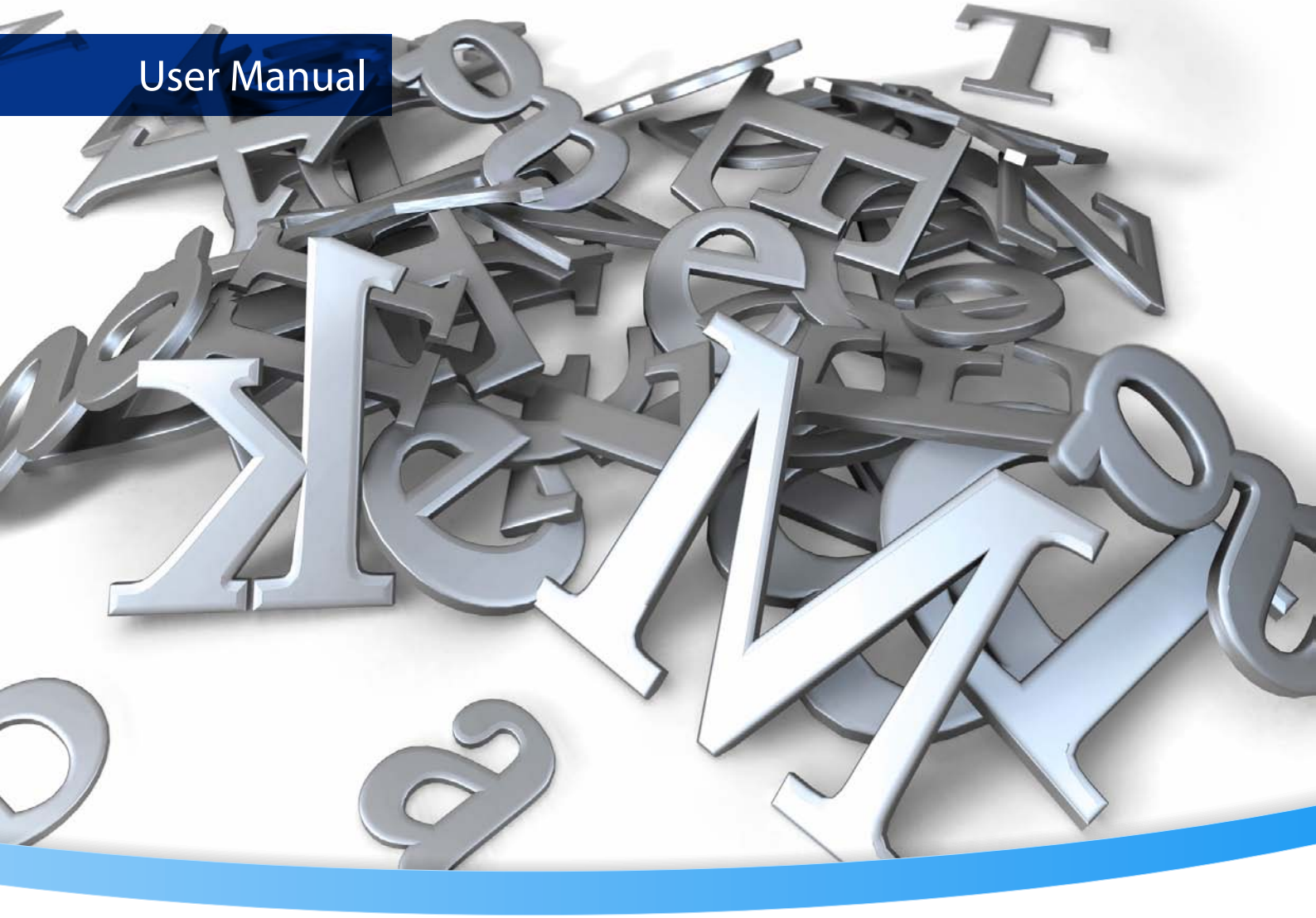
User Manual



3-Heights®
PDF Extract Shell

PDF/A conform

Version 6.27.2

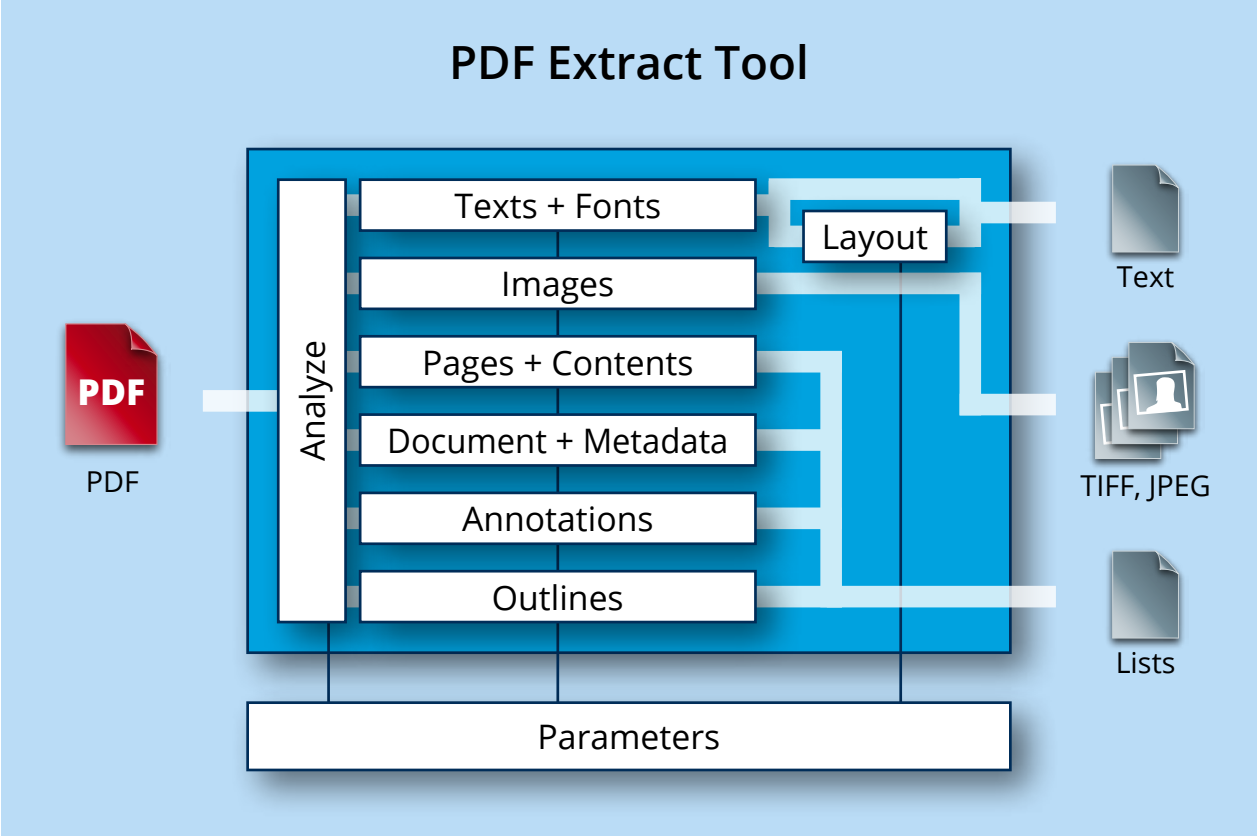
pdftools

Contents

1 Introduction

1.1 Description

The 3-Heights® PDF Extract Shell is a tool for extracting and querying various attributes and page content from a PDF document. This includes text, images, graphic objects, metadata, and embedded fonts, where some object types have additional properties to query. Configurable, intelligent mechanisms significantly increase extraction rates, for instance when extracting text.



1.2 Functions

The 3-Heights® PDF Extract Shell 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.

1.2.1 Features

- Extract text:

    - Extract character by character
    - Extract line by line, with configurable line detection
    - Extract word by word, with configurable word boundary detection
    - Retrieve text attributes such as position, font and font size
    - Automatically apply correct character decoding and produce Unicode output
    - Extract raw character codes
    - Update to-Unicode mapping for fonts from external file
    - Expand common ligatures
- Extract graphics objects (paths) as strings that contain PDF graphics operators
- Extract and store images:
    - Retrieve image attributes such as compression format, position, and transparency masks
- Extract PDF document-level information:
    - Page count
    - PDF version
    - Page labels
    - Creation and modification date
    - Document information such as title, author, subjects, and more
    - Outlines (bookmarks), including destinations
- Extract page information:
    - Media box, crop box, trim box, bleed box, and art box
    - Page rotation
    - Annotations
- Extract and store embedded font files
- Retrieve color space information
- Extract and store embedded files
- Extract and store signatures
- Write CSV output including header line
- Specify a password to decrypt PDF files

1.2.2 Formats

Input Formats:

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

1.2.3 Conformance

Standards:

- ISO 32000-1 (PDF 1.7)
- ISO 32000-2 (PDF 2.0)
- ISO 19005-1 (PDF/A-1)
- ISO 19005-2 (PDF/A-2)
- ISO 19005-3 (PDF/A-3)

1.3 Operating systems

The 3-Heights® PDF Extract Shell is available for the following operating systems:

- Windows Client 7+ | x86 and x64
- Windows Server 2008, 2008 R2, 2012, 2012 R2, 2016, 2019, 2022 | x86 and x64
- Linux:
  - Red Hat, CentOS, Oracle Linux 7+ | x64
  - Fedora 29+ | x64
  - Debian 8+ | x64
  - Other: Linux kernel 2.6+, GCC toolset 4.8+ | x64
- macOS 10.10+ | x64

'+' indicates the minimum supported version.

2 Installation

2.1 Windows

The 3-Heights® PDF Extract Shell comes as a ZIP archive or as an MSI installer.

To install the software, proceed as follows:

1. You need administrator rights to install this software.
2. Log in to your download account at https://www.pdf-tools.com. Select the product "PDF Extract Shell". If you have no active downloads available or cannot log in, please contact pdfsales@pdf-tools.com for assistance.
   You can find different versions of the product available. Download the version that is selected by default. You can select a different version.
   There is an MSI (`*.msi`) package and a ZIP (`*.zip`) archive available. The MSI (Microsoft Installer) package provides an installation routine that installs and uninstalls the product for you. The ZIP archive allows you to select and install everything manually.
   There is a 32 and a 64-bit version of the product available. While the 32-bit version runs on both 32 and 64-bit platforms, the 64-bit version runs on 64-bit platforms only. The MSI installs the 64-bit version, whereas the ZIP archive contains both the 32-bit and the 64-bit version of the product. Therefore, on 32-bit systems, the ZIP archive must be used.
3. If you select an MSI package, start it and follow the steps in the installation routine.
4. If you are using the ZIP archive, uU nzip the archive to a local folder, e.g. `C:\Program Files\PDF Tools AG\`.
   This creates the following subdirectories:

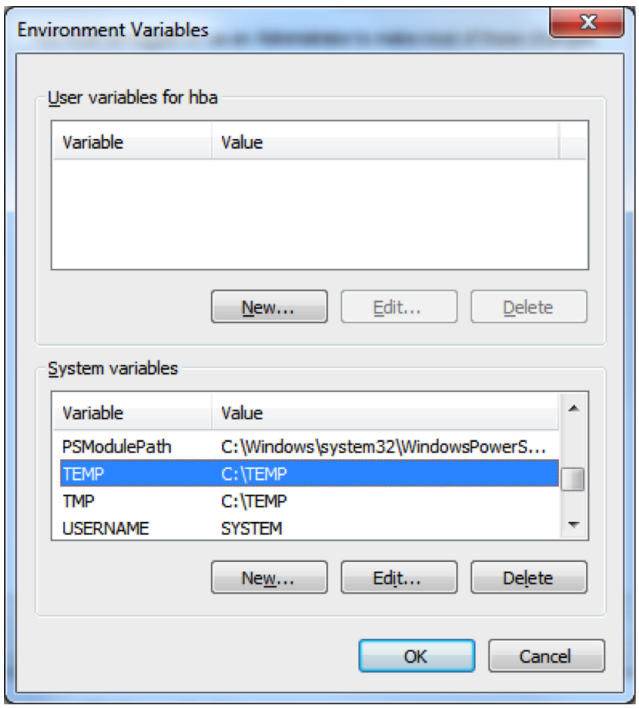
Subdirectory	Description
`bin`	Runtime executable binaries
`doc`	Documentation

5. (Optional) To easily use the 3-Heights® PDF Extract Shell from a shell, the directory needs to be included in the "Path" environment variable.
6. (Optional) Register your license key using the License management.
7. Make sure your platform meets the requirements regarding color spaces described in Color profiles.

2.1.1 How to set the environment variable "Path"

To set the environment variable "Path" in Windows, go to Start → Control Panel (classic view) → System → Advanced → Environment Variables.

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



2.2 Linux and macOS

This section describes installation steps required on Linux or macOS.

Here is an overview of the files that come with the 3-Heights® PDF Extract Shell:

File description

Name	Description
`bin/x64/pdfextract`	Main executable
`doc/*.*`	Documentation

2.2.1 Linux

1. Unpack the archive in an installation directory, e.g. `/opt/pdf-tools.com/`
2. Verify that the GNU shared libraries required by the product are available on your system:

```
ldd pdfextract
```

   If the previous step reports any missing libraries, you have two options:
   a. Download an archive that is linked to a different version of the GNU shared libraries and verify whether they are available on your system. Use any version whose requirements are met. Note that this option is not available for all platforms.
   b. Use your system's package manager to install the missing libraries. It usually suffices to install the package `libstdc++6`.
3. Create a link to the executable from one of the standard executable directories, e.g.

```
ln -s /opt/pdf-tools.com/bin/x64/pdfextract /usr/bin
```

4. Optionally, register your license key using the license manager.
5. Make sure your platform meets the requirements regarding color spaces described in Color profiles.

2.3 Uninstall

If you have used the MSI for the installation, go to Start → 3-Heights® PDF Extract Shell… → Uninstall …

If you have used the ZIP file for the installation, undo all the steps done during installation.

2.4 Color profiles

When extracting images, a color conversion may be neccessary.

For calibrated color spaces (such color spaces with an associated ICC color profile), the color conversion is well defined. For the conversion of uncalibrated device color spaces (DeviceGray, DeviceRGB, DeviceCMYK), however, the 3-Heights® PDF Extract Shell requires apropriate color profiles. Therefore, it is important that the profiles are available and that they describe the colors of the device your input documents are intended for.

If no color profiles are available, default profiles for both RGB and CMYK are generated on the fly by the 3-Heights® PDF Extract Shell.

2.4.1 Default color profiles

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

Windows

1. `%SystemRoot%\System32\spool\drivers\color`
2. directory `Icc`, which must be a direct subdirectory of where the `pdfextract.exe` resides.

Linux and macOS

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

2.4.2 Get other color profiles

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

- https://www.pdf-tools.com/public/downloads/resources/colorprofiles.zip
- https://www.color.org/srgbprofiles.html

3 License management

The 3-Heights® PDF Extract Shell requires a valid license in order to run correctly. If no license key is set or the license is not valid, then the executable will fail and the return code is set to 10.

More information about license management is available in the license key technote.

4 Interface reference

The 3-Heights® PDF Extract Shell is an easy to use tool. However, at some points it could prove helpful if you have a basic understanding about PDF. This manual does not explain any PDF-related features in depth. For further explanation of PDF specific information, see the PDF Reference 1.7.

4.1 pdfextract

When using the listing options such as `-la`, `-lb`, and `-lc`, 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.

4.1.1 `-io` Ignore OCM

Ignore OCM `-io`

If this option is specified, then optional content membership (OCM) is ignored and all content is made visible. While `BeginOCM` and `EndOCM` objects are still extracted when using the `-lp` `-x` options, these objects have no more an effect on the extracted content. For example, when set, then text in a optional content group (OCG, also known as "layer") that is not visible is extracted as well.

4.1.2 `-h` Include a CSV header line

Include a CSV header line `-h`

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

4.1.3 `-la` List annotations

List annotations `-la`

This option lists all annotations including page number, type, position, 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`, and `Widget`. (See table 8.16 in the PDF Reference 1.7.)
- Position and size (Left, Bottom, Right, Top): The rectangle of the annotation. The origin is in the lower left corner of the page as displayed by a viewer. The units are points, which is 1/72 inch (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 1.7.)
- 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.
- Target: The target destination of a link, launch, or remote GoTo annotation. The format is "<targetpage> <destination>". (Refer to chapter 8.2 in the PDF Reference 1.7 for more information on destinations.)

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","",
```

4.1.4 `-laf` List form fields

List form fields `-laf`

This switch lists the form fields in a document. Since form fields are also annotations, they may also be listed using `-la`. The difference is that form fields may be hierarchically nested (parents/children). The listing contains fields that are more related to form fields than annotations. Naturally, annotations that are not form fields, such as 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. The origin is in the lower left corner of the page as displayed by a viewer. The units are points, which correspond to 1/72 inch (A4 = 595x842 points, Letter = 612x792 points).
- Flags: Annotation flags are listed in the PDF Reference 1.7 chapter 9.4 (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. `Tx`, `Btn`, `Chk`, etc.
- FieldFlags: The form field flags are listed in the PDF Reference 1.7 chapter 9.5. Here is an extract:

15	NoToggleToOff
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," "
```

4.1.5 `-lb` List outlines

List outlines `-lb`

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

- Level: The outline root level is 1. The number of a child outline is one level higher than its parent.
- Count: The number of visible children. Not expanded children count negative. (See also chapter G.5 in the PDF Reference 1.7.)
- Destination: The destination type, such as `Fit`, `FitH`, `FitV`, XXY. (See also chapter 8.2 in the PDF Reference 1.7.)
- 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 1.7.)

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
```

4.1.6 `-lc` List color spaces

List color spaces `-lc`

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, usually 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
```

4.1.7 `-ld` List document attributes

List document attributes `-ld`

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

- ClaimedCompliance: The PDF version or PDF\A version that this document claims to conform to. This is any of the following strings:
  - `pdf1.0` (PDF 1.0)
  - `pdf1.1` (PDF 1.1)
  - `pdf1.2` (PDF 1.2)
  - `pdf1.3` (PDF 1.3)
  - `pdf1.4` (PDF 1.4)
  - `pdf1.5` (PDF 1.5)
  - `pdf1.6` (PDF 1.6)
  - `pdf1.7` (PDF 1.7)
  - `pdf2.0` (PDF 2.0)
  - `pdfa-1a` (PDF\A-1a)
  - `pdfa-1b` (PDF\A-1b)
  - `pdfa-2a` (PDF\A-2a)
  - `pdfa-2b` (PDF\A-2b)
  - `pdfa-2u` (PDF\A-2u)
  - `pdfa-3a` (PDF\A-3a)
  - `pdfa-3b` (PDF\A-3b)

- pdfa-3u (PDF\A-3u)
- PageCount: The total number of pages.
- IsLinearized: Set to “Linearized” if the document is linearized (optimized for fast web view), and blank otherwise.
- IsEncrypted: Set to “Encrypted” if encrypted, and blank otherwise.
- IsCollection: Set to “Collection” if the document is a PDF collection, and blank otherwise.
- Title, Author, Subject, Keywords, Creator, Producer: The value of the corresponding document attribute.
- CreationDate, ModificationDate: The date in the format yyyy-mm-dd.
- Metadata: The file name under which XMP metadata is stored when using the `-x` option.

Example: List document attributes

```
pdfextract -u -ld -h exps.pdf
FileName,ClaimedCompliance,PageCount,IsEncrypted,IsLinearized,IsCollection,
Title,Author,Subject,Keywords,Creator,Producer,CreationDate,ModificationDate,
Metadata
"C:\exps.pdf",pdfa-2a,29,,Linearized,,"3-Heights® PDF Extract Shell","PDF
Tools AG","PDF Extract—component for extracting page content (text), resources
(fonts) and other information from PDF documents.","","LuaTeX + ConTeXt Mk
IV","3-Heights(TM) PDF to PDF-A Converter Shell 4.6.26.7
(http://www.pdf-tools.com)",2016-06-20,2016-06-20,
```

4.1.8 `-ldss` List content of the document security store (DSS)

List content of the document security store (DSS) `-ldss`

List certificates, CRLs, and OCSPs of the document security store.

Example: List DSS items and properties:

```
pdfextract -h -ldss document.pdf
Type,Revision,Name,ContentFileName
Cert,1,"Peter Pan",
CRL,1,"Peter Pan's Root CA" (validity 2013-07-31 00:00:00Z - 2018-07-30 23:59:59Z),
OCSP,1,"Peter Pan's OCSP Responder" from 2017-08-23 20:14:34Z,
```

4.1.9 `-lef` List embedded files

List embedded files `-lef`

List all embedded files including name, creation date, and modification date. If the embedded file is extracted using `-x`, it also lists the file name.

Example: Extract and save embedded files

```
pdfextract -x -h -lef input.pdf
Name,CreationDate,ModDate,FileName
"f1.doc","D:20110514063512+01'00'","D:20120104095404+01'00'","f1.doc"
"f2.pdf","D:20070208134624+01'00'","D:20070208134624+01'00'","f2.pdf"
```

4.1.10 -lf List fonts and font properties

List fonts and font properties `-lf`

This option lists all fonts and font 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. Many applications such as Adobe Acrobat remove this information from the font name, and mark the font as subset.
- FontType: The type of the font such as `Type0`, `Type1`, `MMType1`, `TrueType`, `Type3`, `CIDFontType0`, `CIDFontType2`. (See PDF Reference 1.7 Chapter 5.4.)
- Encoding: The encoding, such as WinAnsiEncoding, DifferenceEncoding, MacRomanEncoding, Identity-H. (See PDF Reference 1.7 Appendix D.)
- IsCID: "CID" if the font is a CID font, and blank otherwise.
- IsEmbedded: "Embedded" if the font program is embedded, and blank otherwise.
- IsSubsetted: Returns "Subsetted" if the font is subset and otherwise.
- FontFileName: The name of the font when extraction using the option `-x` is applied. (This value is not listed without `-x`.)

When used in combination with `-r`, then fonts are listed 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, IsSubsetted,
FontFileName
document.pdf,0,"Verdana",TrueType,WinAnsiEncoding,,,,
document.pdf,0,"Verdana-Bold",TrueType,WinAnsiEncoding,,,,
document.pdf,0,"HMAGKB+SymbolMT",CIDFontType2,Identity-H,CID,Subsetted,
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,,Subsetted,Embedded,
document.pdf,0,"CourierNewPSMT",TrueType,WinAnsiEncoding,,,,
document.pdf,0,"HMAMDL+ArialUnicodeMS",CIDFontType2,Identity-H,CID,Subsetted,
Embedded,
```

4.1.11 -li List images and properties

List images and properties `-li`

List images in the PDF document and image 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:

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 1.7, chapter 4.8). Most images, particularly large images, are stored as image XObjects. 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 on different pages require only one resource and therefore keep the file size small.

Listing images by resources returns images from the PDF document's resources, i.e. images from XObjects, but not inline images. These images do not have a well defined 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 both listed this way.

The following properties are extracted for images:

- PageNo: The page number. This value is set to 0 if images are extracted by resources.
- Width, Height: The dimensions in dots (pixels).
- x0, y0: The coordinate of the lower left corner of the image in points. These values are 0 if images are extracted by resources.
- x1, y1: The coordinate of the upper right corner of the image in points. (1 point is 1/72 inch.) 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.
- BitsPerComponent: The number of bits per component, such as 1 for bitonal images or 8 for color and grayscale 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`, and `Null`,.
- Mask: Can take the `Null`, `Stencil`, `Explicit` and `Soft` values. The "ColorSpace" field is set to `Null` for stencil mask images.
- Filter: The image filter, such as `DCTDecode`, `CCITTFaxDecode`, and `FlateDecode`.
- ImageFileName: The name of the image when extraction using the `-x` option is applied.
  For XObject images, the name is `img‹obj number›.‹ext›`. For inline images, it is `imginl‹number›.‹ext›`, where:
  `‹obj number›` is the object number
  `‹number›` is a counter for all inline images
  `‹ext›` is either `jpg` if the image is compressed with a DCT filter, or `tif` in all other cases

Example: List image by resources:

```
pdfextract -h -li -r PDFReference16.pdf
FileName,PageNo,x0,y0,x1,y1,Width,Height,BitsPerComponent,XDPI,YDPI,
ColorSpace,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,
```

4.1.12 -lk Set license key

Set license key `-lk ‹key›`

Pass a license key to the application at runtime, instead of using one that is installed on the system.

```
pdfextract -lk X-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX ...
```

This is required in an OEM scenario only.

4.1.13 -lp List pages and properties

List pages and properties `-lp`

List pages and page 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 defines 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.
- ContentFileName: The name of the text file containing the content when extraction using the `-x` switch is applied. (This value is not listed without `-x`.)

Example: List pages and properties:

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

When combining this switch with `-x`, the content streams of the pages are extracted and written into individual files named `cnt‹N›.txt`, where `‹N›` is the page number, e.g. `"cnt1.txt"`.

4.1.14 -ls List signatures and properties

List signatures and properties `-ls`

List digital signatures and signature properties such as the name of the certificate or the reason why the signature was applied. For unsigned signature form fields, the field name is returned.

Example: List signatures and properties:

```
pdfextract -h -ls document.pdf
Name,Reason,Revision,ContentFileName
"Peter Pan","I am the author of the document",0,
```

4.1.15 -o Write output to file

Write output to file `-o ‹filename›`

With this option, the output can be directed to a file name.

Example: Extract pages and 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.

Example:

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

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

Example: To pipe error and warning messages into a file such as `0x80410042 - E - The content stream contains an invalid operator`.

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

Example: To discard them, use a command like this:

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

4.1.16 -p Specify a password to decrypt the input file

Specify a password to decrypt the input file `-p`

To read PDF documents that require a password to be opened, a password (user or owner password) can be provided using the -p switch.

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

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

4.1.17 -pg List page range

List page range `-pg ‹first page› ‹last page›`

Set a page range. Some listing functions such as fonts or images can be listed by resources (document level) or by page. If the -r switch 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.

4.1.18 -raw Extract resources in raw format

Extract resources in raw format `-raw`

This switch instructs the tool to extract resources in raw format rather than a converted format. Without this switch, font resources are converted to an installable format. It is used in conjunction with -x and the various listing options (-la, -laf, -lb, -lc, -lf, -li, and -lp).

4.1.19 -r Extract by resources

Extract by resources `-r`

Extract data such as images or fonts by resources instead of by page. See -li and -lf switches.

4.1.20 -u Encode output using Unicode

Encode output using Unicode `-u`

The output is written as WinAnsi as default. To write the output as Unicode, use the -u switch.

4.1.21 -v Verbose mode

Verbose mode `-v`

This option turns on the verbose mode.

In the verbose mode, additional information during the processing is written to the shell.

4.1.22 -x Extract and store embedded data

Extract and store embedded data `-x`

This option allows to extract data such as images or fonts. If a document contains an embedded font, then the font is listed with “Embedded” set and the embedded font file can be extracted.

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 object number as file name (e.g. `fnt38.ttf`). Extracted fonts are not installable fonts (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
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, and Height are provided to describe the 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 `-li -x` options. For example, `"img9.tif"`, where the number (9) is the object number for this image. `"imginl2.tif"`, where the number (2) is a counter for inline images.
  - 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: 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"
```

4.2 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. This mode is the default.

Word mode Extract words. Use `-w` to activate this mode.

Text mode Extract all text and take into account the page layout. Use `-t` to activate this mode.

> **Note:** The option `-s` allows you to translate a certain part from the Unicode custom range to WinAnsi codes. It is recommended to enable this option regardless of the extraction mode.

4.2.1 -a Set advance width for text mode

Set advance width for text mode `-a`

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

4.2.2 -c Character mode

Character mode `-c`

With this option, text is extracted character by character.

4.2.3 -fd Directory of pre-installed fonts

Directory of pre-installed fonts `-fd ‹directory›`

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

4.2.4 -h Write a CSV header

Write a CSV header `-h`

Add a CSV (comma-separated values) header as first line. This option can be used in combination with the `-c` or `-w` options, 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)

4.2.5 `-if` Ignore fonts

Ignore fonts `-if`

If this option is set, then changes in the font are ignored when merging text.

Note: If this option is set, then the reported ‹Width› (or in case `-ow` is used ‹XWidth› and ‹YWidth›) of text elements is not correct.

4.2.6 `-l` Line heights for text mode

Line heights for text mode `-l ‹height›`

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. More specifically: 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.

4.2.7 -lk Set license key

Set license key `-lk ‹key›`

Pass a license key to the application at runtime, instead of using one that is installed on the system.

```
pdfextract -lk X-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX ...
```

This is required in an OEM scenario only.

4.2.8 -lt Line height tolerance

Line height tolerance `-lt ‹tolerance›`

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

This switch works in conjunction with the line height switch.

Default: 3 pt

4.2.9 -o Extract text to a file

Extract text to a file `-o ‹filename›`

This option extracts the text to an output file. For example, the following command extracts 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:

Example:

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

4.2.10 -of Factor to use when separating words

Factor to use when separating words `-of ‹factor›`

This option controls the word separation algorithm of the text extraction tool. The parameter is interpreted as a factor, which is multiplied by the width of the space character. If the distance between two characters is greater than the computed result, it is taken as a word boundary.

The default is 0.3.

4.2.11 -or Extract raw string

Extract raw string `-or`

This switch extracts the raw character string of a text as an additional column in the output file. The codes of the character reflect the font's encoding. For fonts with multi-byte encoding, the raw string is empty. The switch does not work in conjunction with the `-sl` switch.

4.2.12 -ow Write widths in X and Y direction separately

Write widths in X and Y direction separately `-ow`

This switch replaces the `Width` column (4th column) by the two columns `XWidth` and `YWidth`.

4.2.13 -p Specify password

Specify password `-p ‹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 "secret".

```
pdtxt -p secret input.pdf
```

4.2.14 -pg Extract a page range

Extract a page range `-pg ‹first page› ‹last page›`

Apply extraction to a selected page range.

Example: Extract text from pages 1 to 2.

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

Default: Extract all pages.

4.2.15 -s Replace symbolic characters

Replace symbolic characters `-s`

Replace symbolic character from the Unicode custom range (0xF000 to 0xF0FF) with WinAnsi codes (0x00 to 0xFF).

Note: It is generally recommended to enable this option.

4.2.16 -sl Replace ligatures

Replace ligatures `-sl`

Ligatures such as ff, fi, fl, ffi, ffl found during text extraction are converted to individual characters ff, fi, fl, etc.

4.2.17 -t Text mode

Text mode `-t`

The text mode lets you extract text from pages and retain the page layout to a certain extent. Depending on the font size, the `-a` option can be used to set the advance width, and the `-l` option to set the line height.

4.2.18 -u Create Unicode text

Create Unicode text `-u`

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
```

4.2.19 -uf Set ToUnicode information

Set ToUnicode information `-uf ‹ToUnicodeFile›`

The configuration file lets you update the mapping from character codes to Unicodes. This mapping does not have to be complete or bijective. Specifically, one character code can map to a sequence of Unicodes. Use this feature if the text is not extractable and you know the encoding used by the creator of the PDF.

Example: Set `ToUnicode` information from file tounicode.txt:

```
pdtxt -uf tounicode.txt input.pdf
```

The `‹ToUnicodeFile›` uses the .ini file syntax, where each section updates the mapping of the respective font.

Example: The following file sets the Unicode of the font “ATTHelv”. This updates character codes 157, 158, 98, and 24 to the Unicode ‘a’, ‘b’, the trademark sign, and the Unicode sequence “Greek capital letter Delta” “combining right arrow above”, respectively.

```
[ATTHelv]
```

```
0x9d = 'a'
0x9e = 'b'
98 = 0x2122
34 = 0x0394 0x20D7
```

4.2.20 -w Word mode

Word mode `-w`

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

4.3 Return codes

All return codes other than `0` indicate an error in the processing.

Return codes

Value	Description
`0`	Success.
`1`	Couldn't open input file.
`3`	Error with given options, e.g. too many parameters.
`4`	PDF input file is encrypted and password is missing or incorrect.
`5`	Extraction error either due to corrupt input PDF or failure when storing an extracted file.
`10`	License error, e.g. invalid license key.

5 Version history

5.1 Changes in versions 6.19–6.27

- **Update** license agreement to version 2.9

5.2 Changes in versions 6.13–6.18

No functional changes.

5.3 Changes in versions 6.1–6.12

No functional changes.

5.4 Changes in version 5

- **New** additional supported operating system: Windows Server 2019.

Shell `pdfextract`

- **New** option `-ldss` to list all items of the document security store (DSS).
- **Improved** option `-ls`:
  - Return the number of the document revision that contains the signature.
  - Return the name of the signing certificate.

5.5 Changes in version 4.12

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

Shell `pdfextract`

- **Improved** extraction performance when listing resources with `-r` for certain documents.

5.6 Changes in version 4.11

- **New** support for reading PDF 2.0 documents.
- **Improved** repair of corrupt image streams.

Shell `pdfextract`

- **Improved** reporting of colorants for pattern color spaces.

Shell `pdtxt`

No functional changes.

5.7 Changes in version 4.10

- **Improved** robustness against corrupt input PDF documents.

Shell `pdfextract`

- **New** exit code 5 indicating an extraction or file save error.
- **Changed** options `-ls` and `-x`: Extraction of signatures for encrypted documents is now possible.

Shell `pdtxt`

- **Changed** option `-uf`: The file to update the ToUnicode font information now supports mappings from a character code to a sequence of Unicodes.
- **New** support of overlapping code ranges in font's ToUnicode tables (used for text extraction).

5.8 Changes in version 4.9

- **Improved** support for and robustness against corrupt input PDF documents.
- **Improved** repair of embedded font programs that are corrupt.
- **New** support for OpenType font collections in installed font collection.

5.9 Changes in version 4.8

- **New** support for ToUnicode mappings that map characters to sequences of Unicodes (longer than 1). This includes special ligatures and surrogate pairs.
- **Improved** space width heuristic. This algorithm is required in order to estimate the width of a space in fonts that contain no space character. The space width is used to detect word breaks for example.
- **Improved** creation of annotation appearances to use less memory and processing time.
- **Added** repair functionality for TrueType font programs whose glyphs are not ordered correctly.

Shell `pdfextract`

- **Changed** option `-ld`: Added document attributes:
  - Report claimed PDF conformance.
  - Report whether document is linearized (fast web view).
  - Report whether document is a collection.

6 Licensing, copyright, and contact

Pdftools (PDF Tools AG) is a world leader in PDF software, delivering reliable PDF products to international customers in all market segments.

Pdftools provides server-based software products designed specifically for developers, integrators, consultants, customizing specialists, and IT departments. Thousands of companies worldwide use our products directly and hundreds of thousands of users benefit from the technology indirectly via a global network of OEM partners. The tools can be easily embedded into application programs and are available for a multitude of operating system platforms.

Contact

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