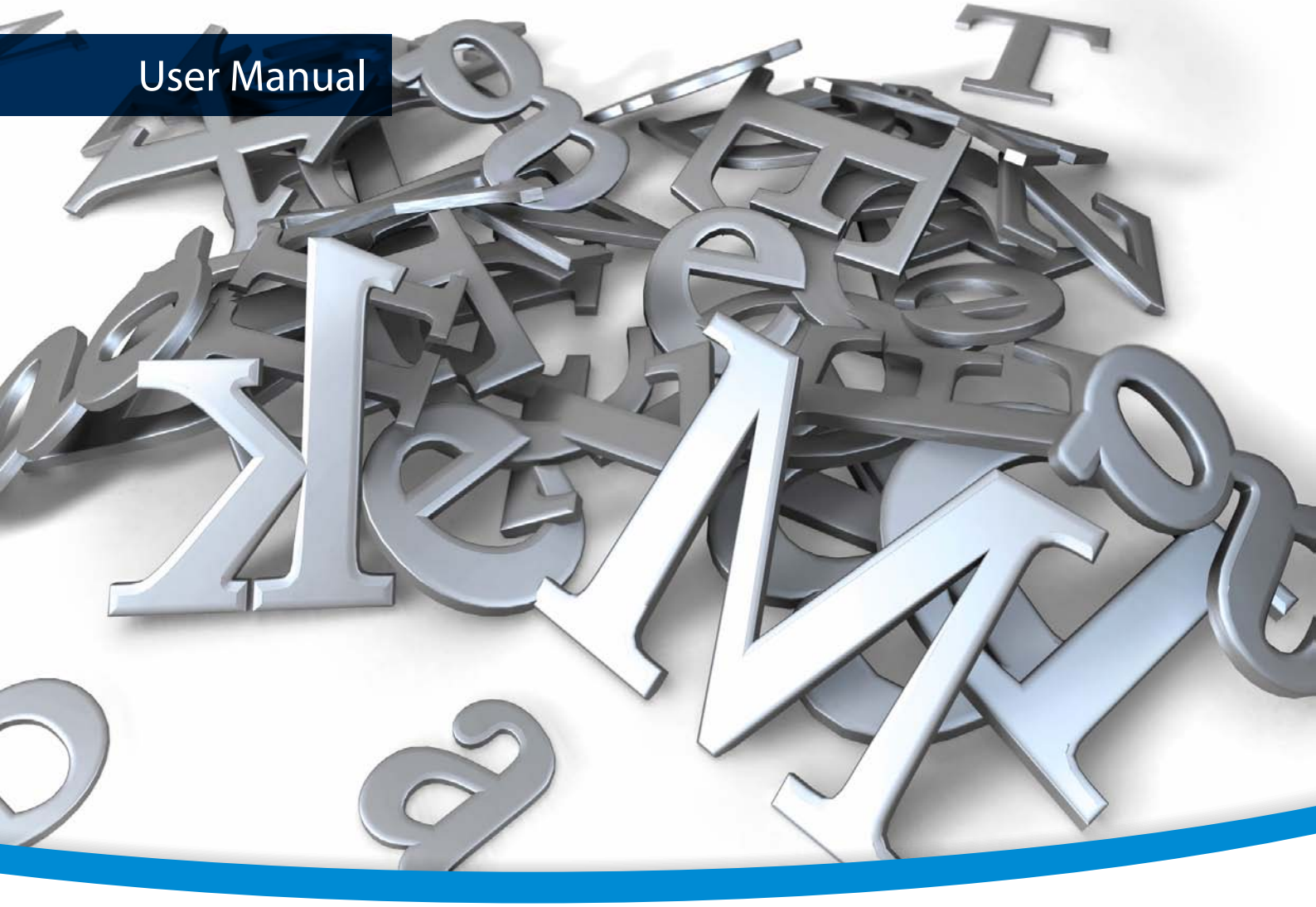
User Manual



4-Heights®
PDF Viewing SDK

For C

Version 2.12.0

PDF-TOOLS.COM
Premium PDF Technology

Contents

1 Introduction

1.1 Description

The 4-Heights® PDF Viewing SDK is a compact, high-performance, high-quality viewer API offering a multitude of navigational and display options for constructing a viewer application for PDF and PDF/A documents.

This product is the successor of the 3-Heights® PDF Viewing API.

1.2 Features

- Structured API with interfaces, enumerations, and callbacks for PDF objects and viewing infrastructure
- Synchronous functions where possible and asynchronous functions with callbacks where necessary
- Support for high resolution displays, e.g. Retina™ displays
- Cache management controls the memory consumption
- Transformation function for mapping screen points to PDF points and vice versa
- API functions to create, change or delete annotations, including:
  - Sticky notes
  - Free-text annotations
  - Stamps (“Draft”, “Approved”, etc.)
  - Ink (pen drawing) annotations
  - Highlighting annotations (highlight, strike through, underline, squiggly underline)
- Obtain the document outline (bookmarks)
- Opening of password protected documents
- Saving of modified documents
- Support for FDF: Keep the document (PDF) and it’s annotations (FDF) in two distinct files

1.3 Supported Input and Output Formats

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

1.4 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)

1.5 Operating Systems

The 4-Heights® PDF Viewing SDK is available for the following operating systems:

- Windows Client 7+ | x86 and x64
- Linux:
  - Red Hat, CentOS, Oracle Linux 8+ | x64
  - Fedora 29+ | x64
  - Debian 10+ | x64
  - Other: Linux kernel 2.6+, GCC toolset 4.8+, glibc 2.27+ | x64
- macOS 10.10+ | x64

'+' indicates the minimum supported version.

2 Installation

2.1 General Installation Steps

The "4-Heights® PDF Viewing SDK For C" comes as a ZIP archive. The installation of the software requires the following steps:

1. Ensure that your system matches one of the supported Operating Systems.
2. Log in to your account on https://www.pdf-tools.com/ and download the ZIP archive and the license key for "PDF Viewing SDK". (See also License Keys.)
3. Unzip the archive to a local directory, e.g, on Windows: `C:\Program Files\PDF Tools AG\`, on Linux: `/opt/pdftools.com/`. This creates the following subdirectories:

Subdirectory	Description
`bin`	Contains the runtime executable binaries for all supported platforms: ▪ `linux/libPdfViewing.so` for 64 bit Linux ▪ `macOS/libPdfViewing.dylib` for 64 bit macOS ▪ `Win32\PdfViewing.dll` for 32 bit Windows ▪ `x64\PdfViewing.dll` for 64 bit Windows
`doc`	Contains documentation
`include`	Contains the header files to include in your C/C++ project. The main header `PdfViewing.h` includes all the other headers.
`lib`	Contains the object file library for Windows to link against in your C/C++ project: ▪ `Win32\PdfViewing.lib` for 32 bit Windows ▪ `x64\PdfViewing.lib` for 64 bit Windows

4. On Windows, you may want to add the `bin\x64` or `bin\Win32` sub-directory to the `%PATH%` environment variable.
   On Linux, you may want to create a link to the shared library from one of the standard library directories, e.g:

```
ln -s /opt/pdf-tools.com/bin/linux/libPdfViewing.so /usr/lib
```

5. Ensure that your platform meets the requirements regarding color profiles. (See Color Profiles.)
6. Ensure that your platform meets the requirements regarding fonts. (See Fonts.)
7. Ensure that the cache directory exists and is writable. (See Special Directories.)

2.2 Color Profiles

If no color profiles are available, default profiles for both RGB and CMYK are generated on the fly by the 4-Heights® PDF Viewing SDK.

2.2.1 Default Color Profiles

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

Windows

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

Linux and macOS

1. $PDF_ICC_PATH if the environment variable is defined
2. the current working directory

2.2.2 Get Other Color Profiles

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

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

2.3 Fonts

Note that on Windows when a font is installed it is by default installed only for a particular user. It is important to either install fonts for all users, or make sure the 4-Heights® PDF Viewing SDK is run under that user and the user profile is loaded.

2.3.1 Font Cache

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

In order to achieve optimal performance, make sure that the cache directory is writable for the 4-Heights® PDF Viewing SDK. Otherwise the font cache cannot be updated and the font directories have to be scanned on each program startup.

The font cache is created in the subdirectory <CacheDirectory>/Installed Fonts of the Cache Directory.

2.4 Special Directories

2.4.1 Directory for temporary files

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

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

Windows

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

Linux and macOS

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

2.4.2 Cache Directory

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

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

Windows

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

Linux and macOS

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

where `<TempDirectory>` refers to the Directory for temporary files.

2.4.3 Font Directories

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

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

Windows

1. `%SystemRoot%\Fonts`
2. User fonts listed in the registry key `\HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Fonts`. This includes user specific fonts from `C:\Users\<user>\AppData\Local\Microsoft\Windows\Fonts` and app specific fonts from `C:\Program Files\WindowsApps`
3. directory `Fonts`, which must be a direct sub-directory of where `PdfViewingAPI.dll` resides.

macOS

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

Linux

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

2.5 License Keys

The 4-Heights® PDF Viewing SDK can only be used with a vaild license key. This key must be set programmatically by using the `Sdk.Initialize` method prior to making any calls to the library. You can download your license key from your account on https://www.pdf-tools.com/.

For licensing questions, please contact pdfsales@pdf-tools.com.

2.6 Un-Install, Install a New Version

Un-installation is done by undoing the steps in General Installation Steps, specifically:

- Remove un-zipped files
- Revert environment variables if necessary
- Remove links if necessary

An update to a new version is done by a drop-in replacement of the existing files. Make sure that all files are updated consistently.

3 C Interface

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 4-Heights® PDF Viewing SDK defines all its used types, such as handles and enumerations, in the header file `PdfViewing_Types.h`. Types are named according to the following naming scheme:

`T‹prefix›_‹name›`

where `‹prefix›` is a shortened namespace prefix and `‹name›` is the name of the type.

Similarly, the 4-Heights® PDF Viewing SDK defines functions, collected in header files `PdfViewing_‹sub-prefix›.h` with the following naming convention:

`‹prefix›_‹type›_‹name›`

where `‹type›` is the "class" name and `‹name›` is the name of the function.

3.2 Library Initialization

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

Before calling any of the other functions, a license key must be set by calling `Viewing_Sdk_Initialize`.

3.3 Objects

Objects in the C interface are represented by object handles. Some types are disposable, which means that they must be closed by calling `‹prefix›_‹type›_Close`

3.4 Properties

"Properties" (a C# term) are modeled with setter and getter functions `‹prefix›_‹type›_Get‹name›` and `‹prefix›_‹type›_Set‹name›`, where `‹name›` is the name of the property.

3.5 Error handling

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

a. If the function'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 function's return type is other than `BOOL` or a pointer, then `Viewing_GetLastError` must be called. If this method returns `eViewing_Error_Success`, then no error has occured.

More information about the error can be from `ViewingGetLastErrorMessage`.

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:

```
// Multibyte encoding:
void ‹prefix›_‹type›_Set‹name›A(T‹prefix›_‹type›* pHandle, const char* szString);
// UTF-16:
void ‹prefix›_‹type›_Set‹name›W(T‹prefix›_‹type›* pHandle, const WCHAR* szString);
#ifdef _UNICODE
#define ‹prefix›_‹type›_Set‹name› ‹prefix›_‹type›_Set‹name›W
#else
#define ‹prefix›_‹type›_Set‹name› ‹prefix›_‹type›_Set‹name›A
#endif
```

3.6.1 String return values

In C, functions that return a string have a special behavior. Instead of returning the string, those functions take a buffer and a 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 buffer argument.

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

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 (`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.)

```
size_t Viewing_Sdk_GetVersionA(char* pBuffer, size_t nBufferSize);
```

```
size_t nBufferSize = Viewing_Sdk_GetVersionA(NULL, 0);
char* pBuffer = malloc(nBufferSize * sizeof char);
nBufferSize = Viewing_Sdk_GetVersionA(pBuffer, nBufferSize);
```

3.7 Streams

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

An implementation for FILE* is provided in the header file PdfViewing_ViewingSys.h. (Search for function ViewingSysCreateFILEStreamDescriptor.)

3.8 Lists

Every list type T‹prefix›_‹list› provides a subset of the following functions, where T‹prefix›_‹eltype› stands for the type of the contained elements:

int ‹prefix›_‹list›_GetCount(T‹prefix›_‹list›*)
Get the number of elements in the list.

Possible errors: eViewing_Error_IllegalState

T‹prefix›_‹eltype›* ‹prefix›_‹list›_Get(T‹prefix›_‹list›*, int index)
Get an element of the list.

Possible errors: eViewing_Error_IllegalState, eViewing_Error_IllegalArgument, eViewing_Error_UnsupportedOperation

BOOL ‹prefix›_‹list›_Add(T‹prefix›_‹list›*, T‹prefix›_‹eltype›*)
Add an element to the end of the list.

Possible errors: eViewing_Error_IllegalState, eViewing_Error_IllegalArgument, eViewing_Error_UnsupportedOperation

4 Version History

4.1 Changes in Version 2

Changes in Version 2.12

Pdf.Geometry.Integer

- **New** struct `Size`.

Pdf.View

- **New** event `ImageRemovedFromCache`.
- **New** property `ViewportSize`.
- **Removed** parameters `ViewportWidth` and `ViewportHeight` from method `GetRenderedResults`.

Pdf.Content

- **New** class `PlacedImage`.
- **New** class `RenderedPage`.
- **New** class `RenderedPageList`.

Pdf.Content.Image

- **Removed** event `Deleting`.
- **New** property `Id`.

Pdf.Content.RenderResults

- **Removed** property `PageRectsOnViewport`.
- **Removed** property `ImageRectsOnViewport`.
- **Removed** property `Images`.
- **New** property `VisiblePages`.

Changes in Version 2.11

PdfViewing

- **New** class `StringList`.
- **New** class `IntegerSet`.

SDK

- **New** property `SupportedFeatures`.
- **New** enumeration `LicenseFeature`.

Pdf.Navigation.FitMode

- **Changed** enum value `0` from name `ActualSize` to name `None`.

Pdf.Document

- **Changed** behaviour of method `SetFitMode`: No longer sets zoom to 100%, when setting to mode `None` (previously `ActualSize`).

Pdf.Annotations.TextStampType

- **Removed** `CustomTextStamp`.
- **Removed** `CustomImageStamp`.

Pdf.Forms

- **Changed** class `Widget`: moved from namespace `Pdf.Annotations` to `Pdf.Forms`.
- **New** class `CheckBox`.
- **New** class `RadioButton`.
- **New** class `ListBox`.
- **New** class `ComboBox`.

Changes in Version 2.10

Pdf.Document

- **Removed** method `GetTextSelection`.

Pdf.View

- **New** method `GetTextSelection`.
- **Changed** event `ViewUpdated` to have a new parameter `PageNumberChanged`.

Pdf.Geometry.Real.Quadrilateral

- **Renamed** to `QuadrilateralOnPage`.
- **Renamed** field `Page` to `PageNumber`.

Pdf.Geometry.Real.PointOnPage

- **Renamed** field `Page` to `PageNumber`.

Pdf.Geometry.Real.RectangleOnPage

- **Renamed** field `Page` to `PageNumber`.

Pdf.Navigation.DirectDestination

- **Renamed** property `Page` to `PageNumber`.

Pdf.Navigation.FitWidthDestination

- **new** property `Top`.
- **new** parameter `top`

Pdf.Navigation.FitHeightDestination

- **new** property `Left`.
- **new** parameter `Left`

Pdf.Annotations.StickyNote

- **changed** parameter `content` to not accept null in method `Create`.

Pdf.Annotations.TextStamp

- **changed** parameter `text` to not accept null in method `CreateRaw`.

Pdf.Annotations.FreeText

- **changed** parameter `richText` to not accept null in method `Create`.

Changes in Version 2.9

Pdf.ViewerController

- **Removed** property `ControllerState`.
- **Removed** event `ControllerStateChanged`.
- **Removed** parameter `ignoreEmbeddedPreferences` from methods `Open`, `OpenUri` and `OpenMem` .
- **Changed** return value of asynchronous methods `Open`, `OpenUri` and `OpenMem` to `Pdf.Document`.
- **Removed** method `CloseDocument`.
- **Removed** property `Document`.
- **Removed** property `View`.

Pdf.Document

- **New** method `RegisterPdfPage`.
- **Changed** parameter `text` in method `SearchText` to no longer be nullable.
- **Changed** parameters `startPoint` and `endPoint` in method `GetTextSelection` to no longer be nullable.

Pdf.View

- **New** new constructor.
- **Changed** parameter `point` in method `GetTextFragmentsOnPoint` to no longer be nullable.
- **Changed** parameter `point` in method `GetAnnotationsOnPoint` to no longer be nullable.
- **Changed** parameter `pointOnViewport` in method `TransformPointFromViewportToPage` to no longer be nullable.
- **Changed** parameter `pointOnViewport` in method `TransformPointFromViewportToSpecificPage` to no longer be nullable.
- **Changed** parameter `rectangleOnViewport` in method `TransformRectangleFromViewportToSpecificPage` to no longer be nullable.
- **Changed** parameter `pointOnPage` in method `TransformPointFromPageToViewport` to no longer be nullable.
- **Changed** parameter `rectangleOnPage` in method `TransformRectangleFromPageToViewport` to no longer be nullable.
- **Changed** parameter `location` in method `ZoomOnLocation` to no longer be nullable.
- **Changed** property `ScrollPosition` to no longer be nullable.
- **Changed** property `ScrollMaxPosition` to no longer be nullable.

Pdf.Geometry.Real.QuadrilateralList

- **Changed** pararmeter `pointOnPage` in method `Contains` to no longer be nullable.

Pdf.Content.TextFragment

- **Changed** property `RectangleOnPage` to no longer be nullable.

Pdf.Navigation.FitMode

- **Removed** enum value `Invalid`.
- **Changed** enum values `ActualSize`, `FitWidth` and `FitPage` to use values in range 0 to 2 instead of 1 to 3.

Pdf.Navigation.PageLayoutMode

- **Removed** enum value `Invalid`.
- **Changed** enum values `SinglePage`, `OneColumn`, `TwoColumnLeft`, `TwoColumnRight`, `TwoPageLeft` and `TwoPageRight` to use values in range 0 to 5 instead of 1 to 6.

Pdf.Annotations.Stamp

- **New** property `Rotation`.

Pdf.Annotations.TextStamp

- **New** parameter `rotation` in method `CreateRaw`.

Pdf.Annotations.CustomTextStamp

- **New** parameter `rotation` in method `Create`.

Pdf.Annotations.CustomImageStamp

- **New** parameter `rotation` in method `Create`.

Pdf.Annotations.FreeText

- **New** parameter `rotation` in method `Create`.
- **New** property `Rotation`.

Pdf.Annotations.TextMarkup

- **New** property `MarkupArea`.

Pdf.Annotations.Link

- **New** property `ActiveArea`.

Changes up to Version 2.8

Pdf.Content.Image

- **New** event `Deleting`.
- **New** method `RemoveImageData`.
- **New** property `HasImageData`.

5 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.

Contact

PDF Tools AG
Brown-Boveri-Strasse 5
8050 Zürich
Switzerland
http://www.pdf-tools.com
pdfsales@pdf-tools.com