pdf-tools.com
Premium PDF Technology

3-Heights™
PDF Extract Tool
User's Manual

Version 1.50

Contact: pdfsupport@pdf-tools.com

Owner: **PDF Tools AG**

Geerenstrasse 33
CH-8185 Winkel
Switzerland

www.pdf-tools.com

Table of Contents

1 Introduction 7

2 Installation 8

3 Getting started 11

4 Users Guide 12

4.1 How to use the component in a new Visual Basic 6 project 12
4.2 How to use the component in an ASP script 13
4.3 How to setup a new Visual Studio C / C++ project 14
4.4 How to use the component in a Delphi project 15

5 Reference Manual 16

5.1 The Document Interface 16
5.1.1 Property String Author 16
5.1.2 Property Date CreationDate 16
5.1.3 Property String Creator 16
5.1.4 Property TPDFErrorCode ErrorCode 16
5.1.5 Property Boolean IsEncrypted 16
5.1.6 Property String Keywords 16
5.1.7 Property Integer MajorVersion 16
5.1.8 Property Date ModDate 16
5.1.9 Property Integer MinorVersion 16
5.1.10 Property IPDFPage Page 17*
5.1.11 Property Long PageCount 17
5.1.12 Property Long PageNo 17
5.1.13 Property String Producer 17
5.1.14 Property String Subject 17
5.1.15 Property String Title 17
5.1.16 Method Void Close() 17
5.1.17 Method Long GetCurrentOutlineLevel() 17
5.1.18 Method IPDFColorSpace GetFirstColorSpaceResource() 17*
5.1.19 Method IPDFFont GetFirstFontResource() 17*
5.1.20 Method IPDFImage GetFirstImageResource() 18*
5.1.21 Method IPDFOutlineItem GetFirstOutlineItem() 18*
5.1.22 Method String GetInfoEntry(String szKey) 18
5.1.23 Method IPDFColorSpace GetNextColorSpaceResource() 18*
5.1.24 Method IPDFFont GetNextFontResource() 18*
5.1.25 Method IPDFImage GetNextImageResource() 18*
5.1.26 Method IPDFOutlineItem GetNextOutlineItem(Long MaxLevel, Boolean ReturnOpenOnly) 18*
5.1.27 Method Boolean Open(String FileName, String Password) 18
5.1.28 Method Boolean OpenMem(Variant MemBlock, String Password) 19
5.2 The Page Interface 20

5.2.1 Property Variant ArtBox 20
5.2.2 Property Variant BleedBox 20
5.2.3 Property IPDFContent* Content 20
5.2.4 Property Variant CropBox 20
5.2.5 Property String DeviceColorant 20
5.2.6 Property IPDFDocument* Document 20
5.2.7 Property Variant MediaBox 20
5.2.8 Property Short Rotate 21
5.2.9 Property Variant TrimBox 21
5.2.10 Method Annotation GetFirstAnnotation() 21
5.2.11 Method Annotation GetNextAnnotation() 21
5.3 The Content Interface 22
5.3.1 Property Boolean BreakWords 22
5.3.2 Property Long Flags 22
5.3.3 Property IPDFGraphicsState* GraphicsState 22
5.3.4 Property IPDFImage* Image 22
5.3.5 Property String Path 22
5.3.6 Property IPDFText* Text 23
5.3.7 Method IPDFImage* GetNextImage() 23
5.3.8 Method String GetNextPath() 23
5.3.9 Method TPDFContentObject GetNextObject() 23
5.3.10 Method IPDFText* GetNextText() 24
5.3.11 Method Void Reset(Boolean AccountForRotate) 24
5.4 The Image Interface 25
5.4.1 Property Alternates 25
5.4.2 Property BitsPerComponent 25
5.4.3 Property IPDFColorSpace* ColorSpace 25
5.4.4 Property Long Height 25
5.4.5 Property Boolean IsBitonal 25
5.4.6 Property Boolean IsColor 25
5.4.7 Property Boolean IsMonochrome 25
5.4.8 Property Variant Mask 25
5.4.9 Property Variant Samples 25
5.4.10 Property Long Width 26
5.4.11 Method ChangeOrientation(TPDFOrientation Orientation) 26
5.4.12 Method IPDFImage* ConvertToRGB() 26
5.4.13 Method Variant GetImage() 26
5.4.14 Method Single GetResolution(IPDFTransformMatrix* Matrix) 26
5.4.15 Method Boolean Store(String FileName) 26
5.4.16 Method Boolean StoreInMemory(String Extension, TPDFCompression Compression) 27
5.5 The Text Interface 28
5.5.1 Property Variant BoundingBox 28
5.5.2 Property Single FontSize 28
5.5.3 Property Integer Length 28
5.5.4 Property Variant NextXPos (deprecated since 1.41) 28
5.5.5 Property Variant NextYPos (deprecated since in 1.41) 28

5.5.6 Property String RawString (removed in 1.41) 28
5.5.7 Property Single Rotation 28
5.5.8 Property TextMatrix (deprecated since 1.41) 28
5.5.9 Property String UnicodeString 29
5.5.10 Property Single Width 29
5.5.11 Property Variant XPos 29
5.5.12 Property Variant YPos 29
5.6 The GraphicsState Interface 30
5.6.1 Property Single CharSpacing 30
5.6.2 Property IPDFTransformMatrix* CTM 30
5.6.3 Property Variant DashArray 30
5.6.4 Property Single DashPhase 30
5.6.5 Property Long FillColorCMYK 30
5.6.6 Property Long FillColorRGB 30
5.6.7 Property IPDFColorSpace* FillColorSpace 30
5.6.8 Property IPDFFont* Font 30
5.6.9 Property Single FontSize 31
5.6.10 Property Single HorizontalScaling 31
5.6.11 Property Single Leading 31
5.6.12 Property Single LineWidth 31
5.6.13 Property Short LineCap 31
5.6.14 Property Short LineJoin 31
5.6.15 Property Single MiterLimit 32
5.6.16 Property Long StrokeColorCMYK 32
5.6.17 Property Long StrokeColorRGB 32
5.6.18 Property IPDFColorSpace* StrokeColorSpace 32
5.6.19 Property Boolean TextKnockout 32
5.6.20 Property Short TextRenderingMode 32
5.6.21 Property Single TextRise 33
5.6.22 Property Single WordSpacing 33
5.7 The Font Interface 34
5.7.1 Property Single Ascent 34
5.7.2 Property Single AvgWidth 34
5.7.3 Property String BaseName 34
5.7.4 Property Single CapHeight 34
5.7.5 Property String Charset 34
5.7.6 Property Single Descent 34
5.7.7 Property Variant Encoding 34
5.7.8 Property Short FirstChar (removed in 1.41) 34
5.7.9 Property Long Flags 34
5.7.10 Property Variant FontBBox 35
5.7.11 Property Variant FontFile 35
5.7.12 Property Integer FontFileType 35
5.7.13 Property Single ItalicAngle 35
5.7.14 Property Short LastChar (removed in 1.41) 35
5.7.15 Property Single Leading 35
5.7.16 Property Single MaxWidth 35

5.7.17 Property Single MissingWidth 35
5.7.18 Property Single StemV 35
5.7.19 Property Single StemH 35
5.7.20 Property String Type 36
5.7.21 Property Variant Widths 36
5.7.22 Property Single XHeight 36
5.8 The ColorSpace Interface 37
5.8.1 Property IPDFColorSpace BaseColorSpace 37*
5.8.2 Property Short ComponentsPerPixel 37
5.8.3 Property Short High 37
5.8.4 Property Boolean IsColor 37
5.8.5 Property Boolean IsIndexed 37
5.8.6 Property Boolean IsMonochrome 37
5.8.7 Property Variant Lookup 37
5.8.8 Property String Name 37
5.9 The TransformMatrix Interface 38
5.9.1 Property a, b, c, d, e, f 38
5.9.2 Property Orientation 38
5.9.3 Property Rotation 38
5.9.4 Property XScaling, YScaling 38
5.9.5 Property XSkew, YSkew 38
5.9.6 Property XTranslation, YTranslation 38
5.10 The Alternate Image Interface 39
5.10.1 Property Boolean DefaultForPrinting 39
5.10.2 Property IPDFImage Image 39*
5.11 The Annotation Interface 40
5.11.1 Property Long Color 40
5.11.2 Property String Contents 40
5.11.3 Property Date Date 40
5.11.4 Property Long Flags 40
5.11.5 Property String Name 40
5.11.6 Property Variant Rect 40
5.11.7 Property String Subtype 40
5.11.8 Property String TextLabel 40
5.12 The Outline Interface 41
5.12.1 Property Long Count 41
5.12.2 Property IPDFDestination Dest 41*
5.12.3 Property String Title 41
5.13 The Destination Interface 42
5.13.1 Property Single Bottom 42
5.13.2 Property Single Left 42
5.13.3 Property Long PageNo 42
5.13.4 Property Single Right 42
5.13.5 Property Single Top 42
5.13.6 Property String Type 42
5.13.7 Property Single Zoom 42

5.14 Enumerations 43
5.14.1 Enum TPDFCompression 43
5.14.2 Enum TPDFContentObject 43
5.14.3 Enum TPDFOrientation 43
5.14.4 Enum TPDFErrorCode 43

6 Interface changes 46

6.1 Changes from 1.40 to 1.41 46
6.2 Changes from 1.41 to 1.50 47

7 Code Samples 47

1 Introduction

The purpose of the 3-Heights™ PDF Extract Tool is to read PDF files and navigate through the documents pages and the outlines. It provides means to extract text, graphics and image objects as well as detailed information about the objects color spaces, fonts etc. The main features of the component are:

- Extract text contained on a PDF page, line-wise and word-wise
- Retrieve text attributes such as position and font
- Extract graphics objects (paths)
- Extract images
- Retrieve PDF image attributes such as format, position and transparency masks
- Retrieve PDF document attributes such as page count, version number, and title
- Retrieve PDF page attributes such as the Crop Box and page rotation
- Retrieve detailed font information from PDF text
- Retrieve detailed graphics state information
- Retrieve detailed color space information
- Specify a password to decrypt PDF files

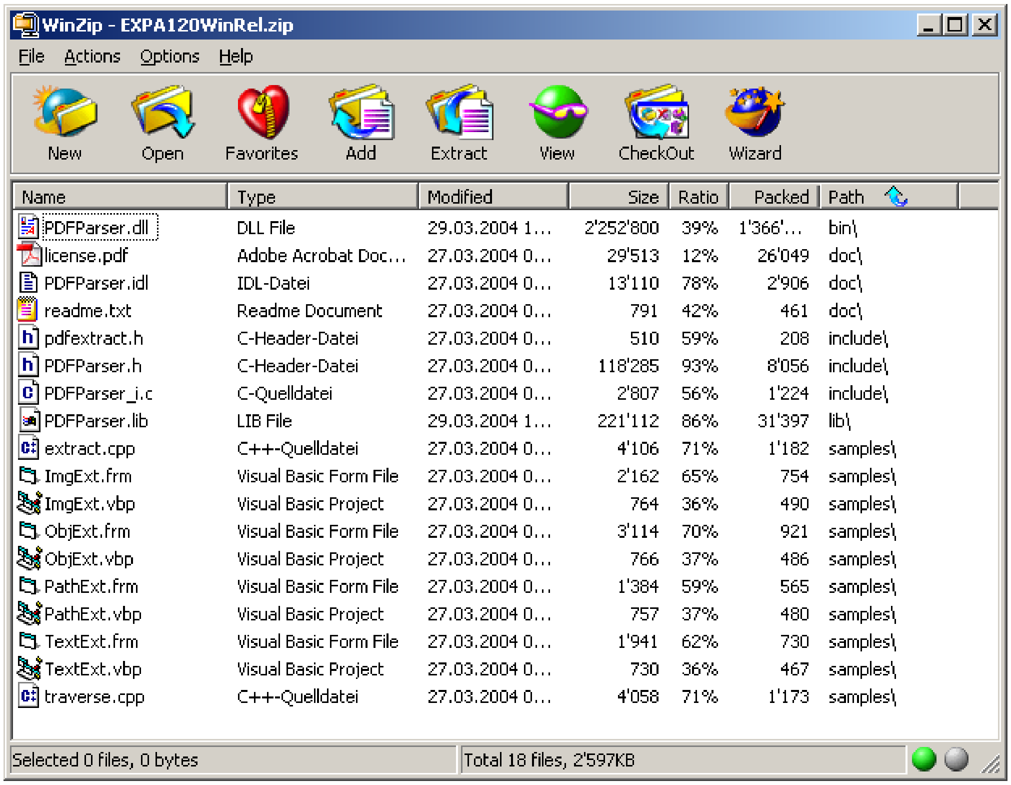
The component provides COM interfaces that can be used in any programming language that supports the COM technology such as C++, Visual Basic, Delphi, ASP and many more. There is also a native C and a Java interface.

There exist interfaces to documents, pages, page contents, images, text, graphic states, fonts, color spaces, transformation matrices, destination, outline, annotations and alternate images.

Most of the objects and their attributes such as the pages of a document or the sample data of an image can be randomly accessed. Some objects such as text, image and graphics objects only can be accessed sequentially while traversing the page content streams.

2 Installation

The retail version of the 3-Heights™ PDF Extract Tool API comes as a ZIP archive containing various files including runtime binary executable code, files required for the developer, documentation and license terms.

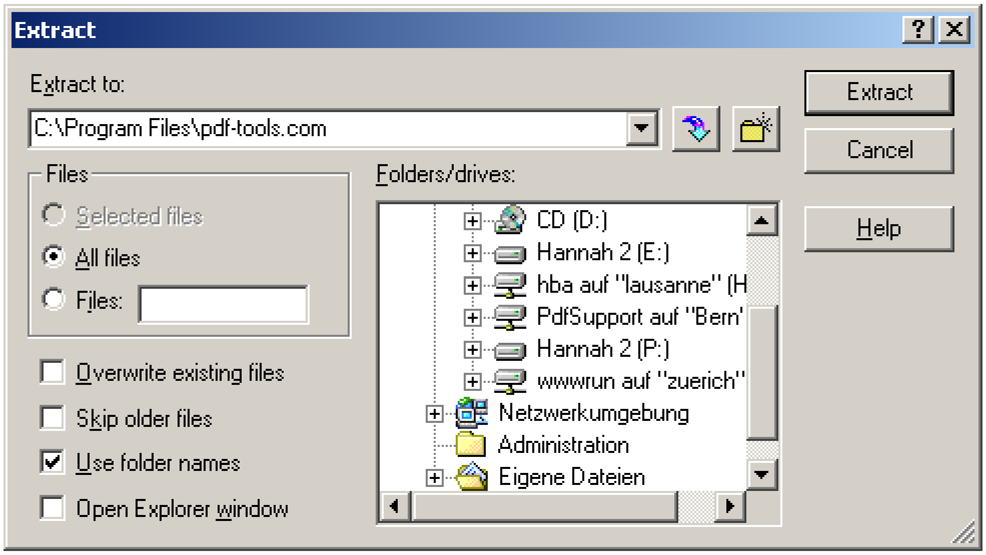


Name	Type	Modified	Size	Ratio	Packed	Path
PDFParser.dll	DLL File	29.03.2004 1...	2'252'800	39%	1'366'...	bin\
license.pdf	Adobe Acrobat Doc...	27.03.2004 0...	29'513	12%	26'049	doc\
PDFParser.idl	IDL-Datei	27.03.2004 0...	13'110	78%	2'906	doc\
readme.txt	Readme Document	27.03.2004 0...	791	42%	461	doc\
pdfextract.h	C-Header-Datei	27.03.2004 0...	510	59%	208	include\
PDFParser.h	C-Header-Datei	27.03.2004 0...	118'285	93%	8'056	include\
PDFParser_i.c	C-Quelldatei	27.03.2004 0...	2'807	56%	1'224	include\
PDFParser.lib	LIB File	29.03.2004 1...	221'112	86%	31'397	lib\
extract.cpp	C++-Quelldatei	27.03.2004 0...	4'106	71%	1'182	samples\
ImgExt.frm	Visual Basic Form File	27.03.2004 0...	2'162	65%	754	samples\
ImgExt.vbp	Visual Basic Project	27.03.2004 0...	764	36%	490	samples\
ObjExt.frm	Visual Basic Form File	27.03.2004 0...	3'114	70%	921	samples\
ObjExt.vbp	Visual Basic Project	27.03.2004 0...	766	37%	486	samples\
PathExt.frm	Visual Basic Form File	27.03.2004 0...	1'384	59%	565	samples\
PathExt.vbp	Visual Basic Project	27.03.2004 0...	757	37%	480	samples\
TextExt.frm	Visual Basic Form File	27.03.2004 0...	1'941	62%	730	samples\
TextExt.vbp	Visual Basic Project	27.03.2004 0...	730	36%	467	samples\
traverse.cpp	C++-Quelldatei	27.03.2004 0...	4'058	71%	1'173	samples\

Selected 0 files, 0 bytes

Total 18 files, 2'597KB

First unzip the file using a tool like WinZip available from WinZip Computing, Inc. at http://www.winzip.com to a directory on your hard disk where your program files reside as indicated below:



Extract

Extract to: C:\Program Files\pdf-tools.com

Files

○ Selected files

● All files

○ Files:

□ Overwrite existing files

□ Skip older files

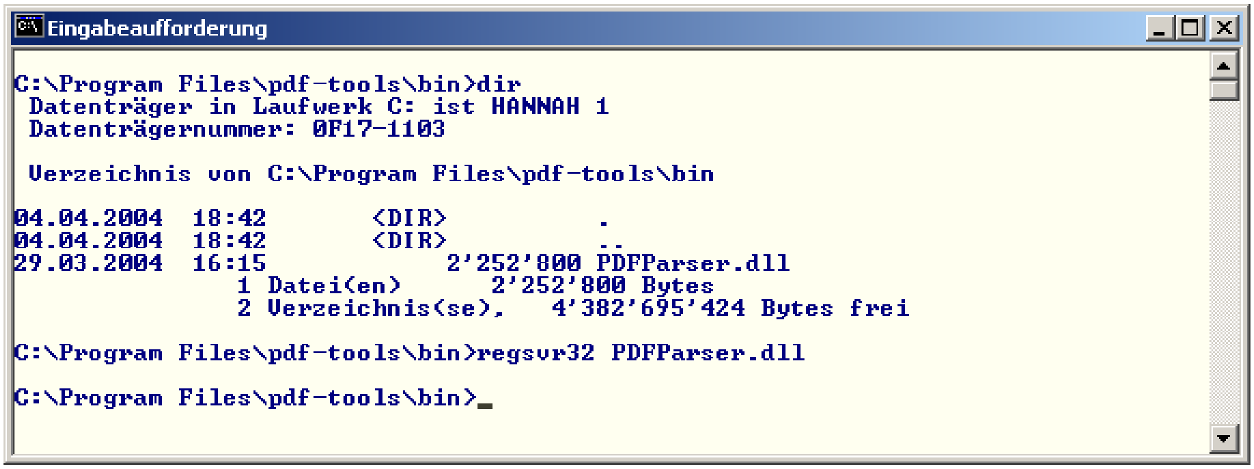
☑ Use folder names

□ Open Explorer window

Check the appropriate option to preserve file paths (folder names). The unzip process now creates the following subdirectories

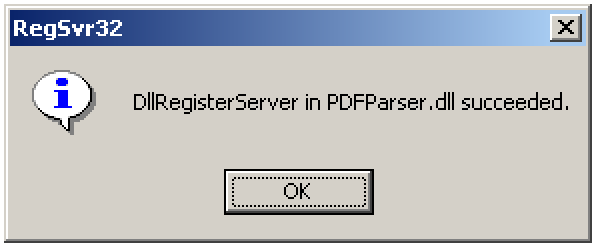
- Bin: Contains the runtime executable binary code.
  - pdparser.dll: This is the DLL that contains the main functionality (required).
  - pdcjk.dll: This DLL contains support for Asian languages (optional). It is loaded from the module path.
  - unicows.dll: This DLL contains the Microsoft Unicode Layer for Windows98 (required for Windows 9x). It is not loaded from module path and should be included in the environment variable PATH.
- Doc: Contains documentation files
- Include : Contains files to include in your C / C++ project
- Lib: Contains a stub library to link with your binary executable
- Samples: Contains Visual Basic 6.0 and C / C++ sample programs

Before you can use the 3-Heights™ PDF Extract Tool API component in your COM application program you have to register the component using the regsvr32 program that is provided with the Windows operating system.



```
C:\Program Files\pdf-tools\bin>dir
 Datenträger in Laufwerk C: ist HANNAH 1
 Datenträgernummer: 0F17-1103

 Verzeichnis von C:\Program Files\pdf-tools\bin

04.04.2004  18:42       <DIR>          .
04.04.2004  18:42       <DIR>          ..
29.03.2004  16:15            2’252’800 PDFParser.dll
               1 Datei(en)      2’252’800 Bytes
               2 Verzeichnis(se),   4’382’695’424 Bytes frei

C:\Program Files\pdf-tools\bin>regsvr32 PDFParser.dll

C:\Program Files\pdf-tools\bin>_
```

If the registration process succeeds the following box is displayed:

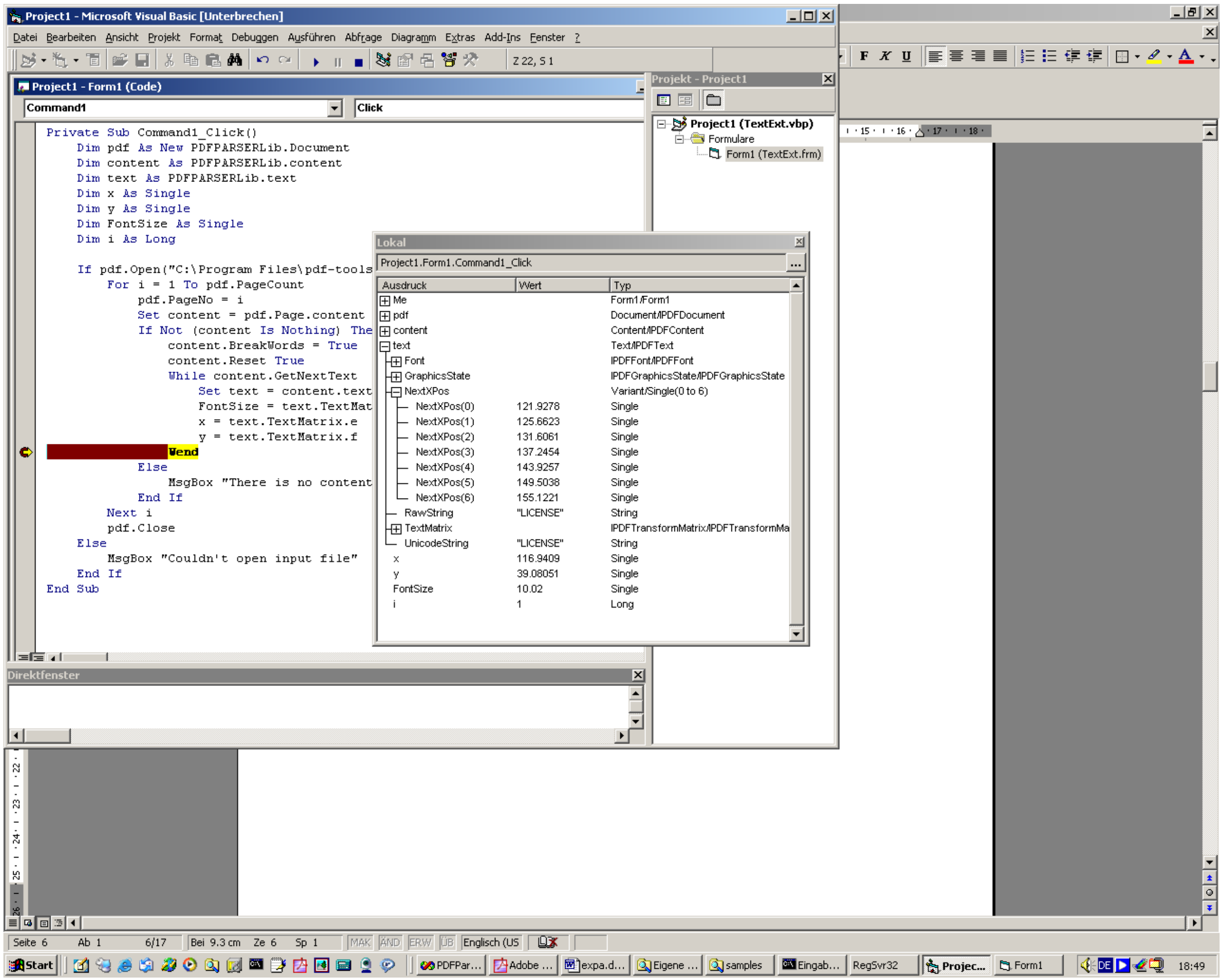


DllRegisterServer in PDFParser.dll succeeded.

The installation process is now complete.

3 Getting started

You can observe the function of the 3-Heights™ PDF Extract Tool API component if you open the TextExt Visual Basic 6.0 project. In order to run the program you have to edit the file name used in the pdf.Open call such that a known document, e.g. the license document is referenced. Now set a break point at the Wend statement and you can now step through the extracted text objects and display the properties in the local window as shown below.


Project1 - Microsoft Visual Basic [Unterbrechen]
Datei Bearbeiten Ansicht Projekt Format Debuggen Ausführen Abfrage Diagramm Extras Add-Ins Fenster ?
Z 22, S 1
Project1 - Form1 (Code)
Command1
Click
Projekt - Project1
Project1 (TextExt.vbp)
Formulare
Form1 (TextExt.frm)
Lokal
Direktfenster
Seite 6 Ab 1 6/17 Bei 9.3 cm Ze 6 Sp 1 MAK ÄND ERW ÜB Englisch (US
Start PDFPar... Adobe ... expa.d... Eigene ... samples Eingab... RegSvr32 Projec... Form1 18:49

```
Private Sub Command1_Click()
    Dim pdf As New PDFPARSERLib.Document
    Dim content As PDFPARSERLib.content
    Dim text As PDFPARSERLib.text
    Dim x As Single
    Dim y As Single
    Dim FontSize As Single
    Dim i As Long

    If pdf.Open("C:\Program Files\pdf-tools
        For i = 1 To pdf.PageCount
            pdf.PageNo = i
            Set content = pdf.Page.content
            If Not (content Is Nothing) The
                content.BreakWords = True
                content.Reset True
                While content.GetNextText
                    Set text = content.text
                    FontSize = text.TextMat
                    x = text.TextMatrix.e
                    y = text.TextMatrix.f
                Wend
            Else
                MsgBox "There is no content
            End If
        Next i
        pdf.Close
    Else
        MsgBox "Couldn't open input file"
    End If
End Sub
```

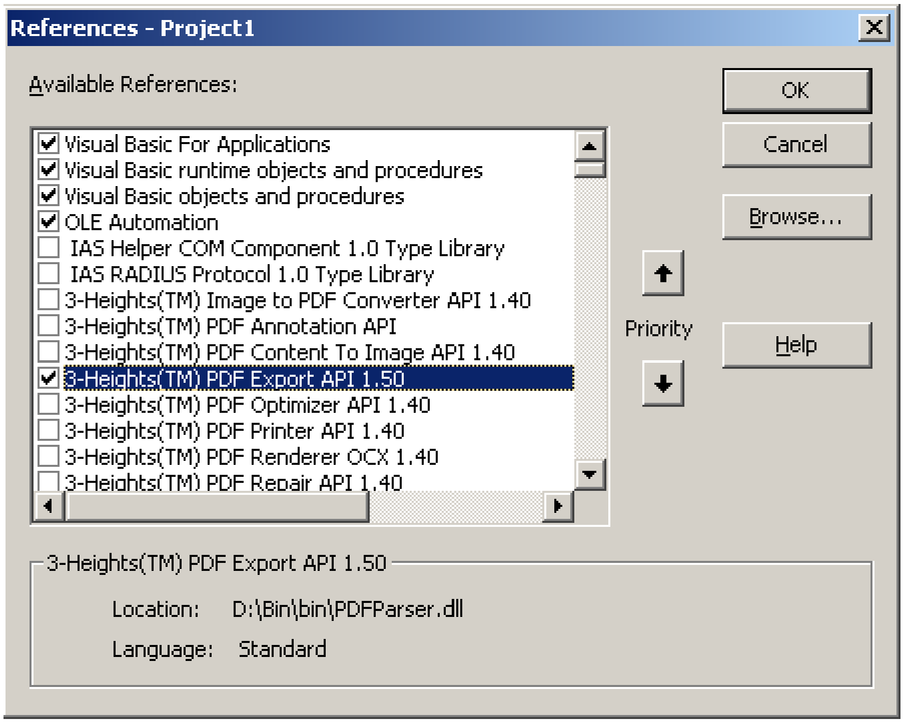
Project1.Form1.Command1_Click

Ausdruck	Wert	Typ
⊞ Me		Form1/Form1
⊞ pdf		Document/IPDFDocument
⊞ content		Content/IPDFContent
⊟ text		Text/IPDFText
⊞ Font		IPDFFont/IPDFFont
⊞ GraphicsState		IPDFGraphicsState/IPDFGraphicsState
⊟ NextXPos		Variant/Single(0 to 6)
NextXPos(0)	121.9278	Single
NextXPos(1)	125.6623	Single
NextXPos(2)	131.6061	Single
NextXPos(3)	137.2454	Single
NextXPos(4)	143.9257	Single
NextXPos(5)	149.5038	Single
NextXPos(6)	155.1221	Single
RawString	"LICENSE"	String
⊞ TextMatrix		IPDFTransformMatrix/IPDFTransformMa
UnicodeString	"LICENSE"	String
x	116.9409	Single
y	39.08051	Single
FontSize	10.02	Single
i	1	Long

4 Users Guide

4.1 How to use the component in a new Visual Basic 6 project

In order to use the component in a Visual Basic 6 project, you have to add the component as a project reference as shown below. The version which is registered will show up.



Available References:

- ☑ Visual Basic For Applications
- ☑ Visual Basic runtime objects and procedures
- ☑ Visual Basic objects and procedures
- ☑ OLE Automation
- □ IAS Helper COM Component 1.0 Type Library
- □ IAS RADIUS Protocol 1.0 Type Library
- □ 3-Heights(TM) Image to PDF Converter API 1.40
- □ 3-Heights(TM) PDF Annotation API
- □ 3-Heights(TM) PDF Content To Image API 1.40
- ☑ 3-Heights(TM) PDF Export API 1.50
- □ 3-Heights(TM) PDF Optimizer API 1.40
- □ 3-Heights(TM) PDF Printer API 1.40
- □ 3-Heights(TM) PDF Renderer OCX 1.40
- □ 3-Heights(TM) PDF Repair API 1.40

3-Heights(TM) PDF Export API 1.50

Location: D:\Bin\bin\PDFParser.dll

Language: Standard

4.2 How to use the component in an ASP script

The PDF Extract component can be accessed in an ASP script using the call *Server.CreateObject* and a class name as parameter. For example to create PDF Extract *Document* object, use a command like this:

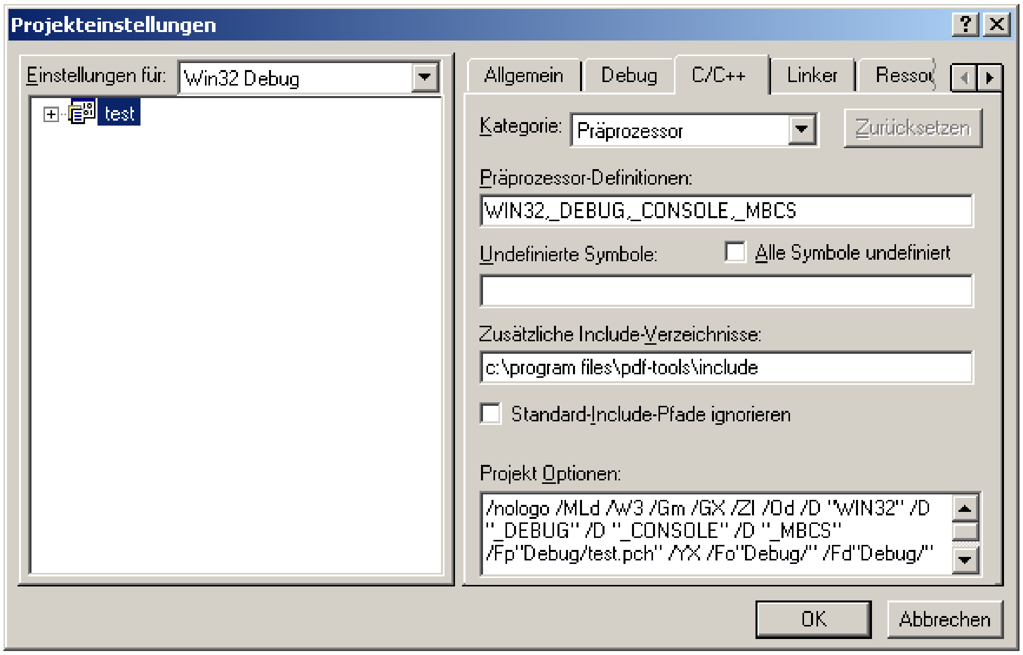
set pdfDoc = Server.CreateObject("PDFParser.Document")

Here is a small ASP sample how to create a Document object and then retrieve the total number of pages in a PDF file. The path to the PDF (“myfile.pdf”) needs to be modified.

```
<%@ Language=VBScript %>
<%
    option explict
    dim pdfDoc
    set pdfDoc = Server.CreateObject("PDFParser.Document")
    if not pdfDoc.Open(Server.Mappath("myfile.pdf")) then
     Response.Write "<p>"
      Response.Write "Could not open  file." & "<br>"
    end if
    Response.Write "<p>"
    Response.Write "Number of pages: " & pdfDoc.PageCount & "<br>"
    Response.Write "</p>"
%>
```

4.3 How to setup a new Visual Studio C / C++ project

When you create a new Visual Studio 6.0 C / C++ project file you have to change the project settings such that the compiler can access the component's include directory as shown below.



Projekteinstellungen

Einstellungen für: Win32 Debug

Kategorie: Präprozessor

Präprozessor-Definitionen: WIN32,_DEBUG,_CONSOLE,_MBCS

Undefinierte Symbole:

□ Alle Symbole undefiniert

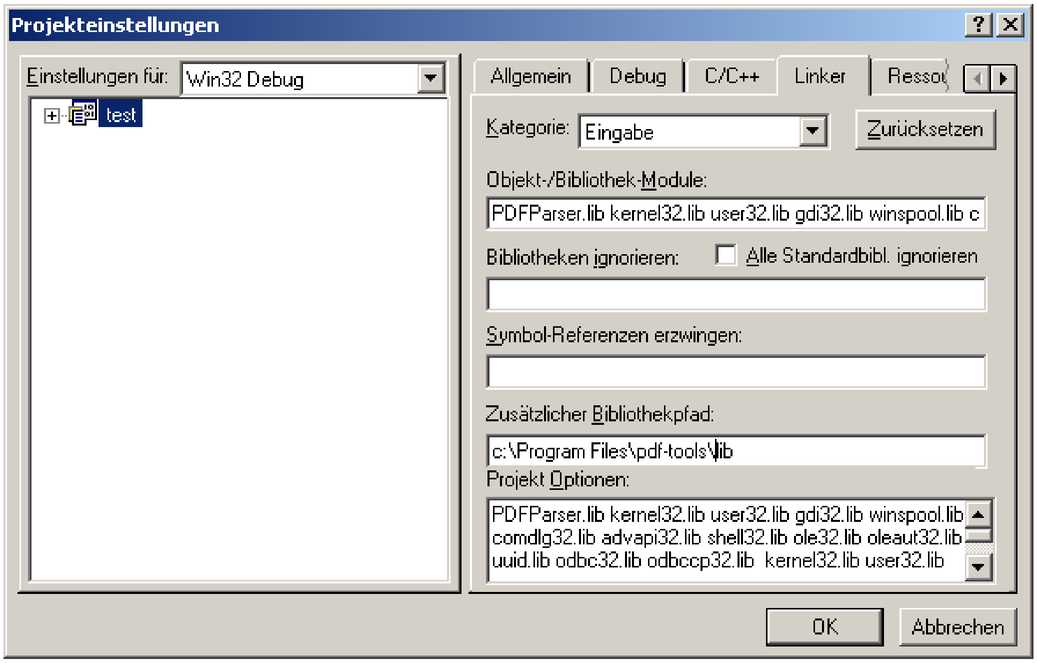
Zusätzliche Include-Verzeichnisse: c:\program files\pdf-tools\include

□ Standard-Include-Pfade ignorieren

Projekt Optionen:

```
/nologo /MLd /W3 /Gm /GX /ZI /Od /D "WIN32" /D
"_DEBUG" /D "_CONSOLE" /D "_MBCS"
/Fp"Debug/test.pch" /YX /Fo"Debug/" /Fd"Debug/"
```

In addition to the include directory you have to tell the linker the location of the stub library and specify the name of the library as an additional library module as shown below.



Projekteinstellungen

Einstellungen für: Win32 Debug

Kategorie: Eingabe

Objekt-/Bibliothek-Module: PDFParser.lib kernel32.lib user32.lib gdi32.lib winspool.lib c

Bibliotheken ignorieren:

□ Alle Standardbibl. ignorieren

Symbol-Referenzen erzwingen:

Zusätzlicher Bibliothekpfad: c:\Program Files\pdf-tools\lib

Projekt Optionen:

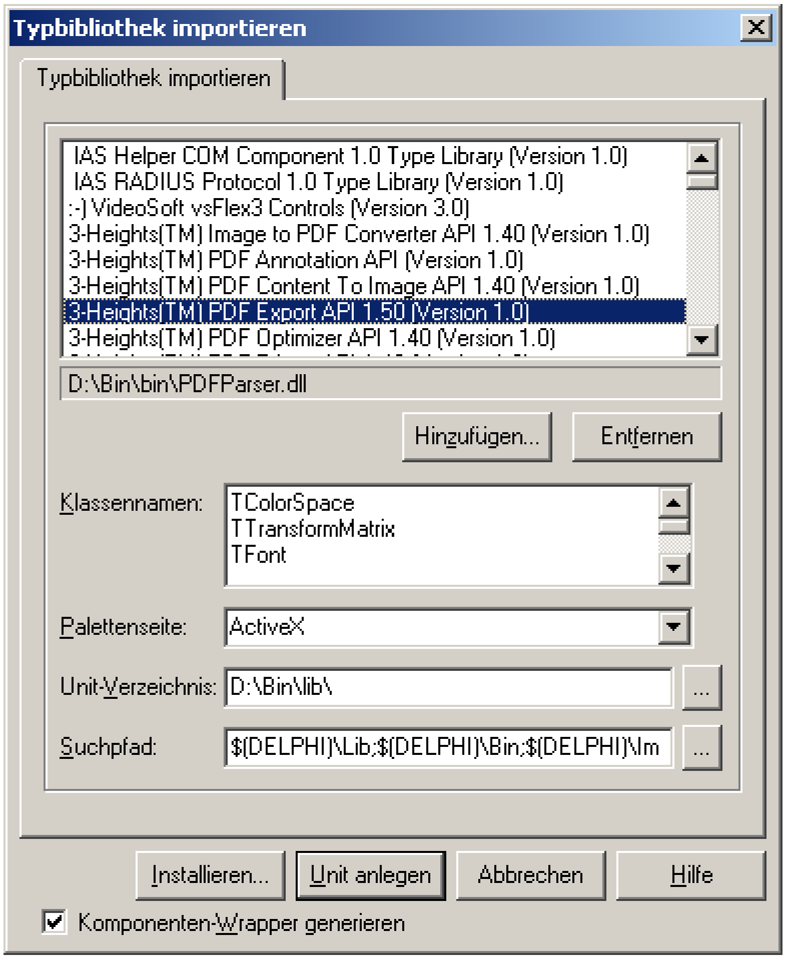
```
PDFParser.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib odbc32.lib odbccp32.lib  kernel32.lib user32.lib
```

4.4 How to use the component in a Delphi project

1. Once the DLL is registered, start your Delphi application.

2. Go to the menu *Project* and select *Import Type Library*.

3. Select the appropriate 3-Heights component and click *Install*. Then click *Create Unit*.



IAS Helper COM Component 1.0 Type Library (Version 1.0)
IAS RADIUS Protocol 1.0 Type Library (Version 1.0)
:-) VideoSoft vsFlex3 Controls (Version 3.0)
3-Heights(TM) Image to PDF Converter API 1.40 (Version 1.0)
3-Heights(TM) PDF Annotation API (Version 1.0)
3-Heights(TM) PDF Content To Image API 1.40 (Version 1.0)
3-Heights(TM) PDF Export API 1.50 (Version 1.0)
3-Heights(TM) PDF Optimizer API 1.40 (Version 1.0)

D:\Bin\bin\PDFParser.dll

Klassennamen: TColorSpace
TTransformMatrix
TFont

Palettenseite: ActiveX

Unit-Verzeichnis: D:\Bin\lib\

Suchpfad: $(DELPHI)\Lib;$(DELPHI)\Bin;$(DELPHI)\Im

☑ Komponenten-Wrapper generieren

4. This should now create a new icon in the menu *ActiveX*.

5. Now you can open the Delphi sample which is included in *samples/Delphi/*, or create a new sample from scratch.

5 Reference Manual

5.1 The Document Interface

5.1.1 Property String Author

This property returns the author from the document's info object.

5.1.2 Property Date CreationDate

This property returns the creation date of the document's info object.

5.1.3 Property String Creator

This property returns the name of the creator of the document's info object.

5.1.4 Property TPDFErrorCode ErrorCode

This property can be accessed to receive the latest error code. Any return value other than 0 indicates that an error occurred. See Enum TPDFErrorCode for a detailed list of error codes, or refer to "pdferror.h" for the latest list.

5.1.5 Property Boolean IsEncrypted

This property returns true if the PDF document is encrypted.

5.1.6 Property String Keywords

This property a string with the keywords of the document's info object.

5.1.7 Property Integer MajorVersion

This property returns the major version of the document. (Ex. PDF Version 1.5 corresponds to Adobe Acrobat 6, the major version is 1, the minor is 5)

5.1.8 Property Date ModDate

This property returns the modification date of the document's info object.

5.1.9 Property Integer MinorVersion

This property returns the minor version of the document.

5.1.10 Property IPDFPage* Page

This property allows to retrieve an interface to the currently selected page of a document.

5.1.11 Property Long PageCount

The property PageCount returns the number of pages of an open document. If the document is closed then zero is returned. This property cannot be set.

5.1.12 Property Long PageNo

This property allows to set and get the currently selected page of an open document given its page number. The numbers are counted from 1 for the first page to the value of the PageCount attribute for the last page. If the document is closed zero is returned.

5.1.13 Property String Producer

This property returns the name of the producer from the document's info object.

5.1.14 Property String Subject

This property returns the subject from the document's info object.

5.1.15 Property String Title

This property returns the title from the document's info object.

5.1.16 Method Void Close()

This Method closes an open document. If the document is already closed the method does nothing.

5.1.17 Method Long GetCurrentOutlineLevel()

This method returns the level of the current outline (bookmark). To root is at level 0.

5.1.18 Method IPDFColorSpace* GetFirstColorSpaceResource()

This method returns an interface to the first color space resource (see ColorSpace Interface).

5.1.19 Method IPDFFont* GetFirstFontResource()

This method returns in interface to the first font resource (see Font Interface).

5.1.20 Method IPDFImage* GetFirstImageResource()

This method returns an interface to the first image resource (see Image Interface).

5.1.21 Method IPDFOutlineItem* GetFirstOutlineItem()

This method returns an interface to the first outline item (see Outline Interface).

5.1.22 Method String GetInfoEntry(String szKey)

This method returns the value of a custom entry in the info object.

5.1.23 Method IPDFColorSpace* GetNextColorSpaceResource()

This method returns an interface to the next color space resource.

5.1.24 Method IPDFFont* GetNextFontResource()

This method returns an interface to the next font resource.

5.1.25 Method IPDFImage* GetNextImageResource()

This method returns an interface to the next image resource.

5.1.26 Method IPDFOutlineItem* GetNextOutlineItem(Long MaxLevel, Boolean ReturnOpenOnly)

This method returns an interface to the next outline item.

- Parameters:
  - MaxLevel (optional, default 20): The maximum level of the depth of the outlines.
  - ReturnOpenOnly (optional, default false): Return only outlines which are opened.

5.1.27 Method Boolean Open(String FileName, String Password)

This method opens a PDF random access disk file, i.e. makes the objects contained in the PDF document accessible. If the document is already open it is closed first.

- Parameters:
  - FileName: the file name and optionally the file path, drive or server string according to the operating systems file name specification rules.
  - Password (optional): the user or the owner password of the encrypted PDF document. If this parameter is left out an empty string is used as a default.

- Return value:
  - True: The file could successfully be opened.
  - False: The file does not exists, it is corrupt, or the password is invalid.

5.1.28 Method Boolean OpenMem(Variant MemBlock, String Password)

This method opens a PDF memory block, i.e. makes the objects contained in the PDF document accessible. If the document is already open it is closed first.

- Parameters:
  - MemBlock: the memory block containing the PDF file given as a one dimensional byte array.
  - Password (optional): the user or the owner password of the encrypted PDF document. If this parameter is left out an empty string is used as a default.
- Return value:
  - True: The file could successfully be opened.
  - False: The file does not exists, it is corrupt, or the password is invalid.

5.2 The Page Interface

5.2.1 Property Variant ArtBox

This property returns the art box rectangle given by the coordinates left, bottom, right, top. The values are returned as an array of four single precision real numbers. The art box is optional, it defines the region that contains meaningful content intended by the creator. If there is no art box set, the crop box is returned. This property cannot be set.

5.2.2 Property Variant BleedBox

This property returns the bleed box rectangle given by the coordinates left, bottom, right, top. The values are returned as an array of four single precision real numbers. The bleed box is optional, it defining the region to which the contents of the page should be clipped when output in a production environment. If there is no bleed box set, the crop box is returned. This property cannot be set.

5.2.3 Property IPDFContent* Content

This property returns an interface to the page's content stream (see Content Interface). This property cannot be set.

5.2.4 Property Variant CropBox

This property returns the crop box rectangle given by the coordinates left, bottom, right, top. The values are returned as an array of four single precision real numbers. The crop box is optional, it defines the range of the visible region of the page. If there is no crop box set, the media box is returned. This property cannot be set.

5.2.5 Property String DeviceColorant

This property returns the device colorant. This property cannot be set.

5.2.6 Property IPDFDocument* Document

This property returns the interface to the page's document (see Document interface). This property cannot be set.

5.2.7 Property Variant MediaBox

This property returns the media box rectangle given by the coordinates left, bottom, right, top. The values are returned as an array of four single precision real numbers. The media box is required, it defines the physical boundaries of the medium on which the page is intended to be displayed or printed. This property cannot be set.

5.2.8 Property Short Rotate

This property returns the page's rotation value. This value is used by viewer programs to turn the page by the given number of degrees while displaying. A positive number turns the page clockwise. The value must be a multiple of 90, i.e. valid values are –270, -180, -90, 0, 90, 180, 270. This property cannot be set.

5.2.9 Property Variant TrimBox

This property returns the trim box rectangle given by the coordinates left, bottom, right, top. The values are returned as an array of four single precision real numbers. The trim box is optional, it defines the intended dimensions of the finished page after trimming. If there is no trim box set, the crop box is returned. This property cannot be set.

5.2.10 Method Annotation GetFirstAnnotation()

This method returns an interface to the first annotation (see Annotation Interface).

5.2.11 Method Annotation GetNextAnnotation()

This method returns an interface to the next annotation.

5.3 The Content Interface

5.3.1 Property Boolean BreakWords

This property serves to control the way how the text extraction algorithm works. Text extraction collects all text objects in a PDF stream and merges them into a single text object as long as a major text attribute such as font size, line coordinate etc. doesn't change. In addition to this the property's value controls the word breaking as follows:

- True: If a word breaking character such as a blank is found or the character spacing exceeds the blank character width the text objects are split up into individual text objects without blanks.
- False: Line breaking characters are treated the same as other characters and character spacing is disregarded.

5.3.2 Property Long Flags

This property returns –1 while content is parsed and the annotation flags when annotations are parsed. (see also Property Long Flags in the Annotation interface)

5.3.3 Property IPDFGraphicsState* GraphicsState

This property returns an interface to the content's graphics state (see GraphicsState Interface). The graphics state is updated each time a GetNextText, GetNextImage, GetNextPath, or GetNextObject method is called. This property cannot be set.

5.3.4 Property IPDFImage* Image

This property returns an interface to the last read image object (see Image Interface). The image object is updated each time a GetNextImage or GetNextObject method is called. This property cannot be set.

5.3.5 Property String Path

This property returns the last read path object in its string form. The path object describes a graphic drawing consisting of stroked lines and curves as well as filled shapes. The string contains the PDF path construction tokens consisting of real value operands (in angle brackets) followed by operator mnemonics:

- Move current point to: <x> <y> m
- Line from current point to: <x> <y> l
- Rectangle: <x> <y> <w> <h> r
- Cubic Bezier curve from current point to: <x1> <y1> <x2> <y2> <x3> <y3> c
- Close figure (move to start of last sub-path): h

- Fill path: f
- Stroke path: s
- Close path (without filling and stroking): n
- Add path to the current clipping region: w

The exact details to the path construction operators can be found in Adobe's PDF Reference Manual. The path object is updated each time a GetNextPath or GetNextObject method is called. This property cannot be set.

5.3.6 Property IPDFText* Text

This property returns an interface to the last read text object (see Text Interface). The text object is updated each time a GetNextText or GetNextObject method is called. This property cannot be set.

5.3.7 Method IPDFImage* GetNextImage()

This method reads the content stream objects until an image object can be returned or the end of the content stream is reached. If an image object could be found, an interface to the image object (see Image Interface) is returned. Its interface can also be retrieved through the content's Image property. The graphics state can be retrieved through the content's GraphicsState property.

5.3.8 Method String GetNextPath()

This method reads the content stream objects until a path object can be returned or the end of the content stream is reached. If a path object could be found, a string representation of a path object is returned. It can also be retrieved through the content's Path property. The graphics state can be retrieved through the content's GraphicsState property.

5.3.9 Method TPDFContentObject GetNextObject()

This method reads the content stream objects until a text, image, or path object can be returned or the end of the content stream is reached.

- Return values:
    - eText: A text object could be composed and its interface can be retrieved through the content's Text property.
    - eImage: An image object could be found and its interface can be retrieved through the content's Image property. The graphics state can be retrieved through the content's GraphicsState property.
    - ePath: A path object could be found and its string representation can be retrieved through the content's Path property. The graphics state can be retrieved through the content's GraphicsState property.
    - eNone: The end of the content stream has been reached and the

content's Path property doesn't return a valid value.

5.3.10 Method IPDFText* GetNextText()

This method reads the content stream objects until a text object can be returned or the end of the content stream is reached. If a text object can be found, an interface to the next read text object (see Text Interface) is returned. In contrast to the GetNextImage and GetNextPath methods this method reads text objects and merges text objects until a major text property (font, line coordinate, etc.) changes or a word break occurs if word breaking is enabled (see Property BreakWords). The current graphic state can be retrieved through the current content object's interface.

5.3.11 Method Void Reset(Boolean AccountForRotate)

This method allows to reset the content extraction process and set the point of extraction to the beginning of the content stream.

- Parameters:
  - AccountForRotate: If this parameter is set to True then the page's rotate property is used to transform the coordinates of the content objects before extraction takes place. If this parameter is set to False the original (non-rotated) coordinates are used. This parameter has an effect on text extraction since vertical lines might be rotated to horizontal lines.

5.4 The Image Interface

5.4.1 Property Alternates

This property returns an array of alternate images (see Interface AlternateImage). An image can have none, one or multiple alternate images.

5.4.2 Property BitsPerComponent

This property returns the number of bits that are used to represent a single color component of an image sample. The number of color components per image data sample can be retrieved through the image's color space interface. This property cannot be set.

5.4.3 Property IPDFColorSpace* ColorSpace

This property returns the interface to the image's color space (see ColorSpace Interface). This property cannot be set.

5.4.4 Property Long Height

This property returns the height of the image in pixels (also called samples). The unit of pixels can be converted to a distance unit such as inch, millimetre etc. using a resolution value, i.e. 72 dpi (dots per inch). This property cannot be set.

5.4.5 Property Boolean IsBitonal

This property is true when the image is bitonal. This property cannot be set.

5.4.6 Property Boolean IsColor

This property is true when the image is color. This property cannot be set.

5.4.7 Property Boolean IsMonochrome

This property is true when the image is monochrome. This property cannot be set.

5.4.8 Property Variant Mask

This property returns the image's explicit mask if available. The mask's sample data is organized the same way as the image data except that the data contains one bit per pixel. A one bit indicates an opaque pixel and a zero bit indicates a transparent pixel.

5.4.9 Property Variant Samples

This property returns the image's data samples in a byte array. The sample data is ordered by line from top to bottom and within a line from left to right. The lines are byte aligned. If the number of bits per component is less than one byte then the samples are ordered beginning with the most significant bit first.

If the image's ImageMask property is set to False then the interpretation of the sample data must be done according to the properties in the image's color space.

If the image's ImageMask property is set to True then the sample data represents a stencil mask. In this case the color space isn't meaningful and the data is organized one bit per pixel. A one bit signifies a transparent pixel and a zero bit signifies a pixel with the current fill color (see GraphicsState Interface).

5.4.10 Property Long Width

This property returns the width of the image in pixels (also called samples). The unit of pixels can be converted to a distance unit such as inch, millimetre etc. using a resolution value, i.e. 72 dpi (dots per inch). This property cannot be set.

5.4.11 Method ChangeOrientation(TPDFOrientation Orientation)

This property sets the orientation of the image. This value has to be set prior to "Store". The orientation of the image can be retrieved from "GraphicsState.ctm.Orientation". See Enum TPDFErrorCode.

5.4.12 Method IPDFImage* ConvertToRGB()

This method converts the image to an RGB image and returns its interface. The conversion uses the image's color space to interpret the sample data. Calibrated color spaces are converted to RGB values according to the sRGB color standard. Device color space are converted using pre-defined color profiles.

5.4.13 Method Variant GetImage()

This method returns the image from memory which was previously saves using the method StoreInMemory. The image is returned as a byte array.

5.4.14 Method Single GetResolution(IPDFTransformMatrix* Matrix)

This method returns the resolution of an image in dpi.

- Parameters:
  - Matrix: The transformation matrix of the image.
- Return values:
  - The calculated resolution in dpi.

5.4.15 Method Boolean Store(String FileName)

This methods stores the image in TIFF format to a disk file.

- Parameters:

  - FileName: The name of the disk file include path, drive, or Server string according to the operating system's naming rules.

- Return values:
  - True: The file has successfully been written.
  - False: An error has occurred and the disk file is unusable.

5.4.16 Method Boolean StoreInMemory(String Extension, TPDFCompression Compression)

This methods stores the image in memory. The saved image can be retrieved using the method GetImage.

- Parameters:
  - Extension: The type of the image is defined by its extension (for example ".jpg", ".tif", ".png", etc.
  - Compression (optional): The compression type (for TIFF images).
- Return values:
  - True: The image has successfully been saved.
  - False: The image could not be saved.

5.5 The Text Interface

5.5.1 Property Variant BoundingBox

This property returns the smallest rectangle that encloses the text. This property cannot be set.

5.5.2 Property Single FontSize

This property returns the size of the font in points. The size can also be interpreted as the height of the text. This property cannot be set.

5.5.3 Property Integer Length

This property returns the number of characters in the string. This property cannot be set.

5.5.4 Property Variant NextXPos (deprecated since 1.41)

This property returns the horizontal position of each character in the text string as an array of single precision real numbers. The horizontal and vertical position of the first character can be retrieved from the text matrix which is a property of the text's graphics state interface. The first element of the array returns the horizontal position of the second character and the second element the horizontal position of the third character and so on. The vertical position remains the same for all characters in the string. This property cannot be set.

5.5.5 Property Variant NextYPos (deprecated since in 1.41)

This property is similar to NextXPos, but returns the Y positions instead.

5.5.6 Property String RawString (removed in 1.41)

This property returns the unprocessed, undecoded string of character codes. The number of bytes per character is given by the associated font. The raw string is always returned regardless of the font information. This property cannot be set.

5.5.7 Property Single Rotation

This property returns the rotation of the string in radiant (rad). This property cannot be set. (2 pi rad = 360°)

5.5.8 Property TextMatrix (deprecated since 1.41)

This property returns an interface to a transformation matrix called text matrix. The values of the matrix are related to the first character of the text string. The text matrix property contains the cumulated transformations of the page rotate attribute, the graphics state current transform matrix, the text matrix of the current text object, and the text rendering matrix. In short it includes the complete transformation from text units to the page units. This property cannot be set.

5.5.9 Property String UnicodeString

This property returns the text as a Unicode UTF-16 encoded string. The number of bytes per character is a multiple of two. For most languages such as English a character can be mapped to a single 16-Bit Unicode value. Complex languages such as Chinese can return multiple 16-Bit values per character. Some text strings, however, cannot be correctly mapped or cannot be mapped at all. The former is the case if e.g. the PDF creator program didn't use correct names for the character in the font encoding (see Font Interface). The latter is the case if e.g. the PDF creator program didn't embed Unicode mapping information for a symbolic font. This property cannot be set.

5.5.10 Property Single Width

This property returns the width of the string in points. This property cannot be set.

5.5.11 Property Variant XPos

This property returns the X position of the first character in the text token.

5.5.12 Property Variant YPos

This property returns the Y position of the first character in the text token.

5.6 The GraphicsState Interface

5.6.1 Property Single CharSpacing

This property returns the current space between two characters of a text string as a single precision real number in text units. This property cannot be set.

5.6.2 Property IPDFTransformMatrix* CTM

This property returns an interface to the current transform matrix. The transform describes the transformation of the graphic object's coordinates from user units to page units including the effect of the page rotate attribute if requested (see method Reset() of the Content Interface). This property cannot be set.

5.6.3 Property Variant DashArray

This property returns the dash array of a line dash pattern. The line dash pattern controls the pattern of dashes and gaps used to stroke paths.

5.6.4 Property Single DashPhase

This property returns the dash phase of a line dash pattern. The dash phase is the offset of the pattern and can be larger as the pattern itself.

5.6.5 Property Long FillColorCMYK

This property returns the CMYK color quad for filling operations. The color value is obtained by converting the color values of the FillColor property by means the FillColorSpace. The CMYK quads are encoded using the following formula: Quad = (((C * 256) + M) * 256 + Y) * 256 + K. This property cannot be set.

5.6.6 Property Long FillColorRGB

This property returns the RGB color triple for filling operations. The color value is obtained by converting the color values of the FillColor property by means the FillColorSpace. The RGB triples are encoded using the following formula: Triple = ((R * 256) + G) * 256 + B. This property cannot be set.

5.6.7 Property IPDFColorSpace* FillColorSpace

This property returns an interface to the current color space that is used for filling operations (see ColorSpace Interface). The color space is used to interpret color values of the FillColor property. This property cannot be set.

5.6.8 Property IPDFFont* Font

This property returns an interface to the text's font object that describe the character encoding as well as the shape of the character glyphs. This property cannot be set.

5.6.9 Property Single FontSize

This property returns the current font size for text strings as a single precision real number in text units. It doesn’t include any scaling factors from coordinate transforms such as from the current transform matrix or the text matrix. In order to obtain the font size in page units the values of the current text matrix have to be examined. This property cannot be set.

5.6.10 Property Single HorizontalScaling

This property returns the current horizontal scaling factor that describes the amount of horizontal stretching of a text string. A value of greater than 1.0 stretches the string whereas a value of less than 1.0 lets the string appear as condensed. This property cannot be set.

5.6.11 Property Single Leading

This property returns the current leading (line spacing) of a text string as a single precision number in text units. This property cannot be set.

5.6.12 Property Single LineWidth

This property returns a single precision real number in user units. The property cannot be set.

5.6.13 Property Short LineCap

This property returns the line cap style. The line cap style specifies the shape to be used at the end of open sub-paths and dashes when they are stroked. This property cannot be set.

0: Butt cap
1: Round cap
2: Projecting square cap

5.6.14 Property Short LineJoin

This property returns the line join styles. The line join styles specifies the shape to be used at the corners of paths that are stroked. This property cannot be set.

0: Miter join
1: Round join
2: Bevel join

5.6.15 Property Single MiterLimit

This property returns the miter limit. The miter limit imposes a maximum on the ratio of the miter length to the line width, which can be fairly large when two line segments meet at a sharp angle. When the limit is exceeded, the join is converted from a miter to a bevel. This property cannot be set.

5.6.16 Property Long StrokeColorCMYK

This property returns the CMYK color quad for stroking operations. The color value is obtained by converting the color values of the StrokeColor property by means the StrokeColorSpace. The CMYK quads are encoded using the following formula: Quad = (((C * 256) + M) * 256 + Y) * 256 + K. This property cannot be set.

5.6.17 Property Long StrokeColorRGB

This property returns the RGB color triple for stroking operations. The color value is obtained by converting the color values of the StrokeColor property by means the StrokeColorSpace. The RGB triples are encoded using the following formula: Triple = ((R * 256) + G) * 256 + B. This property cannot be set.

5.6.18 Property IPDFColorSpace* StrokeColorSpace

This property returns an interface to the current color space that is used for stroking operations (see ColorSpace Interface). The color space is used to interpret color values of the StrokeColor property. This property cannot be set.

5.6.19 Property Boolean TextKnockout

This property returns the text knockout flag. This Boolean flag determines what text elements are considered elementary objects for purposes of color compositing in the transparent imaging model. This property cannot be set.

5.6.20 Property Short TextRenderingMode

This property returns a value that indicates whether the text should be stroked, filled, used as a clip path or some combination of the three. The meaning of the values in detail is:

- 0: Fill text.
- 1: Stroke text.
- 2: Fill, then stroke text.
- 3: Neither fill nor stroke text (invisible).
- 4. Fill text and add path for clipping.
- 5: Stroke text and add path for clipping.
- 6: Fill, then stroke text and add path for clipping.

- 7: Add path for clipping.

This property cannot be set.

5.6.21 Property Single TextRise

This property returns a single precision real number in unscaled text units that indicates by which amount the base line of the text is moved up or down. It is most commonly used to display subscripts and superscripts. This property cannot be set.

5.6.22 Property Single WordSpacing

This property returns the current space between two words of a text string as a single precision real number in text units. This property cannot be set.

(For further information about the Graphic State, see PDF Reference, chapter 4.3.)

5.7 The Font Interface

5.7.1 Property Single Ascent

This property returns the Ascent value. This value represents the maximum height above the baseline reached by the glyphs in the font, excluding the height of glyphs for accented characters.

5.7.2 Property Single AvgWidth

This property returns the average with of the glyphs in the font.

5.7.3 Property String BaseName

This property returns the font name.

5.7.4 Property Single CapHeight

This property returns the height of the top of flat capital letters, measured from the baseline.

5.7.5 Property String Charset

This property returns a string listing the character names defined in a font subset. This property is only useful for Type1 fonts.

5.7.6 Property Single Descent

This property returns the Descent value. This negative number represents the maximum depth below the baseline reached by the glyphs in the font.

5.7.7 Property Variant Encoding

5.7.8 Property Short FirstChar (removed in 1.41)

This property returns the first (lowest) character code in the font.

5.7.9 Property Long Flags

This property returns the flags of the font. The flags are:

1	FixedPitch	All glyphs have the same width.
2	Serif	Glyphs have serifs.
3	Symbolic	The font contains characters outside the standard Latin character set.
4	Script	Glyphs resemble cursive handwriting.
6	NonSymbolic	Font uses standard Latin character set or a subset of it.

7	Italic	Glyphs are italic.
17	AllCap	Font has not lowercase letters.
18	SmallCap	Lowercase letters are small uppercase letters.
19	ForceBold	If set, bold glyphs are painted bold even at very small text size.

5.7.10 Property Variant FontBBox

This property returns the font bounding box. The font bounding box is the rectangle in which all glyphs would fit, with they were placed on top of each other with their origins at the same point.

5.7.11 Property Variant FontFile

This property returns a stream that contains a Type1 font program.

5.7.12 Property Integer FontFileType

This property returns the type of the font.

5.7.13 Property Single ItalicAngle

The counterclockwise angle of the dominant vertical strokes of the font.

5.7.14 Property Short LastChar (removed in 1.41)

This property returns the number of the last (highest) character code in the font.

5.7.15 Property Single Leading

This property returns the desired spacing between baselines of consecutive lines of text.

5.7.16 Property Single MaxWidth

This property returns the maximum width of the glyphs in the font.

5.7.17 Property Single MissingWidth

This property returns the value of the width which is used for character codes for which the glyph is missing in the font directory’s Width array.

5.7.18 Property Single StemV

This property returns the horizontal thickness of the dominant vertical stems of the glyphs in the font.

5.7.19 Property Single StemH

This property returns the vertical thickness of the dominant horizontal stems of the glyphs in the font.

5.7.20 Property String Type

This property returns the font type as string.

5.7.21 Property Variant Widths

This property returns an array which contains the widths of the glyphs.

5.7.22 Property Single XHeight

This property returns the maximum height of flat non-ascending lowercase letters (such as the letter x) measured from the baseline.

(For further information about font descriptors, see PDF Reference, chapter 5.7.)

5.8 The ColorSpace Interface

5.8.1 Property IPDFColorSpace* BaseColorSpace

This property returns an IPDFColorSpace interface to the base color space if it is existing.

5.8.2 Property Short ComponentsPerPixel

This property returns the number of components per pixel. This property cannot be set.

5.8.3 Property Short High

This property returns the highest value of the indexed colors. It is 0 when no indexed color space is used. This property cannot be set.

5.8.4 Property Boolean IsColor

This property is true when the color space is color. This property cannot be set.

5.8.5 Property Boolean IsIndexed

This property is true when the image uses indexed colors. This property cannot be

5.8.6 Property Boolean IsMonochrome

This property is true when the color space is monochrome. This property cannot be set.

5.8.7 Property Variant Lookup

This property returns the lockup table. This property cannot set.

5.8.8 Property String Name

This property returns the name of the color space as string (for example "DeviceGrey", "DeviceRGB" or "Indexed").

5.9 The TransformMatrix Interface

5.9.1 Property a, b, c, d, e, f

The transformation matrix in PDF is specified by six numbers. All information about orientation, rotation, scaling, skewing and translation can be calculated based on these six numbers. However PDF Extract also provides properties which compute these values.

The values e and f represent the translation. In a matrix [1 0 0 1 e f], *e* is the distance on the x-axis from the left side page border, *f* is the distance on the y-axis from the bottom. (0,0) is in the lower left corner, on an page with a size of A4 portrait, (595,842) is in the upper right corner.

The scale factor in a matrix [a 0 0 d 0 0] can be obtained from the values *a* and *d* for x and y scaling respectively. With respect to fonts, *d* represents the font size of horizontal text.

A rotation of the axis by an angle α counter clockwise is produced by a matrix [cos α sin α -sin α cos α 0 0].

More detailed information can be found in the PDF Reference manual chapter 4.2.2.

5.9.2 Property Orientation

This property returns the orientation rounded to the next 90 degrees. The orientation is an enumeration with eight different values (rotation times flipping). The values are: eBottomLeft, eBottomRight, eLeftBottom, eLeftTop, eRightBottom, eRightTop, eTopLeft, eTopRight.

5.9.3 Property Rotation

This property return the rotation angle of the matrix counter clockwise. This is equal to the minimum of XSkew and –YSkew.

5.9.4 Property XScaling, YScaling

This properties return the x and y scaling factor.

5.9.5 Property XSkew, YSkew

This properties return the x and y-axis skewing. The transformation matrix [1 tan α tan β1 0 0] skews the x-axis by α and the y-axis by β. Skewing sometimes is used to transform a regular font to italic.

5.9.6 Property XTranslation, YTranslation

This properties return the X and Y translation. These are the same values as returned by the properties *e* and *f*.

5.10 The Alternate Image Interface

5.10.1 Property Boolean DefaultForPrinting

This property returns true if the alternate image is set as default for printing. This property cannot be set.

5.10.2 Property IPDFImage* Image

This Property returns an interface to the alternate image (see Image Interface). This property cannot be set.

5.11 The Annotation Interface

5.11.1 Property Long Color

This property returns to color of the annotation.

5.11.2 Property String Contents

This property returns the content of the annotation.

5.11.3 Property Date Date

This property returns the date of the annotation.

5.11.4 Property Long Flags

This property returns the flags of the annotation as 32bit integer.

1 Invisible
2 Hidden (PDF 1.2)
3 Print (PDF 1.2)
4 NoZoom (PDF 1.3)
5 NoRotate (PDF 1.3)
6 NoView (PDF 1.3)
7 ReadOnly (PDF 1.3)
8 Locked (PDF 1.4)
9 ToggleNoView (PDF 1.5)

5.11.5 Property String Name

This property returns the name of the annotation as string.

5.11.6 Property Variant Rect

This property returns the rectangle of the annotation as x1, y1, x2, y2. Where x1, y1 is the lower left corner of the annotation and x2, y2 the upper right corner. The lower left corner of the page is at 0, 0 (or according to media and crop-box).

5.11.7 Property String Subtype

This property returns the type of the annotation as string, such as “Widget”, “Square”, “PopUp”, “FreeText”, “Ink”, etc.

5.11.8 Property String TextLabel

This property returns the text label of the annotation as string. This label is usually used for the name of the author.

5.12 The Outline Interface

5.12.1 Property Long Count

This property returns the number of children of the current outline. A negative number means the child tree is not opened.

5.12.2 Property IPDFDestination* Dest

This property returns an interface to the destination (see Destination Interface).

5.12.3 Property String Title

This property returns the title of the outline.

5.13 The Destination Interface

Note that the properties Bottom, Left, Right and Top of the destination interface have different meanings depending on the Type of the destination.

5.13.1 Property Single Bottom

This property returns the Bottom value.

5.13.2 Property Single Left

This property returns the Left value.

5.13.3 Property Long PageNo

This property returns the target page number.

5.13.4 Property Single Right

This property returns the Right value.

5.13.5 Property Single Top

This property returns the Top value.

5.13.6 Property String Type

This property returns the type of the destination, such as “XYZ”, “Fit”, “FitH”, “FitR”, etc.

5.13.7 Property Single Zoom

This property returns the Zoom value of the destination. A value of 0 has means the zoom level is left as is. It has the same meaning as a null value, the returns value will be 0 in both cases. A value of 1 means 100% magnification.

5.14 Enumerations

5.14.1 Enum TPDFCompression

eComprRaw = 0
eComprJPEG = 1
eComprFlate = 2
eComprLZW = 3
eComprGroup3 = 4
eComprGroup3_2D = 5
eComprGroup4 = 6
eComprJBIG2 = 7
eComprJPEG2000 = 8
eComprUnknown = 9

Note that not all image formats/color depths support all compression types.

5.14.2 Enum TPDFContentObject

Const eNone = 0
Const eText = 1
Const eImage = 2
Const ePath = 3

There are more enumerations, however these are not relevant for the image to PDF or image to image class.

5.14.3 Enum TPDFOrientation

eTopLeft=1 (default)
eTopRight=2
eBottomRight=3
eBottomLeft=4
eLeftTop=5
eRightTop=6
eRightBottom=7
eLeftBottom=8

5.14.4 Enum TPDFErrorCode

The following list contains the error codes of *all* 3-Heights products.

PDF general codes

- PDF_S_SUCCESS = 0x00000000: The operation was completed successfully.
- PDF_E_FATAL = 0x8041ffff: Unknown fatal error.

PDF stream module

- PDF_E_FILEOPEN = 0x80410001: The file couldn't be opened.
- PDF_E_FILECREATE = 0x80410002: The file couldn't be created.
- PDF_E_SETPOS = 0x80410003: The read / write position pointer couldn't be set.

PDF file module

- PDF_E_HEADER = 0x80410010: The PDF header was not found.
- PDF_E_EOF = 0x80410011: The PDF end of file mark was not found.
- PDF_E_XREF = 0x80410012: The xref table contains an error.

- PDF_E_TRAILER = 0x80410013: The trailer dictionary is missing or invalid.
- PDF_E_ROOT = 0x80410014: The root object was not found.
- PDF_E_PASSWORD = 0x80410015: The authentication failed due to a wrong password.
- PDF_E_CORRUPT = 0x80410016: The file is corrupt and cannot be repaired.

PDF parser module

- PDF_E_OBJNO = 0x80410021: The object number is missing.
- PDF_E_GENNO = 0x80410022: The generation number is missing.
- PDF_E_IDENTITY = 0x80410023: The object's identity doesn't match with the reference's.
- PDF_E_OBJ = 0x80410024: The "obj" keyword is missing.
- PDF_E_NULL = 0x80410025: The object is empty (null).
- PDF_E_LENGTH = 0x80410026: The "Length" attribute of the stream object is wrong.
- PDF_E_ENDSTREAM = 0x80410027: The "endstream" keyword is missing.
- PDF_E_ENDOBJ = 0x80410028: The "endobj" keyword is missing.

PDF document module

- PDF_E_PAGE = 0x80410031: The page doesn't exist (null).
- PDF_E_PAGETYPE = 0x80410032: The page has a missing or invalid "Type" attribute.
- PDF_E_PAGEKIDS = 0x80410033: The page has a missing or invalid "Kids" attribute.
- PDF_E_PAGECOUNT = 0x80410034: The page has a missing or invalid "Count" attribute.
- PDF_E_PAGEPARENT = 0x80410035: The page has a missing or invalid "Parent" attribute.
- PDF_E_PAGERES = 0x80410036: The page has a missing or invalid "Resource" attribute.
- PDF_E_PAGEMEDIABOX = 0x80410037: The page has a missing or invalid "MediaBox" attribute.
- PDF_I_OPEN = 0x00410001: The document is going to be opened.
- PDF_I_PAGE = 0x00410002: The page is going to be read.

PDF content module

- PDF_E_TOKEN = 0x80410041: An unexpected token was found.

- PDF_E_OPERATOR = 0x80410042: The content stream contains an invalid operator.
- PDF_E_OPERANDS = 0x80410043: The operator has an invalid number of operands.
- PDF_E_OPNDNUMBER = 0x80410044: The operand must be a number.
- PDF_E_OPNDSTRING = 0x80410045: The operand must be a string.
- PDF_E_OPNDNAME = 0x80410046: The operand must be a name.
- PDF_E_OPNDARRAY = 0x80410047: The operand must be an array.
- PDF_E_OPNDDICT = 0x80410048: The operand must be a dictionary.
- PDF_E_UNKCOLORSPACE = 0x80410049: The name of the color space is unknown.

PDF rendering module

- PDF_E_PAINTMISSING = 0x80411001: The path is constructed but the painting operator is missing.
- PDF_E_PATTERNTYPE = 0x80411002: The type of pattern is unknown or not yet implemented.
- PDF_E_SAMPLELENGTH = 0x80411003: The length of the image's sample stream is not correct.
- PDF_E_CHARPROC = 0x80411004: The glyph's charproc was not found.
- PDF_E_SHADINGTYPE = 0x80411005: The type of shading is unknown or not yet implemented.
- PDF_E_FONTINSTALL = 0x80411006: The font program couldn't be installed successfully.
- PDF_E_STANDARDFONT = 0x80411007: The font program of a standard font was not found.
- PDF_W_SYSTEMFONT = 0x00411008: The font uses a pre-installed system font.

PDF_W_FONTREPLACE = 0x00411009: The font was replaced by an other font.

6 Interface changes

6.1 Changes from 1.40 to 1.41

This is a list of interface changes from version 1.40 (DLL 1.4.0.21) to version 1.41 (DLL 1.4.1.24).

Annotation Interface

New: Property String TextLabel

ColorSpace Interface

New: Property Colorant

Content Interface

New: Property Long Flags

Destination Interface

New: Property Zoom

Font Interface

Removed: Property FirstChar, Property LastChar

Image Interface

New: Method Boolean StoreInMemory, Method Variant GetImage

Page Interface

New: Property Variant BleedBox, Property Variant TrimBox, Property Variant ArtBox, Property String DeviceColorant

Text Interface

New: Property Variant BoundingBox, Property Single FontSize, Property Long Length, Property Single Rotation, Property Integer Width, Property Variant XPos, Property Variant YPos

Removed: Property String RawString, Property TextMatrix, Property Variant NextXPos, Property Variant NextYPos

No changes in the following interfaces: AlternateImage, Document, GraphicsState, OutlineItem , TransformationMatrix

The properties TexMatrix, NextXPos, NextYPos are deprecated.

6.2 Changes from 1.41 to 1.50

There are no interface changes from DLL 1.4.1.24 to DLL 1.5.0.1.

7 Code Samples

There are various code samples in the ZIP file of both, the evaluation and the release version of the 3-Heights™ PDF Extract Tool API.