



3-Heights™ PDF to EMF Converter API

Version 4.12.26.6

Contents

1	Introduction	4
1.1	Description	4
1.2	Functions	4
1.2.1	Features	4
1.2.2	Formats	4
1.3	Interfaces	4
1.4	Operating Systems	4
2	Installation and Deployment	5
2.1	Windows	5
2.2	Interfaces	6
2.2.1	Development	6
2.2.2	Deployment	7
2.3	Interface Specific Installation Steps	8
2.3.1	COM Interface	8
2.3.2	C Interface	9
2.4	Uninstall, Install a New Version	9
2.5	Color Profiles	9
2.5.1	Default Color Profiles	9
2.5.2	Get Other Color Profiles	9
2.6	Fonts	10
2.6.1	Font Cache	10
2.6.2	Font Configuration File fonts.ini	10
2.7	Note about the Evaluation License	11
2.8	Special Directories	11
2.8.1	Directory for temporary files	11
2.8.2	Cache Directory	11
2.8.3	Font Directories	12
3	License Management	13
3.1	License Installation and Management	13
3.1.1	Graphical License Manager Tool	13
	List all installed license keys	13
	Add and delete license keys	13
	Display the properties of a license	13
3.1.2	Command Line License Manager Tool	14
	List all installed license keys	14
	Add and delete license keys	14
	Display the properties of a license	14
3.2	License Selection and Precedence	15
3.2.1	Selection	15
3.2.2	Precedence	15
3.3	Key Update	16
3.4	License activation	16
3.4.1	Activation	16
3.4.2	Reactivation	17
3.4.3	Deactivation	17
3.5	Proxy Setting	17
3.6	Offline Usage	18

3.6.1	First Step: Create a Request File	18
3.6.2	Second Step: Use Form on Website	19
3.6.3	Third Step: Apply the Response File	19
3.7	License Key Versions	19
3.8	License Key Storage	19
3.8.1	Windows	19
3.9	Troubleshooting	20
3.9.1	License key cannot be installed	20
3.9.2	License is not visible in license manager	20
3.9.3	License is not found at runtime	20
3.9.4	Eval watermark is displayed where it should not	20
3.9.5	Activation is not recognized	21
3.9.6	Activation is invalidated too often	21
3.9.7	Connection to the licensing service fails	22
3.9.8	Offline usage fails due to a request/response mismatch	22
4	User's Guide	23
4.1	Font and Text Issues	23
4.1.1	Handle Non-Embedded Fonts	23
	Font Replacement Strategy	23
4.1.2	Handle Embedded Fonts	24
4.2	Call Sequence	24
4.3	Error Handling	24
5	Interface Reference	26
5.1	Pdf2Emf Interface	26
5.1.1	Close	26
5.1.2	CreateSurface	26
5.1.3	CreateSurface2	26
5.1.4	DestroySurface	26
5.1.5	ErrorCode	26
5.1.6	FontCount	27
5.1.7	Height	27
5.1.8	Open	27
5.1.9	Options	28
5.1.10	PageCount	28
5.1.11	PermissionFlags	28
5.1.12	ProductVersion	28
5.1.13	RenderPage	28
5.1.14	SaveTo	29
5.1.15	SaveFontTo	29
5.1.16	SetLicenseKey	29
5.1.17	Width	29
5.2	Enumerations	29
5.2.1	TPDFRenderOption Enumeration	29
5.2.2	TPDFErrorCode Enumeration	31

6	Version History	33
6.1	Patches in Version 4.12	33
6.2	Changes in Version 4.12	33
6.3	Changes in Version 4.11	33
6.4	Changes in Version 4.10	33
6.5	Changes in Version 4.9	33
6.6	Changes in Version 4.8	33
7	Licensing, Copyright, and Contact	34

1 Introduction

1.1 Description

The 3-Heights™ PDF to EMF Converter API converts PDF files into single page EMF files.

1.2 Functions

1.2.1 Features

- Create EMF documents
- Convert individual pages
- List and extract embedded fonts
- Replace embedded fonts by pre-installed fonts
- Replace fonts by vector outlines
- Read encrypted PDF documents
- Set surface dimensions
- Support for all languages, such as Chinese, Japanese, Korean

1.2.2 Formats

Input Formats

- PDF 1.x (PDF 1.0, ..., PDF 1.7)
- PDF 2.0

Output Formats

- EMF (Enhanced Metafile)

Compliance

Standards:

- ISO 32000-1 (PDF 1.7)
- ISO 32000-2 (PDF 2.0)

1.3 Interfaces

The following interfaces are available:

- C
- COM

1.4 Operating Systems

The 3-Heights™ PDF to EMF Converter API 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

2 Installation and Deployment

2.1 Windows

The 3-Heights™ PDF to EMF Converter API comes as a ZIP archive.

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 to EMF Converter API". 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.

The product comes as a ZIP archive containing all files.

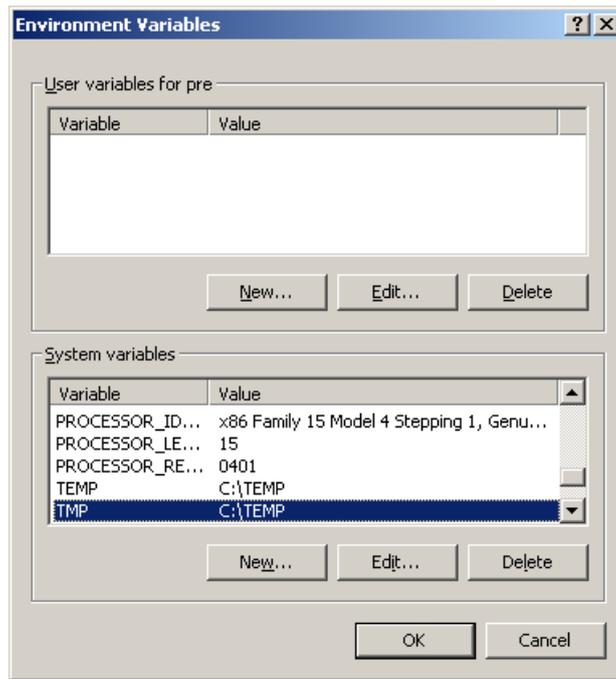
There are 32 and 64-bit versions of the product available. While the 32-bit version runs on both, 32 and 64-bit platforms, the 64-bit version runs on 64-bit platforms only. The ZIP file contains both the 32-bit and the 64-bit version of the product.

3. 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.
include	Contains header files to include in your C/C++ project.
lib	Contains the object file library to include in your C/C++ project.
samples	Contains sample programs in various programming languages

4. (Optional) Register your license key using the [License Management](#).
5. Identify which interface you are using. Perform the specific installation steps for that interface described in chapter [Interface Specific Installation Steps](#)
6. Ensure the system environment variable TMP exists and points to an existing directory. This directory is required to temporarily install fonts that are embedded in PDF documents.
Control Panel → System → Advanced → Environment Variables



7. Ensure the cache directory exists as described in chapter [Special Directories](#).
8. Make sure your platform meets the requirements regarding color spaces and fonts described in chapters [Color Profiles](#) and [Fonts](#) respectively.

2.2 Interfaces

The 3-Heights™ PDF to EMF Converter API provides two different interfaces. The installation and deployment of the software depend on the interface you are using. The table below shows the supported interfaces and examples with which programming languages they can be used.

Interface	Programming Languages
COM	<p>The component object model (COM) interface can be used with any COM-capable programming language, such as:</p> <ul style="list-style-type: none"> ■ MS Visual Basic ■ MS Office Products such as Access or Excel (VBA) ■ C++ ■ VBScript ■ others <p>This interface is available in the Windows version only.</p>
C	<p>The native C interface is for use with C and C++.</p>

2.2.1 Development

The software developer kit (SDK) contains all files that are used for developing the software. The role of each file with respect to the four different interfaces is shown in table [Files for Development](#). The files are split in four categories:

Req. This file is required for this interface.

Opt. This file is optional. See also table [File Description](#) to identify which files are required for your application.

Doc. This file is for documentation only.

Empty field An empty field indicates this file is not used at all for this particular interface.

Files for Development

Name	COM	C
bin\ <platform>\pdf2emfapi.dll< td=""><td>Req.</td><td>Req.</td></platform>\pdf2emfapi.dll<>	Req.	Req.
bin\ <platform>\inet.dll< td=""><td>Opt.</td><td>Opt.</td></platform>\inet.dll<>	Opt.	Opt.
doc*.pdf	Doc.	Doc.
doc\Pdf2EmfAPI.idl	Doc.	
include\pdf2emfapi_c.h		Req.
include*.*		Opt.
lib\ <platform>\pdf2emfapi.lib< td=""><td></td><td>Req.</td></platform>\pdf2emfapi.lib<>		Req.
samples*.*	Doc.	Doc.

The purpose of the most important distributed files of is described in table [File Description](#).

File Description

Name	Description
bin\ <platform>\pdf2emfapi.dll< td=""><td>This is the DLL that contains the main functionality (required).</td></platform>\pdf2emfapi.dll<>	This is the DLL that contains the main functionality (required).
bin\ <platform>\inet.dll< td=""><td>This DLL implements https: and ftp: connections using the Internet Explorer. It is loaded from module path.</td></platform>\inet.dll<>	This DLL implements https: and ftp: connections using the Internet Explorer. It is loaded from module path.
doc*.*	Various documentations.
include*.*	Contains files to include in your C / C++ project.
lib\ <platform>\pdf2emfapi.lib< td=""><td>The object file library needs to be linked to the C/C++ project.</td></platform>\pdf2emfapi.lib<>	The object file library needs to be linked to the C/C++ project.
samples*.*	Contains sample programs in different programming languages.

2.2.2 Deployment

For the deployment of the software only a subset of the files are required. Which files are required (Req.), optional (Opt.) or not used (empty field) for the two different interfaces is shown in the table below.

¹ These files must reside in the same directory as Pdf2EmfAPI.dll.

Files for Deployment

Name	COM	C
bin\ <platform>\pdf2emfapi.dll< td=""><td>Req.</td><td>Req.</td></platform>\pdf2emfapi.dll<>	Req.	Req.
bin\ <platform>\inet.dll< td=""><td>Opt.</td><td>Opt.</td></platform>\inet.dll<>	Opt.	Opt.

The deployment of an application works as described below:

1. Identify the required files from your developed application (this may also include color profiles).
2. Identify all files that are required by your developed application.
3. Include all these files into an installation routine such as an MSI file or simple batch script.
4. Perform any interface-specific actions (e.g. registering when using the COM interface).

Example: This is a very simple example of how a COM application written in Visual Basic 6 could be deployed.

1. The developed and compiled application consists of the file `application.exe`. Color profiles are not used.
2. The application uses the COM interface and is distributed on Windows only.
 - The main DLL `Pdf2EmfAPI.dll` must be distributed.
3. All files are copied to the target location using a batch script. This script contains the following commands:

```
copy application.exe %targetlocation%\.  
copy Pdf2EmfAPI.dll %targetlocation%\.
```

4. For COM, the main DLL needs to be registered in silent mode (`/s`) on the target system. This step requires Power-User privileges and is added to the batch script.

```
regsvr32 /s %targetlocation%\Pdf2EmfAPI.dll.
```

2.3 Interface Specific Installation Steps

2.3.1 COM Interface

Registration Before you can use the 3-Heights™ PDF to EMF Converter API component in your COM application program you have to register the component using the `regsvr32.exe` program that is provided with the Windows operating system. The following command shows the registration of `Pdf2EmfAPI.dll`. Note that in Windows Vista and later, the command needs to be executed from an administrator shell.

```
regsvr32 "C:\Program Files\PDF Tools AG\bin\
```

Where `<platform>` is `Win32` for the 32-bit and `x64` for the 64-bit version.

If you are using a 64-bit operating system and would like to register the 32-bit version of the 3-Heights™ PDF to EMF Converter API, you need to use the `regsvr32` from the directory `%SystemRoot%\SysWOW64` instead of `%SystemRoot%\System32`.²

² Otherwise you get the following message: `LoadLibrary("Pdf2EmfAPI.dll") failed - The specified module could not be found.`

If the registration process succeeds, a corresponding dialog window is displayed. The registration can also be done silently (e.g. for deployment) using the switch `/s`.

Other Files The other DLLs do not need to be registered, but for simplicity it is suggested that they reside in the same directory as the `Pdf2EmfAPI.dll`.

2.3.2 C Interface

- The header file `pdf2emfapi_c.h` needs to be included in the C/C++ program.
- The library `Pdf2EmfAPI.lib` needs to be linked to the project.
- The dynamic link library `Pdf2EmfAPI.dll` needs to be in a path of executables (e.g. on the environment variable `%PATH%`).

2.4 Uninstall, Install a New Version

If you have used the ZIP file for the installation: In order to uninstall the product, undo all the steps done during installation, e.g. un-register using `regsvr32.exe /u`, delete all files, etc.

Installing a new version does not require to previously uninstall the old version. The files of the old version can directly be overwritten with the new version.

2.5 Color Profiles

A PDF document may contain graphical objects using various different color spaces and the output file of 3-Heights™ PDF to EMF Converter API may yet use another color space. Therefore often colors have to be converted between different color spaces.

For calibrated color spaces (such color spaces with an associated ICC color profile) the color conversion is well defined. For the conversion of uncalibrated device color spaces (DeviceGray, DeviceRGB, DeviceCMYK) however, the 3-Heights™ PDF to EMF Converter API requires appropriate color profiles. Therefore it is important, that the profiles are available and that they describe the colors of the device your input documents are intended for.

If no color profiles are available, default profiles for both RGB and CMYK are generated on the fly by the 3-Heights™ PDF to EMF Converter API.

2.5.1 Default Color Profiles

If no particular color profiles are set default profiles are used. For device RGB colors a color profile named "`sRGB Color Space Profile.icm`" and for device CMYK a profile named "`USWebCoatedSWOP.icc`" are searched for in the following directories:

Windows

1. `%SystemRoot%\System32\spool\drivers\color`
2. directory `Icc`, which must be a direct sub-directory of where the `Pdf2EmfAPI.dll` resides.

2.5.2 Get Other Color Profiles

Most systems have pre-installed color profiles available, for example on Windows at `%SystemRoot%\system32\spool\drivers\color\`. Color profiles can also be downloaded from the links provided in the directory `bin\Icc\` or from the following websites:

- <http://www.pdf-tools.com/public/downloads/resources/colorprofiles.zip>
- <http://www.color.org/srgbprofiles.html>
- https://www.adobe.com/support/downloads/iccprofiles/iccprofiles_win.html

2.6 Fonts

PDF documents may contain both embedded and non-embedded fonts. When rendering non-embedded fonts the best result can be achieved, if the font is available on the system. Therefore it is important to make sure the [Font Directories](#) contain all fonts required.

For more information on how to cope with font issues, please refer to section [Font and Text Issues](#).

2.6.1 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 to EMF Converter API. 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](#).

2.6.2 Font Configuration File fonts.ini

The font configuration file is optional. It can be used to control the mapping of fonts used in the PDF to fonts pre-installed on the system.

The file `fonts.ini` must reside at the following location :

Windows: In a directory named `Fonts`, which must be a direct sub-directory of where `Pdf2EmfAPI.dll` resides.

It consists of two sections: `[fonts]` and `[replace]`. Both sections are used to map fonts in the PDF to fonts in the installed font collection on the operating system. This comes into play when the font in the PDF document does not have an embedded font program, or the embedded font is not usable.

The mapping only works if the font types of the specified fonts are matching; e.g. if the font in the PDF is a symbolic font, such as "Symbol" or "ZapfDingbats", the mapped font must be symbolic too.

The section `[fonts]` is only considered if the font-matcher does not find an appropriate font amongst the existing installed fonts. It is suggested to only use this section.

The section `[replace]` is stronger and applied before the font-matcher. This means a font will be replaced as defined, even if the correctly installed font is available on the system.

Syntax: The syntax of the mapping file is as follows

```
[ fonts ]
PDF_font_1=installed_font_1{,font_style}
PDF_font_2=installed_font_2{,font_style}
[ replace ]
PDF_font_n=installed_font_n{,font_style}
```

`PDF_font_*` is the name of the font in the PDF.

This name can be found in one of the following ways:

- Use any tool that can list fonts. Such as 3-Heights™ PDF Extract or 3-Heights™ PDF Optimizer. Ignore possible prefixes of font subsets. A subset prefix consists of 6 characters followed by the plus sign. For example "KHFOKE+MonotypeCorsiva", in this case only use "MonotypeCorsiva" as font name in the mapping file.

- Open the document with Adobe Acrobat, use the "MarkUp Text Tool", mark the text of which you would like to know the font name, right-click it, select "Properties..."

installed_font_* is the font family name of the installed font.

To retrieve this name, find the font in the Windows' font directory and open it by double-clicking. The first line in the property window displays the font family name (this may vary depending on the operating system). The font family name does not include font styles; so an example of a font family name is "Arial", but not "Arial Italic".

font_style is an optional style, that is added coma-separated after the font family name.

The style is always one word. Examples of font styles are "Italic", "Bold", "BoldItalic". Omit the font style, if it is "Regular" or "Normal".

Remove blanks from all font names, i.e. in both the **PDF_font_*** and the **installed_font_***.

Example:

```
[ fonts ]
Ryumin-Light=MSMincho
GothicBBB-Medium=MSGothic
[ replace ]
ArialIta=Arial,BoldItalic
```

2.7 Note about the Evaluation License

With the evaluation license the 3-Heights™ PDF to EMF Converter API automatically adds a watermark to the output files.

2.8 Special Directories

2.8.1 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.

2.8.2 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

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

2.8.3 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 Pdf2EmfAPI.dll resides.

3 License Management

The 3-Heights™ PDF to EMF Converter API requires a valid license in order to run correctly. If no license key is set or the license is not valid, then most of the interface elements documented in [Interface Reference](#) will fail with an error code and error message indicating the reason.

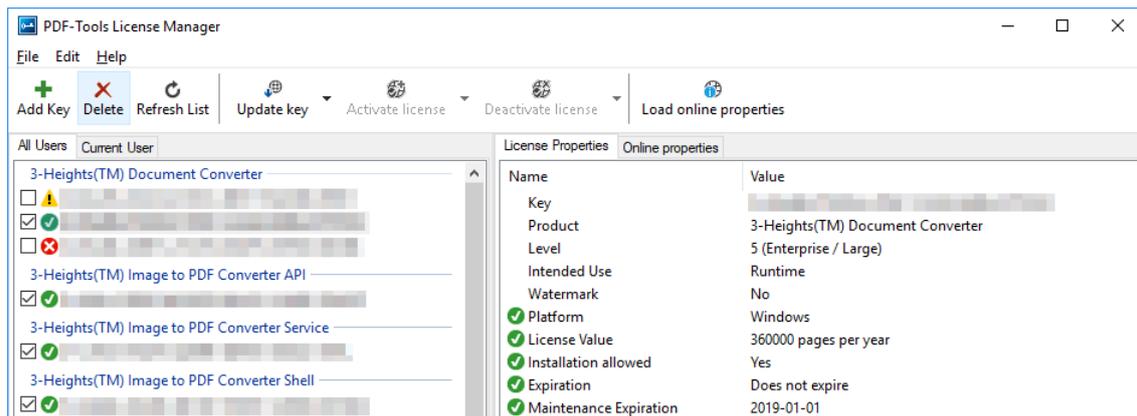
3.1 License Installation and Management

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 [SetLicenseKey](#) method. 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
```

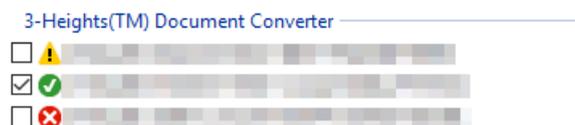
3.2 License Selection and Precedence

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:

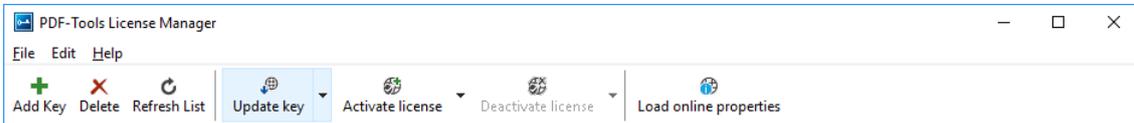
1. License key passed at runtime.
2. License selected for the current user
3. License selected for the current user ([legacy key format](#))
4. License selected for all users
5. License selected for all users ([legacy key format](#))

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

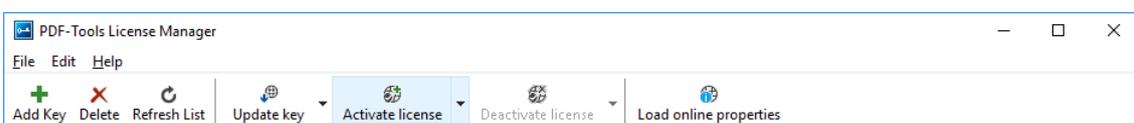
The License can be activated directly within the license manager. Every activation increases the activation count of the license by 1.

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
```

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:

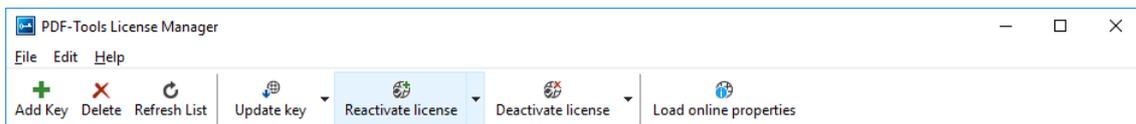
```
licmgr activate -cd 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX  
licmgr activate -c "custom comment" 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

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:

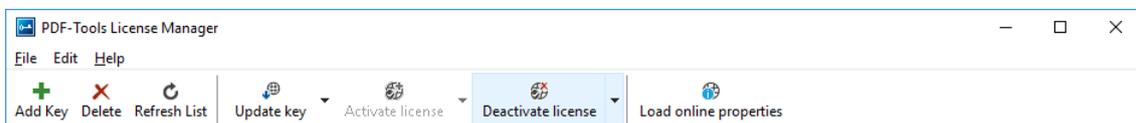
```
licmgr activate 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

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:

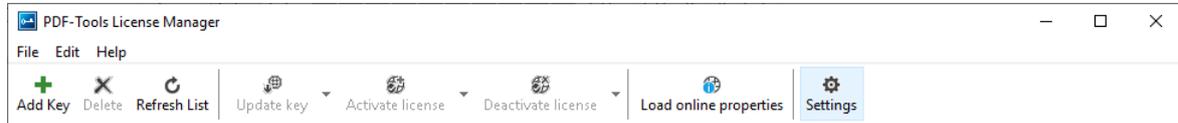
```
licmgr deactivate 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

3.5 Proxy Setting

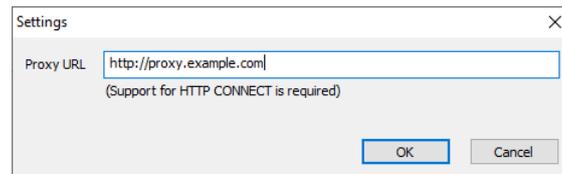
A proxy URL can be configured for computers that cannot access the internet without a web proxy.

Note: The proxy must allow connections via HTTP CONNECT to the server www.pdf-tools.com:443.

In the Graphical User Interface press the button "Settings" in the toolbar:



and enter the proxy URL in the respective field:



3.6 Offline Usage

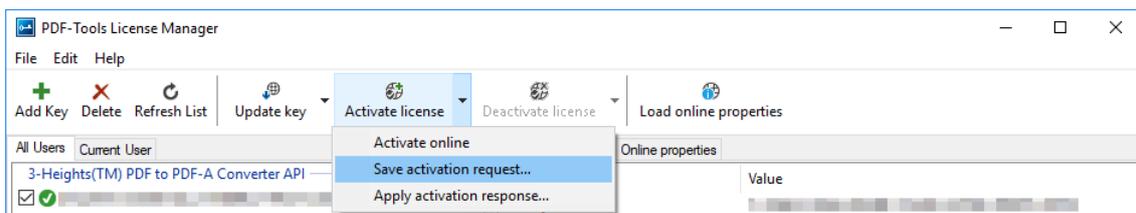
The following actions in the license manager need access to the internet:

- [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.

3.6.1 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-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.

3.6.2 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)

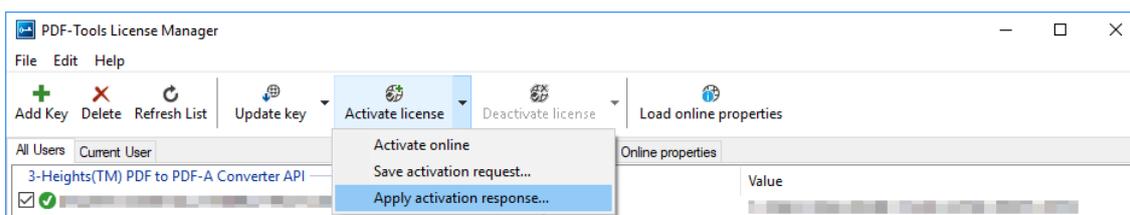
Upload your license request. For more information and instructions please check the manual of your product.



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

3.6.3 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
```

3.7 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.

3.8 License Key Storage

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

3.8.1 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)

3.9 Troubleshooting

3.9.1 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.

3.9.2 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.

3.9.3 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.

3.9.4 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.

3.9.5 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.

3.9.6 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.

3.9.7 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.

3.9.8 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 3.6](#) 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 User's Guide

4.1 Font and Text Issues

1. For issues with text using non-embedded fonts:
 1. Ensure the required fonts are available on the system (see Chapter [Fonts](#)).
 2. See Section [Handle Non-Embedded Fonts](#).
2. For issues with text using embedded fonts:
 1. Ensure embedded fonts are used (i.e. `eOptionNoEmbedded` is not set).
 2. Ensure the two system environment variables TEMP and TMP exist and point to an existing directory. These variables not being set is a common error source for service applications that run under a user that has no temporary directory and thus cannot install fonts. See also Chapter [Installation and Deployment](#).
 3. See Section [Handle Embedded Fonts](#).

4.1.1 Handle Non-Embedded Fonts

A PDF document can contain embedded fonts. EMF files created by this API do not contain embedded fonts. Instead they can be saved as a separate file. These fonts must temporarily be installed (e.g. double click them). This way the EMF file can refer to these fonts and display them correctly.

Font Replacement Strategy

This section describes the exact behavior of font handling of the rendering engine. It is rather technical and it is not required to be understood in order to properly use the software.

The following steps are performed sequentially in the search of a font. If a font is found, the search is stopped; otherwise the next step is performed.

1. If the font is not embedded or `eOptionPreInstalled` is set:
 - a. If the font name appears in the `[replace]` section in the configuration file `fonts.ini` the name is replaced and looked up in the installed font collection.
 - b. If it is a standard font³ it is replaced by the equivalent TrueType font name and it is looked up in the installed font collection.
 - c. If the font name appears in the `[fonts]` section in the configuration file `fonts.ini` the name is replaced and looked up in the installed font collection.
 - d. If the font has "Italic" or "Bold" in its name the font without these styles is looked up in the installed font collection.
2. If a font name is looked up in the installed font collection then the name comparison is performed as follows:
 - a. PostScript name.
 - b. TrueType name without blanks (a missing style is interpreted as "Regular" or "Normal").
 - c. TrueType name without modifications.
3. If the font is embedded, it is converted to a Windows compatible font and temporarily installed. If `eOption-NoEmbedded` is used then the glyphs of the fonts are converted to either bitmaps or outlines⁴. If `eOptionOut-lines` is used then the glyphs are converted to outlines only.
4. If the font is not embedded and the Unicodes are available then the nearest font from the installed font collection is tailored to the metrics of the font.

³ e.g. Times-Roman, Helvetica, Courier

⁴ The outline of a glyph is a vector graphic without any reference to the original font program.

4.1.2 Handle Embedded Fonts

The following list provides possible work-arounds if text is printed incorrectly. Options should be tried in ascending order.

1. Using the option `eOptionNoEmbedded` inhibits all embedded fonts from being used in the spool file and the printer hardware. Instead the glyphs are converted to either bitmaps or outlines. Using the option `eOptionOutlines` at the same time the conversion is restricted to outlines.
2. Using the option `eOptionPreInstalled` inhibits embedded fonts which have the same name as the corresponding installed font from being used. This option can also be used to reduce the number of fonts in a spool file if the printer hardware memory capacity is limited.
3. Pre-render the page in a bitmap and send the pre-rendered image to the printer (`eOptionBitmap`). This results in large spool files.

4.2 Call Sequence

The standard calling sequence to convert a multipage PDF file to multiple single page EMF files and the fonts is shown below:

```
Open
for i = 1 to PageCount
{
  CreateSurface
  RenderPage i
  SaveTo
  for j = 0 to FontCount
  {
    SaveFontTo j
  }
  DestroySurface
}
Close
```

There is a sample project in C and in VB available. It is suggested to take a look at them.

4.3 Error Handling

Most methods of the 3-Heights™ PDF to EMF Converter API can either succeed or fail depending on user input, state of the PDF to EMF Converter API, or the state of the underlying system. It is important to detect and handle these errors, to get accurate information about the nature and source of the issue at hand.

Methods communicate their level of success or failure using their return value. Which return values have to be interpreted as failures is documented in the chapter [Interface Reference](#). To identify the error on a programmatic level, check the property [ErrorCode](#).

Example:

```
public Boolean Open(string file, string password)
{
  if (!doc.Open(file, password))
  {
    if (doc.ErrorCode == PDFErrorCode.PDF_E_PASSWORD)
```

```
{
    password = InputBox.Show("Password incorrect. Enter correct password:");
    return Open(file, password);
}
else
{
    MessageBox.Show(String.Format("Error {0}", doc.ErrorCode));
    return false;
}
}
[...]
```

5 Interface Reference

5.1 Pdf2Emf Interface

5.1.1 Close

Method: Boolean Close()

Close an opened input file. If the document is already closed the method does nothing.

Returns:

True The file was closed successfully.

False Otherwise.

5.1.2 CreateSurface

Method: Boolean CreateSurface(Long Left, Long Top, Long Right, Long Bottom)

Create an EMF surface. Coordinate values are in points.

5.1.3 CreateSurface2

Method: Boolean CreateSurface2(String Printer, Long Left, Long Top, Long Right, Long Bottom)

Create an EMF surface. Coordinate values are in points.

5.1.4 DestroySurface

Method: Boolean DestroySurface()

Destroy the current surface.

5.1.5 ErrorCode

Property (get): TPDFErrorCode ErrorCode

This property can be accessed to receive the latest error code. This value should only be read if a function call on the PDF to EMF Converter API has returned a value, which signals a failure of the function (see chapter [Error Handling](#)). See also enumeration [TPDFErrorCode](#). PDF-Tools error codes are listed in the header file `bseerror.h`. Please note that only few of them are relevant for the 3-Heights™ PDF to EMF Converter API.

Note: This property does not exist in the COM interface.

5.1.6 FontCount

Property (get): Long `FontCount`

Return the number of embedded fonts in the document. At the beginning the number of fonts is zero. With each call to [RenderPage](#) the number of fonts is increased by the number of fonts found on the rendered page.

You can retrieve all fonts of a document when all pages have been rendered or, incrementally, by retrieving the newly added fonts after each rendered page.

5.1.7 Height

Method: Long `Height(Long PageNo)`

Return the height in 1/100 mm.

5.1.8 Open

Method: Boolean `Open(String Filename, String Password)`

Open a PDF file, i.e. make the objects contained in the document accessible. If another document is already open, it is closed first.

Parameters:

Filename [`String`] The file name and optionally the file path, drive or server string according to the operating systems file name specification rules.

Password [`String`] (optional) The user or the owner password of the encrypted PDF document. If this parameter is left out an empty string is used as a default.

Returns:

True The file could be successfully opened.

False The file does not exist, it is corrupt, or the password is not valid. Use the property [ErrorCode](#) for additional information.

5.1.9 Options

Property (get, set): Long Options

Get and set the rendering options. See also enumeration [TPDFRenderOption](#).

5.1.10 PageCount

Property (get): Long PageCount

Get the number of pages of an open document. If the document is closed or if the document is a collection (also known as PDF Portfolio) then this property is 0.

5.1.11 PermissionFlags

Property (get): Long PermissionFlags

The property returns the permission flags of the open document.

5.1.12 ProductVersion

Property (get): String ProductVersion

Get the version of the 3-Heights™ PDF to EMF Converter API in the format "A.C.D.E".

5.1.13 RenderPage

Method: Boolean RenderPage(Long PageNo, Long Left, Long Top, Long Right, Long Bottom)

Render page PageNo of the PDF document to the EMF.

Returns:

True The page was rendered successfully.

False The page could not be rendered.

5.1.14 SaveTo

```
Method: Boolean SaveTo(String FileName)
```

Save the EMF to file.

5.1.15 SaveFontTo

```
Method: Boolean SaveFontTo(Long FontNo, String FileName)
```

Extract and save the embedded fonts as True Type fonts. Use [FontCount](#) to retrieve the number of embedded fonts prior to calling this function.

5.1.16 SetLicenseKey

```
Method: Boolean SetLicenseKey(String LicenseKey)
```

Set the license key.

Note: This method does not exist in the COM interface.

5.1.17 Width

```
Method: Long Width(Long PageNo)
```

Return the width in 1/100 mm.

5.2 Enumerations

5.2.1 TPDFRendererOption Enumeration

Renderer options are set using the property [Options](#). To combine multiple options use a bitwise OR operator. To disable an option use the bitwise AND NOT operators.

Example: Visual Basic

Enable or disable an option, and leave all other options untouched:

```
' Enable high quality rendering (anti-aliasing)  
.Options = .Options OR eOptionHighQuality  
' Disable high quality rendering (anti-aliasing)
```

```
.Options = .Options AND NOT eOptionHighQuality
```

Example: C/C++

```
int iOptions = Pdf2EmfGetOptions (pDocument);  
// Enable high quality rendering (anti-aliasing)  
Pdf2EmfSetOptions (pDocument, iOptions | eOptionHighQuality);  
// Disable high quality rendering (anti-aliasing)  
Pdf2EmfSetOptions (pDocument, iOptions & ~eOptionHighQuality);
```

The following list includes renderer options that are relevant for the 3-Heights™ PDF to EMF Converter API. Note that there are more enumerations available, but they are unrelated to this API.

TPDFRendererOption Table

TPDFRendererOption	
eOptionAutoAccurateMode	Detect content that cannot be rendered using RenderingMode eModeFast (GDI) and switch to eModeAccurate (GDI+) automatically, e.g. to render transparent tiling patterns. Does not have any effect if eModeAccurate was already set.
eOptionBanding	The data is sent in small chunks. This is mainly for older printers with limited memory.
eOptionBitmap	The pages are rendered in a bitmap, which then is sent to the device.
eOptionDisableAnnots	When setting this option flag then annotations are not drawn.
eOptionDisableBPC	If this option flag is set then the black point compensation feature is disabled when converting colors e.g. from CMYK to RGB.
eOptionDisableContent	When setting this option flag then only form fields and annotations are drawn without the underlying page content.
eOptionDisableFilter	Disable image filtering. Images are scaled using the nearest-neighbor algorithm, which improves performance at the cost of rendering quality.
eOptionDisablePatterns	Disable patterns.
eOptionDrawPopups	Draw pop up windows of annotations, such as sticky notes.
eOptionJPEG	If a printer supports JPEG compression, then the pages can be sent with JPEG compression to reduce the size of the spool file.
eOptionHighQuality	Anti-aliasing for text and path objects and filtering of image objects can be turned off and on with this option.
eOptionNoBackground	Do not paint a white background. With this option, the background of the output will be transparent, if supported.
eOptionNoEmbedded	Do not use embedded fonts. Instead fonts from the operating system's font directory are used (%Systemroot%\fonts).

TPDFRendererOption Table

<code>eOptionOutlines</code>	Convert fonts into vector graphics.
<code>eOptionPreInstalled</code>	Replace embedded fonts with a pre-installed font if the same font is already installed on the OS.
<code>eOptionPrint</code>	Draw the document as it was intended for printing. Otherwise, the document is drawn as it is shown in an interactive viewer. For example, this has an effect on which annotations are visible.
<code>eOptionPrintOnlySig</code>	Draw the digital signature appearance only (without any status appearances, e.g. valid or invalid).
<code>eOptionDoNotPrintSig</code>	Do not draw digital signature appearances.
<code>eOptionTransparency</code>	Deprecated option that has no effect.
<code>eOptionTrueType</code>	CFF and Type1 fonts are converted to True Type fonts. This option overrules option <code>eOptionType1</code> .
<code>eOptionType1</code>	CFF fonts are converted to Type1 fonts.
<code>eOptionUseFastImages</code>	Always print images in fast mode. This should help resolving performance issues with complex images and image masks of documents that are to be printed in accurate mode.

Recommended Settings

```
Options = eOptionTrueType + eOptionNoEmbedded + eOptionPreInstalled
```

5.2.2 TPDFErrorCode Enumeration

All `TPDFErrorCode` enumerations start with a prefix, such as `PDF_`, followed by a single letter which is one of `S`, `E`, `W` or `I`, an underscore and a descriptive text.

The single letter gives an indication of the severity of the error. These are: Success, Error, Warning and Information. In general, an error is returned if an operation could not be completed. A warning is returned if the operation was completed, but problems occurred in the process.

A list of all error codes is available in the C API's header file `bseerror.h`, the javadoc documentation of `com.pdftools.NativeLibrary.ERRORCODE` and the .NET documentation of `Pdftools.Pdf.PDFErrorCode`. Note that only a few are relevant for the 3-Heights™ PDF to EMF Converter API, most of which are listed here:

TPDFErrorCode Table

TPDFErrorCode	Description
<code>PDF_S_SUCCESS</code>	The operation was completed successfully.
<code>LIC_E_NOTINIT, ... LIC_E_LEVEL</code>	Various license management related errors.

TPDFErrorCode Table

PDF_E_FILEOPEN	Failed to open the file.
PDF_E_FILECREATE	Failed to create the file.
PDF_E_PASSWORD	The authentication failed due to a wrong password.
PDF_E_UNKSECHANDLER	The file uses a proprietary security handler, e.g. for a proprietary digital rights management (DRM) system.

6 Version History

Some of the documented changes below may be preceded by a marker that specifies the interface technologies the change applies to. E.g. [C, Java] applies to the C and the Java interface.

6.1 Patches in Version 4.12

Note that the version number of the initial “final release” is 4.12.26.3.

Patch 4.12.26.4

- **Improved** error messages for failed HTTP connections in various situations (including license manager).
- **Added** missing documentation and release note for the proxy setting in the GUI license manager.
- **Improved** license reactivation behavior of the commandline license manager (licmgr): The server is now only contacted if necessary.
- **Improved** behavior of license manager when dealing with licenses of unreleased products.

6.2 Changes in Version 4.12

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

6.3 Changes in Version 4.11

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

6.4 Changes in Version 4.10

- **Improved** robustness against corrupt input PDF documents.
- **Removed** the font ZapfDingbats.ttf from the product kit as it is not required anymore.

6.5 Changes in Version 4.9

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

6.6 Changes in Version 4.8

- [C, COM] **New** property `ProductVersion`.

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 to EMF Converter API 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