters:

```
licmgr
```

List all installed license keys

```
licmgr list
```

The currently active license for a specific product is marked with a * on the left side.

Example:

```
>licmgr list
Local machine:
  Product Name:
    1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
    1-YYYYY-YYYYY-YYYYY-YYYYY-YYYYY-YYYYY-YYYYY
    * 1-ZZZZZ-ZZZZZ-ZZZZZ-ZZZZZ-ZZZZZ-ZZZZZ-ZZZZZ
Current user:
```

Add and delete license keys

Install new license key:

```
licmgr store 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

Delete old license key:

```
licmgr delete 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

Both commands have the optional argument -s that defines the scope of the action:

g For all users

u Current user

Display the properties of a license

```
licmgr info 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

Properties that invalidate the license are marked with an X, properties that require attention are marked with an !. In that case an additional line with a comment is displayed.

Example:

```
>licmgr info 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
- Key:          1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
- Product:     Product Name
- Features:    Feature1,Feature2
- Intended use: Development
- Watermark:   No
- Platform:   Windows
- Installation: Yes
! Activation:  2018-05-07
               (The license has not yet been activated.)
- Expiration:  Does not expire
- Maintenance: 2019-04-27
```

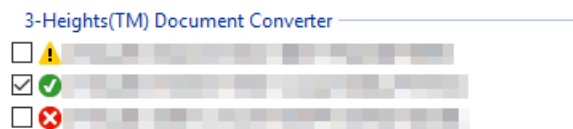
3.2 License Selection and Precedence

3.2.1 Selection

If multiple keys for the same product are installed in the same scope, only one of them can be active at the same time.

Installed keys that are not selected are not considered by the software!

In the Graphical User Interface use the check box on the left side of the license key to mark a license as selected.



With the Command Line Interface use the `select` subcommand:

```
licmgr select 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

3.2.2 Precedence

License keys are considered in the following order:

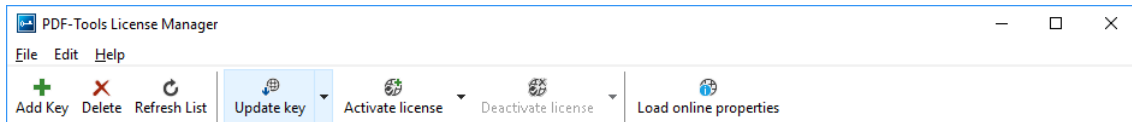
1. License key passed at runtime.
2. License selected for the current user
3. License selected for the current user ([legacy key format](#))
4. License selected for all users
5. License selected for all users ([legacy key format](#))

The first matching license is used, regardless whether it is valid or not.

3.3 Key Update

If a license property like the maintenance expiration date changes, the key can be update directly in the license manager.

In the Graphical User Interface select the license and press the button "Update Key" in the toolbar:



With the Command Line Interface use the update subcommand:

```
licmgr update 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

3.4 License activation

New licenses keys have to be activated (except for OEM licenses).

Note: Licenses that need activation have to be installed in the license manager and must not be passed to the component at runtime.

The license activation is tied to a specific computer. If the license is installed at user scope, the activation is also tied to that specific user. The same license key can be activated multiple times, if the license quantity is larger than 1.

Every license key includes a date, after which the license has to be activated, which is typically 10 days after the issuing date of the key. Prior to this date, the key can be used without activation and without any restrictions.

3.4.1 Activation

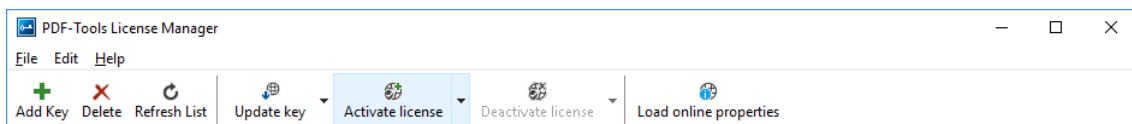
The License can be activated directly within the license manager. Every activation increases the activation count of the license by 1.

It is recommended to add a comment to the activation request which helps keeping track of all activations for a specific license key. In case of problems it also helps us providing support.

The comment is stored in the activation database as long as the license key remains activated. Upon deactivation it is deleted from the database immediately.

All activations and the corresponding comments can be examined using the **Load online properties** function of the license manager. The information is accessible to anyone with access to the license key.

In the Graphical User Interface select the license and press the button "Activate license" in the toolbar:



It is recommended to add a comment to the activation request by using the subsequent dialog box.

With the Command Line Interface use the activate subcommand:

```
licmgr activate 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

Note that the key has to be installed first.

It is recommended to add a comment to the activation request by using the `-c` or `-cd` option:

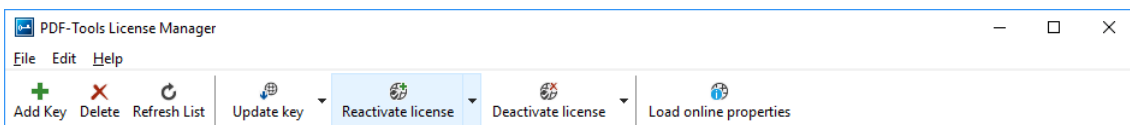
```
licmgr activate -cd 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
licmgr activate -c "custom comment" 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

3.4.2 Reactivation

The activation is tied to specific properties of the computer like the MAC address or host name. If one of these properties changes, the activation becomes invalid and the license has to be reactivated. A reactivation does **not** increase the activation count on the license.

The process for reactivation is the same as for the activation.

In the Graphical User Interface the button "Activate license" changes to "Reactivate license":



With the Command Line Interface the subcommand `activate` is used again:

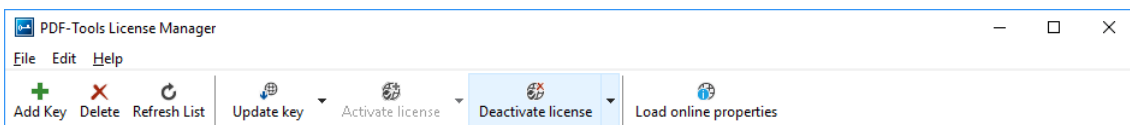
```
licmgr activate 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

3.4.3 Deactivation

To move a license to a different computer, it has to be deactivated first. Deactivation decreases the activation count of the license by 1.

The process for deactivation is similar to the activation process.

In the Graphical User Interface select the license and press the button "Deactivate license" in the toolbar:



With the Command Line Interface use the `deactivate` subcommand:

```
licmgr deactivate 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

3.5 Offline Usage

The following actions in the license manager need access to the internet:

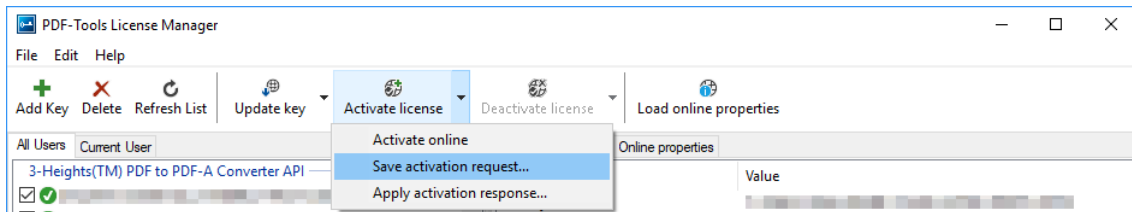
- [License Activation](#)
- [License Reactivation](#)

- [License Deactivation](#)
- [Key Update](#)

On systems without internet access, a three step process can be used instead, using a form on the PDF Tools website.

3.5.1 First Step: Create a Request File

In the Graphical User Interface select the license and use the dropdown menu on the right side of the button in the toolbar:



With the Command Line Interface use the `-fs` option to specify the destination path of the request file:

```
licmgr activate -fs activation_request.bin 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

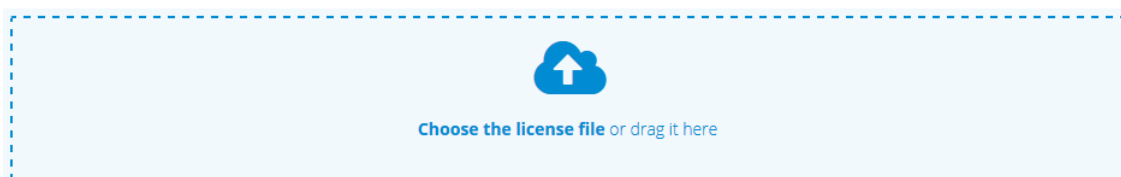
License Deactivation: When saving the deactivation request file, the license is **deactivated immediately** and cannot be used any further. It can however only be activated again after completing the deactivation on the website.

3.5.2 Second Step: Use Form on Website

Open the following website in a web browser: <http://www.pdf-tools.com/pdf20/en/mypdftools/licenses-kits/license-activation/> Upload the request by dragging it onto the marked area:

License activation (offline)

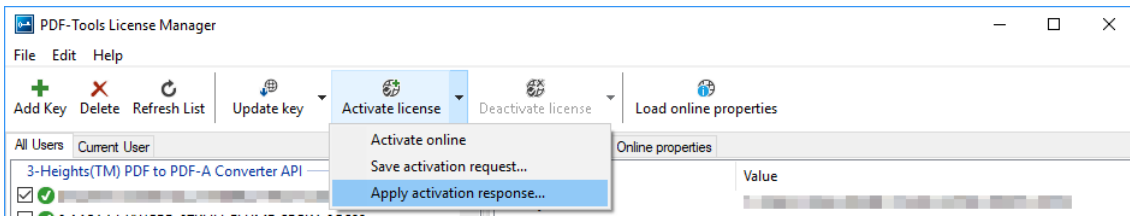
Upload your license request. For more information and instructions please check the manual of your product.



Upon success, the response will be downloaded automatically if necessary.

3.5.3 Third Step: Apply the Response File

In the Graphical User Interface select the license and use the dropdown menu on right side of the button in the toolbar:



With the Command Line Interface use the `-fl` option to specify the source path of the response file:

```
licmgr activate -fl activation_response.bin 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

3.6 License Key Versions

As of 2018 all new keys will have the format 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX. Legacy keys with the old format 0-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX are still accepted for a limited time period.

For compatibility reasons, old and new version keys can be installed side by side and one key of each version can be selected at the same time. In that case, the software always uses the new version.

3.7 License Key Storage

Depending on the platform the license management system uses different stores for the license keys.

3.7.1 Windows

The license keys are stored in the registry:

- "HKLM\Software\PDF Tools AG" (for all users)
- "HKCU\Software\PDF Tools AG" (for the current user)

3.7.2 macOS

The license keys are stored in the file system:

- /Library/Application Support/PDF Tools AG (for all users)
- ~/Library/Application Support/PDF Tools AG (for the current user)

3.7.3 Unix/Linux

The license keys are stored in the file system:

- /etc/opt/pdf-tools (for all users)
- ~/.pdf-tools (for the current user)

Note: The user, group and permissions of those directories are set solely by the license manager tool. It may be necessary to change permissions to make the licenses readable for all users. Example:

```
chmod -R go+rx /etc/opt/pdf-tools
```

3.8 Troubleshooting

3.8.1 License key cannot be installed

The license key cannot be installed in the license manager application. The error message is: "Invalid license format."

Possible causes:

- The license manager application is an older version that only supports the [legacy key format](#).

Solution

Use a current version of the license manager application or use a license key in the legacy key format if available.

3.8.2 License is not visible in license manager

The license key was successfully installed previously but is not visible in the license manager anymore. The software is still working correctly.

Possible causes:

- The license manager application is an older version that only supports the [legacy key format](#).

Solution

Use a current version of the license manager application.

3.8.3 License is not found at runtime

The license is not found at runtime by the software. The error message is: "No license key was set."

Possible causes:

- The license key is actually missing (not installed).
- The license key is installed but not selected in the license manager.
- The application is an older version that only supports the [legacy key format](#), while the license key has the new license format.

Solution

Install and select a valid license key that is compatible with the installed version of the software or use a newer version of the software. The new license key format is supported starting with version 4.10.26.1

For compatibility reasons, one license key of each format can be selected at the same time.

3.8.4 Eval watermark is displayed where it should not

The software prints an evaluation watermark onto the output document, even if the installed license is a productive one.

Possible causes:

- There is an evaluation license key selected for the **current user**, that takes precedence over the key for **all users**.

Note: The software might be run under a different user than the license manager application.

- An evaluation license key that is passed at runtime takes precedence over those selected in the license manager.
- There is an evaluation license key selected with a [newer license format](#) that takes precedence over the key in the older format.
- The software was not restarted after changing the license key from an evaluation key to a productive one.

Solution

Disable or remove all evaluation license in all scopes, check that no evaluation key is passed at runtime and restart the software.

3.8.5 Activation is not recognized

The license is installed and activated in the license manager, but the software does not recognize it as activated. The error message is: "The license has not been activated."

Possible causes:

- There is an unregistered license key selected for the **current user**, that takes precedence over the key for **all users**. This leads to an error even if the same license is registered for all users.

Note: The software might be run under a different user than the license manager application.

- A license key that is passed at runtime takes precedence over those selected in the license manager. This leads to an error even if the same license is registered in the license manager.

Note: Licenses that need activation have to be installed in the license manager and must not be passed to the component at runtime.

- The software was not restarted after activating the license.

Solution

Disable, remove or activate all unregistered licenses in all scopes, check that no key is passed at runtime and restart the software.

3.8.6 Activation is invalidated too often

The license activation is invalidated regularly, for no obvious reason.

Possible causes:

- The MAC address used for computing the machine fingerprint is not static. This may happen e.g. for virtual network adapters with dynamic MAC address (VPN, Juniper, ...).

Solution

Update to a newer version (≥ 4.12) of the PDF Tools product, deactivate the license key using the new license manager and activate it again. After that, an improved fingerprinting algorithm is used.

Deactivation and activation have to be **executed separately**, a reactivation of the license in one step does not change the fingerprinting algorithm and thus does not solve the problem.

Note: After this procedure, older products might not recognize the activation as valid anymore. Reactivating the license using an old license manager will revert the activation to the old fingerprinting algorithm.

As an alternative, remove any virtual network adapter with a dynamic MAC address.

3.8.7 Connection to the licensing service fails

The license activation/deactivation/update fails because the license manager cannot reach the licensing server.

The error message depends on the platform and the exact error condition.

Possible causes:

- The computer is not connected to the internet.
- The connection is blocked by a corporate firewall.

Solution

Make sure that the computer is connected to the internet and that the host `www.pdf-tools.com` is reachable on port 443 (HTTPS).

If this is not possible, try [Offline Usage](#) instead.

3.8.8 Offline usage fails due to a request/response mismatch

The offline license activation/deactivation/update fails because the response file does not match the request file.

The error message is: "Mismatch between request and response."

Possible causes:

- The response file is applied to a different machine than the request file was created.
- The response file is applied to a different user than the request file was created.
- The response file was applied to a specific user while the request was created for all users, or vice versa.
- The response file is applied to the wrong license key.
- Another request file has been created between creating the request file and applying the response file.
- The license key was updated between creating the request file and applying the response file.
- The license key was removed and re-added between creating the request file and applying the response file.

Solution

Delete any old request and response files to make sure they are not used by accident.

Retry the entire process as outlined in [chapter 3.5](#) and refrain from making any other license-related actions between creating the request file and applying the response file.

Make sure that the response file is applied to exactly the same license key in exactly the same location (machine, all users or specific user) where the request file was created.

4 Getting Started

4.1 Basics

4.1.1 Usage

The usage of the 3-Heights™ PDF OCR Shell is:

```
pdfocr [options] input.pdf output.pdf
```

A simple command to process a document requires four parameters: The name of the PDF input file, the PDF output file, an OCR engine name, and a processing mode.

Example: Read the input document `input.pdf`, create a new document `output.pdf`, and use the OCR Service to process images.

```
pdfocr -ocr service -oim update input.pdf output.pdf
```

In order to list all available features type `pdfocr` without any parameters.

4.2 Specify the Folder of the Output File

The output folder can simply be added in front of the output file name

```
pdfocr input.pdf myfolder\output.pdf
```

or absolute (Windows):

```
pdfocr input.pdf C:\myfolder\output.pdf
```

4.3 Process Description

The following is a simplified process description of the 3-Heights™ PDF OCR Shell:

1. **Open document:** The input document is opened.
2. **Process document:** The document is processed and the result written to the output.
 1. **Process pages**
 1. Analyze page: The page's content is analyzed in order to determine, whether processing by the OCR engine is required or not (see chapter [Page Analysis](#) below).
 2. OCR page (optional)
 1. Determine optimal OCR resolution
 2. Render page: The page is converted to an image using the 3-Heights™ PDF Rendering Engine 2.0.
 3. Send image to OCR engine.
 4. Process OCR results (see chapter [OCR Result Processing](#) below).
 3. Copy page: The page is copied to the output. If available, information from the OCR engine is added.
 2. **Process embedded files:** Embedded files can be processed recursively or copied as-is.

Page Analysis

The page's content is analyzed. The modes of ocr parameters specified for images ([-oim](#)), text ([-otm](#)), and pages ([-opm](#)) are evaluated. The page is processed by the OCR engine if required by any of the modes.

OCR Result Processing

Each OCR result object is processed as follows:

1. If it is a barcode, it is processed according to the [barcode mode](#).
2. If its location is on an image on the page, it is processed according to the [image ocr mode](#).
3. If it corresponds to text on the page, it is processed according to the [text ocr mode](#).
4. Otherwise it is added as OCR text to the page, if the [page ocr mode](#) is not none.

4.4 Example

Example: Example that shows how to add OCR text to images.

```
pdfocr -ocr service -ocp "PredefinedProfile=DocumentConversion_Accuracy" -ocl "..." -oim
update input.pdf output.pdf
```

4.5 Use Cases

This chapter describes some common use cases.

4.5.1 How to make text extractable

This example shows how text in a PDF document can be made extractable. This is suitable both for born-digital and scanned documents.

Note: For tagged input documents, the configuration described in [How to tag scans for accessibility \(PDF/A level A\)](#) should be used.

OCR Engine Configuration

Abbyy FineReader 11

The following profile configuration `abbyy11_text.ini` is optimized to extract as much text as possible:

```
[PagePreprocessingParams]
CorrectOrientation=TRUE
[PageAnalysisParams]
DetectVerticalEuropeanText=TRUE
[ObjectsExtractionParams]
DetectTextOnPictures = TRUE
```

Use the profile using the OCR engine running on the OCR Service:

```
pdfocr -ocr service -ocp "Profile=C:\path\to\abby11_text.ini" -ocl "..." ...
```

Note: It is important that C:\path\to\abby11_text.ini must be the path to the configuration file on the OCR Service and the OCR Service process must have read permissions.

Processing Configuration

1. **Detect text contained in images:** For documents that contain images, processing of images can be activated by setting an image OCR mode (see [-oim](#)):

```
pdfocr ... -oim update ...
```

2. **Make text extractable:** For documents that contain non-extractable text, processing of text can be activated by setting a text OCR mode (see [-otm](#)):

```
pdfocr ... -otm update ...
```

3. **Make other visible text extractable:** For documents that contain other forms of visible text, pages can be OCR processed by setting a page OCR mode (see [-opm](#)):

```
pdfocr ... -opm ifNoText ...
```

Process Document

```
pdfocr -ocr service -ocp "Profile=C:\path\to\abby11_text.ini" -ocl "..." -oim update  
input.pdf output.pdf
```

4.5.2 How to tag scans for accessibility (PDF/A level A)

“Tagging” adds structural information to a PDF. This information can be used e.g. to read the document to the visually impaired.

The 3-Heights™ PDF OCR Shell supports tagging scans, e.g. such that they can be converted to PDF/A level A.

Tagging of scans that contain no figures or pictures, will conform to the PDF/UA specification (ISO 14289-1). For figures and pictures an alternative representation or replacement text must be included. Because this information is not provided by the OCR engine, tagging of such files cannot conform to PDF/UA.

Prerequisites

1. The OCR engine must provide structural information for OCR results. We recommend Abby FineReader 11 or newer.
2. If the 3-Heights™ OCR Service is used, it must be newer than 4.11.21.0.

OCR Engine Configuration

Abby FineReader 11

The following profile configuration `abbyy11_tagging.ini` is suitable:

```
[PagePreprocessingParams]
CorrectOrientation=TRUE
```

This is essentially the predefined profile "DocumentConversion_Accuracy", but can also handle rotated pages.

Use the profile using the OCR engine running on the OCR Service:

```
pdfocr -ocr service -ocp "Profile=C:\path\to\abbyy11_tagging.ini" -ocl "." ...
```

Note: It is important that `C:\path\to\abbyy11_tagging.ini` must be the path to the configuration file on the OCR Service and the OCR Service process must have read permissions.

Processing Configuration

1. Activate processing of images by setting an image OCR mode:

```
pdfocr ... -oim update ...
```

2. (Optional) Enable scan enhancements:

```
pdfocr ... -oca -occs ...
```

3. Activate creation of tagging information by setting the tagging mode:

```
pdfocr ... -tm update ...
```

Process Document

```
pdfocr -v -ocr service -ocp "Profile=C:\path\to\abbyy11_tagging.ini" -ocl "." -oim
update -oca -occs -tm update input.pdf output.pdf
```

Error Handling

Tagging errors are classified as warnings by the 3-Heights™ PDF OCR Shell. Because tagging is crucial for this process, tagging warnings returned must be checked and treated as errors. In case of OCR warnings, the return code of 3-Heights™ PDF OCR Shell will be 7. Check for tagging errors returned and printed out by the option `-v`.

4.5.3 How to detect barcodes

This example shows how to detect and extract barcodes and QR codes.

OCR Engine Configuration

There are two OCR engines available that support barcode recognition.

Barcodes OCR Engine

The OCR engine “barcodes” is a specialized plugin for barcode and QR code recognition.

No engine parameters are required. However, in order to speed up the recognition process, it can be limited to a specific set of code types:

```
pdfocr -ocr barcodes -ocp "BarcodeTypes=QRCode" ...
```

Abby FineReader 11 Engine

The predefined profile “BarcodeRecognition_Accuracy” is optimized for this purpose and will detect all supported types of barcodes and QR codes.

Use the profile using the OCR engine running on the OCR Service:

```
pdfocr -ocr service -ocp "PredefinedProfile=BarcodeRecognition_Accuracy" ...
```

Note: This profile detects barcodes only and cannot be used to make any text extractable. However, with a profile file barcode and text recognition can be performed in one step

Processing Configuration

1. **Activate OCR processing:** Activate OCR processing of all pages:

```
pdfocr ... -opm all ...
```

2. **Enable barcode extraction:** Write all detected barcodes in XML format to the file barcodes.xml:

```
pdfocr ... -obx barcodes.xml ...
```

Process Document

```
pdfocr -ocr barcodes -ocp "BarcodeTypes=QRCode" -opm all -obx barcodes.xml input.pdf  
output.pdf
```

5 Interface Reference

Switches are options that are provided with the command to define how the document should be processed.

Switches can occur in two forms: As stand-alone option, such as `-v` (verbose mode) or they may require a parameter, such as `-pw password` (set password to read encrypted input document).

The last two parameters of the command line should always be the input and the output document.

Switches are parsed from left to right. If the same switch is applied multiple times the last set value is applied.

5.1 OCR Engine Configuration

5.1.1 `-le` List OCR Engines

List OCR Engines `-le`

OCR engines are accessed through the corresponding OCR interface DLLs. At present interfacing the following engines are supported:

Abby FineReader 11 OCR Engine This engine is accessed by the OCR interface DLL `pdfocrpluginAbby11.ocr`.

Abby FineReader 10 OCR Engine This engine is accessed by the OCR interface DLL `pdfocrpluginAbby10.ocr`.

3-Heights™ OCR Service This service is accessed by the OCR interface DLL `pdfocrpluginService.ocr`. The service accesses the Abby FineReader 10 or 11 OCR Engine.

The OCR interface DLLs are provided by the 3-Heights™ PDF OCR Shell. The OCR engine is provided as a separate product, such as 3-Heights™ OCR Enterprise Add-On.

Here is an example of listing available OCR engines:

```
pdfocr -le
List of available OCR engines:
- abby11
- abby10
- service
End of list.
```

In order to make use of the OCR engine, the OCR interface DLL and the OCR engine must be installed. The switch `-le` lists all available OCR interface DLLs. It does not verify the corresponding OCR engine is installed and can be initialized. The OCR engine is actually accessed when using the switch `-ocr`.

5.1.2 `-ocr` Load OCR Engine

Load OCR Engine `-ocr <name>`

If a PDF document has to be made fully text searchable even if the text is part of a raster image then the images which are contained in the PDF document must be run through an OCR engine. With this switch the user can select

an OCR engine, e.g. Abbyy11, and instruct the tool to embed the recognized text as a hidden layer on top of the image. If the add-in is not found or the engine cannot be initialized (because it is not installed or the license key is not valid) then an error message is issued.

The name of the OCR engine can be retrieved using the switch `-le`. If the switch `-ocr` is not used, no OCR is applied.

Example: The following switch sets the OCR engine to the OCR Service

```
pdfocr -ocr service input.pdf output.pdf
```

See also documentation for the 3-Heights™ OCR Add-On.

5.1.3 `-ocl` Set OCR Language

```
Set OCR Language -ocl <languages>
```

In order to optimize the performance of the OCR engine, it can be given hints what languages are used. The default language of the Abbyy FineReader 11 OCR Engine is English. This switch can only be used if the switch `-ocr` is set. This setting depends on the OCR engine.

The following switch set the languages to English and German:

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

See also documentation for the 3-Heights™ OCR Add-On.

5.1.4 `-ocp` Set OCR Parameters

```
Set OCR Parameters -ocp <params>
```

Using this switch, OCR engine specific parameters (key/value pairs) can be set to optimize the performance.

The following switch sets a predefined profile (i.e. a configuration setting) which is optimized for creating electronic archives with high accuracy:

```
pdfocr -ocr abbyy11 -ocp "PredefinedProfile = DocumentArchiving_Accuracy"  
input.pdf output.pdf
```

See also documentation for the 3-Heights™ OCR Add-On.

5.2 Image OCR Options

The image OCR parameters control under what conditions and how images should be processed.

Note: The options `-oca` and `-occs` have an effect only, if:

1. The page is a scan and not born-digital.
2. The page is processed by the OCR engine, which depends on the `-oim` set.
3. The required information is provided by the OCR engine, which depends on the type and settings of the engine.

5.2.1 `-oim` Image OCR Mode

Image OCR Mode `-oim <mode>`

The mode according to which images are processed. See chapter [Process Description](#) for a description on how setting this option affects OCR processing.

Available values for `<mode>` are:

none (**default**) Do not process images.

update Only process images that have no OCR text.

replace Process all images and remove existing OCR text.

remove Remove existing OCR text.

ifNoText Process images only if document contains no text.

5.2.2 `-oca` Rotate Scan

Rotate Scan `-oca`

Rotate scan according to the orientation detected by the OCR engine.

5.2.3 `-occs` Deskew Scan

Deskew Scan `-occs`

Deskew scan according to the angle detected by the OCR engine.

5.3 Text OCR Options

5.3.1 `-otm` Text OCR Mode

Text OCR Mode `-otm <mode>`

The mode according to which text is processed. See chapter [Process Description](#) for a description on how setting this option affects OCR processing.

Available values for `<mode>` are:

none (**default**) Do not process text.

update Only OCR process text that is not extractable.

For all characters that have no meaningful Unicode, OCR processing is used to determine the Unicode. This is the recommended mode to make text extractable.

Note that making text extractable requires many OCR operations. The reason is that of all characters multiple instances must be recognized, to deal with erroneous OCR recognitions.

replace OCR process all text.

OCR is used to determine the Unicode of all characters, that is even if they seemingly have Unicode information. This is useful for documents that possibly contain wrong Unicode information. Wrong Unicode information is typically created by flawed PDF creators or to obfuscate text (i.e. to prevent copy-and-paste or search operations).

For documents that contain correct Unicode information, this mode produces the same result as the mode Update. The rare exceptions are special fonts for which the OCR engine produces wrong results, which might happen for some decorative or handwritten fonts. The main disadvantage of the mode Replace over Update is, that more OCR operations are required.

5.4 Page OCR Options

5.4.1 -opm Page OCR Mode

Page OCR Mode -opm <mode>

The mode according to which pages are processed. See chapter [Process Description](#) for a description on how setting this option affects OCR processing.

Available values for <mode> are:

none **(default)** Do not process pages.

all Process all pages that are not empty.

ifNoText Process all pages that contain content but no text.

addResults Do not trigger OCR processing of pages. But if pages are OCR processed, e.g. due to another OCR mode, add results as OCR text to pages.

5.4.2 -tm Tagging mode

Tagging mode -tm <mode>

The mode according to which tagging information is processed.

Available values for <mode> are:

none **(default)** Do not add tagging information.

update Update existing tagging information.

5.5 Barcode Options

5.5.1 -obx Process Recognized Barcodes

Process Recognized Barcodes -obx <stream>

The mode according to which barcodes are processed.

<stream> may be either the string `xmp`, in which case recognized barcodes are embedded into the document's XMP metadata, or a file name, in which case recognized barcodes are written to the file in XML format.

5.6 General Options

5.6.1 -ocd OCR resolution

```
OCR resolution -ocd <d> <dmin> <dmax>
```

Each page's optimal OCR resolution is determined automatically, such that all images and text can be recognized. The default resolution <d> is chosen, if it is within the range of optimal resolutions. The range of allowed resolutions can be chosen using <dmin> and <dmax>.

Default: `-ocd 300 200 400`

5.6.2 -pef Process Embedded Files

```
Process Embedded Files -pef
```

Process embedded files recursively. Otherwise embedded files are copied as-is.

5.6.3 -pw Read an Encrypted PDF File

```
Read an Encrypted PDF File -pw <password>
```

A PDF document that has a user password (the password to open the document) can only be processed when either the user or the owner password is provided. The password can be provided using the option `-pw` followed by the password.

Example: The input PDF document is encrypted with a user password. Either the user or the owner password of the input PDF is "mypassword". The command to process such an encrypted file is:

```
pdfocr -pw mypassword input.pdf output.pdf
```

When a PDF is encrypted with a user password and the password is not provided or is incorrect, the 3-Heights™ PDF OCR Shell cannot read and process the file. Instead it will generate the following error message:

```
Password wasn't correct.
```

5.6.4 -v Verbose Mode

```
Verbose Mode -v
```

This option turns on the verbose mode.

5.6.5 -lk Set License Key

```
Set License Key -lk <key>
```

Pass a license key to the application at runtime instead of using one that is installed on the system.

```
pdfocr -lk X-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX . . .
```

This is required in an OEM scenario only.

5.7 Frequent Error Source

It may happen that you type a command, or copy it from somewhere and it doesn't work even though it seems to be correct. A common reason is that the dash (-) which is used for most parameters is accidentally mistaken by an em dash (—).

5.8 Return Codes

All return codes other than 0 indicate an error in the processing.

In case of an error, an error message is written to the standard error stream (stderr). It is highly recommended to check the command's return value and in case of an error, log the error output.

Return Codes

Value	Description
0	Success.
1	Couldn't open input file.
2	PDF output file could not be created.
3	Error with given options, e.g. too many parameters.
4	PDF input file is encrypted and password is missing or incorrect.
5	OCR infrastructure error. The file could not be processed because of an error in the OCR infrastructure, e.g. page credits of OCR engine used.
6	OCR processing error. The file could not be processed because of an error related to the input file or the processing options set.
7	OCR warnings occurred. I.e. the file has been processed successfully, but warnings occurred. Use -v to get a list of them. This return code may be ignored and treated the same as 0.
10	License error, e.g. invalid license key.

6 Version History

6.1 Changes in Version 4.12

- **Improved** image ocr mode to cache OCR text of images. Images that occur on multiple pages must be OCR processed once only.
- **New** specialized OCR plugin "barcodes" to recognize barcodes and QR codes without an additional OCR engine.
- **New** OCR plugin "abbyy12" for the ABBYY FineReader 12 engine.
- **Improved** memory consumption of product.
- **New** detection of OCR text that is under images.
- **Improved** ocr result processing, notably the accuracy of associating OCR text to existing text on page.
- **New** parameter addResults for page ocr mode -opm.

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™ PDF OCR Shell 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