Contents

1 Introduction .................................................................................................................. 6
  1.1 Description ............................................................................................................. 6
  1.2 Functions .............................................................................................................. 6
  1.2.1 Features ........................................................................................................... 6
  1.2.2 Formats ........................................................................................................... 7
  1.2.3 Conformance .................................................................................................... 7
  1.3 Interfaces .............................................................................................................. 7
  1.4 Operating Systems ............................................................................................... 7

2 Installation and Deployment ......................................................................................... 8
  2.1 Windows ............................................................................................................... 8
  2.2 Zip Archive .......................................................................................................... 8
  2.2.1 Development ................................................................................................... 9
  2.2.2 Deployment ..................................................................................................... 10
  2.3 NuGet Package .................................................................................................... 11
  2.4 Interface Specific Installation Steps ..................................................................... 11
    2.4.1 Java Interface ................................................................................................. 11
    2.4.2 .NET Interface ............................................................................................... 12
    2.4.3 C Interface .................................................................................................... 12
  2.5 Uninstall, Install a New Version .......................................................................... 12
  2.6 Note about the Evaluation License ...................................................................... 12

3 License Management ..................................................................................................... 13
  3.1 License Installation and Management .................................................................. 13
    3.1.1 Graphical License Manager Tool .................................................................. 13
    3.1.1.1 List all installed license keys ................................................................. 13
    3.1.1.2 Add and delete license keys ................................................................. 13
    3.1.1.3 Display the properties of a license ..................................................... 13
    3.1.2 Command Line License Manager Tool ....................................................... 14
    3.1.2.1 List all installed license keys ................................................................. 14
    3.1.2.2 Add and delete license keys ................................................................. 14
    3.1.2.3 Display the properties of a license ..................................................... 14
  3.2 License Selection and Precedence ........................................................................ 15
    3.2.1 Selection ....................................................................................................... 15
    3.2.2 Precedence .................................................................................................... 15
  3.3 Key Update .......................................................................................................... 16
  3.4 License activation ................................................................................................. 16
    3.4.1 Activation ..................................................................................................... 16
    3.4.2 Reactivation .................................................................................................. 17
    3.4.3 Deactivation ................................................................................................. 17
  3.5 Proxy Setting ........................................................................................................ 17
  3.6 Offline Usage ....................................................................................................... 18
    3.6.1 First Step: Create a Request File ............................................................... 18
    3.6.2 Second Step: Use Form on Website ............................................................ 19
    3.6.3 Third Step: Apply the Response File ......................................................... 19
  3.7 License Key Versions .......................................................................................... 19
  3.8 License Key Storage ............................................................................................ 19
    3.8.1 Windows ...................................................................................................... 19
  3.9 Troubleshooting ................................................................................................. 20
License key cannot be installed ................................................................. 20
License is not visible in license manager .................................................. 20
License is not found at runtime ................................................................. 20
Eval watermark is displayed where it should not ....................................... 21
 Activation is not recognized ................................................................. 21
Activation is invalidated too often ............................................................ 22
Connection to the licensing service fails .................................................. 22
Offline usage fails due to a request/response mismatch ............................ 22

4 Getting Started ....................................................................................... 24
4.1 Mixed Raster Content (MRC) .............................................................. 24
4.1.1 Phase 1: Recognizing Photographic Pictures ................................ 24
4.1.2 Phase 2: Segmentation into Layers ............................................... 24
4.1.3 Phase 3: PDF Construction ............................................................ 25
4.2 Internal Engine ................................................................................... 25
4.2.1 RecognizeBlankPages ................................................................. 25
4.2.2 BlankPageMargin ......................................................................... 25
4.2.3 DisableMaskEmbedding .............................................................. 26
4.2.4 RecognizePictures ....................................................................... 26
4.3 Garbage collection and closing objects ............................................. 26

5 Programming Interfaces .......................................................................... 27
5.1 .NET Interface .................................................................................... 27
5.1.1 IDisposable Objects ..................................................................... 27
5.1.2 Streams ....................................................................................... 27
5.1.3 Error handling ........................................................................... 27
5.1.4 Lists ............................................................................................ 27
5.2 Java Interface ..................................................................................... 27
5.2.1 AutoCloseable Objects ............................................................... 27
5.2.2 Properties .................................................................................. 28
5.2.3 Error handling ........................................................................... 28
5.2.4 Streams ..................................................................................... 28
5.2.5 Lists .......................................................................................... 28
5.3 C Interface ........................................................................................ 28
5.3.1 Namespaces, classes and methods .............................................. 28
5.3.2 Library Initialization .................................................................... 28
5.3.3 Objects ....................................................................................... 29
5.3.4 Properties .................................................................................. 29
5.3.5 Error handling ........................................................................... 29
5.3.6 Strings ....................................................................................... 29
5.3.7 Stream ....................................................................................... 30
5.3.8 Lists .......................................................................................... 30
5.3.9 List Interface .............................................................................. 30
5.3.10 Count ....................................................................................... 30
5.3.11 Get .......................................................................................... 30
5.3.12 Append ..................................................................................... 30

© PDF Tools AG – Premium PDF Technology
# Interface Reference

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Common methods</td>
<td>31</td>
</tr>
<tr>
<td>6.1.1</td>
<td>CheckLicense</td>
<td>31</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Close</td>
<td>31</td>
</tr>
<tr>
<td>6.1.3</td>
<td>LicenseKey</td>
<td>31</td>
</tr>
<tr>
<td>6.1.4</td>
<td>ProductVersion</td>
<td>32</td>
</tr>
<tr>
<td>6.2</td>
<td>Converter Interface</td>
<td>32</td>
</tr>
<tr>
<td>6.2.1</td>
<td>CreatePdf</td>
<td>32</td>
</tr>
<tr>
<td>6.2.2</td>
<td>AddTiff</td>
<td>33</td>
</tr>
<tr>
<td>6.2.3</td>
<td>SetXmp</td>
<td>33</td>
</tr>
<tr>
<td>6.2.4</td>
<td>SetOutputIntent</td>
<td>34</td>
</tr>
<tr>
<td>6.2.5</td>
<td>SetDefaultColorSpace</td>
<td>34</td>
</tr>
<tr>
<td>6.2.6</td>
<td>ClosePDF</td>
<td>35</td>
</tr>
<tr>
<td>6.3</td>
<td>Compressor Interface</td>
<td>35</td>
</tr>
<tr>
<td>6.3.1</td>
<td>OpenTiff</td>
<td>35</td>
</tr>
<tr>
<td>6.3.2</td>
<td>SaveTiff</td>
<td>36</td>
</tr>
<tr>
<td>6.3.3</td>
<td>CloseTiff</td>
<td>36</td>
</tr>
<tr>
<td>6.4</td>
<td>Recognizer Interface</td>
<td>37</td>
</tr>
<tr>
<td>6.4.1</td>
<td>OpenTiff</td>
<td>37</td>
</tr>
<tr>
<td>6.4.2</td>
<td>SaveTiff</td>
<td>37</td>
</tr>
<tr>
<td>6.4.3</td>
<td>OcrEngines</td>
<td>38</td>
</tr>
<tr>
<td>6.4.4</td>
<td>CloseTiff</td>
<td>38</td>
</tr>
<tr>
<td>6.5</td>
<td>Importer Interface</td>
<td>39</td>
</tr>
<tr>
<td>6.5.1</td>
<td>CreateTiff</td>
<td>39</td>
</tr>
<tr>
<td>6.5.2</td>
<td>AddImg</td>
<td>39</td>
</tr>
<tr>
<td>6.5.3</td>
<td>CloseTiff</td>
<td>40</td>
</tr>
<tr>
<td>6.6</td>
<td>Merger Interface</td>
<td>40</td>
</tr>
<tr>
<td>6.6.1</td>
<td>CreateTiff</td>
<td>40</td>
</tr>
<tr>
<td>6.6.2</td>
<td>AddTiff</td>
<td>41</td>
</tr>
<tr>
<td>6.6.3</td>
<td>CloseTiff</td>
<td>41</td>
</tr>
<tr>
<td>6.7</td>
<td>Splitter Interface</td>
<td>42</td>
</tr>
<tr>
<td>6.7.1</td>
<td>OpenTiff</td>
<td>42</td>
</tr>
<tr>
<td>6.7.2</td>
<td>PageCount</td>
<td>42</td>
</tr>
<tr>
<td>6.7.3</td>
<td>SavePages</td>
<td>42</td>
</tr>
<tr>
<td>6.7.4</td>
<td>CloseTiff</td>
<td>43</td>
</tr>
<tr>
<td>6.8</td>
<td>Scanner Interface</td>
<td>43</td>
</tr>
<tr>
<td>6.8.1</td>
<td>SetSource</td>
<td>43</td>
</tr>
<tr>
<td>6.8.2</td>
<td>ShowUI</td>
<td>44</td>
</tr>
<tr>
<td>6.8.3</td>
<td>Scan</td>
<td>44</td>
</tr>
<tr>
<td>6.8.4</td>
<td>Capabilities</td>
<td>45</td>
</tr>
<tr>
<td>6.8.5</td>
<td>Sources</td>
<td>45</td>
</tr>
<tr>
<td>6.9</td>
<td>StringList Interface</td>
<td>45</td>
</tr>
<tr>
<td>6.9.1</td>
<td>Get</td>
<td>45</td>
</tr>
<tr>
<td>6.9.2</td>
<td>Size</td>
<td>45</td>
</tr>
<tr>
<td>6.10</td>
<td>Structures</td>
<td>46</td>
</tr>
<tr>
<td>6.10.1</td>
<td>ConverterParameters Struct</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>Resolution</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>OptionalContentGroup</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>IgnoreOcr</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>Conformance</td>
<td>46</td>
</tr>
<tr>
<td>6.10.2</td>
<td>CompressorParameters Struct</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>BinaryCompression</td>
<td>46</td>
</tr>
</tbody>
</table>
GrayscaleCompression .......................... 47
LabCompression ............................. 47
PalettedCompression ......................... 47
Rgb_YcbcrCompression ....................... 47
CmykCompression ............................. 47
MrcCompression ............................. 47
BinaryCompQuality ......................... 47
GrayscaleCompQuality ....................... 47
LabCompQuality ............................. 47
Rgb_YcbcrCompQuality ....................... 47
CmykCompQuality ............................. 47
MrcCompQuality ............................. 47
BinaryDownsampling ......................... 47
GrayscaleDownsampling ...................... 47
LabDownsampling ........................... 47
Rgb_YcbcrDownsampling ...................... 47
CmykDownsampling ........................... 47
MrcDownsampling ........................... 47
Recompress ................................. 47
PerformMrc ................................. 47
BinarizationThreshold ....................... 48
UpgradeJpeg ................................. 48

6.10.3 RecognizerParameters Struct ................. 49
OcrEngineName ............................... 49
OcrParameters ............................... 49
OcrLanguages ................................. 50
BinarizationPriorOcr ......................... 50
BinarizationThreshold ....................... 50

6.10.4 ImporterParameters Struct .................. 50
Resolution .................................. 50
UpgradeJpeg ................................. 50

6.10.5 MergerParameters Struct .................... 51
RemoveBlankPages ........................... 51

6.10.6 SplitterParameters Struct .................... 52
PageNumberStart ............................. 52
PageNumberEnd ............................... 52

6.11 Enumerations ................................ 52
6.11.1 Conformance Enumeration ................... 52
6.11.2 BoolEnum Enumeration ..................... 52
6.11.3 Compression Enumeration ................... 53
6.11.4 ErrorCode Enumeration ..................... 53
    Logic errors ................................ 53
    Environmental errors ....................... 53

7 Version History ................................ 55
7.1 Changes in Version 6 ......................... 55
7.2 Changes in Version 5 ......................... 55
7.3 Changes in Version 4.12 ....................... 55
7.4 Changes in Version 4.11 ....................... 55
7.5 Changes in Version 4.10 ....................... 55
7.6 Changes in Version 4.9 ......................... 55
7.7 Changes in Version 4.8 ......................... 55
1 Introduction

1.1 Description

The 3-Heights™ TIFF Toolbox API enables to process and convert TIFF files. Examples of such processing functions are the merging and splitting of multi-page files, the conversion of other raster image formats such as JPEG into TIFF and the conversion of TIFF files into PDF/A. The optional OCR add-in makes output files searchable in full text mode.

PDF/A has been acknowledged world-wide as the ISO standard for long-term archiving since 2005. The TIFF to PDF/A converter is used to convert images into a standardized format, for instance for electronic archiving or electronic data exchange.

The 3-Heights™ TIFF Toolbox API is characterized by a robust design, high throughput, ability to process large files and accurate image reproduction.

1.2 Functions

The 3-Heights™ TIFF Toolbox API consists of the following interfaces

- **Converter** Convert a multi-page TIFF file into a PDF/A-1 or PDF/A-2 file.

- **Compressor** Re-compress a multi-page TIFF file using the selected algorithms. Split each page of a multi-page TIFF file with the Mixed Raster Content (MRC) technique into foreground, background and mask layer, compress these layers, sample-down the foreground and background.

- **Importer** Import raster image formats such as JPEG into a TIFF file.

- **Merger** Merge single- or multi-page TIFF files into one multi-page TIFF file.

- **Recognizer** Perform optical character recognition using the optional OCR add-in, calculate mask or detect photographic pictures. Embed the recognized text in the resulting TIFF file.

- **Splitter** Split a multi-page TIFF file into single-page TIFF files.

- **Scanner** Scan image from a TWAIN or WIA scanner into one multi-page file.

1.2.1 Features

- Support for mixed raster content (MRC) layers during compression and conversion to PDF/A.
- Support for optical character recognition (OCR) and searchable PDFs through optional OCR add-in.
- Support for recognition of photographic picture regions.
- The JPEG import supports the RGB and CMYK color spaces of the JFIF and the Adobe Photoshop file formats.
- Support for "old-JPEG" according to TIFF V6 and "new-JPEG" according to Tech. Note #2 including the conversion from old to new format.
- Compression support for JPEG2000 and JBIG2 in addition to the TIFF V6 algorithms.
- Support for ICC color profiles.
- Support for XMP metadata.
- Substitution of a default resolution where missing.
Support for merging and splitting of multipage TIFFs.
Support for scanning with scanners that provide the TWAIN or WIA interface.

1.2.2 Formats

Input Formats
- JPEG
- TIFF V6 and Tech. Note #2

Output formats
- TIFF V6 and Tech. Note #2
- PDF 1.x (PDF 1.0, …, PDF 1.7)
- PDF 2.0
- PDF/A-1, PDF/A-2, PDF/A-3

1.2.3 Conformance

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)
- TIFF V6 (http://www.adobe.com)
- Tech. Note #2 (http://www.iijg.org)
- RFC 2301 (https://www.ietf.org/)

1.3 Interfaces

The following interfaces are available:
- C
- Java
- .NET Framework
- .NET Core

1.4 Operating Systems

The 3-Heights™ TIFF Toolbox API is available for the following operating systems:
- Windows Client 7+ | x86 and x64

‘+’ indicates the minimum supported version.

1 Limited supported OS versions. Operating Systems
2 Installation and Deployment

2.1 Windows

The 3-Heights™ TIFF Toolbox API comes as a ZIP archive or as a NuGet package.

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 “TIFF Toolbox API”. If you have no active downloads available or cannot log in, please contact pdfsales@pdf-tools.com for assistance.
   You will find different versions of the product available. We suggest to download the version, which is selected by default. A different version can be selected using the combo box.
   The product comes as a Zip Archive containing all files, or as a NuGet Package containing all files for development in .NET.
   There is a 32 and a 64-bit version 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 archive as well as the NuGet package contain both the 32-bit and the 64-bit version of the product.
3. If you are using the ZIP archive, do the following. Unzip the archive to a local folder, e.g. C:\Program Files\PDF Tools AG\. This creates the following subdirectories (see also Zip Archive):

<table>
<thead>
<tr>
<th>Subdirectory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin</td>
<td>Contains the runtime executable binaries.</td>
</tr>
<tr>
<td>doc</td>
<td>Contains documentation.</td>
</tr>
<tr>
<td>include</td>
<td>Contains header files to include in your C/C++ project.</td>
</tr>
<tr>
<td>jar</td>
<td>Contains Java archive files for Java components.</td>
</tr>
<tr>
<td>lib</td>
<td>Contains the object file library to include in your C/C++ project.</td>
</tr>
<tr>
<td>samples</td>
<td>Contains sample programs in various programming languages</td>
</tr>
</tbody>
</table>

4. The usage of the NuGet package is described in section NuGet Package.
5. (Optional) Register your license key using the License Management.
6. Identify which interface you are using. Perform the specific installation steps for that interface described in Interface Specific Installation Steps.

2.2 Zip Archive

The 3-Heights™ TIFF Toolbox API provides three 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.
## Programming Languages

### .NET

The MS software platform .NET can be used with any .NET capable programming language such as:
- C#
- VB .NET
- J#
- others

For a convenient way to use this interface, see [NuGet Package](#).

### Java

The Java interface.

### C

The native C interface is for use with C and C++.

---

### 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

<table>
<thead>
<tr>
<th>Name</th>
<th>.NET</th>
<th>Java</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bin\&lt;platform\&gt;\TiffToolboxAPI.dll</code></td>
<td>Req.</td>
<td>Req.</td>
<td>Req.</td>
</tr>
<tr>
<td><code>bin\*NET.dll</code></td>
<td>Req.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>bin\*NET.xml</code></td>
<td>Doc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>doc\javadoc\*.*</code></td>
<td>Doc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>include\tifftoolboxapi_c.h</code></td>
<td>Req.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>include\*.*</code></td>
<td>Opt.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>jar\TiffToolboxAPI.jar</code></td>
<td>Req.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>lib\&lt;platform\&gt;\TiffToolboxAPI.lib</code></td>
<td>Req.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The purpose of the most important distributed files of is described in table **File Description**.
### File Description

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bin\&lt;platform&gt;\TiffToolboxAPI.dll</code></td>
<td>This is the DLL that contains the main functionality (required), where <code>&lt;platform&gt;</code> is either Win32 or x64 for the 23-bit or the 64-bit library respectively.</td>
</tr>
<tr>
<td><code>bin\*NET.dll</code></td>
<td>The .NET assemblies are required when using the .NET interface. The files <code>bin\*NET.xml</code> contain the corresponding XML documentation for MS Visual Studio.</td>
</tr>
<tr>
<td><code>bin\&lt;platform&gt;\*.ocr</code></td>
<td>These are OCR plugin DLLs that are used in combination with the 3-Heights™ OCR Enterprise Add-On which can be purchased as a separate product.</td>
</tr>
<tr>
<td><code>doc\*.*</code></td>
<td>Various documentations.</td>
</tr>
<tr>
<td><code>include\*.*</code></td>
<td>Contains files to include in your C / C++ project.</td>
</tr>
<tr>
<td><code>lib\&lt;platform&gt;\TiffToolboxAPI.lib</code></td>
<td>The object file library needs to be linked to the C/C++ project.</td>
</tr>
<tr>
<td><code>jar\TiffToolboxAPI.jar</code></td>
<td>The Java API archive.</td>
</tr>
<tr>
<td><code>samples\*.*</code></td>
<td>Contains sample programs in different programming languages.</td>
</tr>
</tbody>
</table>

### 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 three different interfaces is shown in the table below.

<table>
<thead>
<tr>
<th>Name</th>
<th>.NET</th>
<th>Java</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bin\&lt;platform&gt;\TiffToolboxAPI.dll</code></td>
<td>Req.</td>
<td>Req.</td>
<td>Req.</td>
</tr>
<tr>
<td><code>bin\*NET.dll</code></td>
<td>Req.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>jar\TiffToolboxAPI.jar</code></td>
<td></td>
<td></td>
<td>Req.</td>
</tr>
</tbody>
</table>

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).

---

2 These files must reside in the same directory as TiffToolboxAPI.dll.
2.3 NuGet Package

NuGet is a package manager that facilitates the integration of libraries for the software development in .NET. The nuget package for the 3-Heights™ TIFF Toolbox API contains all the libraries needed, managed and native.

**Installation**  Download the package PdfTools.TiffToolbox.6.6.0.nupkg from your account on [https://www.pdf-tools.com/](https://www.pdf-tools.com/) to some suitable location.

In Visual Studio click on “Tools” and then “Options”. Select “NuGet Package Manager” and add the location of the downloaded package in “Package Sources”.

Right-click on a .NET project in Visual Studio and select “Manage NuGet Packages...”. Finally, select the package source that was defined above and browse to the desired package.


The required native libraries are loaded automatically. All project platforms are supported, including “AnyCPU”. In order to use the software, you must first install a license key for the 3-Heights™ TIFF Toolbox API. To do this you have to download the product kit and use the license manager in it. See also License Management.

**Note:** This NuGet package is only supported on a subset of the operating systems supported by .NET Core. See also Operating Systems.

2.4 Interface Specific Installation Steps

2.4.1 Java Interface

The 3-Heights™ TIFF Toolbox API requires Java version 7 or higher.

**For compilation and execution**  When using the Java interface, the Java wrapper jar\TiffToolboxAPI.jar needs to be on the CLASSPATH. This can be done by either adding it to the environment variable CLASSPATH, or by specifying it using the switch -classpath:

```
javac -classpath ";C:\Program Files\PDF Tools AG\jar\TiffToolboxAPI.jar" sampleApplication.java
```

**For execution**  Additionally the library TiffToolboxAPI.dll needs be in one of the system's library directories or added to the Java system property java.library.path. This can be achieved by either adding it dynamically at program startup before using the API, or by specifying it using the switch -Djava.library.path when starting the Java VM. Choose the correct subdirectory (x64 or Win32 on Windows) depending on the platform of the Java VM.

```
java -classpath ";C:\Program Files\PDF Tools AG\TiffToolboxAPI.jar" ^
```

---

3 On Windows defined by the environment variable PATH and e.g. on Linux defined by LD_LIBRARY_PATH.

4 If the wrong data model is used, there is an error message similar to this: “Can’t load IA 32-bit .dll on a AMD 64-bit platform”
2.4.2 .NET Interface

The 3-Heights™ TIFF Toolbox API does not provide a pure .NET solution. Instead, it consists of a native library and .NET assemblies, which call the native library. This has to be accounted for when installing and deploying the tool. It is recommended to use the NuGet Package. This ensures the correct handling of both the .NET assemblies and the native library.

Alternatively, the files in the Zip Archive can be used directly in a Visual Studio project targeting .NET Framework 2.0 or later. To achieve this, proceed as follows.

The .NET assemblies (*.NET.dll) are to be added as references to the project; They are needed at compile time. TiffToolboxAPI.dll is not a .NET assembly, but a native library. It is not to be added as a reference to the project. Instead, it is loaded during execution of the application.

For the operating system to find and successfully load the native library TiffToolboxAPI.dll, it must match the executing application's bitness (32-bit versus 64-bit) and it must reside in either of the following directories:

- In the same directory as the application that uses the library.
- In a subdirectory win-x86 or Pathwin-x64 for 32-bit or 64-bit applications respectively.
- In a directory that is listed in the PATH environment variable

In Visual Studio, when using the platforms "x86" or "x64", the above can be achieved by adding the 32-bit or 64-bit TiffToolboxAPI.dll respectively as an “existing item” to the project, and setting its property “Copy to output directory” to true. When using the “AnyCPU” platform, then you have to make sure by some other means that both the 32-bit and the 64-bit TiffToolboxAPI.dll are copied to subdirectories win-x86 and win-x64 of the output directory respectively.

2.4.3 C Interface

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

2.5 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. If using the COM interface, the new DLL must be registered, un-registering the old version is not required.

2.6 Note about the Evaluation License

With the evaluation license the 3-Heights™ TIFF Toolbox API automatically adds a watermark to the output files.
3 License Management

The 3-Heights™ TIFF Toolbox 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 LicenseKey property. 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 shell tool is available in the Utilities & Tools section of your My PDF Tools customer account.

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

```
licmgr
```

### List all installed license keys

```
licmgr list
```

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

**Example:**

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

### Add and delete license keys

Install new license key:

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

Delete old license key:

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

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

- `g` For all users
- `u` Current user

### Display the properties of a license

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

```plaintext
>licmgr info 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
- Key:          1-XXXXX-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:

```plaintext
licmgr select 1-XXXXX-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:

![License Manager Interface](image)

**With the Command Line Interface** use the update subcommand:

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

3.4 License activation

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

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

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

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

3.4.1 Activation

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

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

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

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

**In the Graphical User Interface** select the license and press the button “Activate license” in the toolbar:

![License Manager Interface](image)

It is recommended to add a comment to the activation request by using the subsequent dialog box.
With the Command Line Interface use the activate subcommand:

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

Note that the key has to be installed first.

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

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

### 3.5 Proxy Setting

A proxy URL can be configured for computers that cannot access the internet without a web proxy.
Note: The proxy must allow connections via HTTP CONNECT to the server www.pdf-tools.com:443.

In the Graphical User Interface press the button “Settings” in the toolbar:

and enter the proxy URL in the respective field:

3.6 Offline Usage

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

- License Activation
- License Reactivation
- License Deactivation
- Key Update

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

3.6.1 First Step: Create a Request File

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

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

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

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

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

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-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)
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.
- The software was not restarted after registering the license. Note that for web applications the webserver, and not merely the application, must be restarted.

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:
- One of the properties used to calculate the system fingerprint is changing frequently.

Solution Update to a newer version 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.

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 as 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 Getting Started

4.1 Mixed Raster Content (MRC)

Some raster images—typically scanned documents—consist mainly of text, possibly in several colors and interspersed with some pictures. Such images are difficult to compress with one single compression type because of the diverse or even conflicting features of different parts of the image.

The MRC technique is a way of breaking such images down into parts, such that each part is well suited for one type of a compression algorithm. With this approach, the resulting file size often can be reduced without significantly reducing the visual quality of the document.

**Note:** MRC optimization can only be enabled for continuous images, i.e. not for bi-tonal images and images with an indexed color space.

4.1.1 Phase 1: Recognizing Photographic Pictures

In this phase, the Recognizer computes a bi-tonal mask. Optionally rectangular areas containing photographic features are detected. The mask and the information about the location of the detected photographic pictures is embedded in the resulting TIFF.

It is possible, that actual photographic regions present in the input image are not recognized correctly. This can happen for example if a photographic region contains parts with uniform color.

4.1.2 Phase 2: Segmentation into Layers

In this phase the Compressor is used.

All photographic pictures that have been recognized in phase 1 are removed from the input image and are embedded as separate images in the resulting TIFF file. Thereafter the image is supposed not to contain photographic features anymore. Instead, the image is assumed to consist of text and graphic, potentially with varying color.

Now, using the mask from phase 1 the whole image is separated into three layers, a foreground, a background and a mask layer. The mask, which can be thought of as a bi-tonal image tells for each pixel whether to show the corresponding pixel of the foreground layer or the background layer.

**Example:**

Let the image consist of a yellow background with black paragraph text and a title text in red. Then the resulting background layer contains the yellow color only. The foreground layer contains the black text color where the paragraph text is located and the red text color where the title is located. In the mask, pixels for which the foreground layer should be displayed are set to 1, the others are set to 0. I.e. the mask contains 1's where the black and the red text is and 0's everywhere else.

In the resulting TIFF the foreground layer, the background layer and the mask are stored as three images. The mask can be compressed differently to the foreground and background layer. Since all the detailed features have been moved to the mask, it makes sense to down-sample the foreground and background layers and use a low image quality. The mask on the other hand is usually stored with a lossless compression type optimized for text.
4.1.3 Phase 3: PDF Construction

In this phase the TIFF-file that resulted from phase 1 (recognition of photographic pictures and recognition of mask) and phase 2 (the segmentation into foreground and background layers and into a mask) is now used to construct a PDF-file by using the Converter. If in phase 1, a single photographic region covering the entire image is detected, then the original image is used and the reconstruction is finished. Otherwise, the construction first places the background layer, followed by the foreground layer with the mask. Finally, if any isolated photographic pictures are found they are placed at their respective locations on top of the foreground layer.

4.2 Internal Engine

The internal engine comprises recognition functionality for the Recognizer, such as mask recognition or recognition of photographic picture regions.

In general, the string parameter for the OCR engines is composed by a sequence of Key-Value pairs that are separated by semicolons (;). In order to form the string parameter the following keys are supported by the internal engine.

4.2.1 RecognizeBlankPages

**Key:** RecognizeBlankPages  Type: Boolean  Default: false

Recognize blank pages of a certain file. A blank page is considered to be a page with a uniform coloring containing only slight noise. Colored, grayscale and bi-tonal pages can be subject to blank page recognition. The value of the Key-Value pair takes either true or false.

**Example:** Choose internal engine and recognize blank pages. Set parameters in RecognizerParameters as follows:

```plaintext
ocrEngineName = "internal";
ocrParameters = "RecognizeBlankPages=true";
```

Blank pages can then be removed from a tiff file using the Merger.

4.2.2 BlankPageMargin

**Key:** BlankPageMargin  Type: double  Default: 0.02
Set the ratio the margin takes with respect to the corresponding page length. The margin is excluded from the analysis whether a page is blank. The allowed values range from 0 to 0.5. This parameter is only active if at the same time the value of RecognizeBlankPages is true.

**Example:** Choose internal engine and analyze if a page is blank without taking into account a margin of 5% on every side. Set parameters in RecognizerParameters as follows:

```plaintext
ocrEngineName = "internal";
ocrParameters = "RecognizeBlankPages=true;BlankPageMargin=0.05";
```

Blank pages can then be removed from a tiff file using the Merger.

### 4.2.3 DisableMaskEmbedding

<table>
<thead>
<tr>
<th>Key:</th>
<th>DisableMaskEmbedding</th>
<th>Type: Boolean</th>
<th>Default: false</th>
</tr>
</thead>
</table>

If this option is set to true, no mask is embedded in the output TIFF. If this option is not set, a mask is embedded by default. The value of the Key-Value pair takes either true or false.

**Example:** Choose internal engine, recognize photographic pictures, but don’t embed mask in output. Set parameters in RecognizerParameters as follows:

```plaintext
ocrEngineName = "internal";
ocrParameters = "DisableMaskEmbedding=true;RecognizePictures=true";
```

### 4.2.4 RecognizePictures

<table>
<thead>
<tr>
<th>Key:</th>
<th>RecognizePictures</th>
<th>Type: Boolean</th>
<th>Default: false</th>
</tr>
</thead>
</table>

Recognize photographic picture regions.

**Example:** Choose internal engine and recognize photographic pictures. Set parameters in RecognizerParameters as follows:

```plaintext
ocrEngineName = "internal";
ocrParameters = "RecognizePictures=true";
```

The recognition of photographic picture regions works only for colored pictures.

### 4.3 Garbage collection and closing objects

Every interface object is considered being a resource that needs to be closed after use. Most objects are closed automatically, at the latest when the owning document is closed, in C# and Java possibly earlier by the garbage collector.
5 Programming Interfaces

Where possible and useful the 3-Heights™ TIFF Toolbox API uses language specific features. This means that some parts of the API use different syntax for different programming languages.

5.1 .NET Interface

5.1.1 IDisposable Objects

Objects that must be closed explicitly implement the IDisposable interface. Instead of calling Dispose() directly, it is recommended to use the “using” statement:

```csharp
using (Document document = ...) {
    ...
} // document.Dispose() is called implicitly here
```

See also Garbage collection and closing objects.

5.1.2 Error handling

Errors are reported using exceptions.

The three logic error codes are mapped to the corresponding native exception classes:

- `IllegalArgument` maps to `System.ArgumentException`
- `IllegalState` maps to `System.InvalidOperationException`
- `UnsupportedOperation` maps to `System.NotSupportedException`

The rest of the error codes is modeled using a single exception class `PdfTools.ErrorCodeException` that provides access to the underlying error code and message.

5.1.3 Streams

The native stream interface `System.IO.Stream` is used.

5.1.4 Lists

Lists implement the native list interface `System.Collections.Generic.IList<T>`.

5.2 Java Interface

5.2.1 AutoCloseable Objects

Objects that must be closed explicitly implement the AutoCloseable interface. Instead of calling close() directly, it is recommended to use the “try-with-resources” statement:
5.2.2 Properties

Properties are modeled with setter and getter methods.

5.2.3 Error handling

Errors are reported using exceptions.

The three logic error codes are mapped to the corresponding native runtime exception classes and are not checked:

- `IllegalArgumentException` maps to `java.lang.IllegalArgumentException`
- `IllegalStateException` maps to `java.lang.IllegalStateException`
- `UnsupportedOperationException` maps to `java.lang.UnsupportedOperationException`

The rest of the error codes is modeled using a single checked exception class `com.pdf_tools.ErrorCodeException` that provides access to the underlying error code and message.

5.2.4 Streams

The native stream interfaces cannot be used, because they are lacking two important features:

- The PDF file format is based on random access. Native Java streams have only limited support for this.
- The ability to read from an output stream is crucial for processing large files.

Instead we provide a custom stream interface `com.pdf_tools.Stream`, which has a similar interface as `java.io.RandomAccessFile`.

An implementation for files is provided, backed by `java.io.RandomAccessFile`.

5.2.5 Lists

Lists implement the native Java list interface `java.util.List`.

5.3 C Interface

5.3.1 Namespaces, classes and methods

In most languages, namespaces and classes are used to model the interfaces.

The exception is C, where this is modeled with function prefixes and functions operating on handles. The prefix of all functions of the 3-Heights™ TIFF Toolbox API is `Tiff`, for types it is `TTiff` and for enum values `eTiff`.

5.3.2 Library Initialization

The first method called must be `TiffInitialize`. Failing to invoke this function results in undefined behavior. Similarly, the last method must be `TiffUninitialize`.

See also Garbage collection and closing objects.
5.3.3 Objects

Objects in the C interface are represented by object handles. After use, all object handles returned by the 3-Heights™ TIFF Toolbox API must be closed with TiffClose.

5.3.4 Properties

Properties are modeled with setter and getter methods.

5.3.5 Error handling

After a having called a method, an error should be detected as follows:

a. If the method's return type is BOOL or a pointer and the return value is FALSE or NULL respectively, then an error has occurred.

b. If the method's return type is other than BOOL or a pointer, then TiffGetLastError must be called to detect whether an error has occurred.

More information about the error can be retrieved by using the functions TiffGetLastError and TiffGetLastErrorMessage.

5.3.6 Strings

All functions involving strings are provided in two different flavors:

- UTF-16 function with suffix W, using WCHAR as parameter type.
- Multibyte character set function with suffix A, using char as parameter type. The concrete character set that is used depends on the platform:
  - On Windows, the current ANSI code page (CP_ACP) is assumed.
  - On Linux or macOS, the current C encoding (LC_CTYPE) is used.

In addition to the effective function names with suffix, there's a macro without suffix for each function pair: It either resolves to the W variant (if _UNICODE is defined), or to the A variant (if _UNICODE is not defined).

Example: Signature of an API string property setter, where <String> stands for the property's name:

```c
void TiffSet<String>A(const char* szString); // Multibyte encoding
void TiffSet<String>W(const WCHAR* szString); // UTF-16
#define TiffSet<String>W TiffSet<String>A
#else
#define TiffSet<String>W TiffSet<String>A
#endif
```

String return values

Functions that return a string are treated specially in C. Instead of returning the string, those functions take a buffer and size as last parameters and write into that buffer. The return value is the amount of data written to the buffer.

To determine the required buffer size, the function has to be called with NULL as argument.

Calling the function with a buffer size that is too small results in a ePdfErrorIllegalState.

Multibyte character set functions (with suffix A) that return a string can fail to encode the string in the current operating systems' encoding. In case of such a failure, the return value is 0 and no error code is set. In order to prevent
such failures, it is recommended to use the UTF-16 \( \text{w} \) functions on Windows or to use operating systems with a Unicode code page.

**Example:** Signature and usage of an API string property getter (error handling is omitted), where \(<\text{String}>\) stands for the property's name:

```c
size_t TiffGet< String >A(char* pBuffer, size_t nBufferSize);
```

```c
size_t nBufferSize = TiffGet< String >A(NULL, 0);
char* pBuffer = malloc(nBufferSize * sizeof(char));
nBufferSize = TiffGet< String >A(pBuffer, nBufferSize);
```

### 5.3.7 Streams

Streams are modeled by means of a set of callbacks and a context pointer, grouped in a struct `TPdfStreamDescriptor`.

An implementation for `FILE*` is provided in the header file `pdfdecl.h`. (Search for function `PdfCreateFILEStreamDescriptor`.)

### 5.3.8 Lists

Lists in C are implemented like any other interface.

**List Interface**

Every list type provides a subset of the following properties and methods, where \(<\text{ElementType}>\) stands for the type name of the contained elements:

**Count**

- **Property (get):** `int Count`

  The number of elements of the list.

**Get**

- **Method:** `\(<\text{ElementType}>\) Get(int index)`

  Error codes: `IllegalState, IllegalArgument, UnsupportedOperation`

  Get an element of the list.

**Append**

- **Method:** `Append(<\text{ElementType}> element)`

  Error codes: `IllegalState, IllegalArgument, UnsupportedOperation`

  Append an element to the list.
6 Interface Reference

Note: This manual mainly describes the C# interface. Other interfaces (Java, C) however work similarly, i.e. they have calls with similar names and the call sequence to be used is the same as with C#.

6.1 Common methods

Note: In this section common static methods are listed. They can be called in every interface.

6.1.1 CheckLicense

Method: void CheckLicense()

Static

Error code: License

Check if the product is properly licensed.
This method can be used to perform a license checky without actually opening or creating a document, e.g. when starting a GUI application or a service.

Error Code:

License  The product is not properly licensed.

6.1.2 Close

Method: void Close()

Error code: IllegalState

Close the object and release all native resources associated with it.
After closing the object and releasing all native resources, any call to a method of this object will result in an IllegalState error.

Error Code:

IllegalState  The object has already been closed.

6.1.3 LicenseKey

Property (set): String LicenseKey

Static

Error code (set): License
Set the license key.

Note: License keys that require activation can only be installed in the license manager. Setting them at runtime is not supported.

Error Code:

License The license key is not valid.

6.1.4 ProductVersion

Property (get): String ProductVersion

Get the version of the 3-Heights™ TIFF Toolbox API in the format “A.C.D.E”.

6.2 Converter Interface

The Converter Interface converts a stream of a multi-page TIFF file into a stream of a PDF/A-2 conforming document.

If a TIFF page contains mixed raster content (MRC) layers (background, foreground, mask) they are converted into equivalent objects in the PDF document. The same is true for TIFF pages which contain alpha channels.

If a TIFF page contains embedded OCR text it is converted into an invisible text layer to make the resulting PDF document searchable. Note that only embedded OCR text that was created using the Converter Interface.

Tools which create embedded OCR text from other vendors are not supported, since the embedding of OCR is not part of the TIFF V6 standard and all vendors are using proprietary tags.

If a TIFF page contains XMP metadata stream it is added as a Metadata entry in the PDF page object.

In addition to the device specific color spaces the Converter Interface also supports calibrated color spaces and ICC color profiles.

The Converter Interface minimizes the re-compression of streams. If the page is not compressed it will not be compressed in the resulting output. However, if the page uses LZW compression it will be re-compressed using FLATE, since LZW is not allowed in PDF/A-2.

6.2.1 CreatePdf

Method: void CreatePdf(Stream stream, ConverterParameters convParms)

Error codes: IO, License, IllegalArgument, IllegalState, UnsupportedOperation

Create the output PDF file.

Parameters:

stream [Stream] The stream of the output PDF file.

convParms [ConverterParameters] Parameter object for the conversion.
**Error Codes:**

IO  Error occurred writing to the stream.

License  License not valid.

IllegalArgumentException  The stream argument is `null`.

IllegalStateException  The output document has already been created.

UnsupportedOperationException  Creation of output document failed.

### 6.2.2 AddTiff

**Method:**  
```java
void AddTiff(Stream stream)
```

**Error codes:**  
IO, UnsupportedOperation, IllegalArgumentException, IllegalStateException

Add TIFF file to PDF output file.

**Parameter:**

stream  [Stream]  The stream of the input TIFF file.

**Error Codes:**

IO  Error occurred reading from the stream.

UnsupportedOperationException  Adding of input TIFF document failed

IllegalArgumentException  The stream argument is `null`.

IllegalStateException  The output document has already been closed.

### 6.2.3 SetXmp

**Method:**  
```java
void SetXmp(Stream stream)
```

**Error codes:**  
IO, IllegalStateException, IllegalArgumentException, UnsupportedOperationException

Set XMP metadata.

**Parameter:**

stream  [Stream]  The stream of the XMP metadata.

**Error Codes:**

IO  Error occurred reading from the stream.
IllegalState   The output document has not been created yet.

IllegalArgument   The stream argument is null.

UnsupportedOperation   Setting XMP metadata failed.

### 6.2.4 SetOutputIntent

**Method:** Boolean SetOutputIntent(Stream stream)

**Error codes:** IO, IllegalState, UnsupportedOperation

The output intent represents the output color profile. Color profiles are usually provided with the OS. On Windows for example, they can be found at C:\Windows\System32\spool\drivers\color. Alternatively profiles can be found here:

- [http://www.color.org/srgbprofiles.html](http://www.color.org/srgbprofiles.html)

Please note that most color profiles are copyrighted, therefore you should read the license agreements on the above links before using the color profiles.

**Parameter:**

**stream**   [Stream]   The stream of the color profile. An example could be:

C:\Windows\System32\spool\drivers\color\sRGB Color Space Profile.icm

If PDF/A conformance is selected and no output intent is defined, then CMYK USWebCoatedSWOP.icc is embedded as default output intent.

This method must be called after CreatePdf has been called.

### 6.2.5 SetDefaultColorSpace

**Method:** void SetDefaultColorSpace(Stream stream)

**Error codes:** IO, IllegalState, IllegalArgument, UnsupportedOperation

This color space is used in the output PDF for pages that have no ICC profile available in the corresponding page in the input TIFF. A default color space profile can be set for both RGB and CMYK.

**Parameter:**

**stream**   [Stream]   The stream to the color space profile.

**Error Codes:**

IO   Error occurred reading from the stream.

IllegalState   The output document has not been created yet.
IllegalArgument  The stream argument is null.

UnsupportedOperation  Setting default color space failed.

### 6.2.6 ClosePDF

**Method:** void ClosePDF()

**Error code:** IllegalState

Close output PDF file. In comparision to the Close method the object itself is not closed and hence can be used for further processing.

**Error Code:**

IllegalState  The output document has already been closed.

### 6.3 Compressor Interface

The Compressor Interface compresses the pages of the input file with the selected compression algorithms.

For each of the TIFF classes Binary, Grayscale, Lab, Palette, RGB & YCbCr, Separated (CMYK) and MRC a different compression algorithm can be specified. The tool also allows for specifying a class specific quality measure for lossy compression algorithms.

The Compressor Interface allows for upgrading embedded JPEG streams in the TIFF V6 format to the newer format specified in the independent JPEG group's Technical Note #2.

The Compressor Interface provides the possibility of using the mixed raster content (MRC) technique, i.e. continuous images can be converted into background, foreground and mask layers (Section 4.1). The computation of a mask is done in the Recognizer. If no mask is present, the Compressor interface computes one when performing MRC. Photographic picture regions that are recognized by the Recognizer are not subject to MRC. The foreground and background layers can additionally be down-sampled.

The Compressor Interface processes the stream of an input single- or multi-page TIFF file. The output file contains the same number of pages but with differently compressed image data. If MRC layers are being created, the output file contains for every page images for the mask, the foreground and background layer, and the isolated photographic pictures that were recognized by the Converter.

### 6.3.1 OpenTiff

**Method:** void OpenTiff(Stream stream)

**Error codes:** IO, License, IllegalArgument, IllegalState

Open the input TIFF file.

**Parameter:**

stream  [Stream]  The stream of the input TIFF file.
6.3.2 SaveTiff

Method: void SaveTiff(Stream stream, CompressorParameters compParms)

Error codes: IO, IllegalArgument, IllegalState, UnsupportedOperation

Compress and save the TIFF file.

Parameters:

stream [Stream] The stream of the output TIFF file.
compParms [CompressorParameters] Parameter object for the compression.

Error Codes:

IO Error occurred writing to the stream.
IllegalArgument The stream argument is null.
IllegalState The input document has already been closed.
UnsupportedOperation Creation of output document failed.

6.3.3 CloseTiff

Method: void CloseTiff()

Error code: IllegalState

Close input TIFF file. In comparison to the Close method the object itself is not closed and hence can be used for further processing.

Error Code:

IllegalState The output document has already been closed.
6.4 Recognizer Interface

The Recognizer Interface recognizes the text using an OCR engine and embeds the recognized text. Furthermore a binary mask is calculated and photographic picture regions are detected.

In order to use text recognition one of the supported OCR engines must be used. Currently the engines FineReader 10 and 11 from ABBYY are supported. These engines are available as separate product kits.

Furthermore there is an internal engine which provides some recognition functionality (see Section 4.2), such as recognition of a binary mask or recognition of photographic picture regions.

Mask computation is done by default. An embedded mask can then be used in the Compressor to perform MRC (Section 4.1).

The Recognizer Interface processes the stream of one input single- or multi-page TIFF file. The output stream describes a TIFF file containing the same number of pages. If the OCR process for a specific page succeeds then the original image is replaced by the one delivered by the OCR engine, if any. The new image is compressed using a lossless algorithm such as CCITT G4 and LZW depending on the image class.

6.4.1 OpenTiff

Method: void OpenTiff(Stream stream)

Error codes: IO, License, IllegalArgumentException, IllegalState

Open the input TIFF file.

Parameter:

stream [Stream] The stream of the input TIFF file.

Error Codes:

IO Error occurred reading from the stream.

License License not valid.

IllegalArgumentException The stream argument is null.

IllegalStateException The input document has already been opened.

6.4.2 SaveTiff

Method: void SaveTiff(Stream stream, Stream xmlStream, RecognizerParameters recogParms)

Error codes: IO, IllegalArgumentException, IllegalStateException, UnsupportedOperation
Recognize text and / or photographic picture regions and save resulting TIFF file.

**Parameters:**

**stream**  [Stream]  The stream of the output TIFF file.

**xmlStream**  [Stream]  The stream of the XML file that stores the information of the recognized text and / or photographic image regions. If this stream is `null` no information of the recognition is returned.

**recogParms**  [RecognizerParameters]  Parameter object for the recognition of text and / or photographic image regions.

**Error Codes:**

**IO**
- Error writing to the stream.
- Error writing to the XML stream.

**IllegalArgumentException**
- The `stream` argument is `null`.
- The `recogParms.ocrParameters` argument has a value that is not supported.
- The `recogParms.ocrLanguages` argument has a value that is not supported.
- The `recogParms.ocrEngineName` argument has a value that is not supported.

**IllegalStateException**  The input document has already been closed.

**UnsupportedOperationException**  Creation of output document failed.

### 6.4.3 OcrEngines

**Property (get):**  `StringList OcrEngines`

Get list of OCR engines.

### 6.4.4 CloseTiff

**Method:**  `void CloseTiff()`  
**Error code:**  `IllegalStateException`

Close input TIFF file. In comparison to the `Close` method the object itself is not closed and hence can be used for further processing.

**Error Code:**

**IllegalStateException**  The output document has already been closed.
6.5 Importer Interface

The Importer Interface imports a stream of a JPEG, TIFF or PDF image and embeds it into a stream of a TIFF container. The JPEG streams are embedded conforming to the TIFF V6 specification. The Interface supports JPEG streams conforming to the JFIF and Adobe Photoshop format by interpreting the corresponding application specific markers.

If the input stream contains ICC profiles or XMP metadata streams they are embedded separately using the appropriate tags. All other application specific markers are ignored.

If the resolution information is missing in the JPEG stream a default resolution is used. The color space is retrieved from the application specific markers and, if missing, from the number of color channels.

6.5.1 CreateTiff

Method: void CreateTiff(Stream stream, ImporterParameters importParms)

Error Codes: IO, License, IllegalArgument, IllegalState, UnsupportedOperation

Create the output TIFF file.

Parameters:

stream [Stream]  The stream of the output TIFF file.

importParms [ImporterParameters]  Parameter object for the conversion.

Error Codes:

IO  Error occurred writing to the stream.

License  License not valid.

IllegalArgument  The stream argument is null.

IllegalStateException  The output document has already been created.

UnsupportedOperation  Creation of output document failed.

6.5.2 AddImg

Method: void AddImg(Stream stream)

Error Codes: IO, UnsupportedOperation, IllegalArgument, IllegalState

Add Image (JPEG, TIFF or PDF) to TIFF output file.

Parameter:

Error Codes:

**IO** Error occurred reading from the stream.

**UnsupportedOperation** Adding of input Image failed.

**IllegalArgumentException** The stream argument is null.

**IllegalStateException** The output document has already been closed.

### 6.5.3 CloseTiff

**Method:** void CloseTiff()

**Error code:** **IllegalStateException**

Close output TIFF file. In comparison to the `Close` method the object itself is not closed and hence can be used for further processing.

**Error Code:**

**IllegalStateException** The output document has already been closed.

### 6.6 Merger Interface

The Merger Interface merges streams of single- and multi-page TIFF files into a stream of a large multi-page file without further processing except for embedded JPEG streams.

There is a special processing of embedded V6 JPEG streams. Instead of just copying the corresponding tags to the output the JPEG stream is re-assembled from the corresponding tags and then embedded into the output page conforming to the TIFF V6 specification. This fixes known problems with badly created TIFF V6 files (see Technical Note #2 for more information on this) and increases the interoperability with the most common readers (viewers) in use.

### 6.6.1 CreateTiff

**Method:** void CreateTiff(Stream stream, MergerParameters mergeParms)

**Error codes:** IO, License, IllegalArgumentException, IllegalStateException, UnsupportedOperationException

Create the output TIFF file.

**Parameters:**

**stream** [Stream] The stream of the output PDF file.

**mergeParms** [MergerParameters] Parameter object that tells whether blank pages should be removed.
Error Codes:

IO  Error occurred writing to the stream.

License  License not valid.

IllegalArgumentException  The stream argument is null.

IllegalStateException  The output document has already been created.

UnsupportedOperationException  Creation of output document failed.

6.6.2 AddTiff

Method:  void AddTiff(Stream stream)

Error codes:  IO,UnsupportedOperation,IllegalArgumentException,IllegalStateException

Add TIFF to output TIFF file.

Parameter:

stream  [Stream]  The stream of the input TIFF file.

Error Codes:

IO  Error occurred reading from the stream.

UnsupportedOperationException  Adding of input TIFF document failed

IllegalArgumentException  The stream argument is null.

IllegalStateException  The output document has already been closed.

6.6.3 CloseTiff

Method:  void CloseTiff()

Error code:  IllegalState

Close output TIFF file. In comparison to the Close method the object itself is not closed and hence can be used for further processing.

Error Code:

IllegalStateException  The output document has already been closed.
6.7 Splitter Interface

The Splitter Interface selects a certain page from the input stream of a multi-page TIFF file and returns a stream of the corresponding single-page file without further processing except for embedded JPEG streams.

There is a special processing of embedded V6 JPEG streams. Instead of just copying the corresponding tags to the output the JPEG stream is re-assembled from the corresponding tags and then embedded into the output page conforming to the TIFF V6 specification. This fixes known problems with badly created TIFF V6 files (see Technical Note #2 for more information on this) and increases the interoperability with the most common readers (viewers) in use.

6.7.1 OpenTiff

**Method:** void OpenTiff(Stream stream)

**Error codes:** IO, License, IllegalArgument, IllegalState

Open the input TIFF file.

**Parameter:**

stream [Stream] The stream of the input TIFF file.

**Error Codes:**

IO Error occurred reading from the stream.
License License not valid.
IllegalArgument The stream argument is null.
IllegalState The input document has already been opened.

6.7.2 PageCount

**Property (get):** int PageCount

Get list of OCR engines.

6.7.3 SavePages

**Method:** void SavePages(Stream stream, SplitterParameters splitParms)

**Error codes:** IO, IllegalArgument, IllegalState, UnsupportedOperation
Pick all pages in a certain page range of the input TIFF file and save them as new TIFF file.

**Parameters:**

- **stream**  
  [Stream]  
  The stream of the output TIFF file.

- **splitParms**  
  [SplitterParameters]  
  Parameter object that contains page range.

**Error Codes:**

- **IO**  
  Error occurred writing to the stream.

- **IllegalArgumentException**  
  The **stream** argument is **null**.

- **IllegalStateException**  
  The input document has already been closed.

- **UnsupportedOperationException**  
  Creation of output document failed.

### 6.7.4 CloseTiff

**Method:**  
`void CloseTiff()`  

**Error code:**  
`IllegalStateException`

Close input TIFF file. In comparision to the `Close` method the object itself is not closed and hence can be used for further processing.

**Error Code:**

- **IllegalStateException**  
  The output document has already been closed.

### 6.8 Scanner Interface

This interface allows to scan with scanners that provide the TWAIN or WIA interface. One can specify custom scan settings in a user interface and save them in a capability file.

#### 6.8.1 SetSource

**Method:**  
`void SetSource(String source)`  

**Error codes:**  
`IllegalStateException, IllegalArgumentException, IllegalArgumentException`

Set TWAIN or WIA scanner to scan with. A list of available scanners can be obtained from `Sources`.

**Parameter:**

- **source**  
  [String]  
  Name of the scanner.
### Error Codes:

**IllegalState**  
The scanner has already been closed.

**IllegalArgument**  
The scanner could not be found.

### 6.8.2 ShowUI

**Method:**  
`Boolean ShowUI()`  
**Error code:**  
`IllegalState`

Show user interface that allows to configure scan settings. Get the saved settings with property `Capabilities`.

**Returns:**

- **true**: User interface button clicked "OK". 
- **false**: User interface button clicked "Cancel".

**Error Code:**

**IllegalState**  
The scanner has already been closed.

### 6.8.3 Scan

**Method:**  
`void Scan(Stream stream)`  
**Error codes:**  
`IO, License, IllegalState, IllegalArgument, UnsupportedOperation`

Scan into output stream. Use `SetSource` to set the TWAIN or WIA scanner.

**Parameter:**

- **stream**  
  `[Stream]`  
  Stream to scan into.

**Error Codes:**

- **IO**  
  Error writing to the `stream`.

- **License**  
  License not valid.

- **IllegalState**  
  The scanner has already been closed.

- **IllegalArgument**  
  The `stream` is `null`.

- **UnsupportedOperation**  
  The output file couldn’t be created.
6.8.4 Capabilities

**Property (get, set):** Stream Capabilities

**Error codes (set):** IO, IllegalState, IllegalArgument, UnsupportedOperation

Get or set capability stream which stores scan settings. They can be customized by ShowUI in a user interface.

**Error Codes:**

- **IO**  Error writing the Capabilities stream.
- **IllegalState**  The scanner has already been closed.
- **IllegalArgument**  The Capabilities stream is null.
- **UnsupportedOperation**  The Capabilities could not be set.

6.8.5 Sources

**Property (get):** StringList Sources

Get list of available TWAIN or WIA sources. Only scanners that match the architecture of the API are listed.

6.9 StringList Interface

6.9.1 Get

**Method:** String Get(int index)

**Error code:** IllegalArgument

Pick certain element of list.

**Parameter:**

- **index**  [int]  Index of list element to be picked.

**Error Code:**

- **IllegalArgument**  The index is out of bounds.

6.9.2 Size

**Method:** int Size()
Get number of listed elements.

## 6.10 Structures

### 6.10.1 ConverterParameters Struct

#### Resolution

<table>
<thead>
<tr>
<th>Property (get, set):</th>
<th>double Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>96.0</td>
</tr>
</tbody>
</table>

If a TIFF page does not contain resolution tags they are defined by the parameter value of this property. The value’s unit is dots per inch (DPI).

#### OptionalContentGroup

<table>
<thead>
<tr>
<th>Property (get, set):</th>
<th>BoolEnum OptionalContentGroup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>False</td>
</tr>
</tbody>
</table>

PDF/A-2 supports optional content sometimes referred to as layers. If this property is used in conjunction with MRC layers, then the output document contains the optional content groups “Colored Text”, “Black Text”, “Foreground” and “Background” which can be individually turned on and off.

#### IgnoreOcr

<table>
<thead>
<tr>
<th>Property (get, set):</th>
<th>BoolEnum IgnoreOcr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>False</td>
</tr>
</tbody>
</table>

Ignore OCR data that was recognized by the Recognizer. Photographic image regions are not ignored by this option.

#### Conformance

<table>
<thead>
<tr>
<th>Property (get, set):</th>
<th>Conformance Conformance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>PDF/A-2b</td>
</tr>
</tbody>
</table>

PDF Conformance level. See Conformance.

### 6.10.2 CompressorParameters Struct

#### BinaryCompression

<table>
<thead>
<tr>
<th>Property (get, set):</th>
<th>TiffCompression BinaryCompression</th>
</tr>
</thead>
</table>

Compression algorithm for binary images.
**GrayscaleCompression**

Property (get, set): TiffCompression GrayscaleCompression

Compression algorithm for grayscale images.

**LabCompression**

Property (get, set): TiffCompression LabCompression

Compression algorithm for lab images.

**PalettedCompression**

Property (get, set): TiffCompression PalettedCompression

Compression algorithm for paletted images.

**Rgb_YcbcrCompression**

Property (get, set): TiffCompression Rgb_YcbcrCompression

Compression algorithm for RGB and YCbCr images.

**CmykCompression**

Property (get, set): TiffCompression CmykCompression

Compression algorithm for CMYK images.

**MrcCompression**

Property (get, set): TiffCompression MrcCompression

Compression algorithm for MRC images.

**BinaryCompQuality**

Property (get, set): double BinaryCompQuality

Default: 80

Compression quality for binary images. The value range is from 1 to 100.

**GrayscaleCompQuality**

Property (get, set): double GrayscaleCompQuality

Default: 80
Compression quality for grayscale images. The value range is from 1 to 100. The quality is effective only when using a lossy compression algorithm for the specific class.

**LabCompQuality**

<table>
<thead>
<tr>
<th>Property (get, set):</th>
<th>double LabCompQuality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>80</td>
</tr>
</tbody>
</table>

Compression quality for lab images. The value range is from 1 to 100. The quality is effective only when using a lossy compression algorithm for the specific class.

**Rgb_YcbcrCompQuality**

<table>
<thead>
<tr>
<th>Property (get, set):</th>
<th>double Rgb_YcbcrCompQuality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>80</td>
</tr>
</tbody>
</table>

Compression quality for RGB and YCbCr images. The value range is from 1 to 100. The quality is effective only when using a lossy compression algorithm for the specific class.

**CmykCompQuality**

<table>
<thead>
<tr>
<th>Property (get, set):</th>
<th>double CmykCompQuality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>80</td>
</tr>
</tbody>
</table>

Compression quality for CMYK images. The value range is from 1 to 100. The quality is effective only when using a lossy compression algorithm for the specific class.

**MrcCompQuality**

<table>
<thead>
<tr>
<th>Property (get, set):</th>
<th>double MrcCompQuality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>10</td>
</tr>
</tbody>
</table>

Compression quality for MRC foreground and background images. The binary mask cannot be compressed. The value range is from 1 to 100. The quality is effective only when using a lossy compression algorithm (see table) for the specific class.

**BinaryDownsampling**

<table>
<thead>
<tr>
<th>Property (get, set):</th>
<th>double BinaryDownsampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Down sampling factor for binary images. A value of 2 e.g. means that the resulting image dimension and resolution are divided by two.

**GrayscaleDownsampling**

<table>
<thead>
<tr>
<th>Property (get, set):</th>
<th>double GrayscaleDownsampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Down sampling factor for grayscale images. A value of 2 e.g. means that the resulting image dimension and resolution are divided by two.

**LabDownsampling**

| Property (get, set): double LabDownsampling |
|-----------------------------|-----------------------------|
| Default: 1.0 |

Down sampling factor for lab images. A value of 2 e.g. means that the resulting image dimension and resolution are divided by two.

**Rgb_YcbcrDownsampling**

| Property (get, set): double Rgb_YcbcrDownsampling |
|-----------------------------|-----------------------------|
| Default: 1.0 |

Down sampling factor for RGB and YCbCr images. A value of 2 e.g. means that the resulting image dimension and resolution are divided by two.

**CmykDownsampling**

| Property (get, set): double CmykDownsampling |
|-----------------------------|-----------------------------|
| Default: 1.0 |

Down sampling factor for CMYK images. A value of 2 e.g. means that the resulting image dimension and resolution are divided by two.

**MrcDownsampling**

| Property (get, set): double MrcDownsampling |
|-----------------------------|-----------------------------|
| Default: 1.0 |

Down sampling factor for MRC background and foreground images. The binary mask is not subjected to downsampling. A value of 2 e.g. means that the resulting image dimension and resolution are divided by two.

**Recompress**

| Property (get, set): BoolEnum Recompress |
|-----------------------------|-----------------------------|
| Default: 1.0 |

Recompress streams that have already been compressed with a lossy algorithm (JPEG, JPEG2000) in order to reduce the size by specifying either a different algorithm or a different quality.

**PerformMrc**

| Property (get, set): BoolEnum PerformMrc |
|-----------------------------|-----------------------------|
| Default: False |
Separate the image into background, foreground and mask layer.

**BinarizationThreshold**

**Property (get, set):** double BinarizationThreshold

The threshold value determines which pixel is associated to the foreground layer respectively to the background layer. (Black is 0, white is 255.) The threshold value should be set near to the average gray level of the image to achieve the best compression quality and ratio. The default threshold value is computed automatically from the image sample data.

**UpgradeJpeg**

**Property (get, set):** BoolEnum UpgradeJpeg

Default: False

Re-writes TIFF pages containing JPEG streams according to the TIFF V6 specification in the newer format conforming to the Technical Note #2 of the independent JPEG group. The re-formating does not involve any re-compression and thus is lossless.

### 6.10.3 RecognizerParameters Struct

**OcrEngineName**

**Property (get, set):** String OcrEngineName

Default: Nothing

One of the supported OCR engines can be selected. Currently the values "abbyy10", "abbyy11" and "service" are supported. In order to function correctly the tool requires the selected engine add-in to be installed.

In addition the value "internal" is supported. In this case no add-in is needed. For more information see Section 4.2.

**OcrParameters**

**Property (get, set):** String OcrParameters

Default: Nothing

Is an OCR engine specific string parameter. For more information on this, refer to the documentation of the engine add-in.

**OcrLanguages**

**Property (get, set):** String OcrLanguages

Default: Nothing

Receive a language identifier. The language is a hint for the OCR engine to recognize the text in the given language.
**BinarizationPriorOcr**

**Property (get, set):** `BoolEnum BinarizationPriorOcr`  
Default: `False`  

Convert color and gray scale image data to bi-tonal samples before sending the data to the OCR engine.

**BinarizationThreshold**

**Property (get, set):** `double BinarizationThreshold`  
Default: `128`  

Define the black/white threshold.

### 6.10.4 ImporterParameters Struct

**Resolution**

**Property (get, set):** `double Resolution`  
Default: `96.0`  

Recompress streams that have already been compressed with a lossy algorithm (JPEG, JPEG2000, JBIG2) in order to reduce the size by specifying either a different algorithm or a different quality.

**UpgradeJpeg**

**Property (get, set):** `BoolEnum UpgradeJpeg`  
Default: `False`  

Re-writes TIFF pages containing JPEG streams according to the TIFF V6 specification in the newer format conforming to the Technical Note #2 of the independent JPEG group. The re-formatting does not involve any re-compression and thus is lossless.

### 6.10.5 MergerParameters Struct

**RemoveBlankPages**

**Property (get, set):** `BoolEnum RemoveBlankPages`  
Default: `False`  

Pages that have been recognized by the Recognizer to be blank are removed from the TIFF file.
6.10.6 SplitterParameters Struct

**PageNumberStart**

<table>
<thead>
<tr>
<th>Property (get, set):</th>
<th>int PageNumberStart</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>FirstPage</td>
</tr>
</tbody>
</table>

Start number of page selection range. If this property is not set, by default the first page will be used.

**PageNumberEnd**

<table>
<thead>
<tr>
<th>Property (get, set):</th>
<th>int PageNumberEnd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>LastPage</td>
</tr>
</tbody>
</table>

End number of page selection range. If this property is not set, by default the last page will be used.

6.11 Enumerations

6.11.1 Conformance Enumeration

- **Unknown** The conformance is unknown or automatically determined.
- **Pdf10** PDF Version 1.0
- **Pdf11** PDF Version 1.1
- **Pdf12** PDF Version 1.2
- **Pdf13** PDF Version 1.3
- **Pdf14** PDF Version 1.4 (corresponds to Acrobat 5)
- **Pdf15** PDF Version 1.5
- **Pdf16** PDF Version 1.6 (corresponds to Acrobat 7)
- **Pdf17** PDF Version 1.7
- **Pdf20** PDF Version 2.0
- **PdfA1B** PDF/A-1b, ISO 19005-1, Level B conformance
- **PdfA1A** PDF/A-1a, ISO 19005-1, Level A conformance
- **PdfA2B** PDF/A-2b, ISO 19005-2, Level B conformance
- **PdfA2U** PDF/A-2u, ISO 19005-2, Level U conformance
- **PdfA2A** PDF/A-2a, ISO 19005-2, Level A conformance
- **PdfA3B** PDF/A-3b, ISO 19005-3, Level B conformance
- **PdfA3U** PDF/A-3u, ISO 19005-3, Level U conformance
- **PdfA3A** PDF/A-3a, ISO 19005-3, Level A conformance

6.11.2 BoolEnum Enumeration

- **Compression_Undef** Not defined.
6.11.3 Compression Enumeration

**CompressionUndef**  Compression not defined.
**CompressionUncompressed**  No compression.
**CompressionCcittGroup31d**  CCITT Fax Group 3 1d.
**CompressionCcittGroup3**  CCITT Fax Group 3.
**CompressionCcittGroup4**  CCITT Fax Group 4.
**CompressionLzw**  Lempel-Ziv-Welch.
**CompressionJpeg**  Joint Photographic Expert Group. Lossy compression.
**CompressionJpegTn2**  Joint Photographic Expert Group Tech. Note #2. Lossy compression.
**CompressionAdobeDeflate**  Adobe Deflate.
**CompressionPackBits**  Pack Bits.
**CompressionDeflate**  Deflate.
**CompressionJpeg2000**  Wavelet transformation. Lossy compression.
**CompressionJbig2**  Joint Bi-level Image Experts Group. Compression not lossy.

6.11.4 ErrorCode Enumeration

See Programming Interfaces for more information about how these codes are mapped to exceptions in the .NET and Java interface.

**Logic errors**

These codes denote errors in the application program logic and should never happen at runtime.

**UnsupportedOperation**  The requested method or property is not supported.
**IllegalState**  The object is in a state, where the requested object or property cannot be called.
**IllegalArgument**  The method was called using an illegal argument.

**Environmental errors**

**Generic**  The error is not further specified.
**Fatal**  A fatal error occurred.
**License**  Licensing error.
**NotFound**  The requested item or resource could not be found.
**IO**  Error while reading or writing from a stream.
**UnknownFormat**  The format is unknown.
**Corrupt**  The data is corrupt.
**Password**  The resource or document is protected by a password.
Conformance  A conformance mismatch happened.

UnsupportedFeature  The file contains an unsupported feature.

Infrastructure  An infrastructure error occurred.

Processing  The file cannot be processed.

Exists  The item already exists.
7 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.

7.1 Changes in Version 6

- [.NET] **Improved** ErrorCodeException which is now serializable (except for the .NET Standard 1.0 target).
- [Java] **Changed** minimal supported Java language version to 7 [previously 6].
- [NET] **New** availability of this product as NuGet package for Windows.
- [NET] **New** support for .NET Core versions 1.0 and higher. The support is restricted to a subset of the operating systems supported by .NET Core, see Operating Systems.
- [NET] **Changed** platform support for NuGet packages: The platform "AnyCPU" is now supported for .NET Framework projects.
- [Java] **Changed** inheritance. All interfaces except StringList now inherit from AutoCloseable and therefore can be used in a try-resource clause.
- [Java] **New** static method fromValue for all non-flags enums.

**Interface StringList**

- [Java] **Deprecated** method close.
- [NET] **Deprecated** method Dispose.

7.2 Changes in Version 5

- **New** additional supported operating system: Windows Server 2019.
- **Changed** behavior when reading a TIFF. The value Relative from tag ResolutionUnit is now interpreted as Inch.
- [NET] **Changed** TiffToolboxNET.dll library. Cross-product functionality is outsourced into common library PdfCommonNET.dll.

**Interface Compressor**

- **Improved** MRC background and foreground layer coloring.

7.3 Changes in Version 4.12

- **New** HTTP proxy setting in the GUI license manager.
- **New** Interface Scanner to scan images from a TWAIN or WIA scanner.

**Interface Recognizer**

- **New** key BlankPageMargin for the formation of the OcrParameters property in struct RecognizerParameters. The corresponding value denotes the relative margin. The margin is excluded from the analysis if a page is blank.
Interface Scanner

- **New** method `SetSource` to set TWAIN or WIA scanner to scan with.
- **New** method `ShowUI` to show user interface that allows to configure scan settings.
- **New** method `Scan` to scan into output stream.
- **New** property `Capabilities` to get and set capability stream.
- **New** property `Sources` to get list (see `StringList`) of available TWAIN/WIA sources.

### 7.4 Changes in Version 4.11

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

### 7.5 Changes in Version 4.10

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

### 7.6 Changes in Version 4.9

- **Improved** metadata generation for standard PDF properties.
- [C] **Changed** return value `pfGetLength` of `TPDFStreamDescriptor` to `pos_t`.

Interface Converter

- [.NET, C, Java] **New** method `SetDefaultColorSpace`: This color space is used in the output PDF for pages that have no ICC profile available in the corresponding page in the input TIFF. A default color space profile can be set for both RGB and CMYK.
- [Java] **New**: `MemoryStream` implementation for `Stream` interface.

### 7.7 Changes in Version 4.8

#### All interfaces

- [.NET, C, Java] **New** property `ProductVersion` to identify the product version.
- [.NET] **New** static property setter `LicenseKey`.

#### Interface Compressor

- [.NET, C, Java] **New** method `SetOutputIntent` to set the output intent.

---

1 This has no effect on neither the .NET, Java, nor COM API
8 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™ TIFF Toolbox 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 Bachenbüelach
Switzerland
http://www.pdf-tools.com
pdfsales@pdf-tools.com