

User Manual

# 3-Heights™ PDF Viewer API

Version 4.12.26.6



# Contents

|          |                                       |           |
|----------|---------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                   | <b>5</b>  |
| 1.1      | Description                           | 5         |
| 1.2      | Functions                             | 5         |
| 1.2.1    | Viewing                               | 5         |
| 1.2.2    | Navigation                            | 6         |
| 1.2.3    | Printing                              | 6         |
| 1.2.4    | Annotate Document                     | 7         |
| 1.2.5    | Formats                               | 7         |
| 1.2.6    | Compliance                            | 7         |
| 1.3      | Interfaces                            | 7         |
| 1.4      | Operating Systems                     | 7         |
| 1.5      | Compatibility Note                    | 8         |
| <b>2</b> | <b>Installation and Deployment</b>    | <b>9</b>  |
| 2.1      | Windows                               | 9         |
| 2.2      | Interfaces                            | 10        |
| 2.2.1    | Development                           | 10        |
| 2.2.2    | Deployment                            | 11        |
| 2.3      | Interface Specific Installation Steps | 12        |
| 2.3.1    | COM Interface                         | 12        |
| 2.3.2    | C Interface                           | 13        |
| 2.4      | Uninstall, Install a New Version      | 13        |
| 2.5      | Color Profiles                        | 13        |
| 2.5.1    | Default Color Profiles                | 13        |
| 2.5.2    | Get Other Color Profiles              | 14        |
| 2.6      | Fonts                                 | 14        |
| 2.6.1    | Font Cache                            | 14        |
| 2.6.2    | Font Configuration File fonts.ini     | 14        |
| 2.7      | Note about the Evaluation License     | 15        |
| 2.8      | Special Directories                   | 15        |
| 2.8.1    | Directory for temporary files         | 15        |
| 2.8.2    | Cache Directory                       | 16        |
| 2.8.3    | Font Directories                      | 16        |
| <b>3</b> | <b>License Management</b>             | <b>17</b> |
| 3.1      | License Installation and Management   | 17        |
| 3.1.1    | Graphical License Manager Tool        | 17        |
|          | List all installed license keys       | 17        |
|          | Add and delete license keys           | 17        |
|          | Display the properties of a license   | 17        |
| 3.1.2    | Command Line License Manager Tool     | 18        |
|          | List all installed license keys       | 18        |
|          | Add and delete license keys           | 18        |
|          | Display the properties of a license   | 18        |
| 3.2      | License Selection and Precedence      | 19        |
| 3.2.1    | Selection                             | 19        |
| 3.2.2    | Precedence                            | 19        |
| 3.3      | Key Update                            | 20        |
| 3.4      | License activation                    | 20        |

|          |  |           |
|----------|--|-----------|
| 3.4.1    | Activation .....   | 20        |
| 3.4.2    | Reactivation .....   | 21        |
| 3.4.3    | Deactivation .....   | 21        |
| 3.5      | Proxy Setting .....  | 21        |
| 3.6      | Offline Usage .....  | 22        |
| 3.6.1    | First Step: Create a Request File .....                      | 22        |
| 3.6.2    | Second Step: Use Form on Website .....                       | 23        |
| 3.6.3    | Third Step: Apply the Response File .....                    | 23        |
| 3.7      | License Key Versions .....                                   | 23        |
| 3.8      | License Key Storage .....                                    | 23        |
| 3.8.1    | Windows .....  | 23        |
| 3.9      | Troubleshooting .....  | 24        |
| 3.9.1    | License key cannot be installed .....                        | 24        |
| 3.9.2    | License is not visible in license manager .....              | 24        |
| 3.9.3    | License is not found at runtime .....                        | 24        |
| 3.9.4    | Eval watermark is displayed where it should not .....        | 24        |
| 3.9.5    | Activation is not recognized .....                           | 25        |
| 3.9.6    | Activation is invalidated too often .....                    | 25        |
| 3.9.7    | Connection to the licensing service fails .....              | 26        |
| 3.9.8    | Offline usage fails due to a request/response mismatch ..... | 26        |
| <b>4</b> | <b>Programming Interfaces .....</b>                          | <b>27</b> |
| 4.1      | Visual Basic .....   | 27        |
| 4.2      | .NET .....   | 28        |
| 4.3      | MFC .....  | 30        |
| <b>5</b> | <b>User's Guide .....</b>                                    | <b>31</b> |
| 5.1      | Open a PDF Document .....                                    | 31        |
| 5.2      | Navigate .....   | 31        |
| 5.3      | Start a Print Job .....                                      | 32        |
| 5.3.1    | What Is a Print Job? .....                                   | 32        |
| 5.3.2    | Individual Pages .....                                       | 32        |
| 5.3.3    | Entire Document .....  | 33        |
| 5.3.4    | Print to the Default Printer .....                           | 33        |
| 5.4      | Sticky Note Annotations .....                                | 33        |
| 5.5      | How to use the Property Options .....                        | 34        |
| 5.6      | Error Handling .....   | 35        |
| <b>6</b> | <b>Interface Reference .....</b>                             | <b>36</b> |
| 6.1      | PDFViewer Interface .....                                    | 36        |
| 6.1.1    | BeginDocument .....  | 36        |
| 6.1.2    | Border .....   | 36        |
| 6.1.3    | Center .....   | 36        |
| 6.1.4    | Close .....  | 37        |
| 6.1.5    | ClosePrinter .....   | 37        |
| 6.1.6    | CompareDocuments .....                                       | 37        |
| 6.1.7    | ConvertPt .....  | 38        |
| 6.1.8    | Copies .....   | 38        |
| 6.1.9    | CreateAnnotation .....                                       | 38        |
| 6.1.10   | CursorMode .....   | 39        |
| 6.1.11   | DataType .....   | 39        |
| 6.1.12   | DefaultSource .....  | 39        |
| 6.1.13   | DPI .....  | 40        |

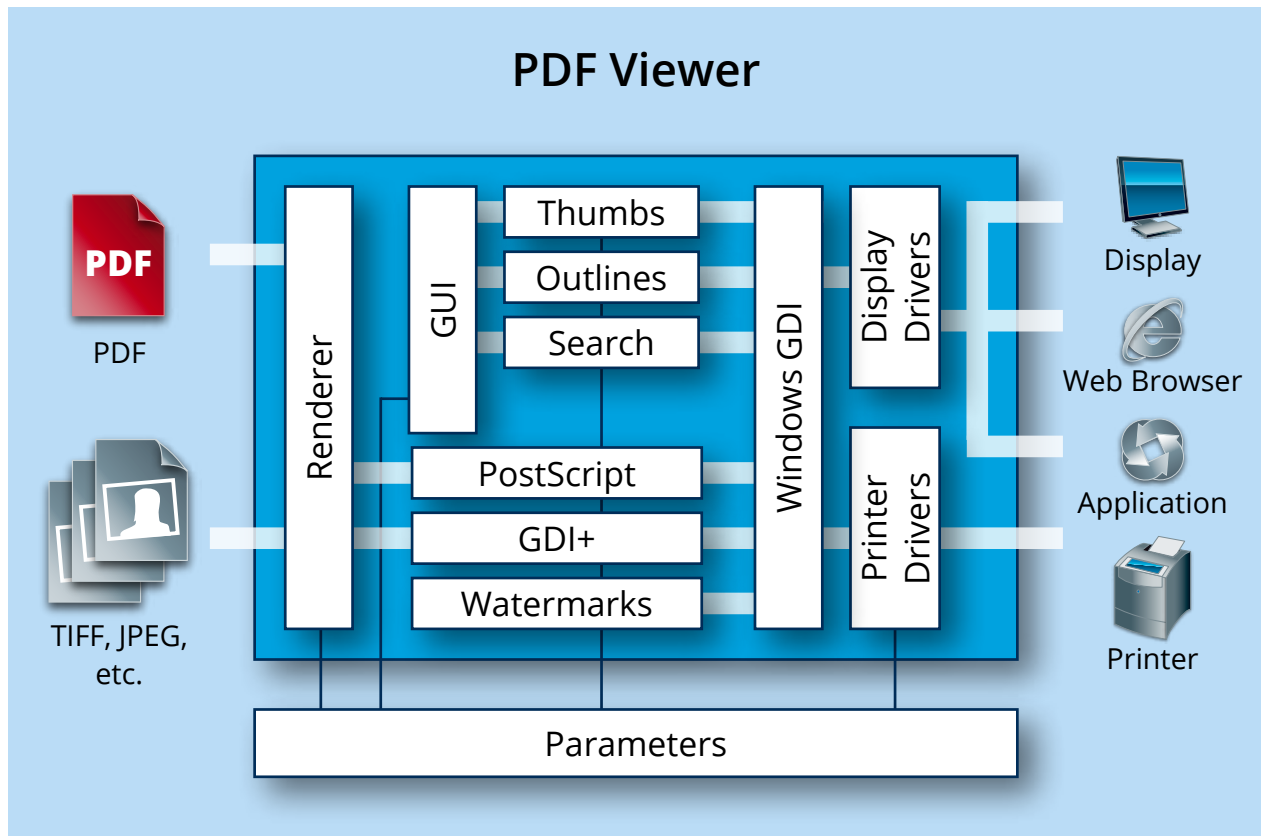
|        |                                  |    |
|--------|----------------------------------|----|
| 6.1.14 | Duplex .....                     | 40 |
| 6.1.15 | EndDocument .....                | 40 |
| 6.1.16 | ErrorCode .....                  | 41 |
| 6.1.17 | FitMode .....                    | 41 |
| 6.1.18 | FitPage .....                    | 41 |
| 6.1.19 | GetDefaultPrinter .....          | 41 |
| 6.1.20 | GetPageRect .....                | 42 |
| 6.1.21 | GoBack .....                     | 42 |
| 6.1.22 | Goto .....                       | 42 |
| 6.1.23 | LayoutMode .....                 | 43 |
| 6.1.24 | LicenseIsValid .....             | 43 |
| 6.1.25 | MarkRectangle .....              | 44 |
| 6.1.26 | Modified .....                   | 44 |
| 6.1.27 | OffsetX,OffsetY .....            | 45 |
| 6.1.28 | Open .....                       | 45 |
| 6.1.29 | OpenMem .....                    | 45 |
| 6.1.30 | OpenMem2 .....                   | 46 |
| 6.1.31 | OpenPrinter .....                | 46 |
| 6.1.32 | Options .....                    | 47 |
| 6.1.33 | Orientation .....                | 47 |
| 6.1.34 | Page .....                       | 47 |
| 6.1.35 | PageAtPos .....                  | 48 |
| 6.1.36 | PageCount .....                  | 48 |
| 6.1.37 | PagesPerWindow .....             | 48 |
| 6.1.38 | PanX .....                       | 48 |
| 6.1.39 | PanY .....                       | 49 |
| 6.1.40 | PaperSize .....                  | 49 |
| 6.1.41 | PrintDocument .....              | 49 |
| 6.1.42 | PrintDocumentDlg .....           | 50 |
| 6.1.43 | PrintPage .....                  | 50 |
| 6.1.44 | RenderingMode .....              | 50 |
| 6.1.45 | ProductVersion .....             | 51 |
| 6.1.46 | Rotate .....                     | 51 |
| 6.1.47 | SaveAs .....                     | 51 |
| 6.1.48 | ScaleXY .....                    | 52 |
| 6.1.49 | SearchFirst .....                | 52 |
| 6.1.50 | SearchNext .....                 | 53 |
| 6.1.51 | SetAnnotPropertiesStr .....      | 53 |
| 6.1.52 | SetCMSEngine .....               | 53 |
| 6.1.53 | SetCursor .....                  | 54 |
| 6.1.54 | SetLicenseKey .....              | 54 |
| 6.1.55 | SetUILanguage .....              | 54 |
| 6.1.56 | SizeX,SizeY .....                | 55 |
| 6.1.57 | TreeViewWidth .....              | 55 |
| 6.1.58 | ViewerOptions .....              | 55 |
| 6.1.59 | XDPI .....                       | 55 |
| 6.1.60 | YDPI .....                       | 56 |
| 6.1.61 | Zoom .....                       | 56 |
| 6.2    | Events .....                     | 56 |
| 6.3    | Enumerations .....               | 57 |
| 6.3.1  | TPDFAnnotation Enumeration ..... | 57 |
| 6.3.2  | TPDFCursorMode Enumeration ..... | 58 |

|          |  |           |
|----------|--|-----------|
| 6.3.3    | TPDFErrorCode Enumeration .....                | 59        |
| 6.3.4    | TPDFFitMode Enumeration .....                  | 60        |
| 6.3.5    | TPDFLayoutMode Enumeration .....               | 60        |
| 6.3.6    | TPDFPermission Enumeration .....               | 60        |
| 6.3.7    | TPDFRenderOption Enumeration .....             | 61        |
| 6.3.8    | TPDFRenderingMode Enumeration .....            | 63        |
| 6.3.9    | TPDFRotateMode Enumeration .....               | 63        |
| 6.3.10   | TPDFViewerOption Enumeration .....             | 63        |
| <b>7</b> | <b>Tips, Tricks, and Troubleshooting .....</b> | <b>65</b> |
| 7.1      | Performance .....                              | 65        |
| 7.2      | Font and Text Issues .....                     | 65        |
| 7.2.1    | Handle Non-Embedded Fonts .....                | 65        |
|          | Viewing vs Printing .....                      | 65        |
|          | Font Replacement Strategy .....                | 65        |
| 7.2.2    | Handle Embedded Fonts .....                    | 66        |
| 7.3      | Transparency .....                             | 66        |
| <b>8</b> | <b>Version History .....</b>                   | <b>67</b> |
| 8.1      | Patches in Version 4.12 .....                  | 67        |
| 8.2      | Changes in Version 4.12 .....                  | 67        |
| 8.3      | Changes in Version 4.11 .....                  | 67        |
| 8.4      | Changes in Version 4.10 .....                  | 67        |
| 8.5      | Changes in Version 4.9 .....                   | 67        |
| 8.6      | Changes in Version 4.8 .....                   | 68        |
| <b>9</b> | <b>Licensing, Copyright, and Contact .....</b> | <b>69</b> |
| <b>A</b> | <b>Default Values .....</b>                    | <b>70</b> |
| A.1      | Duplex Modes .....                             | 70        |
| A.2      | Paper Bins .....                               | 70        |
| A.3      | Paper Sizes .....                              | 70        |

# 1 Introduction

## 1.1 Description

The 3-Heights™ PDF Viewer API is an independent, compact, high-quality PDF viewer. It offers a multitude of functions such as display options, navigation, optional printing and support for non-PDF formats such as TIFF, JPEG and other raster image formats. The PDF Viewer is characterized mainly by its ease of integration and handling and high performance.



The Viewer can be used as an intranet website tool, for instance, or as a standalone application or software solution component. The Viewer can be started outside of the programming environment in under a second via the command line, and integrates seamlessly in modern development environments as an API.

## 1.2 Functions

The 3-Heights™ PDF Viewer API offers many options for displaying PDF documents in a manner that suits requirements: files can be viewed in single page or multi-page mode, and the navigation supports links, bookmarks and calling up pages arbitrarily. The search function can locate strings of text in the document. The tool supports PDF documents in many languages: the reproduction of Japanese, Chinese, Korean, Russian, etc., is no problem at all. The PDF Viewer can also print out and save PDF documents.

### 1.2.1 Viewing

- Display PDF file in single page or multi-page mode
- Enter password to decrypt PDF documents

- Display image files such as TIFF, JPEG, JBIG2, JPEG2000, PNG and GIF
- Open files from file system or from internet (HTTP, HTTPS, FTP)
- Query the number of pages in a document
- Set scalability (page size, page width, actual size, any size)
- Set X,Y resolution in dpi
- Set various display options such as image filters
- Reproduce documents with Chinese, Japanese and Korean fonts (CJK)
- Read document from file or memory
- Convert viewer coordinates to PDF coordinates
- Set number of pages per window
- Rotate and display the page
- Show annotations
- Support for "FDF" file format
- Save the PDF or image files on display
- Set user rights for the saved new version of the PDF file
- Compare two documents and highlight differences
- Display, open and save embedded files, e.g. of PDF Portfolios

## 1.2.2 Navigation

- Search PDF documents for a specific string of text
- Show and hide windows for bookmarks and page navigation
- Jump to a bookmark's location
- Jump to the location of a link within the document
- Move windows vertically and horizontally using the mouse or keyboard
- Enlarge and reduce windows with the aid of a zoom rectangle or zoom factor
- Freely select any page in the document for display
- Query the position of the cursor
- Query the position of the scrollbars
- Intercept events (e.g. mouse events, keyboard events, zoom events)
- Highlight rectangle in color
- Set mouse cursor mode
- Show and hide scrollbars
- Activate or deactivate links
- Activate mouse wheel (for scrolling) or deactivate
- Replace embedded fonts with preinstalled fonts for viewing and printing
- Display fonts as vector graphics
- Navigate within embedded files, e.g. of PDF Portfolios

## 1.2.3 Printing

- Print the PDF or image files on display to any local or remote Windows printer
- RAW or EMF mode for printing
- Duplex printing
- Set paper tray
- Windows 9x compatibility mode to support legacy printers
- Set paper size
- Set number of copies
- Centered printing
- Set print orientation (landscape, portrait, printer default, automatic)

- Query local Windows standard printer
- Optional suppression of digital signatures during printing
- Option to pre-render pages and send them to the printer as bitmaps
- Black point compensation during printing

## 1.2.4 Annotate Document

- Create, modify and delete annotations on PDF documents.
- Enable and disable possibility to edit annotations.
- Support Sticky Notes, Text and Highlight Annotations.
- Configure default values for properties of new annotations that cannot be overridden by the user.
- Load annotations from and store annotations to FDF file.
- Save files on the file system or on the internet (WebDAV)

## 1.2.5 Formats

### Input Formats

- PDF 1.x (PDF 1.0, ..., PDF 1.7)
- PDF 2.0
- PDF/A-1, PDF/A-2, PDF/A-3
- Raster images (e.g. TIFF, JPEG, PNG, GIF, BMP)
- Forms Data Format (FDF)

### Output Formats

- The PDF Viewer can save files in their original input format.
- Forms Data Format (FDF)

## 1.2.6 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 Interfaces

There is a C and a COM interface available.

## 1.4 Operating Systems

The 3-Heights™ PDF Viewer 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



## 1.5 Compatibility Note

1. In versions prior to 2010 it was distinguished between the standard and the pro version. Certain features (such as printing or viewing raster images) were only supported in the pro version. At present there is only one version, which supports all features.
2. The Viewer control is an ActiveX control. It is provided in two different versions: as 32 and 64 bit. Since it runs in-process, the 64 bit version can only run with a 64 bit application. This means for development with MS Studio, which is a 32 bit application, the 32 bit version of the Viewer control must be used. For deployment however, the 64 bit version can be used.

## 2 Installation and Deployment

### 2.1 Windows

The 3-Heights™ PDF Viewer 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 Viewer API”. If you have no active downloads available or cannot log in, please contact [pdfsales@pdf-tools.com](mailto: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.

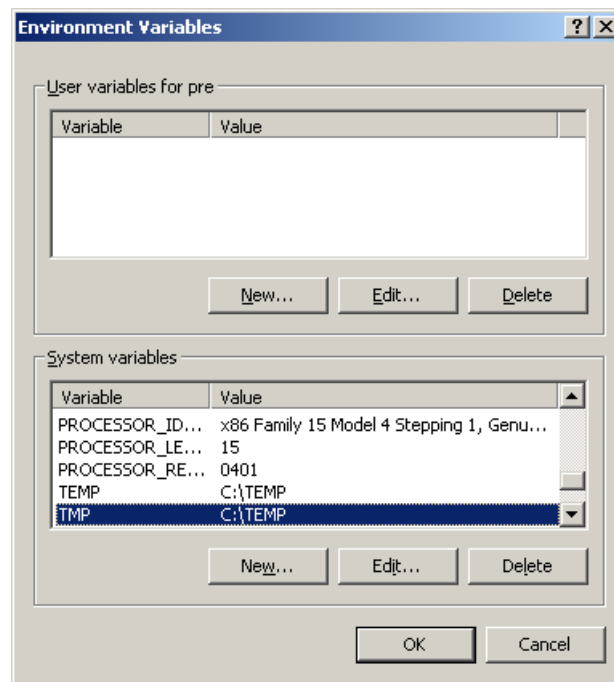
3. Unzip the archive to a local folder, e.g. C:\Program Files\PDF Tools AG\.

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. 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



6. Ensure the cache directory exists as described in chapter [Special Directories](#).
7. 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 Viewer 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"> <li>■ MS Visual Basic</li> <li>■ MS Office Products such as Access or Excel (VBA)</li> <li>■ C++</li> <li>■ VBScript</li> <li>■ others</li> </ul> <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>\viewerOCX.dll | Req. | Req. |
| bin\<platform>\INET.dll      | Opt. | Opt. |
| doc\*.pdf                    | Doc. | Doc. |
| doc\viewerAPI.idl            | Doc. |      |
| include\viewer_c.h           |      | Req. |
| include\*.*                  |      | Opt. |
| lib\<platform>\viewerOCX.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>\viewerOCX.dll | This is the DLL that contains the main functionality (required).  |
| bin\<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>\viewerOCX.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.

<sup>1</sup> These files must reside in the same directory as viewerOCX.dll.

## Files for Deployment

| Name                         | COM  | C    |
|------------------------------|------|------|
| bin\<platform>\viewerOCX.dll | Req. | Req. |
| bin\<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 `viewerOCX.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 viewerOCX.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%\viewerOCX.dll.
```

## 2.3 Interface Specific Installation Steps

### 2.3.1 COM Interface

**Registration** Before you can use the 3-Heights™ PDF Viewer 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 `viewerOCX.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\<platform>\viewerOCX.dll"
```

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 Viewer API, you need to use the `regsvr32` from the directory `%SystemRoot%\Syswow64` instead of `%SystemRoot%\System32`.<sup>2</sup>

<sup>2</sup> Otherwise you get the following message: `LoadLibrary("viewerOCX.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 `viewerOCX.dll`.

## 2.3.2 C Interface

- The header file `viewer_c.h` needs to be included in the C/C++ program.
- The library `lib\viewerOCX.lib` needs to be linked to the project.
- The dynamic link library `viewerOCX.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 device of 3-Heights™ PDF Viewer 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 Viewer 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.

**Note:** When setting an alternative color management system such as Neugebauer, no color profiles are required.

If no color profiles are available, default profiles for both RGB and CMYK are generated on the fly by the 3-Heights™ PDF Viewer 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 `viewerOCX.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](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 Viewer 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 `viewerOCX.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 Viewer API automatically adds a watermark to the displayed pages.

## 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 viewerOCX.dll resides.

## 3 License Management

The 3-Heights™ PDF Viewer 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.

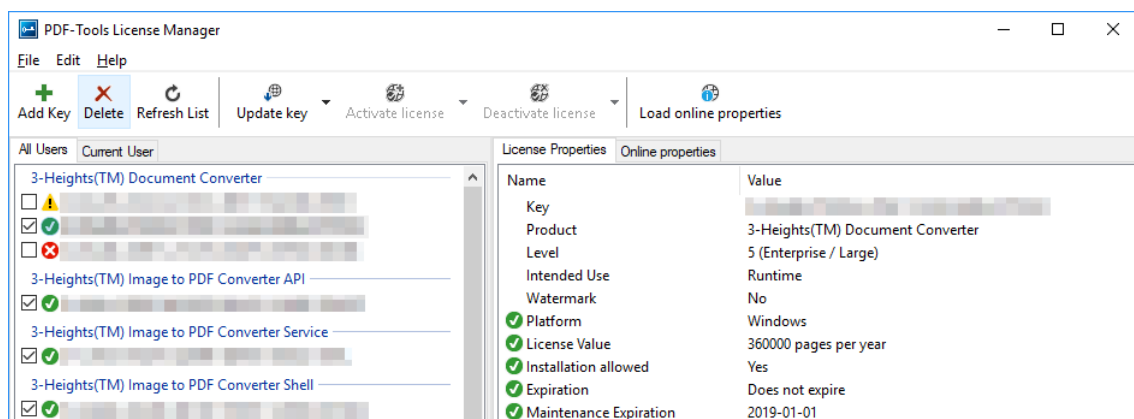
### 3.1 License Installation and Management

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
    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
```

Delete old license key:

```
licmgr delete 1-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
```

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
```

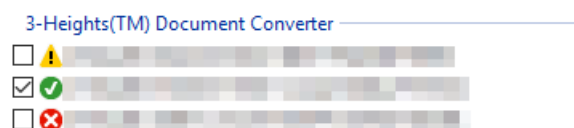
## 3.2 License Selection and Precedence

### 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:

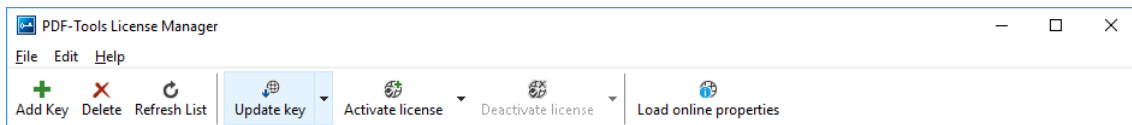
1. License key passed at runtime.
2. License selected for the current user
3. License selected for the current user ([legacy key format](#))
4. License selected for all users
5. License selected for all users ([legacy key format](#))

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
```

## 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

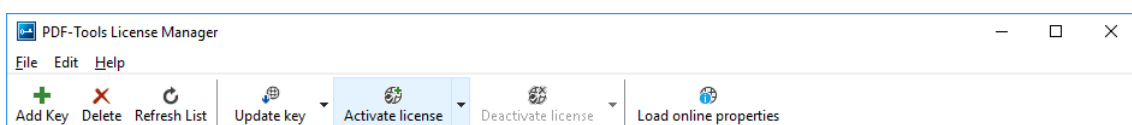
The License can be activated directly within the license manager. Every activation increases the activation count of the license by 1.

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:

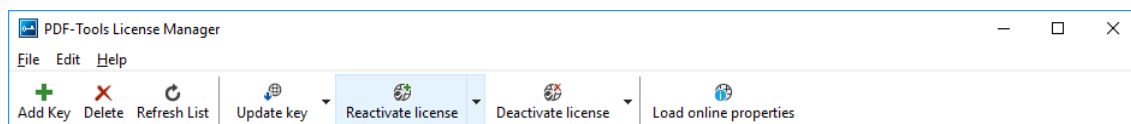
```
licmgr activate -cd 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX  
licmgr activate -c "custom comment" 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

## 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:

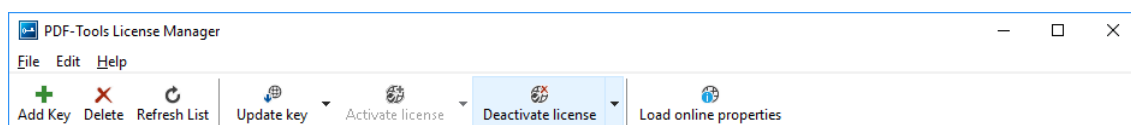
```
licmgr activate 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

## 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:

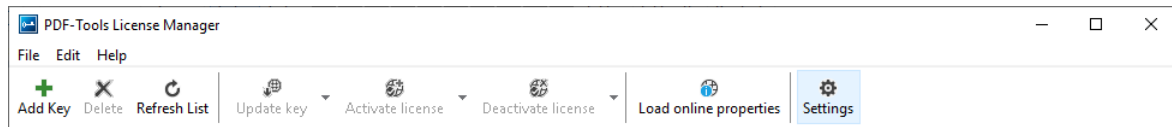
```
licmgr deactivate 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

## 3.5 Proxy Setting

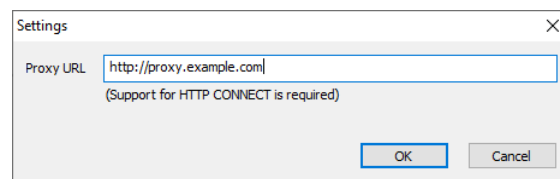
A proxy URL can be configured for computers that cannot access the internet without a web proxy.

**Note:** The proxy must allow connections via HTTP CONNECT to the server [www.pdf-tools.com:443](http://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

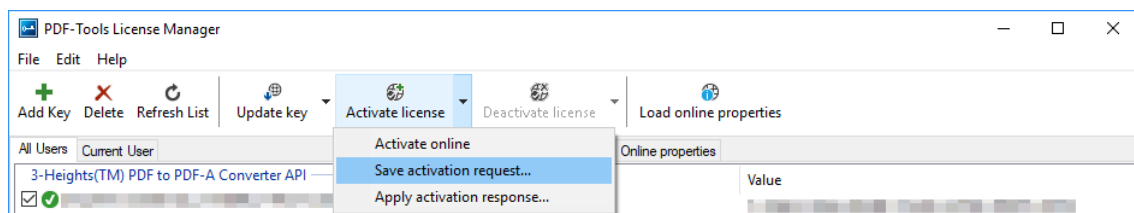
The following actions in the license manager need access to the internet:

- [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
```

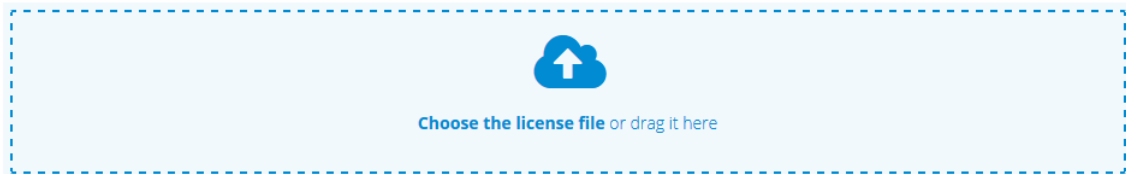
**License Deactivation:** When saving the deactivation request file, the license is **deactivated immediately** and cannot be used any further. It can however only be activated again after completing the deactivation on the website.

## 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)

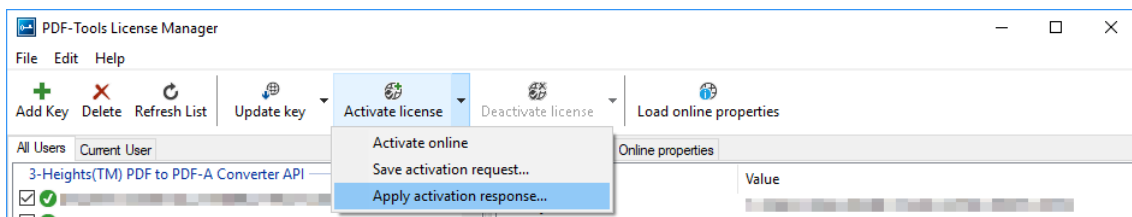
Upload your license request. For more information and instructions please check the manual of your product.



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 ( $\geq 4.12$ ) 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.

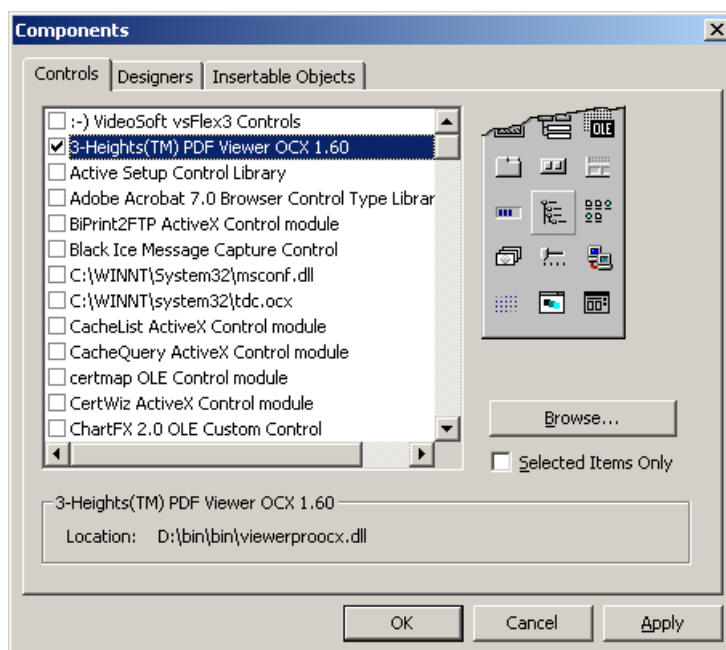
Make sure that the response file is applied to exactly the same license key in exactly the same location (machine, all users or specific user) where the request file was created.

## 4 Programming Interfaces

### 4.1 Visual Basic

After installing and registering the PDF Viewer, you find a Visual Basic example `PDFViewer.vbp` in the directory `samples\VB\`. You can either use this sample as a basis for an application, or you can start from scratch.

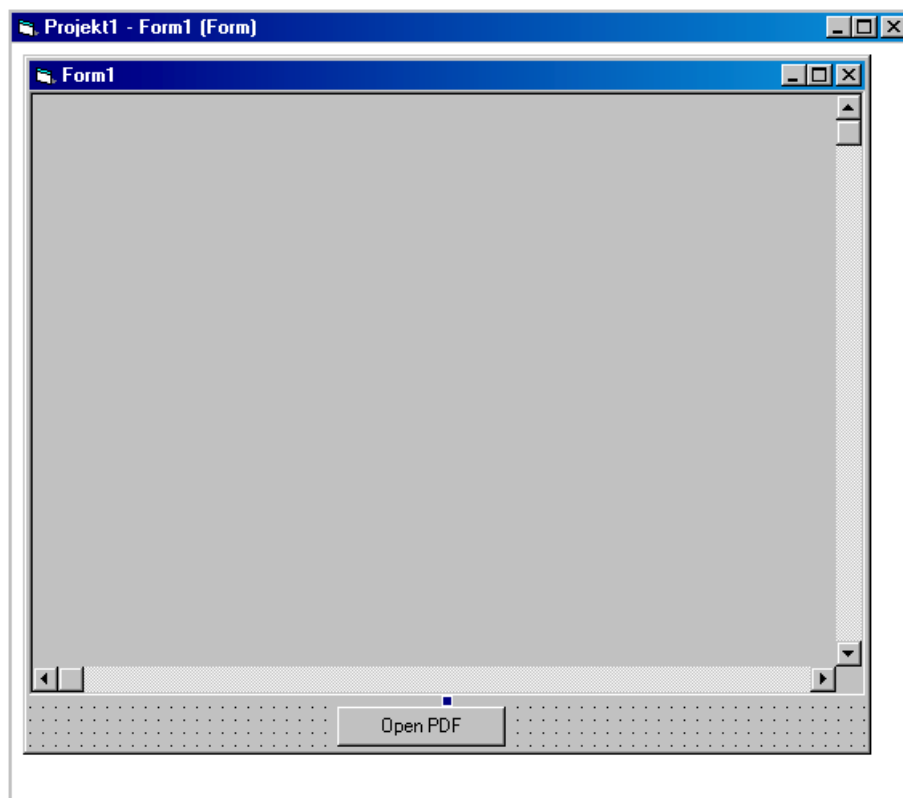
If you start from scratch, first create a new Standard-Exe Visual Basic 6 project. Add the 3-Heights™ PDF Viewer API OCX component:



This will add the PDF Viewer API object icon, which looks like this:



Add PDF Viewer object and a command button, so that you receive a form that looks similar as the one shown below:



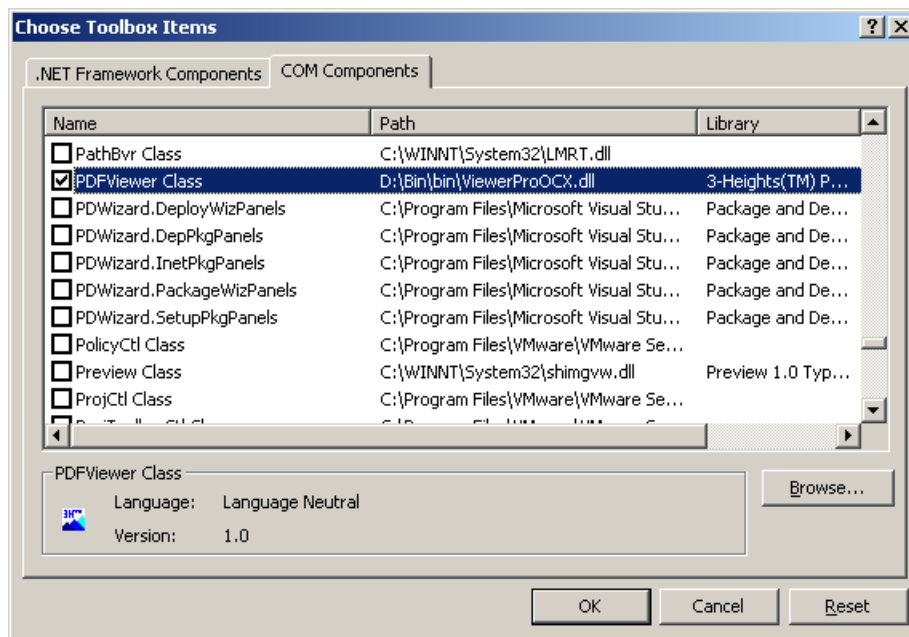
Now double-click the command button, add an open function and your PDF Viewer API is already completed. The open function is the first function described in the next chapter [User's Guide](#).

## 4.2 .NET

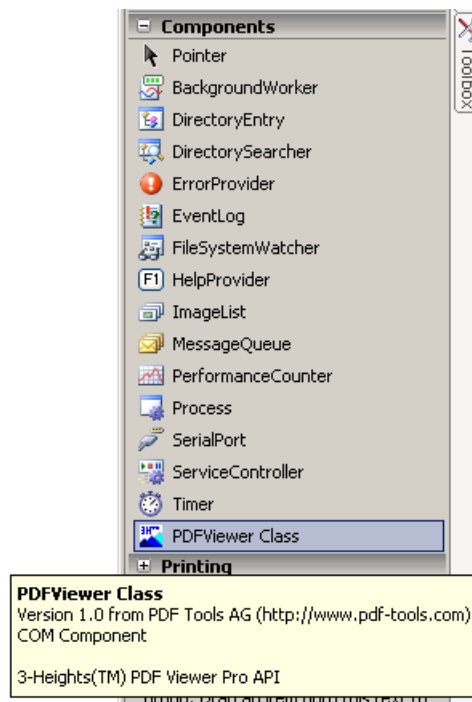
The 3-Heights™ PDF Viewer API does not provide a native .NET interface. In a .NET project the component is used through its COM interface.

Therefore the component first needs to be registered as described in the chapter [COM Interface](#).

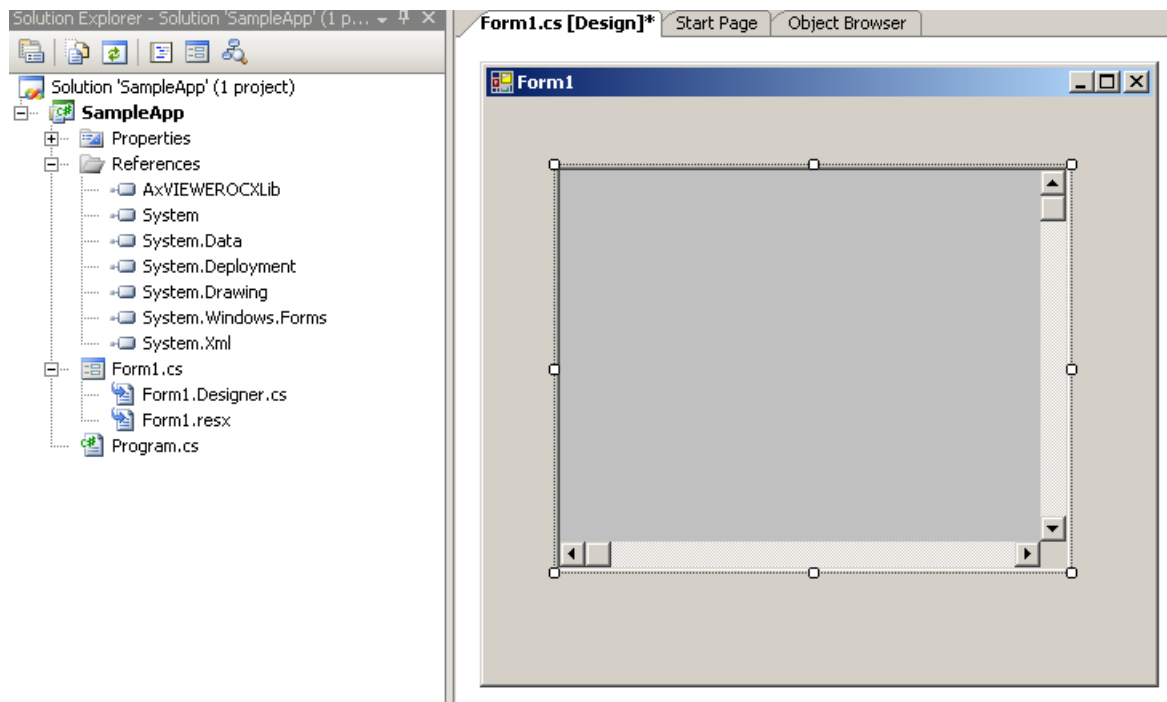
When using the component in a new project, add the PDFViewer Class to the Toolbox. To do so, right-click on the expanded MS Studio's Toolbox and select "Choose Items...". In the dialog select the COM component "PDF Viewer Class" as shown in the screenshot below:



After this step the PDFViewer Class is available in MS Studio's Toolbox as indicated in the screenshot below:



Use this class to draw a PDFViewer component onto a form. This step creates the COM-Interop DLL Interop.VIEWEROCXLib.dll and refers to it as AxVIEWEROCXLib in the project. The screenshot below shows this.



The `Interop.VIEWEROCXLib.dll` is automatically generated when launching an MS Studio project that uses the `PDFViewer` Class and `ViewerOCX.dll` is registered on that OS.

## 4.3 MFC

Please refer to the sample “mfcviewer”, which is created using the MFC wizard and is provided with the software package.

## 5 User's Guide

Most samples in this guide are written in Visual Basic. The call sequence, however, for any other programming language is the same.

### 5.1 Open a PDF Document

Documents can be opened either from file using the method [Open](#) or from memory using the method [OpenMem](#). A password is required and must be provided if the file is encrypted with a user password. For documents that have no user password set, use the default password - an empty string. Here is a simple VB example that shows how to open a PDF file from disk.

**Example:** Open PDF from File System.

```
private Sub Open_Click()  
    If Not ViewerControl.Open(App.Path & "\input.pdf", "") Then  
        MsgBox "Failed to open input file"  
    End If  
End Sub
```

ViewerControl is an object of type PDFViewer. Its name is defined by the name of the viewer control. The sample checks whether opening the input file was successful or not. If not, it shows a message box.

[OpenMem](#) is usually used when a PDF document is already available in memory, e.g. is read from a data base or is passed in-memory from another application.

The following example shows how a PDF document can be opened from memory by reading it from file and writing it in a byte array and then reading that byte array using [OpenMem](#).

**Example:** Open PDF from Memory.

```
Private Sub OpenMem_Click()  
    Dim bChar() As Byte  
    Dim lFileLenght As Long  
    Open App.Path & "\input.pdf" For Binary As #1  
    lFileLenght = LOF(1)  
    ReDim bChar(lFileLenght - 1)  
    Get #1, , bChar  
    Close #1  
    ViewerControl.OpenMem bChar, ""  
End Sub
```

Using [OpenMem](#) is especially useful if the file is already in memory. This can either happen when the application just created the file or if the same file is used multiple times.

### 5.2 Navigate

There are various ways how the user can navigate in a document with the viewer control. The application programmer has all options to enable, disable or limit the user's navigation. The following features can be used to navigate:

- Use the property [Page](#) to set page number to be displayed. The page range goes from 1 to [PageCount](#).
- Use the mouse in cursor mode [eCursorModeMove](#) (default) to scroll the page by pressing the left mouse button and moving the mouse up and down. Change the cursor mode using the property [CursorMode](#).



- Use horizontal and vertical scroll bars. Scroll bar functionality can be disabled using the property [ViewerOptions](#) and disable (AND NOT<sup>3</sup>) [eViewerOptionScrollbars](#).
- Use the mouse wheel to scroll. The mouse wheel can be disabled using the property [ViewerOptions](#) and enable (OR) [eViewerOptionDisableMouseWheel](#).
- Use the method [Goto](#) to move to a specific location and zoom level within the document.
- Use the outlines (bookmarks) or thumbnails from the navigation panel at left hand side of the control to random access a position.
  - The navigation panel can be disabled using the property [ViewerOptions](#) and disable [eViewerOptionPages](#) and [eViewerOptionOutlines](#) from it.
  - Enable [eViewerOptionPages](#), disable [eViewerOptionOutlines](#) to show thumbnails.
  - Enable [eViewerOptionOutlines](#), disable [eViewerOptionPages](#) to show outlines; if there are not outlines in the document, the navigation panel is not displayed.
- Internal and external links within the content of the PDF document can also be used to navigate.
  - Internal links (Goto) are disabled using [eViewerOptionDisableGoto](#).
  - External links (GotoR) are disabled using [eViewerOptionDisableGotoR](#).
  - URI are disabled using [eViewerOptionDisableURI](#).
- The search functions can indirectly be used to navigate.

## 5.3 Start a Print Job

A print job can be started using [BeginDocument](#). The parameter of [BeginDocument](#) is the name of the print job. To define the end the print job, use [EndDocument](#).

The method [PrintDocument](#) starts a print job, prints, and ends the print job. It cannot be combined with other calls, such as [BeginDocument](#) or [EndDocument](#).

### 5.3.1 What Is a Print Job?

A print job is a series of pages that are printed as one job. All pages of a print job are printed before the next print job starts.

The order in which individual print jobs are processed by the printer device is defined by the spooler. The order in which print jobs are created doesn't have to be the same order as they are printed. Small print jobs may receive higher priority and can overtake large print jobs. An exception to this are linked print jobs.

### 5.3.2 Individual Pages

The 3-Heights™ PDF Viewer API can print the document on display or a page thereof. It is not intended to be used for automated bulk printing.

The following sample shows how to open a printer, how to start and name a print job and how to print two copies. It assumes a document is already opened and displayed.

#### Example:

```
Private Sub Print_Click()
    ' replace the printer name with your printer name
    ViewerControl.OpenPrinter "HP LaserJet 4050 Series PS"
    ' start a new print job called "MyJob"
    ViewerControl.BeginDocument "MyJob"
```

<sup>3</sup> Options are enabled using bitwise OR and disabled using bitwise AND NOT. See also chapter [TPDFRendererOption](#).

```

' loop through the number of copies
NumberOfCopies = 2
For iCopies = 1 To NumberOfCopies
    ' loop through the pages
    For iPage = 1 To ViewerControl.PageCount
        ViewerControl.PrintPage iPage
    Next iPage
Next iCopies
' end the print job
ViewerControl.EndDocument
' close the printer and delete the temporary font files
ViewerControl.ClosePrinter
End Sub

```

### 5.3.3 Entire Document

The following sample shows how to print a document (multiple times). The advantage when with this sample is that the document has to be sent only once, whereas in the previous sample, the pages are sent multiple times, which lowers the performance.

#### Example:

```

Private Sub PrintDocument_Click()
    ' set amount of copies
    ViewerControl.Copies = 2
    ' print the document
    ViewerControl.PrintDocument "HP LaserJet 4050 Series PS", "MyJob"
End Sub

```

### 5.3.4 Print to the Default Printer

The Windows' default printer can be received using the method [GetDefaultPrinter](#). An alternative way to select the default printer is passing an empty string as the name of the printer.

#### Example:

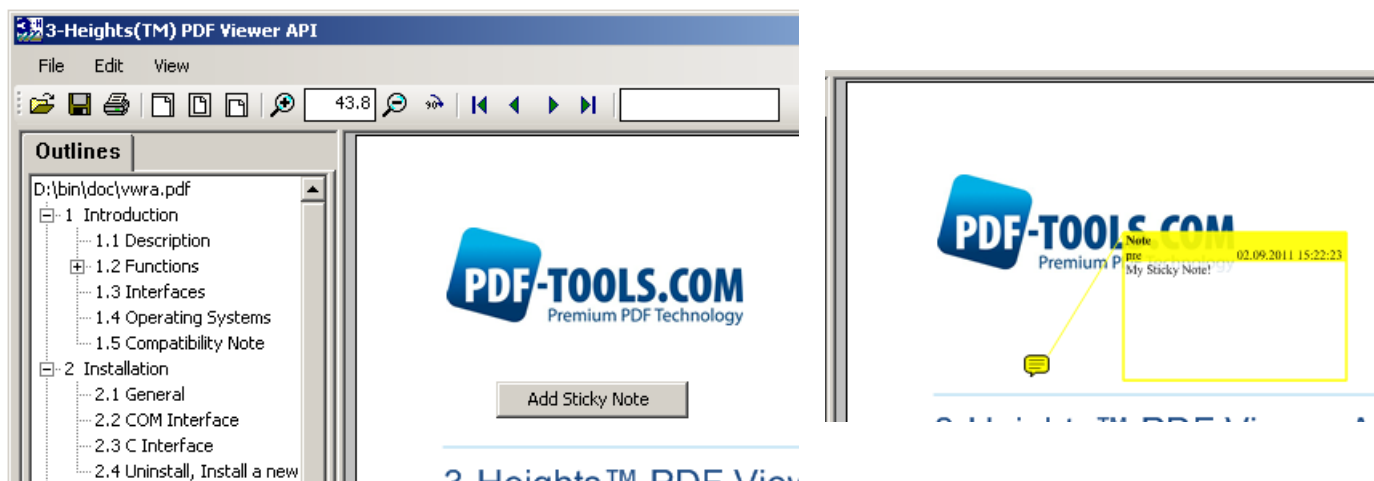
```

Private Sub PrintToDefaultPrinter_Click()
    ' 1st possibility to print to the default printer:
    PrinterName = ViewerControl.GetDefaultPrinter
    ViewerControl.PrintDocument PrinterName, "MyJob"
    ' 2nd possibility to print to the default printer:
    ViewerControl.PrintDocument "", "MyJob"
End Sub

```

## 5.4 Sticky Note Annotations

Use the control's property [ViewerOptions](#) to enable or disable a right-click context menu "Add Sticky Note" that allows a user to add a sticky note annotation.



The following C# code shows annotations are enabled and disabled.

```
// enable
PdfViewer1.ViewerOptions |=
(int)VIEWEROCXLib.TPDFViewerOption.eViewerOptionEnableAnnotEdit;
// disable
PdfViewer1.ViewerOptions &=
~(int)VIEWEROCXLib.TPDFViewerOption.eViewerOptionEnableAnnotEdit;
```

How to save annotations is described in the function [SaveAs](#).

## 5.5 How to use the Property Options

The property [Options](#) can be used to set various flags.

Options can be enabled (bitwise or) or disabled (bitwise and not). The default value is set to [eOptionTransparency](#) + [eOptionBanding](#) + [eOptionHighQuality](#) + [eOptionBilinear](#). To enable or disable a particular flag, a code like the sample below can be used. This will ensure resetting a flag does not change the values of other flags.

### Visual Basic 6 - Code Snippet:

```
'Enable Banding
printer.Options = printer.Options Or eOptionBanding
'Disable Banding
printer.Options = printer.Options And Not eOptionBanding
```

### C/C++ - Code Snippet:

```
int iOptions = PDFPrnGetOptions(pDocument);
// Enable Banding
PDFPrnSetOptions(pDocument, iOptions | eOptionBanding);
// Disable Banding
PDFPrnSetOptions(pDocument, iOptions & ~eOptionBanding);
```

### C# - Code Snippet:

```
// Enable Banding
printer.Options |= (int) eOptionBanding;
// Disable Banding
printer.Options &= ~(int) eOptionBanding;
```

## 5.6 Error Handling

Most methods of the 3-Heights™ PDF Viewer API can either succeed or fail depending on user input, state of the PDF Viewer 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 (!viewer.Open(file, password))
    {
        if (viewer.ErrorCode == PDFErrorCode.PDF_E_PASSWORD)
        {
            password = InputBox.Show("Password incorrect. Enter correct password:");
            return Open(file, password);
        }
        else
        {
            MessageBox.Show(String.Format(
                "Error {0}: {1}", viewer.ErrorCode, viewer.ErrorMessage));
            return false;
        }
    }
    [...]
}
```

## 6 Interface Reference

The following section lists all methods and properties of the COM interface of the 3-Heights™ PDF Viewer API.

### 6.1 PDFViewer Interface

#### 6.1.1 BeginDocument

**Method:** Boolean `BeginDocument(String DocumentName)`

Start a new printer job. All pages within one print job are printed successively, e.g. cannot be interrupted by another print job. The printer must be previously chosen with [OpenPrinter](#). During or before the beginning of the print job, a PDF or image document can be opened from file or memory and closed.

The end of the print job is marked with [EndDocument](#).

**Parameter:**

**DocumentName** [String] The name of the print job

**Returns:**

**True** Successfully connected to printer and started a print job.

**False** Otherwise.

#### 6.1.2 Border

**Property (get, set):** Long `Border`  
Default: **300**

The PDF Viewer control displays pages on top of a gray background. The border defines the width between the edge of the control and the page, as well as the distance between the pages in the continuous page mode. This property must be set before opening a document.

#### 6.1.3 Center

**Property (get, set):** Boolean `Center`  
Default: **False**

Set or get the center mode. When set to **True**, the document is horizontally and vertically centered on the page. When set to **False**, the document is printed to the upper left corner of the page.

## 6.1.4 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.

## 6.1.5 ClosePrinter

**Method:** Boolean `ClosePrinter()`

Close the connection to the printer. It deletes temporarily installed font files.

### Returns:

**True** The connection could successfully be closed.

**False** The connection could not be closed.

## 6.1.6 CompareDocuments

**Method:** Boolean `CompareDocuments(String LeftFileName, String LeftPassword, String RightFileName, String RightPassword)`

Compare two documents based on their rendered visual differences. The three shortcut commands Ctrl + Shift + S, Ctrl + Shift + D and Ctrl + Shift + F allow to switch between displaying only one of the two documents or a difference view.

### Parameters:

**LeftFileName** [String] Name or path of the left file to compare.

**LeftPassword** [String] Password of the left file. If the file is not password protected, one may use an empty string as this parameter.

**RightFileName** [String] Name or path of the right file to compare.

**RightPassword** [String] Password of the Right file. If the file is not password protected, one may use an empty string as this parameter.

### Returns:

**True** Successfully compared documents.

**False** Otherwise

## 6.1.7 ConvertPt

**Method:** Boolean `ConvertPt(Long Page, Single x, Single y)`

Convert document coordinate (for example returned by events) to page coordinates.

**0, 0** for document coordinates is the lower left corner of the first page. The borders are considered for document coordinates.

Page coordinates are raw (untransformed) user space coordinates of the PDF document. The units are PDF points (1 point = 1/72 inch, A4 = 595x842 points, Letter = 612x792 points). Page coordinates do not consider the border.

### Returns:

**True** Conversion successful.

**False** Otherwise.

### Parameters:

**Page** [Long] The page number

**x** [Single] The X coordinate

**y** [Single] The Y coordinate

## 6.1.8 Copies

**Property (get, set):** Integer `Copies`

Default: **-1**

Set or get the number of copies. This property should be used in combination with `PrintDocument`. If the value of this property is set to **-1**, the number of copies defined in the printer is applied.

## 6.1.9 CreateAnnotation

**Method:** Boolean `CreateAnnotation(TPDFAnnotation Type)`

Enable the user to create a new annotation of the specified type using the mouse. This allows an application to offer buttons, menu items or keyboard shortcuts to activate the annotation create mode, e.g. to create a sticky note. See also enumeration [TPDFAnnotation](#).

### 6.1.10 CursorMode

**Property (get, set):** [TPDFCursorMode](#) [CursorMode](#)  
Default: [eCursorModeMove](#)

Set or get the current cursor mode. There are five supported cursor modes. The default mode is the move mode. Most events can be caught in the mode [eCurserModeNoop](#). The [eCursorModeMark](#) can be used to catch the event method [MarkRectangle](#). See also enumeration [TPDFCursorMode](#).

### 6.1.11 DataType

**Property (get, set):** [String](#) [DataType](#)  
Default: `""`

Set or get the data type of the spool file. There are two valid data types: `"raw"` and `"emf"`.

**"raw"** `"raw"` is with respect to the printer language. E.g. if `"raw"` is used for a PCL printer, a PCL file is created and if `"raw"` is used for a Postscript printer, a Postscript file is created.  
For local printers the [DataType](#) should be set to `"raw"`.

**"emf"** If `"emf"` is used, an EMF<sup>4</sup> file is generated. The EMF file can be sent over a network and at its destination the (remote-) printer driver converts it to a raw file.  
For network printers `"emf"` should be used and it comes with the following two advantages:

- It takes less bandwidth to send the spool file over the network, because and EMF file is smaller than raw spool file.
- The workload is balanced: On the host where the Printer API resides, the EMF file is generated, on the host where the printer resides, the EMF file is converted to a raw file.

**""** If [DataType](#) is set to an empty string or [Nothing](#), then the data type is inherited from the printer's setting of the current user. In any situation where the current user settings are not well defined (e.g. IIS), the [DataType](#) should be set explicitly to either `"raw"` or `"emf"`. Some printer drivers will only allow the setting of additional options if the datatype is set to EMF or RAW explicitly. For these drivers use [Nothing](#).

### 6.1.12 DefaultSource

**Property (get, set):** [Integer](#) [DefaultSource](#)  
Default: `-1`



The default source defines from which input tray the paper shall be selected. For default values see [Paper Bins](#). There is no property to set the output paper tray. In order to set the output paper tray, use the device mode functionality.

### 6.1.13 DPI

**[Deprecated] Property (get):** Long [DPI](#)  
Default: **108**

Get the resolution in dots per inch (DPI). For accessing the DPI refer to [XDPI](#) and [YDPI](#).

### 6.1.14 Duplex

**Property (get, set):** Integer [Duplex](#)  
Default: **-1**

Set or get the duplex mode. For Windows default values, see [Duplex Modes](#). It is suggested to use the default values **1**, **2** and **3**.

|           |                     |
|-----------|---------------------|
| <b>-1</b> | Use printer default |
| <b>1</b>  | Simplex             |
| <b>2</b>  | Vertical Duplex     |
| <b>3</b>  | Horizontal Duplex   |

### 6.1.15 EndDocument

**Method:** Boolean [EndDocument\(\)](#)

Define the end of the printer job. After calling [EndDocument](#), the print job is no longer under the control of the 3-Heights™ PDF Viewer API.

When printing directly to a printer (i.e. not using a spooler), [EndDocument](#) means the entire print job is on the printer.

When using the spooler, [EndDocument](#) means the entire print job is in the queue of the spooler. It does not imply that it is already being printed.

#### Returns:

**True** The print job was submitted and the connection to the printer could successfully be closed.

**False** Otherwise.

## 6.1.16 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 Viewer 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 Viewer API.

## 6.1.17 FitMode

**Property (get, set):** `TPDFFitMode FitMode`  
Default: `eFitModeWidth`

The fit mode defines how the pages are displayed in the viewer component. The available fit modes are explained in the enumeration [TPDFFitMode](#). The fit mode can be altered at any time.

## 6.1.18 FitPage

**Property (get, set):** `Boolean FitPage`  
Default: `False`

The fit-page property defines how the PDF page should fit the paper size. Allowed values are:

- |              |  |
|--------------|--|
| <b>True</b>  | The page is resized so that both, page width and height fit on the printable part of the paper supported by the printer device. The ratio width to height remains unchanged. |
| <b>False</b> | The size of the page remains unchanged. If part of the content is outside the printable area (i.e. close to the border of the page) it will not be printed.                  |

## 6.1.19 GetDefaultPrinter

**Method:** `String GetDefaultPrinter()`

Return the name of the default printer, if there is a default printer installed on the system. If there is no default printer defined, it returns an empty string.

### Returns:

The name of the default printer.

## 6.1.20 GetPageRect

**Method:** Boolean GetPageRect(Long iPageNo, Variant varRect)

Get the size of a page in points.

Units: For PDF, 1 point = 1/72 inch; a page of size A4 portrait has normally the values (0, 0, 595, 842), for letter they are (0, 0, 612, 792). For image documents the resolution of the image is considered. If the image does not have a defined resolution, a default resolution of 96 DPI is assumed, which means 1 point = 1/96 inch.

### Returns:

**True** The page exists and the rectangle is filled.

**False** Otherwise.

### Parameters:

**iPageNo** [Long] The page number, page 1 is the first page in the document

**varRect** [Variant] The rectangle with the coordinates x1, y1, x2, y2, where the first two represent the position of the lower left corner, and the last two the position of the upper right corner of the page.

See also example in the function [Goto](#).

## 6.1.21 GoBack

**Method:** Boolean GoBack()

When viewing an embedded document, use the GoBack method in order to navigate back to the document in which the embedded document is contained.

### Returns:

**True** The parent document was opened successfully.

**False** Could not navigate back, e.g. because current document is not an embedded document or user canceled operation due to pending modifications.

## 6.1.22 Goto

**Method:** Boolean Goto(Long iPageNo, Single x, Single y, Single z)

Go to a page and position in the document and set the zoom factor.

### Parameters:

**iPageNo** [Long] The page number, page 1 is the first page in the document.

- x** [Single] The x-coordinate of the upper left corner.
- y** [Single] The y-coordinate of the upper left corner.
- z** [Single] The zoom factor; **1** is 100%, note this is different from the property **Zoom**, where **100** is 100%.

### Returns:

**True** The page and position was set successfully.

**False** The page or position does not exist.

**Example:** The following VB 6 sample goes to page 2, positions in the middle of the page and sets the zoom factor to half the page width. This means—depending on the shape of the viewer control—the lower right quadrant of page 2 is displayed.

```
' Get size of page 2
Dim r As Variant
If Not ViewerCtrl1.GetPageRect(2, r) Then
    MsgBox "Page 2 does not exist"
End If
' Get the zoom level required to display half the page width
ViewerCtrl1.Page = 2
ViewerCtrl1.FitMode = eFitModeWidth
Dim z As Double
z = ViewerCtrl1.Zoom * 2 / 100 '%'
' Set position to middle of 2nd page
If Not ViewerCtrl1.Goto(2, r(0)+(r(2)-r(0))/2, r(1)+(r(3)-r(1))/2, z) Then
    MsgBox "Position does not exist"
End If
```

## 6.1.23 LayoutMode

**Property (get, set):** TPDFLayoutMode LayoutMode  
Default: eLayoutDocument

The layout mode determines whether pages are displayed continuously or not. Consequently there are two available layout modes, these are: eLayoutModeDocument and eLayoutModePage, see [TPDFLayoutMode](#).

The layout mode may be changed whilst a document is being displayed. The layout mode has no impact on printing.

## 6.1.24 LicenseIsValid

**Property (get):** Boolean LicenseIsValid  
Static

Check if the license is valid.

## 6.1.25 MarkRectangle

**Method:** `Void MarkRectangle(Single X, Single Y, Single Width, Single Height, Long Color, Long Width)`

Define a rectangle at a specified location with a specified color on the foreground of the currently displayed document. For languages that support default parameters, the parameter for the color is optional.

The values of x and y define the location of the bottom left corner in raw (untransformed) PDF user space coordinates. The units are PDF points.<sup>5</sup> The value of w and h define the width and height of the rectangle.

### Parameters:

**X** [Single] The x-coordinate of the lower left corner

**Y** [Single] The y-coordinate of the lower left corner

**Width** [Single] The width of the rectangle in points

**Height** [Single] The height of the rectangle in points

**Color** [Long] The color of the marked rectangle as ARGB<sup>6</sup> value. An alpha value of 0 means the marked rectangle is transparent, an alpha value of 255 means it is opaque.

Compatibility Note: Previous version of the Viewer did not support an alpha channel.

**Width** [Long] (optional): This value defines the line width of the rectangle in points. If 0 is specified (default), the rectangle is filled.

Compatibility Note: In previous version of the Viewer, the line width had different units (1/35 points) or could not be set at all.

**Example:** Set the color to a semitransparent green, outlined rectangle.

```
ViewerControl.MarkRectangle 300, 300, 100, 100, &H7F00FF00, 6
```

## 6.1.26 Modified

**Property (get):** Boolean `Modified`  
Default: `False`

This property indicates whether the document contains unsaved changes, e.g. because its annotations have been edited.

<sup>5</sup> 1 point = 1/72 inch; A4 = 595 by 842 points; letter = 612 by 792 points

<sup>6</sup> Note that Windows uses the order alpha, blue, green, red for ARGB.

## 6.1.27 OffsetX, OffsetY

**Property (get, set):** Long `OffsetX`  
Default: 0

**Property (get, set):** Long `OffsetY`  
Default: 0

Set or get the X and Y-offset of the page on the paper. Units: 1/100 millimeters.

## 6.1.28 Open

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

Open a PDF file or raster image 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.

## 6.1.29 OpenMem

**Method:** Boolean `OpenMem(Variant MemBlock, String Password)`

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

### Parameters:

**MemBlock** [Variant] The memory block containing the PDF file given as a one dimensional byte array.

**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 document could be successfully opened.

**False** The document could not be opened, it is corrupt, or the password is not valid.

## 6.1.30 OpenMem2

```
Method: Boolean OpenMem2(Variant MemBlockPDF, Variant MemBlockFDF, String Password)
```

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

### Parameters:

**MemBlockPDF** [Variant] The memory block containing the PDF file given as a one dimensional byte array.

**MemBlockFDF** [Variant] The memory block containing the FDF file given as a one dimensional byte array.

**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 document could be successfully opened.

**False** The document could not be opened, it is corrupt, or the password is not valid.

## 6.1.31 OpenPrinter

```
Method: Boolean OpenPrinter(String PrinterName)
```

Open a printer. Available printers are listed in the "Printers" window of the Windows' "Control Panel".

### Parameter:

**PrinterName** [String] The name of the printer. The name is the same as shown on the Settings/Printer window, for example "HP LaserJet 4050 Series PS". It is not the same as the network name. Network printers could look like this: "\\PrinterServer\HP LaserJet 4250 PCL 6".

### Returns:

**True** The printer could be successfully opened.

**False** Otherwise.

### 6.1.32 Options

**Property (get, set):** TPDFRendererOption Options  
Default: eOptionBanding + eOptionBicubic + eOptionHighQuality

Set or get a specific rendering option.

Use bitwise "OR" to add an option.

Use bitwise "AND NOT" to remove an option.

For more information on the options available in the 3-Heights™ PDF Viewer API and how to use the this property please see [TPDFRendererOption](#).

### 6.1.33 Orientation

**Property (get, set):** Integer Orientation  
Default: -1

Set or get the orientation of the paper. Allowed values are:

|    |                               |
|----|-------------------------------|
| -2 | Automatic                     |
| -1 | Use printer setting (default) |
| 1  | Force portrait                |
| 2  | Force landscape               |

When not specified, the PDF Printer API uses the setting of the printer.

**Compatibility Note:** In older versions, the default was set to automatic, which places the page on the paper such that it fits best.

### 6.1.34 Page

**Property (get, set):** Long Page  
Default: 0

Set or get the first visible page in the currently opened document. This value is automatically updated by the control, to always return the first visible page. (**First** is in this case defined as in the page with the lowest page number)



### 6.1.35 PageAtPos

**Property (get):** Long PageAtPos

Default: 1

Get the page at a relative view port position (0: top, 1: bottom). This property can be used to detect which pages are displayed, if multiple pages are displayed and the property [Page](#) may not be accurate enough.

#### Example:

```
PageAtPos(0) = 4  
PageAtPos(0.25) = 5  
PageAtPos(0.5) = 5  
PageAtPos(0.75) = 5  
PageAtPos(1) = 6
```

The above values indicate that the pages 4, 5 and 6 are displayed. It is likely that only page number 5 is fully displayed. From page 4 the bottom, and from page 6 the top are displayed.

More precisely: At the very top of the control, page 4 is displayed (this is also returned by the property [Page](#)). However short after that (in the uppermost 25% of the control) page 5 starts. Page 5 is displayed at 25%, in the middle of the control and at 75%. At the bottom the top of page 6 is shown.

### 6.1.36 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.

### 6.1.37 PagesPerWindow

**Property (get, set):** Integer PagesPerWindow

Default: 1

The viewer control can display 1 or 2 pages per window next to each other. The default is 1.

### 6.1.38 PanX

**Property (get, set):** Long PanX

Default: 0

This value represents the position of the horizontal scrollbar. Its units are 1/100 millimeters.

It is 0 at the very left position. If the viewer control is wider than the width of the currently displayed page, then **PanX** becomes negative representing the distance between the left border of the current page and the left border of the viewer control.

### 6.1.39 PanY

**Property (get, set):** Long PanY  
Default: 0

This value represents the position of the vertical scrollbar. Its units are 1/100 millimeters.

Using **LayoutMode** = **eLayoutPage**, it is 0 at the top of the current page. Using **LayoutMode** = **eLayoutDocument**, it is 0 at the top of the first page of the document. If the viewer control is higher than the height of the currently displayed page, then **PanY** becomes negative representing the distance between the upper border of the page and the upper border of the viewer control.

### 6.1.40 PaperSize

**Property (get, set):** Integer PaperSize  
Default: -1

Get or set the paper size. The 118 Windows default paper sizes are listed in chapter [Paper Sizes](#).

Besides the Windows default paper sizes, you may also choose one of the following two codes:

- 1 User default paper size of used printer.
- 2 Select paper size automatically.

### 6.1.41 PrintDocument

**Method:** Boolean PrintDocument(String PrinterName, String DocumentName, String DataType)

Print the currently displayed document.

#### Parameters:

**PrinterName** [String] The name of the printer.

**DocumentName** [String] The name of the print job.

**DataType** [String] The data type of the spool file. It is suggested to use **"raw"** if printing to a local printer, and to use **"emf"** printing to a remote printer. See also property [DataType](#).

#### Returns:

**True** Successfully printed document.

**False** Otherwise.

## 6.1.42 PrintDocumentDlg

**Method:** Boolean `PrintDocumentDlg(String DocumentName)`

Print the currently displayed document. This will open a printing dialog window that allows the user to select the printer, page ranges and properties.

### Returns:

**True** Successfully printed document.

**False** Otherwise.

### Parameter:

**DocumentName** [String] The name of the print job.

## 6.1.43 PrintPage

**Method:** Boolean `PrintPage(Long PageNumber)`

Print a page of the currently opened document. The printer must be selected previously ([OpenPrinter](#)). This method must be called after [BeginDocument](#) and before [EndDocument](#).

### Parameter:

**PageNumber** [Long] This is the number of the page in the PDF file to be printed.

### Returns:

**True** The page was successfully printed.

**False** Otherwise.

## 6.1.44 RenderingMode

**Property (get, set):** TPDFRenderingMode `RenderingMode`  
Default: `eModeFast`

Set or get the rendering mode. The supported rendering modes are listed in the enumeration [TPDFRenderingMode](#). The default and recommended mode is `eModeFast`.

## 6.1.45 ProductVersion

**Property (get):** `String ProductVersion`

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

## 6.1.46 Rotate

**Property (get, set):** `Integer Rotate`

Default: `0`

Rotate pages, a positive number turns the page clockwise. The value must be a multiple of 90, i.e. valid values are `-270`, `-180`, `-90`, `0`, `90`, `180` and `270`. This property is reset to `0` when opening a document.

## 6.1.47 SaveAs

**Method:** `Boolean SaveAs(String FileName, String UserPw, String OwnerPw, TPDFPermission PermissionFlags, Long KeyLength)`

Save the currently opened document.

The target file format must be the same as the file format of the currently opened document. An exception is FDF: If a PDF document is opened, annotations (sticky notes) can be added. The result can be saved as PDF or FDF. If saved as FDF, it contains all annotations, not only those added by the 3-Heights™ PDF Viewer API, but also those contained in the original PDF document.

### Parameters:

**FileName** [`String`] The file name and optionally the file path, drive or server string according to the operating systems file name specification rules. When saving an FDF, this can be either a file path or an URL. When an URL is provided the FDF is uploaded to the server using HTTP put.

The file type is defined via file extension.

**UserPw** [`String`] (optional) Set the user password of the PDF document. If this parameter is omitted, the default password is used. Use `""` to set no password.

**OwnerPw** [`String`] (optional) Set the owner password of the PDF document. If this parameter is omitted, the default password is used. Use `""` to set no password.

**PermissionFlags** [`TPDFPermission`] (optional) The permission flags.

By default no encryption is used (`-1`). The permissions that can be granted are listed at the enumeration [TPDFPermission](#). To not encrypt the output document, set `PermissionFlags` to `ePermNoEncryption`, user and owner password to `""`. In order to allow high quality printing, flags `ePermPrint` and `ePermDigitalPrint` need to be set.

**KeyLength** [Long] (optional) Default set to **128**. With this option the length of the encryption key can be set. Supported are all values from **40** to **128** that are multiples of 8. The two most commonly used values are **40** and **128**.<sup>7</sup> The default value is calculated based on the selected permission flags.

#### Returns:

**True** The opened document could successfully be saved to file.

**False** Otherwise. One of the following occurred<sup>8</sup>:

- The output file cannot be created.
- **PDF\_E\_FILECREATE**: Failed to create the file.

## 6.1.48 ScaleXY

**Property (get, set):** Float **ScaleXY**  
Default: **1.0**

After the page has been scaled to fit the paper size an additional scaling can be specified by using this property. The scale factor is given in percent. A number less than 1 shrinks the page. A number greater than 1 expands the page. This property can optionally be combined with the **FitPage** property.

## 6.1.49 SearchFirst

**Method:** Boolean **SearchFirst**(String **SearchText**)

Mark the **SearchText** in the viewer if the string can be found.

#### Parameter:

**SearchText** [String] The string to be searched for. It may not contain blanks.

#### Returns:

**True** Search string was found.

**False** No occurrence of the search string was found.

<sup>7</sup> **128** bit requires PDF 1.4 or later (Acrobat 5). Acrobat only supports 40 and 128 bit encryption. Other tools, such as the 3-Heights™ tools also support other encryption key lengths.

<sup>8</sup> This is not a complete list. If **SaveAs** returns **False**, it is recommended to abort the processing of the file and log the error code and error message.

## 6.1.50 SearchNext

**Method:** Boolean `SearchNext()`

The method `SearchFirst` marks the first instance of the string, every subsequent call to `SearchNext` jumps to the next instance.

### Returns:

**True** A further occurrence of the search string was found.

**False** No further occurrence of the search string was found.

## 6.1.51 SetAnnotPropertiesStr

**Method:** Boolean `SetAnnotPropertiesStr()`

Set non-overridable default values for the creation of new annotations. For example, this is to ensure that a user can create annotations under his own name only.

## 6.1.52 SetCMSEngine

**Method:** Boolean `SetCMSEngine(String CMSEngine)`

Set the Color Management System (CMS) Engine. The following strings are supported:

**"None"** The algorithms specified in the PDF reference are used. This results in the maximum possible contrast.

**"Neugebauer"** The Neugebauer algorithm efficiently converts CMYK to RGB. It does not need any color profiles. The results, however, look similar to conversion using color profiles.

**"lcms"** (default): Use ICC color profiles. Default profiles are used for all unmanaged device color spaces as described in section [Color Profiles](#).

**FileName** Providing a file name, a configurable version of the Neugebauer algorithm is applied. The coefficients can be defined in the text file. The default Neugebauer coefficients are listed below (Red, Green, Blue; Color):

```
0.996078, 0.996078, 0.996078 ; White
0.000000, 0.686275, 0.937255 ; C
0.925490, 0.149020, 0.560784 ; M
1.000000, 0.949020, 0.066667 ; Y
0.215686, 0.203922, 0.207843 ; K
0.243137, 0.247059, 0.584314 ; CM
0.000000, 0.658824, 0.349020 ; CY
0.066667, 0.176471, 0.215686 ; CK
0.929412, 0.196078, 0.215686 ; MY
0.215686, 0.101961, 0.141176 ; MK
0.200000, 0.196078, 0.125490 ; YK
0.266667, 0.266667, 0.274510 ; CMY
```

```
0.133333, 0.098039, 0.160784 ; CMK  
0.074510, 0.180392, 0.133333 ; CYK  
0.215686, 0.121569, 0.113725 ; MYK  
0.125490, 0.121569, 0.121569 ; CMYK
```

The Neugebauer algorithm mixes the colors based on the amount of color and the corresponding weighted coefficient. Altering the values for a pure color specifically changes the result for this pure color.

The color transition remains smooth.

### 6.1.53 SetCursor

**Method:** Boolean SetCursor(TPDFCursorMode CursorMode, Long Cursor)

Set the cursor of a specific cursor mode.

#### Parameters:

**CursorMode** [TPDFCursorMode] The cursor mode, for which the new cursor should be set.

**Cursor** [Long] The new cursor may be any of the standard resource identifiers supported by the Windows LoadCursor function.

#### Example:

```
ViewerControl.SetCursor(/* eCursorModeMove */ 0, /* IDC_HAND */ 32512)  
ViewerControl.SetCursor(/* eCursorModeLink */ 4, /* IDC_ARROW */ 32649)
```

### 6.1.54 SetLicenseKey

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

Set the license key.

### 6.1.55 SetUILanguage

**Method:** Boolean SetUILanguage(String Language)

Set the language of the user interface of the viewer control.

Supported languages are:

| Languages |          |
|-----------|----------|
| en        | English  |
| de        | German   |
| ja        | Japanese |

By default, the locale of your application is used. If this is not available, the fallback is English.

## 6.1.56 SizeX, SizeY

**Property (get, set):** Long `SizeX`  
 Default: 0

**Property (get, set):** Long `SizeY`  
 Default: 0

Even though paper sizes can be set directly in millimeters using these properties, it is suggested to use the property `PaperSize` instead.

## 6.1.57 TreeViewWidth

**Property (get, set):** Long `TreeViewWidth`  
 Default: 200

The viewer control provides an optional navigation control on the left hand side. This navigation control can contain a tree view of the outlines<sup>9</sup> or a page list. The width of this navigation control can be set in pixel using this property.

## 6.1.58 ViewerOptions

**Property (get, set):** `TPDFViewerOptions` `ViewerOptions`  
 Default: `eViewerOptionScrollbars + eViewerOptionOutlines + eViewerOptionDoubleBuffer`

Get or set the viewing options. See enumeration `TPDFViewerOption`.

## 6.1.59 XDPI

**Property (get, set):** Single `XDPI`  
 Default: (calculated)

<sup>9</sup> Outlines are sometimes referred to as "Bookmarks".



Get or set the resolution in the X-axis in DPI.<sup>10</sup>

The default value is calculated and depends on the size of the monitor and resolution. In order for the value to be calculated correctly, it is required that the correct monitor driver is installed. On a 4:3 17 inch monitor with a resolution of 1280x1024, this value is approximately 102.<sup>11</sup>

A potential reason why this value could be calculated falsely is an incorrectly installed monitor driver.

## 6.1.60 YDPI

**Property (get, set):** Single YDPI  
Default: \*calculated\*

Get or set the resolution in the Y-axis in DPI.

The default value is calculated and depends on the size of the monitor and resolution. In order for the value to be calculated correctly, it is required that the correct monitor driver is installed. On a 4:3 17 inch monitor with a resolution of 1280x1024, this value is approximately 108.

See also [XDPI](#).

## 6.1.61 Zoom

**Property (get, set):** Single Zoom  
Default: 100

The zoom level defines how large the document is displayed with respect to its true size. In order to see a document at its true size at 100% zoom, it is required that the properties for [XDPI](#) and [YDPI](#) are correctly set.

## 6.2 Events

There is a series of events which can be caught using the corresponding events handlers. Mouse related events depend on the cursor mode. They will not fire in all cursor modes. In the [eCursorModeNoop](#) most will fire. [OnMarkRect](#) fires in [eCursorModeMark](#).

```
Event OnChar(Long iChar, Long iFlags)
```

```
Event OnGoto(String bstrType, Long pageNo, Single x, Single y, Single z)
Event OnGotoR(String bstrFileName)
Event OnGotoE(String bstrName, String bstrTmpName, String bstrType,
    Long pageNo, Single x, Single y, Single z)
```

```
Event OnKeyDown(Long iVirtKey, Long iFlags)
Event OnKeyUp(Long iVirtKey, Long iFlags)
```

<sup>10</sup> DPI = dots per inch

<sup>11</sup> Note that in Adobe Acrobat this is a fixed value of 96 for X and Y direction.

```
Event OnLButtonDblClk(Single x, Single y)
Event OnLButtonDown(Single x, Single y)
Event OnLButtonUp(Single x, Single y)
```

```
Event OnMarkRect(Single Left, Single Top, Single right, Single bottom)
```

```
Event OnMouseMove(Single x, Single y)
```

```
Event OnPage(Long iPage)
```

```
Event OnRButtonDblClk(Single x, Single y)
Event OnRButtonDown(Single x, Single y)
Event OnRButtonUp(Single x, Single y)
```

```
Event OnURI(Single bstrURI)
```

```
Event OnZoom(Single fZoom)
Event OnZoomRect(Single Left, Single Top, Single right, Single bottom)
```

```
Event OnCursorModeChanged(TPDFCursorMode oldCursorMode,
    TPDFCursorMode newCursorMode)
```

## 6.3 Enumerations

**Note:** Depending on the interface, enumerations may have “TPDF” as prefix (COM, C) or “PDF” as prefix (.NET) or no prefix at all (Java).

### 6.3.1 TPDFAnnotation Enumeration

TPDFAnnotation Table

| Text-Related Annotations |  |
|--------------------------|--|
| eAnnotationStickyNote    | This annotation type represents a “sticky note” attached to a coordinate on a page in the PDF document. When closed, the annotation appears as an icon; when open, it displays a pop-up element containing the text of the note. |

## TPDFAnnotation Table

|                                     |  |
|-------------------------------------|--|
| eAnnotationTextAnnotation           | A free-text annotation displays text directly on the page of a PDF document. Unlike a sticky-note annotation, a free text annotation has no open or closed state; instead of being displayed in a pop-up window, the text is always visible. |
| <b>Graphic Drawing Annotations</b>  |  |
| eAnnotationLine                     | A line annotation displays a single straight line on a page in a PDF document.   |
| eAnnotationSquare                   | A square annotation displays a single rectangle on a page in a PDF document.   |
| eAnnotationCircle                   | A square annotation displays a single circle or ellipse on a page in a PDF document.   |
| <b>Text Markup</b>                  |  |
| eAnnotationHighlight                | A highlight annotation underlays a rectangular area with a color.  |
| eAnnotationUnderline                | A underline annotation draws a single straight line under the text.  |
| <b>Standard Stamps</b>              |  |
| eAnnotationStampApproved            | Green stamp "APPROVED".  |
| eAnnotationStampNotApproved         | Red stamp "NOT APPROVED".  |
| eAnnotationStampDraft               | Blue stamp "DRAFT".  |
| eAnnotationStampFinal               | Green stamp "FINAL".   |
| eAnnotationStampCompleted           | Green stamp "COMPLETED".   |
| eAnnotationStampConfidential        | Blue stamp "CONFIDENTIAL".   |
| eAnnotationStampForPublicRelease    | Blue stamp "FOR PUBLIC RELEASE".   |
| eAnnotationStampNotForPublicRelease | Blue stamp "NOT FOR PUBLIC RELEASE".   |
| eAnnotationStampForComment          | Blue stamp "FOR COMMENT".  |
| eAnnotationStampVoid                | Red stamp "VOID".  |
| eAnnotationStampPreliminaryResults  | Blue stamp "PRELIMINARY RESULTS".  |
| eAnnotationStampInformationOnly     | Blue stamp "INFORMATION ONLY".   |

## 6.3.2 TPDFCursorMode Enumeration

**TPDFCursorMode Table**

| TPDFCursorMode               |  |
|------------------------------|--|
| <code>eCursorModeMove</code> | Allow using the mouse to scroll (default).                               |
| <code>eCursorModeZoom</code> | Mark rectangle and zoom into it.   |
| <code>eCursorModeMark</code> | Mark a rectangle, see method description <a href="#">MarkRectangle</a> . |
| <code>eCursorModeNoop</code> | No operation mode.   |
| <code>eCursorModeLink</code> | Allow clicking on links.   |

### 6.3.3 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 Viewer API, most of which are listed here:

**TPDFErrorCode Table**

| TPDFErrorCode  | Description   |
|--|---|
| <code>PDF_S_SUCCESS</code>                                       | The operation was completed successfully.   |
| <code>LIC_E_NOTSET</code> ,<br><code>LIC_E_NOTFOUND</code> , ... | Various license management related errors.  |
| <code>PDF_E_FILEOPEN</code>                                      | Failed to open the file.  |
| <code>PDF_E_FILECREATE</code>                                    | Failed to create the file.  |
| <code>PDF_E_PASSWORD</code>                                      | The authentication failed due to a wrong password.  |
| <code>PDF_E_UNKSECHANDLER</code>                                 | The file uses a proprietary security handler, e.g. for a proprietary digital rights management (DRM) system.  |
| <code>PDF_E_XFANEEDSRENDERING</code>                             | <p>The file contains unrendered XFA form fields, i.e. the file is an XFA and not a PDF file.</p> <p>The XFA (XML Forms Architecture) specification is referenced as an external document to ISO 32'000-1 (PDF 1.7) and has not yet been standardized by ISO. Technically spoken, an XFA form is included as a resource in a shell PDF. The PDF's page content is generated dynamically from the XFA data, which is a complex, non-standardized process. For this reason, XFA is forbidden by the ISO Standards ISO 19'005-2 (PDF/A-2) and ISO 32'000-2 (PDF 2.0) and newer.</p> |

## 6.3.4 TPDFFitMode Enumeration

TPDFFitMode Table

| TPDFFitMode        |  |
|--------------------|--|
| eFitModeActualSize | Display actual size.                     |
| eFitModePage       | Display whole page.                      |
| eFitModeWidth      | Display width of current page (default). |

## 6.3.5 TPDFLayoutMode Enumeration

TPDFLayoutMode Table

| TPDFLayoutMode  |                             |
|-----------------|-----------------------------|
| eLayoutPage     | Display one page at a time. |
| eLayoutDocument | Display all pages.          |

## 6.3.6 TPDFPermission Enumeration

An enumeration for permission flags. If a flag is set, the permission is granted.

TPDFPermission Table

| TPDFPermissionFlag       | Description   |
|--------------------------|---|
| ePermNoEncryption        | Do not apply encryption.<br><br>This enumeration value shall not be combined with other values. When using this enumeration set both passwords to an empty string or <b>Nothing</b> . |
| ePermNone                | Grant no permissions  |
| ePermPrint               | Low resolution printing   |
| ePermModify              | Changing the document   |
| ePermCopy                | Content copying or extraction   |
| ePermAnnotate            | Annotations   |
| ePermFillForms           | Filling of form fields  |
| ePermSupportDisabilities | Support for disabilities  |
| ePermAssemble            | Document assembly   |

#### TPDFPermission Table

|                   |                          |
|-------------------|--------------------------|
| ePermDigitalPrint | High resolution printing |
| ePermAll          | Grant all permissions    |

Changing permissions or combining multiple permissions is done using a bitwise “or” operator.

**Note:** The special value `ePermNoEncryption` cannot be combined with any other values.

Changing the current permissions in Visual Basic should be done like this:

Allow Printing

```
Permission = Permission Or ePermPrint
```

Prohibit Printing

```
Permission = Permission And Not ePermPrint
```

### 6.3.7 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 = PdfViewerGetViewerOptions(hViewer);
// Enable high quality rendering (anti-aliasing)
PdfViewerSetViewerOptions(hViewer, iOptions | eOptionHighQuality);
// Disable high quality rendering (anti-aliasing)
PdfViewerSetViewerOptions(hViewer, iOptions & ~eOptionHighQuality);
```

The following list includes renderer options that are relevant for the 3-Heights™ PDF Viewer 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. |
| eOptionBilinear         | A bilinear image filter is applied to images to improve the image quality. This option cannot be combined with eOptionBicubic.  |
| 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       | (default) Draw pop up windows of annotations, such as sticky notes.   |
| eOptionHighQuality      | (default) Anti-aliasing for text and path objects and filtering of image objects can be turned off and on with this option.   |
| eOptionNoEmbedded       | Do not use embedded fonts. Instead fonts from the operating system's font directory are used (%Systemroot%\fonts).  |
| eOptionOutlines         | Convert fonts into vector graphics.   |
| eOptionPreInstalled     | Replace embedded fonts with a pre-installed font if the same font is already installed on the OS.   |
| eOptionPrint            | 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.                                      |
| eOptionPrintOnlySig     | Draw the digital signature appearance only (without any status appearances, e.g. valid or invalid).   |
| eOptionDoNotPrintSig    | Do not draw digital signature appearances.  |
| eOptionTransparency     | Deprecated option that has no effect.   |
| eOptionTrueType         | CFF and Type1 fonts are converted to True Type fonts. This option overrides option eOptionType1.  |
| eOptionType1            | CFF fonts are converted to Type1 fonts.   |

**TPDFRendererOption Table**

|                      |   |
|----------------------|---|
| eOptionUseFastImages | 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. |
|----------------------|---|

### 6.3.8 TPDFRenderingMode Enumeration

**TPDFLayoutMode Table**

| TPDFRenderingMode |  |
|-------------------|--|
| eModeAccurate     | The accurate mode is intended for virtual printers such as a TIFF printer. It uses the Windows GDI+ for rendering. This mode allows for image filtering, sub-pixel rendering and anti-aliasing. It should not be applied for physical devices, such as a laser printer, due to the fact that those devices do not support the above features. Using the accurate mode creates generally larger spool files than the fast mode. |
| eModeDirect       | This mode is deprecated.   |
| eModeFast         | The fast mode is the recommended mode for printing to any physical printer device such as a laser printer, or ink jet printer. It uses the Windows GDI for rendering. This mode is generally faster and creates smaller spool files than the accurate mode. Use this mode for high resolution (600 DPI).   |

### 6.3.9 TPDFRotateMode Enumeration

**TPDFViewerOption Table**

| TPDFRotateMode   |  |
|------------------|--|
| eRotateAttribute | Set the rotation to the viewing rotation attribute of the PDF page, i.e. rendering the page with the same rotation as it is displayed in a PDF viewer. |
| eRotatePortrait  | Rotate page to portrait.   |
| eRotateLandscape | Rotate page to landscape.  |
| eRotateNone      | Process the page as it is saved in the pdf file.   |

### 6.3.10 TPDFViewerOption Enumeration

**TPDFViewerOption Table**

| TPDFViewerOption          |  |
|---------------------------|--|
| eViewerOptionDisableGoto  | Disable GoTo links (links within this document).   |
| eViewerOptionDisableGotoR | Disable GoToR links (links outside this document). |



**TPDFViewerOption Table**

|   |   |
|---|---|
| <code>eViewerOptionDisableURI</code>        | Disable URI links (web links).  |
| <code>eViewerOptionDisableLinks</code>      | Disable all types of interactive links within PDF document (GoTo, GoToR, URIs).   |
| <code>eViewerOptionDisableMouseWheel</code> | Disable mouse wheel for scrolling.  |
| <code>eViewerOptionEnableAnnotEdit</code>   | If this option is set the user interface offers commands to add, delete and edit text annotations (sticky notes).   |
| <code>eViewerOptionAnnotations</code>       | Enables pane for displaying and modifying annotations.  |
| <code>eViewerOptionHighlighting</code>      | Show the highlighting menu on the left hand side to allow for searching and marking different texts.  |
| <code>eViewerOptionScrollbars</code>        | Show scrollbars (default).  |
| <code>eViewerOptionOutlines</code>          | Show outlines (also known as Bookmarks) (default).  |
| <code>eViewerOptionPages</code>             | Show thumbnails. Each page is represented as a pre-view image. Enabling this option can reduce the performance. (Compatibility Note: Versions prior to 1.9.10.1 did not support displaying thumbnails, instead they displayed a page number tree.)  |
| <code>eViewerOptionDoubleBuffer</code>      | Enable double buffering. Double buffering is a technique used to minimize visible artifacts from the drawing process (flickering). As opposed to single buffering, the memory used for the displayed page is kept separately for writing and reading. This results in a better viewing quality but also requires more memory and CPU. |
| <code>eViewerOptionEmbeddedFiles</code>     | Show the list of embedded files on the left hand side.  |
| <code>eViewerOptionSaveEF</code>            | Allows the user to save embedded files.   |

# 7 Tips, Tricks, and Troubleshooting

## 7.1 Performance

The 3-Heights™ PDF Viewer API provides a variety of settings to tune the performance. In most cases a simple rule applies: Higher quality takes more resources (memory, CPU) and therefore lowers the performance. The following settings have an impact on the performance:

**Content of the PDF** A document with thousands of objects requires more time for rendering than a page with plain text.

**Resolution** The higher the size of the viewer window the more pixels need to be drawn and the lower the performance.

**Double Buffering** Enabling double buffering requires extra memory and CPU.

**Rendering Mode** Disabling the accurate rendering mode improves the performance at the cost of visual quality. It should only be applied at high zoom levels or high resolutions.

**Filters** The filters for “Bilinear”, “Bicubic” and “HighQuality” require extra CPU.

**Thumbnails** Displaying thumbnails reduces the performance. Use outlines instead.

## 7.2 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 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](#).
  2. See Section [Handle Embedded Fonts](#).

### 7.2.1 Handle Non-Embedded Fonts

#### Viewing vs Printing

First and foremost, one needs to distinguish between incorrect fonts when viewing the document in the viewer control and incorrect fonts when printing to paper using the print functionality of the viewer control. This chapter is about incorrect fonts when viewing the document. If you have correct fonts when viewing, but incorrect fonts when printing, the best choice is probably to update or alter the printer driver.

#### 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 font<sup>12</sup> 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 `eOptionNoEmbedded` is used then the glyphs of the fonts are converted to either bitmaps or outlines<sup>13</sup>. If `eOptionNoOutlines` 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.

## 7.2.2 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 `eOptionNoOutlines` 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.

## 7.3 Transparency

The 3-Heights™ rendering engine supports transparency functions such as a number of blend modes as well as isolated and non-isolated transparency groups, but not transparency in general.

Certain types of tiling and shading patterns may not correctly be reproduced by the rendering engine.

---

<sup>12</sup> e.g. Times-Roman, Helvetica, Courier

<sup>13</sup> The outline of a glyph is a vector graphic without any reference to the original font program.

## 8 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.

### 8.1 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.

### 8.2 Changes in Version 4.12

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

### 8.3 Changes in Version 4.11

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

### 8.4 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.
- [C] **Clarified** Error handling of `TPdfStreamDescriptor` functions.

### 8.5 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.
- [C] **Changed** return value `pfGetLength` of `TPDFStreamDescriptor` to `pos_t`<sup>14</sup>.
- [Java] **New** method to set the language of internal strings to a provided locale (e.g. context menu entries): `setLanguage()`. Currently supported languages: English ('en-US'), German ('de-DE') and French ('fr-FR')
- [Java] **New** method to open a file with a byte buffer: `openMem()`
- [Java] **New** method to open a file with a stream: `openStream()`

<sup>14</sup> This has no effect on neither the .NET, Java, nor COM API

- [Java] **New** method to open a PDF file together with an FDF file: `openMem2()`
- [Java] **New** method to save file in a stream: `saveToMemory()`

## 8.6 Changes in Version 4.8

- **Improved** creation of annotation appearances to use less memory and processing time.
- **Added** repair functionality for TrueType font programs whose glyphs are not ordered correctly.
- [Java] **New** function to create sticky (`eAnnotationText`), highlight (`eAnnotationHighlight`) and freehand (`eAnnotationInk`) annotations. `createAnnotation(TPdfAnnotationType type, int page, Double[] rect)`
- [Java] **New** viewer functionality to move and delete annotations.
- [Java] **New** function to save a document.
- [.NET] **New** field `PageOrder` allows to change the display order of the document pages in the viewer. This is only for viewing purposes and not stored in the document.
- [.NET, C, Java] **New** property `ProductVersion` to identify the product version.
- [.NET] **Deprecated** method `GetLicenseIsValid`.
- [.NET] **New** property `LicenseIsValid`.

## 9 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 Viewer 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](mailto:pdfsales@pdf-tools.com)

# A Default Values

## A.1 Duplex Modes

|   |                   |
|---|-------------------|
| 1 | Simplex           |
| 2 | Vertical Duplex   |
| 3 | Horizontal Duplex |

## A.2 Paper Bins

|    |                 |
|----|-----------------|
| 1  | Upper           |
| 2  | Lower           |
| 3  | Middle          |
| 4  | Manual          |
| 5  | Envelope        |
| 6  | Envelope Manual |
| 7  | Auto            |
| 8  | Tractor         |
| 9  | Small FMT       |
| 10 | Large FMT       |
| 11 | Large Capacity  |
| 12 | undef.          |
| 13 | undef.          |
| 14 | Cassette        |
| 15 | From Source     |

## A.3 Paper Sizes

|    |                              |
|----|------------------------------|
| 1  | Letter 8 1/2 x 11 in         |
| 2  | Letter Small 8 1/2 x 11 in   |
| 3  | Tabloid 11 x 17 in           |
| 4  | Ledger 17 x 11 in            |
| 5  | Legal 8 1/2 x 14 in          |
| 6  | Statement 5 1/2 x 8 1/2 in   |
| 7  | Executive 7 1/4 x 10 1/2 in  |
| 8  | A3 297 x 420 mm              |
| 9  | A3 297 x 420 mm              |
| 10 | A4 Small 210 x 297 mm        |
| 11 | A5 148 x 210 mm              |
| 12 | B4 (JIS) 250 x 354           |
| 13 | B5(JIS)182x257mm             |
| 14 | Folio 8 1/2 x 13 in          |
| 15 | Quarto 215 x 275 mm          |
| 16 | 10x14 in                     |
| 17 | 11x17 in                     |
| 18 | Note 8 1/2 x 11 in           |
| 19 | Envelope # 9 3 7/8 x 8 7/8   |
| 20 | Envelope # 10 4 1/8 x 9 1/2  |
| 21 | Envelope # 11 4 1/2 x 10 3/8 |
| 22 | Envelope # 12 4 1/2 x 11     |
| 23 | Envelope # 14 5 x 11 1/2     |
| 24 | C size sheet                 |
| 25 | D size sheet                 |
| 26 | E size sheet                 |
| 27 | Envelope DL 110 x 220mm      |



|    |                                    |
|----|------------------------------------|
| 28 | Envelope C5 162 x 229 mm           |
| 29 | Envelope C3 324 x 458 mm           |
| 30 | Envelope C4 229 x 324 mm           |
| 31 | Envelope C6 114 x 162 mm           |
| 32 | Envelope C65 114 x 229 mm          |
| 33 | Envelope B4 250 x 353 mm           |
| 34 | Envelope B5 176 x 250 mm           |
| 35 | Envelope B6 176 x 125 mm           |
| 36 | Envelope 110 x 230 mm              |
| 37 | Envelope Monarch 3.875 x 7.5 in    |
| 38 | 63/4Envelope35/8x61/2in            |
| 39 | USStdFanfold147/8x11in             |
| 40 | German Std Fanfold 8 1/2 x 12 in   |
| 41 | German Legal Fanfold 8 1/2 x 13 in |
| 42 | B4 (ISO) 250 x 353 mm              |
| 43 | Japanese Postcard 100 x 148 mm     |
| 44 | 9 x 11 in                          |
| 45 | 10 x 11 in                         |
| 46 | 15 x 11 in                         |
| 47 | Envelope Invite 220 x 220 mm       |
| 48 | RESERVED—DO NOT USE                |
| 49 | RESERVED—DO NOT USE                |
| 50 | Letter Extra 9 \275 x 12 in        |
| 51 | Legal Extra 9 \275 x 15 in         |
| 52 | Tabloid Extra 11.69 x 18 in        |
| 53 | A4 Extra 9.27 x 12.69 in           |
| 54 | Letter Transverse 8 \275 x 11      |

|    |  |
|----|--|
| 55 | A4 Transverse 210 x 297 mm             |
| 56 | Letter Extra Transverse 9\275          |
| 57 | SuperA/SuperA/A4 227 x 356             |
| 58 | SuperB/SuperB/A3 305 x 487             |
| 59 | Letter Plus 8.5 x 12.69 in             |
| 60 | A4 Plus 210 x 330 mm                   |
| 61 | A5 Transverse 148 x 210 mm             |
| 62 | B5 (JIS) Transverse 182 x 257 mm       |
| 63 | A3 Extra 322 x 445 mm                  |
| 64 | A5 Extra 174 x 235 mm                  |
| 65 | B5 (ISO) Extra 201 x 276 mm            |
| 66 | A2 420 x 594 mm                        |
| 67 | A3 Transverse 297 x 420 mm             |
| 68 | A3 Extra Transverse 322 x 445 mm       |
| 69 | Japanese Double Postcard 200 x 148 mm  |
| 70 | A6 105 x 148 mm                        |
| 71 | Japanese Envelope Kaku # 2             |
| 72 | Japanese Envelope Kaku # 3             |
| 73 | Japanese Envelope Chou # 3             |
| 74 | Japanese Envelope Chou # 4             |
| 75 | Letter Rotated 11 x 8 1/2 11 in        |
| 76 | A3 Rotated 420 x 297 mm                |
| 77 | A4 Rotated 297 x 210 mm                |
| 78 | A5 Rotated 210 x 148 mm                |
| 79 | B4 (JIS) Rotated 364 x 257 mm          |
| 80 | B5 (JIS) Rotated 257 x 182 mm          |
| 81 | Japanese Postcard Rotated 148 x 100 mm |

|     |   |
|-----|---|
| 82  | Double Japanese Postcard Rotated 148 x 200 mm         |
| 83  | A6 Rotated 148 x 105 mm                               |
| 84  | Japanese Envelope Kaku # 2 Rotated                    |
| 85  | Japanese Envelope Kaku # 3 Rotated                    |
| 86  | Japanese Envelope Chou # 3 Rotated                    |
| 87  | Japanese Envelope Chou # 4 Rotated 88B6(JIS)128x182mm |
| 89  | B6 (JIS) Rotated 182 x 128 mm                         |
| 90  | 12x11in   |
| 91  | Japanese Envelope You # 4                             |
| 92  | Japanese Envelope You # 4 Rotated                     |
| 93  | PRC 16K 146 x 215 mm                                  |
| 94  | PRC 32K 97 x 151 mm                                   |
| 95  | PRC 32K(Big) 97 x 151 mm                              |
| 96  | PRC Envelope # 1 102 x 165 mm                         |
| 97  | PRC Envelope # 2 102 x 176 mm                         |
| 98  | PRC Envelope # 3 125 x 176 mm                         |
| 99  | PRC Envelope # 4 110 x 208 mm                         |
| 100 | PRC Envelope # 5 110 x 220 mm                         |
| 101 | PRC Envelope # 6 120 x 230 mm                         |
| 102 | PRC Envelope # 7 160 x 230 mm                         |
| 103 | PRC Envelope # 8 120 x 309 mm                         |
| 104 | PRC Envelope # 9 229 x 324 mm                         |
| 105 | PRC Envelope # 10 324 x 458 mm                        |
| 106 | PRC 16K Rotated                                       |
| 107 | PRC 32K Rotated                                       |
| 108 | PRC 32K(Big) Rotated                                  |
| 109 | PRC Envelope # 1 Rotated 165 x 102 mm                 |

|     |  |
|-----|--|
| 110 | PRC Envelope # 2 Rotated 176 x 102 mm  |
| 111 | PRC Envelope # 3 Rotated 176 x 125 mm  |
| 112 | PRC Envelope # 4 Rotated 208 x 110 mm  |
| 113 | PRC Envelope # 5 Rotated 220 x 110 mm  |
| 114 | PRC Envelope # 6 Rotated 230 x 120 mm  |
| 115 | PRC Envelope # 7 Rotated 230 x 160 mm  |
| 116 | PRC Envelope # 8 Rotated 309 x 120 mm  |
| 117 | PRC Envelope # 9 Rotated 324 x 229 mm  |
| 118 | PRC Envelope # 10 Rotated 458 x 324 mm |