



3-Heights™ PDF Extract Shell

User's Manual

Version 1.91

Contact: pdfsupport@pdf-tools.com

Owner: **PDF Tools AG**
Geerenstrasse 33
CH-8185 Winkel
Switzerland
www.pdf-tools.com

June 21, 2010

Table of Contents

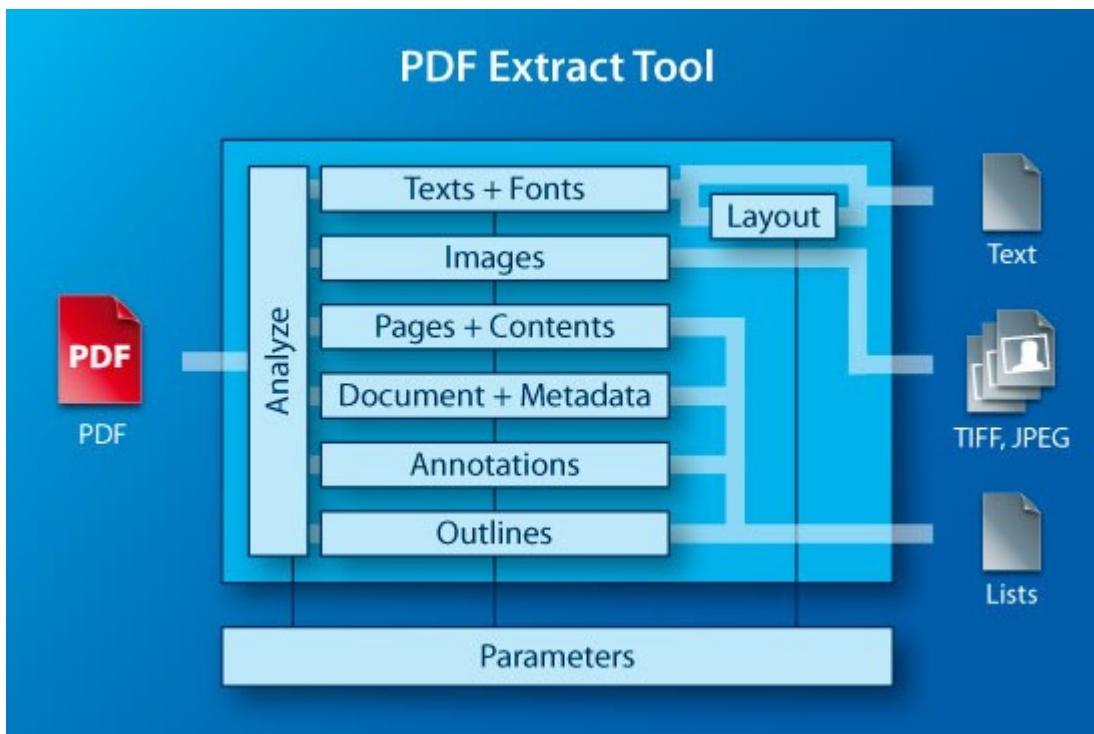
1	Introduction.....	3
1.1	Description	3
1.2	Functions	3
	Features.....	4
	Formats.....	4
	Compliance.....	4
1.3	Operating Systems	4
2	Installation	5
2.1	Installing the 3-Heights™ PDF Extract Shell	5
	How to Set the "Path" Environment Variable.....	5
3	Reference Manual	6
3.1	pdfextract.....	6
	-h Include a CSV Header Line.....	6
	-la List Annotations	6
	-laf List Form Fields	7
	-lb List Outlines.....	8
	-lc List Color Spaces	9
	-ld List Document Attributes	9
	-lf List Fonts and Their Properties	10
	-li List Images and Their Properties	10
	-lp List Pages and Their Properties.....	12
	-ls List Signatures and Their Properties	13
	-o Write Output to File	13
	-p Specify a Password to Decrypt the Input File.....	13
	-pg List Page Range.....	14
	-r Extract by Resources.....	14
	-u Encode Output using Unicode.....	14
	-v Verbose Mode	14
	-x Extract and Store Embedded Data	14
3.2	pdtxt.....	16
	-a Set the Advance Width for Text Mode.....	16
	-c Character Mode.....	16
	-fd Directory of Pre-Installed Fonts	16
	-h Write a CSV Header	16
	-l Line Heights for Text Mode	17
	-lt Line Height Tolerance.....	17
	-o Extract Text to a File.....	17
	-p Specify Password.....	17
	-pg Extract a Page Range	17
	-r Account for Viewer Rotation	18
	-s Replace Symbolic Characters	18
	-sl Replace Ligatures	18
	-t Text Mode	18
	-u Create Unicode Text.....	18
	-w Word Mode	18
3.3	Return Codes	19

1 Introduction

1.1 Description

The 3-Heights™ PDF Extract Tool is a solution for extracting and querying various attributes and page content from a PDF document. This includes texts, images, graphic objects (including paths), metadata and embedded fonts.

It is also possible to query the properties of objects. Intelligent mechanisms significantly increase extraction rates, for instance when extracting text.



1.2 Functions

The PDF Extract Tool is used to extract text, images and graphic objects (including paths) from PDF documents. Text is extractable as lines and as individual words. It is also possible to query information such as position, color, font and font size. Intelligent functions such as heuristics, word formation support and character set interpretation make it possible to restore text that is lacking essential information. The tool can also collect significant data such as position, color space and size when extracting images such as TIFF or JPEG. Querying document attributes such as PDF version, creator, author, title, subject and creation date is also possible. The tool also supports reading encrypted PDF files.

June 21, 2010

Features

- 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

Formats

Input Formats:

- PDF 1.x (e.g. PDF 1.4, PDF 1.5)

Compliance

- Standards: ISO 32000 (PDF 1.7)

1.3 Operating Systems

- Windows 2000, XP, 2003, Vista, 2008, Windows 7 - 32 and 64 bit
- FreeBSD 4.7 for Intel
- HP-UX 11.0 - 32 bit and 32/64 bit Itanium
- IBM AIX (4.3: 32 Bit, 5.1: 64 bit)
- Linux (SuSE and Red Hat on Intel)
- Mac OS X
- Sun Solaris (2.7 and higher)

2 Installation

2.1 Installing the 3-Heights™ PDF Extract Shell

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

1. Download the ZIP archive of the product from your download account at www.pdf-tools.com.
2. Open the ZIP archive.
3. Check the appropriate option to preserve file paths (folder names) and unzip the archive to a local folder (e.g. *C:\program files\pdf-tools*).
4. The unzip process now creates the following subdirectories:
 - *Bin*: Contains the runtime executable binary code
 - *Doc*: Contains documentation files
5. (Optional) In order to allow for starting the 3-Heights™ PDF Extract Tool from a shell without providing a fully qualified path to the executable, the directory where the two executables *pdfextract* and *pdtxt* reside should be included in the "Path" environment variable.

How to Set the "Path" Environment Variable

To set the "Path" environment variable on Windows 2000: Go to **Start -> Settings -> Control Panel -> System -> Advanced -> Environment Variables**

Windows XP: Go to **Start -> Control Panel** (classic view) -> **System -> Advanced -> Environment Variables**.

Select "Path" and **Edit**, then add the directory where *pdtxt.exe* and *pdfextract.exe* are located to the "Path". If the environment variable "Path" does not exist, create it.

3 Reference Manual

The 3-Heights™ PDF Extract Shell is an easy to use tool. However at some points it could prove helpful if the user has a basic understanding about PDF. This manual does not explain any PDF related features in depth.

For further explanation of the PDF specific information, the PDF Reference Manual 1.6 can be used, it is available at the following link and requires Acrobat 6 or later or the 3-Heights™ PDF Viewer to be read:

<http://www.pdf-tools.com/public/downloads/pdf-reference/pdfreference16.pdf>

3.1 pdfextract

When using the listing options such as `-la`, `lb`, `-lc`, etc., the information is provided on the document level. This means items, such as fonts, color spaces or images are listed once per document. If a page range is selected, using the option `-pg`, the information is provided for each page separately. If information is provided on the document level, the page number in the listing is set to 0.

-h Include a CSV Header Line

This option adds a CSV formatted header. The header is written separately for every listing option. It is comma separated.

-la List Annotations

This option lists all annotations including page number, type, position and size, date, color, opacity, label, content and target.

- PageNo: The page number of where the annotation is.
- Type: The type of annotation such as Circle, FreeText, Ink, Highlight, Polygon, Popup, Square, Stamp, Widget, etc. (see table 8.16 in the PDF Reference Manual)
- Position and size(Left, Bottom, Right, Top): The rectangle of the annotation. The origin is in the lower left corner. The dimensions are in points (A4 = 595x842 points, Letter = 612x792 points)
- Date: The date of the annotation. If the date is unavailable, this value is left empty.
- Flags: The annotation flags. (see chapter 8.4.2 in the PDF Reference Manual 1.6)
- Color: The color in RGB, $color = R + (256 * (G + 256 * B))$
- Opacity: The opacity of the annotation. 1 is opaque, 0 is fully transparent.
- Label: The label (usually the author) of the annotation.
- Contents: The contents of the annotation.

June 21, 2010

- Target: The target of a link, launch, or remote GoTo annotations.

Example: List annotations:

```
pdfextract -h -la annotations.pdf
FileName,PageNo,Type,Left,Bottom,Right,Top,Date,Flags,Color,Opacity,Label,Contents,Target
annotations.pdf,1,Widget,59.598,771.687,121.205,788.429,,4,0,1.000,"Button","",
annotations.pdf,1,Widget,60.268,738.205,75.000,754.277,,4,0,1.000,"Checkbox","",
annotations.pdf,1,Widget,65.625,633.071,136.607,649.143,,4,0,1.000,"Textbox","",
annotations.pdf,1,Text,187.500,756.366,207.500,774.366,2004-08-
11,28,65535,1.000,"hba","Sticky note",
annotations.pdf,1,Square,324.277,784.580,397.599,805.670,2004-08-11,4,255,1.000,"hba","",
annotations.pdf,2,Circle,312.893,597.750,376.170,639.598,2004-08-11,4,255,1.000,"hba","",
annotations.pdf,2,Polygon,93.421,607.172,197.602,677.488,2004-08-11,4,255,1.000,"hba","",
annotations.pdf,2,Popup,595.000,508.384,775.000,628.384,,28,0,1.000,"","",
annotations.pdf,2,Stamp,313.137,505.372,566.775,557.198,2004-08-11,4,255,1.000,"hba","Yes",
annotations.pdf,2,Highlight,68.648,565.553,166.917,578.774,2004-08-
11,4,65535,1.000,"hba","",
```

-laf List Form Fields

This switch lists the form fields in a document. Since form fields are also annotations they may also be listed using the switch **-la**. The difference however is, that form fields may be hierarchically nested (parents/children) and that the listing contains fields that are more related to form fields than annotations. Furthermore, annotations that are not form fields, e.g. link annotations, are not listed with this switch.

- Level: The nesting level of the form field.
- Label: The label of the form field, e.g. "Button", "Textbox", "Checkbox", etc.
- Page: The page number, e.g. 1, 2, etc.
- Left, Bottom, Right, Top: The position in PDF points of the form field. (1 points = 1/72 inch. A4 = 595x842 points)
- Flags: Annotation flags are listed in the PDF Reference chapter 8.4 (Table 8.12). Here is an extract:
 - 1 Invisible
 - 2 Hidden
 - 3 Print
 - etc.
- AppearanceState: Corresponds to the "Export Value" of Acrobat.
- FieldType: The type of the form field, e.g. Btn, Chk, etc.
- FieldFlags: The form field annotations are listed in the PDF Reference chapter 8.5 (Table 8.66, 8.71, 8.73). Here is an extract:
 - 15 NoToggleToOff

June 21, 2010

16 Radio
 17 Pushbutton
 26 RadiosInUnison
 etc.

Example: List form fields:

```
pdfextract -h -laf annotations.pdf
FileName,Level,Label,Page,Left,Bottom,Right,Top,Flags,AppearanceState,FieldType,FieldFlags,
Value
"annotations.pdf",1,Button,1,59.598,771.69,121.205,788.43,4,,Btn,65536," "
"annotations.pdf",1,Checkbox,1,60.268,738.21,75,754.28,4,Ja,Btn,0," "
"annotations.pdf",1,Combobox,1,62.277,694.68,127.902,716.11,4,,Ch,131072,"First"
"annotations.pdf",1,Listbox,1,56.25,654.5,126.563,676.6,4,,Ch,0," "
"annotations.pdf",1,Textbox,1,65.625,633.07,136.607,649.14,4,,Tx,0," "
```

-lb List Outlines

This option lists all outlines (bookmarks), including bookmark level, count, title, destination, target page number, target position and zoom.

- Level: The bookmark root level is 1. The number of a child bookmarks is one level higher as its parent.
- Count: The number of visible children. Not expanded children count negative. (see also chapter G.5 in the PDF Reference Manual 1.6)
- Destination: The destination type, such as Fit, FitH, FitV, XXY. (see also chapter 8.2 in the PDF Reference Manual 1.6)
- Target Position and Zoom (Left, Bottom, Right, Top, Zoom): These parameters depend on the destination type. (see also chapter 8.2 in the PDF Reference Manual 1.6)

Example: List outlines:

```
pdfextract -h -lb outlines.pdf
FileName,Level,Count,Title,Destination,PageNo,Left,Bottom,Right,Top,Zoom
outlines.pdf,1,5,"Part 1","FitH",1,0.000,0.000,0.000,844.000,0.000
outlines.pdf,2,0,"Chapter 1","FitH",1,0.000,0.000,0.000,844.000,0.000
outlines.pdf,2,3,"Chapter 2","FitH",2,0.000,0.000,0.000,839.000,0.000
outlines.pdf,3,2,"Sub-Chapter 2.1","FitH",2,0.000,0.000,0.000,700.000,0.000
outlines.pdf,4,0,"Text 2.1.1","FitH",2,0.000,0.000,0.000,505.000,0.000
outlines.pdf,4,0,"Text 2.1.2","FitH",2,0.000,0.000,0.000,505.000,0.000
outlines.pdf,1,2,"Part 2","FitH",3,0.000,0.000,0.000,843.000,0.000
outlines.pdf,2,0,"Chapter 3","FitH",3,0.000,0.000,0.000,676.000,0.000
outlines.pdf,2,0,"Chapter 4","FitH",4,0.000,0.000,0.000,836.000,0.000
```

June 21, 2010

-lc List Color Spaces

This option lists color spaces, including page number, name, number of components, colorants, base name and alternate name.

- PageNo: The page number. This is set to 0 when no page range is defined.
- Name: The name of the color space such as ICCBased, Indexed, Pattern, Separation, etc.
- Number of components: The number, usual 1-4, of components used in the color space.
- Colorants: A description of colorants used, this should correspond to the number of components.
- Base Name, Alternate Name: The name and alternate name of the color space, such as DeviceCMYK, DeviceRGB, DeviceGray, etc.

Example: List color spaces:

```
pdfextract -h -lc PDFReference16.pdf
FileName,PageNo,Name,NoOfComponents,Colorants,BaseName,AlternateName
PDFReference16.pdf,0,Separation,1,All,,DeviceCMYK
PDFReference16.pdf,0,Separation,1,Comment,,DeviceCMYK
PDFReference16.pdf,0,Indexed,1,Index,DeviceCMYK,
PDFReference16.pdf,0,Indexed,1,Index,DeviceCMYK,
PDFReference16.pdf,0,ICCBased,3,Red-Green-Blue,,DeviceRGB
PDFReference16.pdf,0,ICCBased,1,Gray,,DeviceGray
PDFReference16.pdf,0,Pattern,0,,ICCBased,
PDFReference16.pdf,0,ICCBased,4,Cyan-Magenta-Yellow-Black,,DeviceCMYK
```

-ld List Document Attributes

This options lists document attributes, such as number of pages, encryption, document title, document author, subject, keywords, creator, producer, date of creation, modification date.

- PageCount: The total number of pages.
- IsEncrypted: Returns *Encrypted* if encrypted, returns blank if not encrypted.
- Title, Author, Subject, Keywords, Creator, Producer: The value of the corresponding document attribute.
- CreationDate, ModificationDate: The date in the format yyyy-mm-dd.

Example: List document attributes:

```
pdfextract -ld exps.pdf
FileName,PageCount,IsEncrypted,Title,Author,Subject,Keywords,Creator,Producer,CreationDate,
ModificationDate
exps.pdf,17,Encrypted,"3-Heights™ PDF Extract Shell Tool User's Manual","PDF Tools AG","
","Acrobat PDFMaker 7.0.7","PDF PT 3.10p (pdf-tools.com)",2006-10-02,2006-10-02
```

June 21, 2010

-lf List Fonts and Their Properties

This option lists all fonts and their properties, such as page number, name of the font, font type, encoding, CID, embedding, subsetting and file name.

- PageNo: The page number. This is set to 0 when no page range is defined.
- FontName: The name of the font. Subsetting pre-fixes, such as "HMAGKB+" are included. Note that many applications such as Adobe Acrobat remove this information from the font name, and mark the font as subsetting.
- FontType: The type of the font such as Type0, Type1, MMTyep1, TrueType, Type3, TrueType, CIDFontType0, CIDFontType2. (see PDF Reference Manual chapter 5.4)
- Encoding: The encoding, such as WinAnsiEncoding, DifferenceEncoding, MacRomanEncoding, Identity-H. (see PDF Reference Manual Appendix D)
- IsCID: Returns *CID* if the font is CID font, returns blank otherwise.
- IsEmbedded: Returns *Embedded* if the font program is embedded, returns blank otherwise.
- IsSubsetting: Returns *Subsetting* if the font is subsetting, returns blank otherwise.
- FileName: The name of the font when extraction using the option -x is applied. (this value is not listed without -x)

The switch **-r** lists fonts by resources (every font is listed once). Without the switch **-r**, every font is listed for every page.

Example: List all fonts in the PDF document's resources:

```
pdfextract -h -lf -r document.pdf
FileName,PageNo,FontName,FontType,Encoding,IsCID,IsEmbedded,IsSubsetting,FontFileName
document.pdf,0,"Verdana",TrueType,WinAnsiEncoding,,,,
document.pdf,0,"Verdana-Bold",TrueType,WinAnsiEncoding,,,,
document.pdf,0,"HMAGKB+SymbolMT",CIDFontType2,Identity-H,CID,Subsetting,Embedded,
document.pdf,0,"CenturyGothic",TrueType,WinAnsiEncoding,,,,
document.pdf,0,"Verdana",TrueType,WinAnsiEncoding,,,,
document.pdf,0,"TimesNewRomanPSMT",TrueType,WinAnsiEncoding,,,,
document.pdf,0,"Verdana-Italic",TrueType,WinAnsiEncoding,,,,
document.pdf,0,"HMAJDK+Courier",Type1,WinAnsiEncoding,,Subsetting,Embedded,
document.pdf,0,"CourierNewPSMT",TrueType,WinAnsiEncoding,,,,
document.pdf,0,"HMAMD+ArialUnicodeMS",CIDFontType2,Identity-H,CID,Subsetting,Embedded,
```

-li List Images and Their Properties

List images in the PDF document and their properties, such as location, dimensions, bits per component, color space, image mask, image soft mask, filter, resolutions and file name.

Images can be listed in two ways:

June 21, 2010

1. By resources
2. By occurrence on the pages

By resources: Images in PDF can occur in two different ways: As image XObject, or as an inline image. (See also *PDF Reference, chapter 4.8 Images*). Most images, particularly large images, are stored as image XObjects. Their image data is stored as a resource in the PDF. The benefit of storing images like this is that multiple references to the same image, with possibly different resolutions and at different pages only require one resource and therefore keep the file size small.

Listing images by resources returns images from the PDF document's resources. i.e. it returns images from XObjects, but not inline images. These images do not have resolution. These images may be referenced once, multiple times or not at all on the pages of the document.

To list images by resources apply the switch **-r**.

By occurrence on the pages: Every time an image is referenced it is listed. Images from XObjects and inline images are listed this way.

The following properties are returned for extracted images:

- PageNo: The page number. This value is set to 0 if images are extracted by resources.
- Width, Height: The dimensions in dots.
- x0, y0: The coordinate of the lower left corner of the image in points. These values are 0 if images are extracted by resources. (This property was introduced in version 1.8.15.1).
- x1, y1: The coordinate of the upper right corner of the image in points. These values are 0 if images are extracted by resources. Depending on the transformation matrix the x and y values can be rotated, mirrored, etc. (This property was introduced in version 1.8.15.1).
- BitsPerComponent: The number of bits per component, such as 1 for bi-tonal images or 8 for color and grey scale images.
- XDPI, YDPI: The horizontal and vertical resolution in DPI (dots per inch). These values are 0 if images are extracted by resources.
- ColorSpace: The name of the color space such as 'ICCBased', 'Indexed', 'Pattern', 'Separation', 'Null', etc.
- Mask: can have the values 'Null', 'Stencil', 'Explicit' and 'Soft'. The field 'ColorSpace' is set to 'Null' for stencil mask images.
- Filter: The image filter, such as 'DCTDecode', 'CCITTFaxDecode', 'FlateDecode', etc.
- FileName: The name of the image when extraction using the option **-x** is applied. If a DCT compressed images is extracted the image is named *img{obj number}.jpg*, for all other compressions the extension is *.tif* instead (e.g. *img9.jpg*, or *img26.tif*).

Example: List image by resources:

```
pdfextract -h -li -r PDFReference16.pdf
```

```
FileName,PageNo,x0,y0,x1,y1,Width,Height,BitsPerComponent,XDPI,YDPI,ColorSpace,
```

June 21, 2010

```
Mask,Filter,ImageFileName
```

```
"PDFReference16.pdf",0,0,0,1,1,337,256,8,0,0,DeviceGray,,DCTDecode,
```

```
"PDFReference16.pdf",0,0,0,1,1,281,602,1,0,0,DeviceGray,,FlateDecode,
```

```
"PDFReference16.pdf",0,0,0,1,1,281,602,1,0,0,DeviceGray,,CCITTFaxDecode,
```

Example: List image by occurrence on the pages:

```
pdfextract -h -li PDFReference16.pdf
```

```
FileName,PageNo,x0,y0,x1,y1,Width,Height,BitsPerComponent,XDPI,YDPI,ColorSpace,IsMask,
```

```
HasSoftMask,Filter,ImageFileName
```

```
"PDFReference16.pdf",326,225,364,386,486,337,256,8,150,150,DeviceGray,,,DCTDecode,
```

```
"PDFReference16.pdf",486,155,491,222,636,281,602,1,300.04,300.4,DeviceGray,,,FlateDecode,
```

```
"PDFReference16.pdf",486,390,491,457,636,281,602,1,300.04,300.4,DeviceGray,,,CCITTFaxDecode,
```

-lp List Pages and Their Properties

List pages and their properties, such as page number, viewing rotation, media box, crop box, trim box, art box and content.

- PageNo: The page number in the document.
- Rotate: The viewing rotation attribute (0, or a multiple of 90).
- MediaBox: The media box rectangle given by the coordinates left, bottom, right, top. The media box is required, it defines the physical boundaries of the medium on which the page is intended to be displayed or printed.
- CropBox: The crop box rectangle given by the coordinates left, bottom, right, top. 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.
- TrimBox: The trim box rectangle given by the coordinates left, bottom, right, top. 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.
- BleedBox: The bleed box rectangle given by the coordinates left, bottom, right, top. 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.
- ArtBox: The art box rectangle given by the coordinates left, bottom, right, top. 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.
- FileName: The name of the text file containing the content when extraction using the switch **-x** is applied. (this value is not listed without **-x**)

Example of possible output:

Example: List pages and their properties:

```
pdfextract -h -lp document.pdf
```

```
FileName,PageNo,Rotate,MediaBox,CropBox,TrimBox,BleedBox,ArtBox,ContentFileName
```

June 21, 2010

```
document.pdf,1,0;0,0,595,842,0,0,595,842,0,0,595,842,0,0,595,842,0,0,595,842,
document.pdf,2,0;0,0,595,842,0,0,595,842,0,0,595,842,0,0,595,842,0,0,595,842,
document.pdf,3,90;0,0,595,842,0,0,595,842,0,0,595,842,0,0,595,842,0,0,595,842,
```

-ls List Signatures and Their Properties

List digital signatures and their properties, such as the name of the certificate or the reason why the signature was applied.

Example: List signatures and their properties:

```
pdfextract -h -ls document.pdf
```

```
Name,Reason,ContentFileName
```

```
"Peter Pan","I am the author of the document",
```

-o Write Output to File

Using the switch **-o**, followed by a file name, the output can be directed to a file name.

Example: Extract pages and their properties of the document "document.pdf" and write the result in the text file "ListOfPage.txt".

```
pdfextract -h -lp -o ListOfPages.txt document.pdf
```

This is similar as piping the output to a file using the operator **>**.

```
pdfextract -h -lp document.pdf > ListOfPage.txt
```

The error messages and warnings are written to standard error. To pipe these messages into a file use the operator **2>**.

Example: To pipe error and warning messages such as

```
0x80410042 - E - The content stream contains an invalid operator.
```

```
pdfextract -h -lp document.pdf 2> errorlog.txt
```

to discard them use a command like this:

```
pdfextract -h -lp document.pdf 2> Nul
```

-p Specify a Password to Decrypt the Input File

In order to read PDF documents which require a password to be opened, a password (user or owner password) can be provided using the switch **-p**.

Example: The following command opens an encrypted document and retrieves its page information. Either the user or the owner password of that document is "password".

```
pdfextract -p password -h -lp encrypted_document.pdf
```

June 21, 2010

-pg List Page Range

Set a page range. Some listing functions, such as fonts or images, can be listed by resources (document level) or by page. If the switch **-r** is not used, the information is listed separately for each page. The page range is defined by providing the start and end page. -1 defines the last page of the document.

-r Extract by Resources

Extract data (e.g. images or fonts) by resources instead of by page. See switches **-li** and **-lf**.

-u Encode Output using Unicode

The output is written as WinAnsi as default. In order to write the output as Unicode, use the switch **-u**.

-v Verbose Mode

Turn on the verbose mode to get additional information during the processing.

-x Extract and Store Embedded Data

This option allows to extract data, such as images or fonts.

How to extract a font:

If a document contains an embedded font, the flag *Embedded* is set and font name is listed.

Example: Extract and store embedded Data:

```
pdfextract -h -lf -x document.pdf
FileName,PageNo,FontName,FontType,Encoding,IsCID,IsEmbedded,IsSubsetted,FontFileName
document.pdf,0,"Arial-BoldMT",TrueType,WinAnsiEncoding,,,,
document.pdf,0,"TimesNewRomanPS-BoldMT",TrueType,WinAnsiEncoding,,,,
document.pdf,0,"Arial-BlackItalic",TrueType,WinAnsiEncoding,,,,
document.pdf,0,"KHFOKE+MonotypeCorsiva",TrueType,WinAnsiEncoding,,Subsetted,Embedded,fnt38.ttf
...
```

The extracted font is then saved with the corresponding font type and the object number as file name (e.g. "fnt38.ttf").

Note that the extracted fonts are not installable fonts (this is due to copyright reasons).

Example: The switch -x can also be applied to extract page content:

```
pdfextract -h -lp -x document.pdf
FileName,PageNo,Rotate,MediaBox,CropBox,TrimBox,BleedBox,ArtBox,ContentFileName
```

June 21, 2010

```
document.pdf,1,0;0,0,595,842,0,0,595,842,0,0,595,842,0,0,595,842,0,0,595,842,cnt1.txt
document.pdf,2,0;0,0,595,842,0,0,595,842,0,0,595,842,0,0,595,842,0,0,595,842,cnt2.txt
document.pdf,3,90;0,0,595,842,0,0,595,842,0,0,595,842,0,0,595,842,0,0,595,842,cnt3.txt
```

The content of the pages is then written to a corresponding text file (cnt1.txt for page 1, etc). The list contains the page number, the type of content, the coordinates and text. The content is returned in z-order. Which means what is written last (on top) is listed last.

- *PageNo: The page number in the document.*
- *Position: For text and images, the values Left, Bottom, Width, Height are provided to describe their position and dimensions.*
- *Type: The type of content, such as Text, Image, Path or Save and Restore operators.*
- *Text: This value depends on Type.*

Text: The actual text string, e.g. "this is some text".

Image: The name of the image when extracted using options -li -x, e.g. "img9.tif".

Path: The parameter of the path operator, e.g. "256.258 752.02 269.775 0.01 re f " for a filled rectangle.

Save, Restore: Empty

Example of possible output:

```
PageNo,Type,Left,Bottom,Width,Height,Text
3,Text,70.86,743.2,55.995,20.025,"Page 2 "
3,Save,,,,,
3,Image,70.86,70.86,300,441.78,"img9.tif"
3,Restore,,,,,
3,Text,370.86,225.215,4.4536,20.025," "
3,Path,,,,,"256.258 752.02 269.775 0.01 re f "
3,Text,70.86,76.655,110.232,20.025,"this is some text"
```

3.2 pdtxt

The text extraction tool pdtxt can be used to extract text from PDF documents. This tool has different modes:

character mode	extract single characters
word mode	extract words
text mode	extract all text and take into account the page layout

-a Set the Advance Width for Text Mode

This option sets the advance width for the text mode (see option -t). The default value is 7.2 points.

-c Character Mode

With this option, text is extracted character by character.

-fd Directory of Pre-Installed Fonts

Adds the files in a given directory to the installed fonts collection (e.g. C:\Windows\Fonts).

-h Write a CSV Header

Add a CSV (comma separated values) header as first line. This option can be used in combination with the options -c or -w, but not with -t.

The header has the following structure:

PageNo, XPos, YPos, XWidth, FontSize, FontName, Length, Text

PageNo number of current page

XPos X-position, the left border being 0. An A4 page is 595 points wide.

YPos Y-position, the bottom being 0. For an A4 page, the top is at 842 points.

XWidth width of the text tokens in points.

FontSize size of the font (or height of the text tokens) in points.

FontName name of the font

Length number of characters

Text character(s)

June 21, 2010

-l Line Heights for Text Mode

Define the height of a text line. This option is used in combination with the text mode option `-t`. This option can be used to insert blank lines. It takes influence under the following circumstances:

- If the text is written with a large font size, or different font sizes
- If there are blank rows, which need to be considered in the layout
- If multiple parallel columns are used

Example: Set the line height to 20 points. Put in simple words: If two lines of text in the PDF are 20 points apart, they are extracted as two individual lines. If two lines are 40 points apart a blank line is inserted in between them.

```
pdtxt -t -l 20 input.pdf
```

The default is 0, which means no extra rows are ever inserted between text lines.

-lt Line Height Tolerance

Defines the maximum vertical divergence in points of two text tokens that they are still considered to be on the same line.

This switch works in conjunction with the line height switch.

Default: 3 pt

-o Extract Text to a File

This option will extract the text to an output file. For example, the following command will extract the text to the output file "text.txt":

Example: Extract text and write it to the file "text.txt".

```
pdtxt -o text.txt input.pdf
```

Alternatively the output can be piped into a file:

```
pdtxt input.pdf > text.txt
```

-p Specify Password

If the input file is encrypted with a user password, a password needs to be provided to read the input PDF document. This can be either the user or owner password.

Example: Extract text from an encrypted PDF document. Either the user or the owner password of that document is "password".

```
pdtxt -p password input.pdf
```

-pg Extract a Page Range

Apply extraction to a selected page range. Example, extract pages 1 and 2 only:

Example: Extract text from pages 1 to 2.

June 21, 2010

```
pdtxt -pg 1 2 input.pdf
```

Default: Extract all page.

-r Account for Viewer Rotation

Each page in a PDF document can have a page rotation attribute that describes if the page is to be rotated when displayed (for example when a 90 degree rotated portrait is displayed as landscape). pdtxt by default ignores this attribute. Using the option -r, the rotation of the page is taken into account.

-s Replace Symbolic Characters

Replace symbolic character from the Unicode custom range (0xF000..0xFFFF) with WinAnsi codes (0x00..0xFF).

-sl Replace Ligatures

Replace ligatures ff, fi, fl, ffi, ffl with individual characters 'f', 'i' and 'l'.

-t Text Mode

The text mode allows text extraction of pages and retaining the page layout to a certain extent. Depending on the font size, the parameter -a can be used to set the advance width, the option -l to set the line height.

-u Create Unicode Text

Using this option creates the text output in Unicode.

***Example:** Normally shells do not support Unicode, therefore the output should be written to a file like this:*

```
pdtxt -u -o unicode.txt input.pdf
```

-w Word Mode

The word mode extracts text by words. If the font or font size changes, there will be a new word, even when the text appears visually as one word.

3.3 Return Codes

For both, the pdfextract and the pdtxt, all return codes other than "0" indicate an error in the processing.

- 0 Success
- 1 PDF Input File could not be opened or invalid parameters
- 2 Output File could not be created
- 3 Invalid option or option values were entered
- 4 PDF Input File is encrypted and password is incorrect or not provided